


# Leveraging mobility and content caching for proactive load balancing in heterogeneous cellular networks

Sanaullah Manzoor<sup>1</sup>  | Suleman Mazhar<sup>1</sup> | Ahmad Asghar<sup>2</sup> | Adnan Noor Mian<sup>1,3</sup> | Ali Imran<sup>2</sup> | Jon Crowcroft<sup>3</sup>

<sup>1</sup>Department of Computer Science, Information Technology University, Lahore, Pakistan

<sup>2</sup>AI4Networks Lab, Department of Electrical and Computer Engineering, University of Oklahoma, Tulsa, OK

<sup>3</sup>Department of Computer Science and Technology, University of Cambridge, Cambridge, UK

## Correspondence

Sanaullah Manzoor, Department of Computer Science, Information Technology University, Lahore-54000, Pakistan.  
Email: sanaullah.manzoor@itu.edu.pk

## Funding information

Punjab Higher Education Commission (PHEC), Lahore, Pakistan; National Science Foundation, Grant/Award Number: 1619346, 1559483, 1718956, and 1730650

## Abstract

Evolution of cellular networks into dynamic, dense, and heterogeneous networks have introduced new challenges for cell resource optimization, especially in the imbalanced traffic load regions. Numerous load balancing schemes have been proposed to tackle this issue; however, they operate in a reactive manner that confines their ability to meet the top-notch quality of experience demands. To address this challenge, we propose a novel proactive load balancing scheme. Our framework learns users' mobility and demands statistics jointly to proactively cache future contents during their stay at lightly loaded cells, which results in quality of experience maximization and load minimization. System level simulations are performed and compared with the state-of-the-art reactive schemes.

## 1 | INTRODUCTION

Mobile data traffic is increasing exponentially with the expeditious rise in wireless devices and data-hungry applications. It is anticipated that 50 billion devices will be connected and associated to the Internet by 2021, with an average of four devices per person, where each user will be generating nearly 61 GB of data traffic per month.<sup>1,2</sup> At present, about 50% network traffic is only coming from video streaming applications such as *Netflix* and *YouTube*.<sup>3</sup> Moreover, it is also projected that nearly 80% of the future data traffic would be made up of video contents.<sup>1,3</sup> Such applications require uninterrupted content delivery with the ambitious quality of experience (QoE) and ultra-low latency.<sup>4</sup>

These challenges, along with the promise of future cellular networks, necessitate premium quality of service with extremely low latency and unprecedented bandwidth capacity. Despite the advancements in physical layer techniques and higher frequency spectrum exploration, network densification is the most yielding approach to achieve the prodigious capacity gain in the future cellular networks (5G and beyond).<sup>5,6</sup> Under this densification theme, many low-power small base stations (SBS) are overlaid in addition to the conventional macro base station (MBS), and the resultant network is termed as heterogeneous cellular networks (HetNets).<sup>4,5,7</sup> It is intuitive that such massive deployments of nodes will create numerous problems, including uneven traffic load, unfair resource allocation, mobility management, backhaul congestion, etc. Among the aforementioned challenges, imbalanced load distribution between macro and small cells is a long-standing problem in these HetNets, which results in inefficient utilization of systems' installed capacity and users' QoE degradation.<sup>7,8</sup>

Various load balancing (LB) solutions have been proposed (eg, see other works<sup>7-13</sup>) to address the uneven load issue; however, they operate in reactive mode. Under the vast deployment of nodes such as HetNets, these reactive LB policies may become a bottleneck that hinders network to deliver contents seamlessly and leads to worse user experience. The LB mechanism can be improved significantly through proactive traffic offloading by inferring the user entity (UE) behavior, along with network state information such as channel conditions, cell loads, etc. The imbalanced cell load problem in HetNets arises due to state-of-the-art user-cell association procedure, ie, maximum reference signal received power (Max-RSRP).<sup>8</sup> Under this procedure, a UE selects the cell with the highest RSRP value. In HetNets, MBS has higher transmission power than SBS; therefore, MBS attracts more UEs than SBS. Consequently, MBSs become congested, SBSs remain underutilized, and overall network faces severely uneven load distribution.<sup>7-9</sup> In such a scenario, the congested cells also have to accommodate incoming traffic in addition to current UEs with the certain level of QoE provisioning. This will cause the congested cells to become more overloaded and can offer only worse quality of services. To overcome this load imbalance issue, one of the promising solutions is traffic offloading from congested cells to lightly loaded cells.<sup>7,12,14</sup>

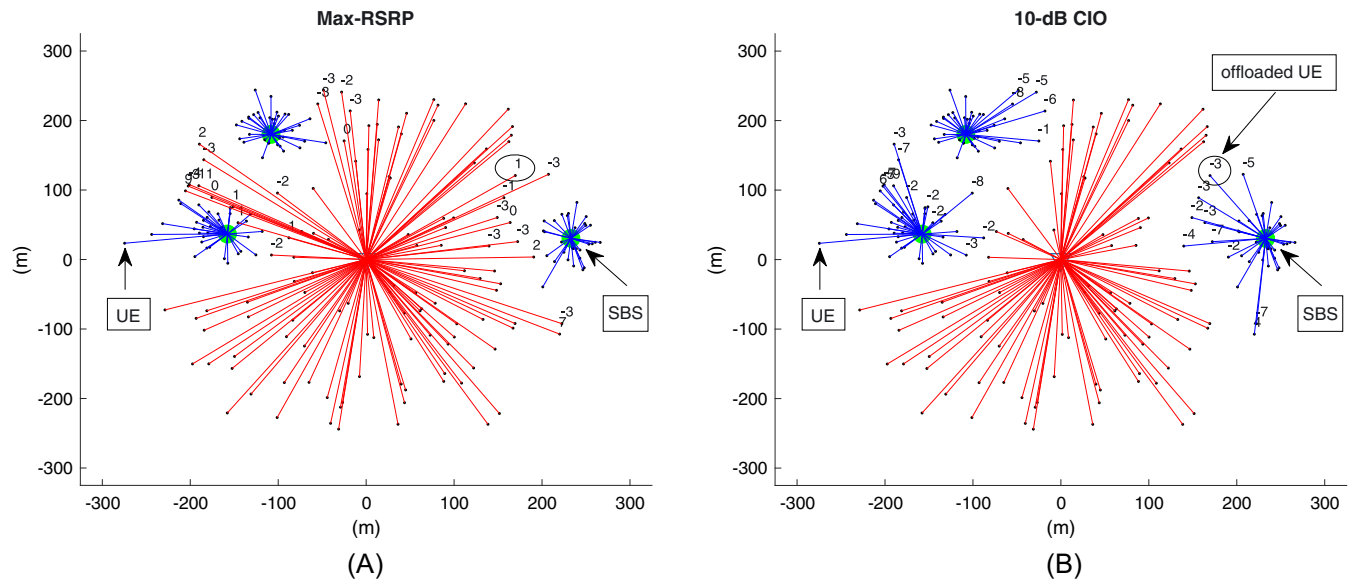
The traffic offloading through cell range expansion has been considered as a widely adopted solution (eg, see other works<sup>7,8,14-16</sup>). In this procedure, cell transmission power is virtually boosted through the assignment of cell individual offset (CIO) parameter.<sup>7-9,14,17</sup> As a result, some traffic from the congested cell are shifted to neighboring cells.<sup>12,14</sup> The main goal of this scheme is to improve users' QoE in such a way that the offloaded UEs can get more physical resource blocks (PRB) from the new cell (if available) and remaining UEs at previously congested cells get relief from the congestion state and experience better downlink (DL) throughput. However, this reactive offloading mechanism activates CIOs after any or more cells become congested, without taking account of cell loads and users' QoE demand information, which might affect the signal-to-interference-and-noise ratio (SINR) of offloaded users adversely.<sup>8</sup>

In order to better understand this reactive offloading phenomenon, we perform simulations and show its effect with the help of an example HetNet scenario in Figure 1. The figure shows a HetNet model having one MBS with 46 dBm transmit power, three SBSs with 30 dBm transmitting power each, and one small cell per sector. Moreover, UEs and SBSs are uniformly placed. We consider 200 UEs in the overall system. We follow the distance-dependent pathloss model  $PL(d) = 128.1 + 37.6\log_{10}(d)$  and  $PL(d) = 140.1 + 36.7\log_{10}(d)$  for MBS and SBS, respectively, where  $d$  is the distance between UE and BS in meters. We consider Log-normal shadowing with standard deviation of 8 dB and noise power spectral density of 174 dBm/Hz. Figure 1A illustrates users' SINR under Max-RSRP procedure, and Figure 1B shows users' SINR after 10 dB CIO activation. Under 10 dB CIO approach, some users are offloaded from MBSs to SBSs. As a result, after reassociation with new cells, offloaded UEs are facing a clear drop in their SINR. From Figure 1A, we can observe that UE in the circle has 1 dB SINR initially, while after reassociation, its SINR becomes  $-3$  dB. This indicates that new BS lacks surplus PRBs to meet UE's QoE demands. The applications running at this UE may be stopped due to inadequate DL rate, which can reduce its confidence over mobile network operator services. Consequently, reactive CIO activation without consideration of UEs' QoE demand, cell load, and the remaining number of PRBs information lead to a decrease in user satisfaction ratio.

To address the above limitations, we propose a novel proactive load balancing (PLB) framework that takes account of user trajectories and traffic requirements jointly to cache user's future contents proactively in the lightly loaded cells other than possibly congested cells. To best of our knowledge, this is the first PLB solution that leverages users' mobility patterns and content demand statistics for proactive traffic offloading. The key idea behind this framework is to make cellular networks autonomous and artificially intelligent so that they can anticipate users' behavior for cell LB optimization and improve users' QoE. Notable contributions of this paper are summarized as follows.

1. We proposed a novel PLB framework that leverages user mobility and content demands jointly to maximize users' downlink (DL) throughput and minimize cell loads.
2. To offload future contents at expected cells, user mobility patterns are modeled through Semi-Markov renewal process and we proposed a novel caching approach using *logdet* optimization of user-file rating matrix.
3. We estimated user satisfaction ratio to demonstrate the effectiveness of proactively offloaded contents.
4. To practically assess the performance of the proposed PLB framework we trained our mobility and caching models using real data traces instead of simulation-based synthetic data.

The rest of the manuscript is organized as follows. Section 2 presents related work. Section 3 explains our PLB framework. Section 4 covers performance evaluation and discussions on results obtained. Section 5 provides analysis and insights into the proposed framework. Finally, Section 6 concludes this work and provides suggestions for future work.



**FIGURE 1** Repercussions of reactive offloading mechanism in the heterogeneous cellular networks (HetNets). A, Maximum reference signal received power (Max-RSRP)-based user association; B, User offloading using 10 dB cell individual offset (CIO)

## 2 | RELATED WORK

A load balancing (LB) scheme aims to optimize cell loads for efficient spectrum capacity utilization and users' QoE maximization. Therefore, LB mechanism is considered as an indispensable approach for network resource optimization.<sup>7</sup> Mainly, LB schemes are devised under certain objectives, such as users' DL rate maximization,<sup>9–12</sup> overall network energy efficiency optimization,<sup>18</sup> cell capacity and coverage optimization,<sup>8</sup> global outage probability minimization<sup>19</sup> and network mobility management.<sup>20</sup> Load balancing schemes can be classified into a reactive or proactive scheme, based on their mode of operation. Reactive LB schemes ignore incipient cell congestion and activate CIOs after one or more cells become congested. Under these schemes, cells are allowed to associate UEs until they are fully loaded despite the availability of lightly loaded cells in the near vicinity. In contrast, proactive schemes aim to offload traffic prior to cell overloading. We now explore insights of existing reactive and proactive LB schemes in terms of their key characteristics and objectives, including (i) user mobility, (ii) content caching, (iii) backhaul, (iv) energy efficiency, (v) cell load prediction, and (vi) whether the LB mechanism is reactive or proactive. Table 1 compares the most prominent LB solutions, which are proposed in the last five years. We see that most of the LB approaches operate in the reactive manner and ignore UE's important information such as mobility and content demands. Majority of the existing works have paid very little attention about prior cell load prediction and energy efficiency perspective for traffic load balancing. Recently, many attempts have been made to incorporate users' mobility for load balancing.<sup>21,22</sup> Similarly, edge caching has gained popularity in the last decade.<sup>23,24</sup> Thus far, these works utilized either content caching or mobility; however, no work has explored a framework for joint consideration of both factors for proactive load balancing. Next, we present the details of some eminent reactive and proactive LB schemes.

**Reactive load balancing:** Reactive load balancing has been studied thoroughly since the emergence of HetNets. Here, we present only the most prominent and recent approaches. Andrews et al<sup>7</sup> provided a detailed overview of the reactive LB mechanism. Ye et al<sup>9</sup> formulated a joint LB and user association problem to maximize user DL throughput. Similarly, Fooladivanda et al<sup>25</sup> formulated a joint resource allocation and user association problem for LB, Zhou et al<sup>12</sup> extended DL rate maximization problem with QoS constraint, and a recent work<sup>11</sup> proposed joint user-cell association and scheduling policy for LB. However, the algorithms proposed in these studies ignore users' mobility information. Therefore, these schemes are valid under static or low mobility scenarios only.<sup>9</sup> On the other hand, Boostanimehr et al<sup>19</sup> defined a joint user association and LB problem to minimize global outage probability. Zhang et al<sup>18</sup> proposed a user-association scheme for LB with energy efficiency constraint. Similarly, Asghar et al<sup>26</sup> proposed user association for LB and capacity optimization by exploiting the information of cell loads, CIOs, and antenna tilt. Notably, these attempts also ignore user mobility and content demands behavior. Very few research efforts have been made by considering user mobility and content caching jointly. Some schemes such as the work of Coucheney et al<sup>20</sup> consider either users' mobility or users' file demand

**TABLE 1** Comparative analysis of existing load balancing schemes

Type	Ref., year	Mobility	Energy eff.	Backhaul	Caching	Load pred.	Objective
Reactive	<sup>28</sup> , 2012	Yes	No	No	No	No	User association to support handovers minimization
	<sup>9</sup> , 2013	No	No	No	No	No	Throughput maximization and fractional PRB allocation for LB
	<sup>22</sup> , 2014	Yes	No	Yes	Yes	No	UE mobility-aware content caching to minimize backhaul load
	<sup>20</sup> , 2014	Yes	No	No	No	No	Mobility load-aware UE-cell association for rate maximization
	<sup>21</sup> , 2015	Yes	No	No	No	No	CRE with UE mobility and DL coverage probability
	<sup>12</sup> , 2015	No	No	Yes	No	No	Optimization of cell loads for user DL rate maximization
	<sup>18</sup> , 2015	No	Yes	No	No	No	Energy efficiency based LB without backhaul
	<sup>29</sup> , 2017	Yes	No	No	No	No	BS queue aware association with PRB scheduling and mobility
	<sup>27</sup> , 2017	No	No	Yes	No	No	Traffic offloading for effective capacity maximization
	<sup>13</sup> , 2018	Yes	No	No	No	No	User behavior aware cell association to reduce handovers
	<sup>8</sup> , 2018	No	No	Yes	No	No	Joint optimization of CIOs, transmission power and antenna tilts
	<sup>11</sup> , 2018	No	No	Yes	No	No	User association and user scheduling for DL rate maximization
Proactive	<sup>30</sup> , 2013	No	No	No	No	No	Proactive vertical handovers initiation for call admission control
	<sup>23</sup> , 2014	No	No	Yes	Yes	No	Proactive content caching to offload SBS traffic
	<sup>31</sup> , 2017	No	No	No	No	No	Proactive user association procedure for macro and SBS
Proposed, 2019	Yes	No	Yes	Yes	Yes	Leveraging mobility and caching jointly for user QoE optimization	

Abbreviations: CIO, cell individual offset; CRE, cell range expansion; DL, downlink; UE, user entity; LB, load balancing; QoE, quality of experience; PRB, physical resource block; SBS, small base station.

information. Besides, there exist few contributions that considered mobility and caching together such as users' mobility-aware caching and user-association<sup>22</sup> and user behavior aware cell-association.<sup>13</sup> The focus of these studies was to reduce mobility handovers and reactive load balancing instead of proactive content caching for better LB gains and QoE optimization. Gholami et al<sup>27</sup> proposed a reactive traffic offloading approach for effective capacity maximization in the HetNets exploiting their user association and power allocation algorithm. However, this work ignores content caching and users' mobility information.

**Proactive load balancing:** PLB can improve the user's QoE through timely initiating the offloading procedure. Very few efforts have been made for proactive offloading as evident from Table 1. To that end, we review existing proactive techniques. Ma and Ma<sup>30</sup> proposed a new proactive load balancing mechanism to initiate vertical handovers prior to admitting call if network resources are not substantial. Proactivity of this work concerns user-cell association policy only and lacks consideration of users' mobility and content demands. The European Telecommunications Standards Institute, along with third-generation partnership project (3GPP), has documented the standardization of multiaccess edge computing (MEC).<sup>32</sup> Recently, some efforts have been made by considering ETSI-recommended MEC architectures for traffic offloading. Moreover, MEC supports content caching for traffic offloading at near proximity of mobile UE such as caching at BS and at UEs' local cache.<sup>33</sup> Jin et al proposed a new elastic virtual content placement algorithm that exploits the MEC paradigm for prefetching the popular contents at the edge server.<sup>34</sup> However, this work ignores users' content preferences and mobility information. Dinh et al<sup>35</sup> introduced a MEC-based dynamic edge caching framework that predicts users' content demands based on blockchain transactions and nonnegative matrix factorization. Their caching framework exploits users' access history to predict their future contents while preserves private data. However, caching framework

pays no attention to users' mobility history. Al Ridhawi et al<sup>36</sup> proposed a MEC-based content caching framework, in which cloud load is shared through edge nodes by data decomposition and replication process. The framework caches most frequently demanded files at the edge nodes and lacks users' mobility modelling. Tang et al<sup>37</sup> proposed a user-centric contents delivery mechanism that exploits joint coordination of SBSs for proactive content caching. The approach considers cache storage capacity and available channel bandwidth as optimization constraints while forgets to include users' mobility information. Park and Song<sup>38</sup> leverages software defined networking and introduced a cooperative base station caching scheme to offload traffic from video streaming contents. Nonetheless, this work ignores incorporation of users' mobility. Said et al<sup>24</sup> proposed a proactive content caching algorithm for device-to-device enabled networks without taking into account of mobility. Similarly, Bastug et al<sup>23</sup> proposed proactive content caching for data offloading at SBSs. This scheme caches only the most popular files rather than users' demand based files. In addition, besides their proactive offloading claim, this work fails to provide any insights into LB gains and also, lacks user mobility information. The work mentioned in the work of Ichkov et al<sup>31</sup> also claims proactive user offloading, but they provide proactiveness at the user-cell association level only. This technique has very less margin for the system to reserve resources in advance because the user-association decision is up to *milliseconds* to *seconds* level decision.

Precisely, none of the above reactive and proactive LB schemes consider users' mobility and content demand information jointly for proactive traffic offloading, which can be exploited to improve users' QoE and cell loads optimization. In addition, under the complex and massive cell deployment such as in HetNets, these inefficient reactive LB policies will become a bottleneck that hinders network to fulfill the seamless connectivity goals. Therefore, current LB approaches fall short of the mark for existing and future high data rate and low latency demanding applications and time-critical machine-to-machine processes due to the following prominent limitations.

- Reactive traffic offloading operates when one or more cells have already become overloaded.
- Traffic offloading valid under stationary or low mobility conditions only.
- Reactive LB schemes are unable to answer *When* to offload traffic from congested cells but instead can provide the answer for *How* and *Where* to offload traffic.
- Caching-based traffic offloading schemes (eg, see the work of Bastug et al<sup>23</sup>) ignore user mobility information and also fail to address important questions *What*, *When*, and *Where* to cache simultaneously.

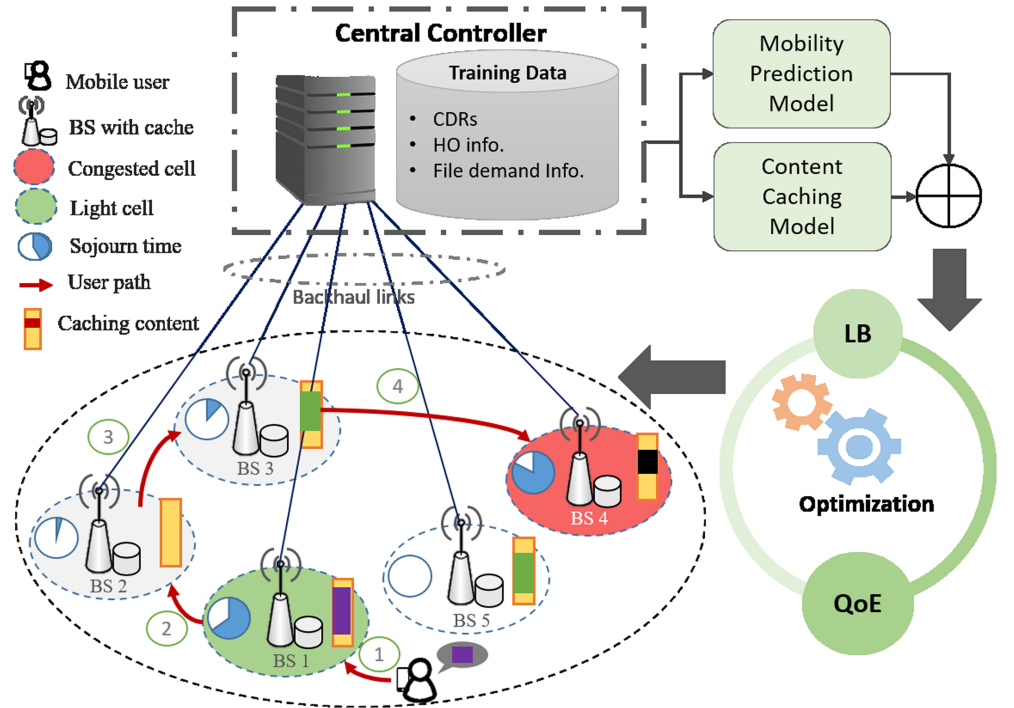
### 3 | PROACTIVE LOAD BALANCING FRAMEWORK

In this section we present the proposed PLB framework in detail. Our PLB framework consists of a central controller and several base stations (BSs) as shown in Figure 2. The central controller keeps a record of meaningful information from each user such as its current connection ID, handover time with the next cell, and file demand statistics, eg, file type and the number of requests against the particular file. The central controller and all BSs are connected through backhaul links. Base stations are leveraged with cache memory storage to cache user's content demands at the network edge. Whenever there is a request for a particular file at a BS, content is directly delivered to the user without consuming backhaul bandwidth if that content is already cached at BS cache; otherwise, content is fetched from the core network. Specifically, our PLB framework possesses the following key components:

- Mobility prediction model to predict users' future cells;
- Caching model to estimate users' future content demands;
- Joint coordination of mobility and caching models for proactive content offloading.

#### 3.1 | System model

We consider the HetNet supervised by a central controller consisting of densely deployed  $B$  cells,  $b = \{1, 2, \dots, B\}$ , and each cell  $b$  is equipped with the local cache memories. In our system, UEs are also enabled with limited cache memories. We also assumed that UEs and BSs placement follow Poisson point process distribution with  $\Omega_u$  and  $\Omega_b$  densities, respectively. It is assumed that there is no coverage overlap between MBSs, and a user can associate to one cell at a time. Figure 3 shows a mobile user trajectory in Voronoi cells. Referring to Figure 3,  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$  represent user sojourn time within consecutive cells along the user trajectory, and they are identically and independently distributed. Before proceeding further, we define these entities. Let  $u = \{1, 2, \dots, U\}$  represent the number of UEs in the system.  $\mathcal{C}$  and  $\mathcal{W}$  denote system bandwidth and bandwidth of a PRB, respectively.  $\mathcal{P}_b$  is the transmission power of BS  $b$ , and  $H_{bu}$  is channel gain between



**FIGURE 2** Proposed proactive load balancing (PLB) framework. BS, base station; QoE, quality of experience

**TABLE 2** Summary of notations

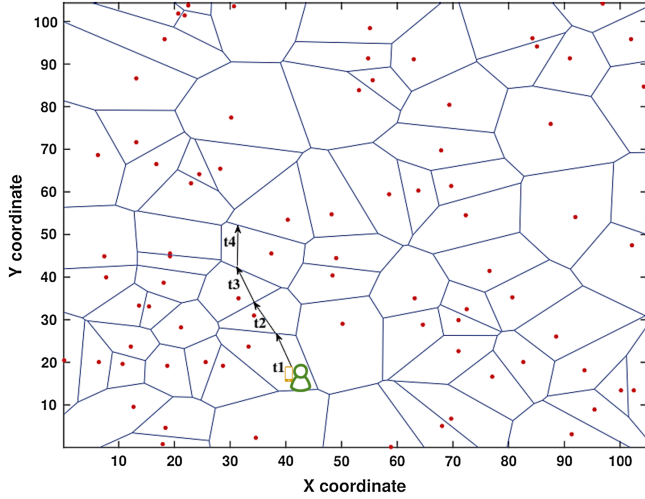
Notation	Description
$b$	Cell (BS)
$\mathcal{C}$	System bandwidth
$\mathcal{W}$	Bandwidth of a PRB
$\mathbf{v}$	Backhaul link capacities vector
$f$	Content file
$\eta_{bu}$	signal to interference and noise ratio (SINR)
$r_{bu}$	Downlink achievable rate
$x_{bu}$	User association indicator
$l_b$	Cell $b$ traffic load
$L_t h$	Cell load threshold
$\Phi_{i,j}$	Probability of transition from $i$ th to $j$ th cell for the user $u$
$p_{i,j}$	Handover probability from cell $i$ th to $j$ th for user $u$
$\theta_{i,j}$	Sojourn time distribution in $i$ th cell when the next cell is $j$ for user $u$
$N_{i,j}$	For user $u$ the number of handovers from cell $i$ to $j$
$N_i$	Total number of handover from cell $i$ for user $u$
$N_{i,j,k}$	Number of handovers with sojourn time equal to or less than $k$
$\mathbf{u}_b$	Vector containing IDs of UEs moving towards congested cell
$\mathbf{C}_b$	Cache decision matrix at cell $b$
$c_{u,f}$	File $f$ is cached for the user $u$

Abbreviations: UE, user entity; PRB, physical resource block.

BS  $b$  and user  $u$ . System has a library of  $F$  content files, which are indexed as  $\mathbf{f} = [f_1, f_2, \dots, f_F]$ . Each file is atomic and has size  $e_i$ , where files size vector is denoted as  $\mathbf{e} = [e_1, e_2, \dots, e_F] \in \mathbb{Z}^+$ . Backhaul link capacities between central controller and small cells are defined as  $\mathbf{v} = [v_1, v_2, \dots, v_B] \in \{0, \mathbb{Z}^+\}$ . Table 2 describes key notations. In this paper, the term BS and cell are interchangeably used.

Under the above system description, when a UE  $u$  is associated with BS  $b$ , its SINR is defined as

$$\eta_{bu} = \frac{P_b \mathcal{H}_{bu}}{\sum_{j \in B/\{b\}} P_j \mathcal{H}_{ju} + \gamma^2}, \quad (1)$$



**FIGURE 3** Illustration of mobile users' trajectory through Voronoi cells

where  $\gamma^2$  is noise power of each PRB for UE  $u$ . For a given  $P_b$ , the maximum achievable DL rate per PRB from BS  $b$  is given by using Shannon capacity theorem

$$R_{bu} = \mathcal{W} \log_2 \left( 1 + \frac{P_b \mathcal{H}_{bu}}{\sum_{j \in B \setminus \{b\}} P_j \mathcal{H}_{ju} + \gamma^2} \right), \quad (2)$$

where  $\mathcal{W}$  is typically  $180\text{kHz}$  in an orthogonal frequency division multiple access-based system and  $\gamma^2$  is additive white Gaussian noise. If UE  $u$  is connected to BS  $b$ , then association indicator,  $x_{bu}$  will be 1, otherwise 0. Moreover, load of the user  $u$  on the cell  $b$  can be defined as  $y_{bu} = \frac{\delta_u}{R_{bu}}$ , where  $\delta_u$  is user practical rate and  $R_{bu}$  is maximum achievable rate according to (2). Now, utilizing  $x_{bu}$  and  $y_{bu}$  of all active UEs, load of the cell  $b$  at time  $t$  is expressed as

$$l_b(t) = \sum_{u \in \mathcal{U}} x_{bu} y_{bu} \quad \forall u \in \mathcal{U}. \quad (3)$$

The term  $l_b(t)$  represents the number of active users in cell  $b$  at time  $t$ . Similarly, at time  $t$  system load vector is given as  $\mathbf{L}(t) = [l_1, l_2, \dots, l_B]$ . Practically, the available DL capacity offered by a cell is equally shared among all associated users; therefore, perceived effective DL rate not only depends upon channel conditions (interference, pathloss, etc) but also on current cell load.<sup>12</sup> Therefore, perceived DL rate can be normalized with cell load, and can be expressed as  $r_{bu} = R_{bu}/l_b(t)$ . Let the vector  $\mathbf{s} = [s_1, s_2, \dots, s_B] \in \{0, \mathbb{Z}\}$  denote cell storage capacities that can provide information coming from the central controller to associated UEs over wireless links with the rates  $r_{b,u}$ . Let  $\mathbf{R}$  be the matrix that represents the rate at user  $u$  from the cell  $b$ , and it is given by

$$\mathbf{R} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_B \end{bmatrix} = \begin{bmatrix} r_{1,1} & \dots & r_{1,U} \\ r_{2,1} & \dots & r_{2,U} \\ \vdots & \vdots & \vdots \\ r_{B,1} & \dots & r_{B,U} \end{bmatrix} \in \{0, \mathbb{Z}^+\}^{B \times U}. \quad (4)$$

### 3.2 | Mobility prediction model

Central controller enabled with mobility management entity exploits user trajectory information to learn and track mobility patterns over the time and constructs users' mobility profile. Mobility profile for a user  $u$  within discrete state space  $b = \{1, 2, \dots, B\}$  is modeled using Semi-Markov renewal process,  $\{X_b, T_b : b \geq 0\}$ , as described in the work of Farooq and Imran,<sup>39</sup> where  $X_b$ ,  $T_b$ , and  $B$  represent the state of  $b$ th transition, the time of  $b$ th transition and total cells, respectively. Each cell (MBS or SBS) represents the state of this Semi-Markov process, and a handover from one cell to another

cell is considered as state transition. Here, we define the probability of transition from  $i$ th to  $j$ th cell for the user  $u$ , which already spent time  $t$  in the  $i$ th cell

$$\begin{aligned}\Phi_{i,j}(t) &= Pr(X_{b+1} = j, T_{b+1} - T_b \leq t | X_b = i) \\ &= p_{i,j} \theta_{i,j}(t),\end{aligned}\quad (5)$$

where

$$p_{i,j} = \lim_{t \rightarrow \infty} \Phi_{i,j}(t), = Pr(X_{b+1} = j | X_b = i), p_{i,j} \in \mathbf{P}_u \quad (6)$$

and

$$\theta_{i,j}(t) = Pr(T_{b+1} - T_b \leq t | X_{b+1} = j, X_b = i), \quad (7)$$

where  $p_{i,j}$  represents handover probability from  $i$ th to  $j$ th cell for each user  $u$  and  $\theta_{i,j}(t)$  denotes the sojourn time distribution in  $i$ th cell when the next cell is  $j$ . For each user  $u \in U$ , central controller constructs probability transition matrix  $\mathbf{P}_u$  and sojourn time distribution matrix  $\Theta_u$ , which are described in the work of Farooq and Imran<sup>39</sup> and given as

$$\mathbf{P}_u = \begin{bmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,B} \\ p_{2,1} & p_{2,2} & \dots & p_{2,B} \\ \cdot & & & \\ \cdot & & & \\ p_{B,1} & p_{B,2} & \dots & p_{B,B} \end{bmatrix} \quad (8)$$

$$\Theta_u = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \dots & \theta_{1,B} \\ \theta_{2,1} & \theta_{2,2} & \dots & \theta_{2,B} \\ \cdot & & & \\ \cdot & & & \\ \theta_{B,1} & \theta_{B,2} & \dots & \theta_{B,B} \end{bmatrix} \quad (9)$$

If we have past handover history (handover-time, cell-ID) of user  $u$ , probability transition matrix  $\mathbf{P}_u$  and sojourn time distribution matrix  $\Theta_u$  can be initialized<sup>39</sup> and given by

$$p_{i,j} = \frac{N_{i,j}}{N_i} \quad (10)$$

$$\theta_{i,j}(k) = \frac{N_{i,j,k}}{N_{i,j}}, \quad (11)$$

where  $N_{i,j}$  represents user  $u$ 's number of handovers from cell  $i$  to  $j$ ,  $N_i$  is the total number of handover from cell  $i$  for user  $u$ , and  $N_{i,j,k}$  is the number of handovers with sojourn time  $t_n \geq k$  from cell  $i$  to  $j$ . Based on previous history of  $p_{i,j}$  and  $\theta_{i,j}$  for each user  $u$ , after every  $t'$  time steps, central controller generates (time, cell-ID) tuple, ie,  $\{T_{HO}, b_{NC}\}$ , where  $T_{HO}$  is next handover time and  $b_{NC}$  is the future cell.

Finally, future BS association  $x_{bu}$ , can be determined using users' future cell information. At time  $t + t'$ , user-cell association matrix  $\mathbf{X}$  can be updated and given as follows:

$$x_{bu} = \left\{ \forall u \in \mathcal{U} | b = \underset{\forall b \in \mathcal{B}}{\mathbf{argmax}} \mathcal{P}_b / l_b \right\} \quad (12)$$

$$\mathbf{X}(t + t') = \begin{bmatrix} x_{1,1} & \dots & x_{1,U} \\ x_{2,1} & \dots & x_{2,U} \\ \vdots & \vdots & \vdots \\ x_{B,1} & \dots & x_{B,U} \end{bmatrix} \in \{0, 1\}^{B \times U}, \quad (13)$$

where (12) represents cell-load aware user-cell association policy. Now, exploiting future user-cell association information future cell loads can be calculated using (3) for time  $t + t'$ . At time  $t + t'$ , to determine whether a cell is congested or not,



it can be estimated using following expression:

$$\varpi_b = \begin{cases} 1, l_b \geq L_{th} \\ 0, l_b < L_{th} \end{cases} \quad \forall b \in B, \quad (14)$$

where  $L_{th}$  is the cell load threshold,  $L_{th} \in (0, 1]$ ; moreover,  $L_{th}$  can be defined by operator based on QoE demand level from the users such as minimum DL rate must be greater than  $\mathbf{JKbps}$ , etc. Moreover,  $\varpi$  is a vector that contains the list of light loaded and congested cells with (0, 1) entries, denoted by  $\varpi = [\varpi_1, \varpi_2, \dots, \varpi_B]$ . By predicting future cell loads, central controller will be able to prepare lightly loaded cells to reduce congestion at fully loaded cells using proactive content caching. From central controller, each cell can get a vector  $\check{y}_b$ , which contains the IDs of UEs moving from lightly loaded cell  $b$  at time  $t$  and will be at congested cell  $b_{NC}$  at time  $t + t'$ .

### 3.3 | Caching model

Central controller can exploit users' file request information to construct content demand profile matrix  $\mathbf{D}$ . Let a user  $u$  request a content item  $f$  from file library  $\mathbf{f}$ . Central controller tracks this demand statistics to rate the demanded files according to the number of times a particular file is requested (such as the most requested file will be rated maximum value, while least demanded or not demanded file will be rated minimum or zero). Therefore, there exist  $U$  users and  $F$  files at the central controller; thus, matrix  $\mathbf{D}$  can be stated as

$$\mathbf{D} = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_U \end{bmatrix} = \begin{bmatrix} d_{1,1} & \dots & d_{1,F} \\ d_{2,1} & \dots & d_{2,F} \\ \vdots & \vdots & \vdots \\ d_{U,1} & \dots & d_{U,F} \end{bmatrix} \in \{0, \mathbb{Z}\}^{U \times F}. \quad (15)$$

Practically,  $F \gg U$ , as every user cannot ask each file from the library  $\mathbf{f}$ ; therefore, central controlled can rate only those files which were ever demanded by users. Thus, matrix  $\mathbf{D}$  will be a sparse matrix with a large number of zeros. To cache users' future demands proactively, there is a need to estimate full demand matrix  $\bar{\mathbf{D}}$ . Without loss of generality, we assume that each user has already spent some time in the system and requested some files from the network; therefore, the central controller is able to construct its demand profile. Moreover, it is supposed that  $\bar{\mathbf{D}}$  matrix will be a low-rank matrix. Finally, the central controller formulates an optimization problem to learn the relationship between a user and its demanded files in order to rate nonrated entries in the  $\mathbf{D}$  matrix, which is given below

$$\begin{aligned} \min_{\bar{\mathbf{D}}} \quad & \log \det ((\bar{\mathbf{D}}^T \bar{\mathbf{D}})^{1/2} + I) \\ \text{s.t.} \quad & \bar{D}_{uf} = D_{uf}, \quad (u, f) \in \mathfrak{O}, \end{aligned} \quad (16)$$

where  $\mathfrak{O}$  is set of observed entries and  $I$  is identity matrix. Moreover, function  $\log \det$  is tighter rank approximation than nuclear norm and is defined  $\log \det((\bar{\mathbf{D}}^T \bar{\mathbf{D}})^{1/2} + I) \leq \|\bar{\mathbf{D}}\|_*$ . The aforementioned problem can be solved using different algorithms such as nuclear norm minimization solution<sup>40</sup> or through least-square minimization.<sup>41</sup> However, an efficient algorithm, defined in the work of Kang et al<sup>42</sup> is exploited to learn missing user-file ratings in the original matrix  $\mathbf{D}$ . Thus, complete user-item based demand matrix is given by

$$\bar{\mathbf{D}} = \begin{bmatrix} \bar{D}_1 \\ \bar{D}_2 \\ \vdots \\ \bar{D}_U \end{bmatrix} = \begin{bmatrix} \bar{d}_{1,1} & \dots & \bar{d}_{1,F} \\ \bar{d}_{2,1} & \dots & \bar{d}_{2,F} \\ \vdots & \vdots & \vdots \\ \bar{d}_{U,1} & \dots & \bar{d}_{U,F} \end{bmatrix} \in [0, 1]^{U \times F}. \quad (17)$$

Inspired by the findings of an empirical study, given in the work of Zoha et al,<sup>43</sup> we aim to reserve system resources in advance when congestion is expected to happen in the future. Cell loads follow temporal fluctuations; exploiting this information can help the system to enable proactive response. Following this, our ultimate objective is to cache future

demands of those users that are currently residing in lightly loaded cells at time  $t$ , and their next destination at time  $t + t'$  is a congested cell. The motivation behind this aim is that these lightly loaded cells can download a large amount of users' future data from the central controller without degrading QoE of associated users. Therefore, the central controller can proactively cache most expected demand files ( $Top - N$ ), represented by  $n \in \lambda = 1, 2, \dots, N$ , from the users by sorting  $\bar{D}$  in the decreasing order. Eventually, the central controller can transfer these files to lightly loaded cells during off-peak hours; the factor  $N$  depends on the cell storage capacity  $s_b$ . Now, at each lightly loaded cell  $b$ , there will be caching decision matrix  $\mathbf{C}_b$ , which contains these  $Top - N$  files for each user

$$\mathbf{C}_b = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_U \end{bmatrix} = \begin{bmatrix} c_{1,1} & \dots & c_{1,F} \\ c_{2,1} & \dots & c_{2,F} \\ \vdots & \vdots & \vdots \\ c_{U,1} & \dots & c_{U,F} \end{bmatrix} \in \{0, 1\}^{U \times F}, \quad (18)$$

where  $c_{u,f} = 1$  means the file  $f$  is cached for the user  $u$ .

### 3.4 | Proactive content offloading for load balancing

This section covers joint coordination of mobility and caching models. Leveraging Equation (18), each lightly loaded cell contains  $\mathbf{C}_b$  files to offload contents proactively. Herewith, these  $\mathbf{C}_b$  files need to be transferred to users before they leave the current cell  $b$ . Thus, files delivery time is bounded by users' sojourn time  $\theta_{b,b_{NC}}(t)$  (start of sojourn time,  $t_{so1}$  and end of sojourn  $t_{so2}$ ) with the achievable DL rate  $r_{bu}$ . Suppose we have complete future contents of user  $u$ ,  $\lambda_u \subseteq \bar{D}$ , at time  $t$  which he/she will be demanding at time  $t + t'$ ; then, the satisfaction ratio is given by

$$\zeta_u(\lambda) = \frac{1}{N} \sum_{n \in \lambda} \mathbf{Y} \left\{ \frac{e_n}{t_{so2} - t_{so1}} \geq r_{b,u} \right\}, \quad (19)$$

where  $e_n$  is length of  $n$ th file and  $\mathbf{Y}\{ \dots \}$  is indicator function, which returns 1 if statement holds otherwise 0. An optimization problem can be formulated in order to maximize the file satisfaction ratio  $\zeta_u$  at cell  $b$  for each user  $u$ , which is given by

$$\begin{aligned} \max_{\theta_{b,b_{NC}}, r_{b,u}} \zeta_u(\lambda) &= \frac{1}{N} \sum_{n \in \lambda} \mathbf{Y} \left\{ \frac{e_n}{t_{so2} - t_{so1}} \geq r_{b,u} \right\} \\ \text{subject to: } \mathbf{o} &\leq o^{\max}, \\ r_{b,u} &\leq R^{\max}, \end{aligned} \quad (20)$$

where  $o^{\max}$  is capacity constraints of user storage space and  $R^{\max}$  is maximum achievable DL rate from cell  $b$  to user  $u$ . For each user  $u$ , even if we know exact  $t_n$ , sojourn time at cell  $b$ , and expected demands files  $\lambda_u$ , all the downloading file request cannot be fulfilled due to DL rate  $r_{b,u}$  and user storage capacity  $\mathbf{o}$  constraints. Employing brute-force search would be hard due to combinatorial nature of this problem. Therefore, to reduce search space we can employ sojourn time threshold  $\beta$ , to select only those UEs which have sojourn time,  $t_n \geq \beta$ . Now, Algorithm 1 can be used to transfer  $\mathbf{f}$  files to each UE from a BS  $b$ . Algorithm 1 caches files with the highest popularities until the storage capacity  $\mathbf{o}$  of the cells are achieved. Basically, it sorts the files based on their popularities in the descending order then start caching files until the cache is filled. Hence, this algorithm maximizes our objective function by storing the files based on their popularities and cell cache size.

**Algorithm 1** Proactive content caching

---

```

1: Input: caching decision matrix at cell  $C_b$ , vector of lightly loaded cells  $B = \|\bar{\omega}\|$ , user-cell association matrix  $\mathbf{X}$ ,
   caching files  $\mathbf{f}$ , file size  $\mathbf{e}$ , cell cache storage vector  $\mathbf{o}$ 
2: Initialize  $\mathbf{S} \leftarrow \mathbf{0}_{U \times F}$ ,  $\bar{o} \leftarrow \mathbf{0}_{B \times 1}$ 
3: for  $b = 1, \dots, B$  do
4:   Get  $\mathbf{q}$ 
5:   Get files matrix of UE  $\|\mathbf{q}\|$ ,  $\mathbf{G} = \mathbf{C}_b(\mathbf{q}, :)$ 
6:    $U = \|\mathbf{q}\|$ 
7:   for  $u = 1, \dots, U$  do
8:      $[\mathbf{v}, \mathbf{i}] \leftarrow \text{SORT}(g_u)$ ,           Sorting files in descending order
9:      $F = \|\mathbf{i}\|$ 
10:    for  $f = 1, \dots, F$  do
11:       $k \leftarrow i_f$ ,           Gets index of most popular file
12:      if  $e_k + \bar{o}_u \leq o_u$  then
13:         $S_{u,f} \leftarrow 1$ , Sets elements of user cache matrix to 1
14:         $\bar{o}_u \leftarrow \bar{o}_u + e_f$ , Increase current storage size
15:      else
16:        break,           Stops if storage reaches maximum capacity
17:      end if
18:    end for
19:  end for
20: end for

```

---

The proposed PLB framework utilizes mobility prediction and content caching models jointly to optimize cell loads by caching users' future contents in the lightly loaded cells if their next probable destination is a congested cell. Figure 2 shows PLB framework with five cells. The central controller contains two binary vectors  $\bar{\omega}$  and  $\omega$  that represent the loads of lightly loaded cells and congested cells respectively. Moreover, at each BS  $b \in \bar{\omega}$ , there will be a vector  $\mathbf{q}$  that contains IDs of those users, which are currently at lightly loaded cell  $b$  but after  $t + t'$  time will be at cell  $b' \in \omega$ . From Figure 2, cell 4 (red colored) is congested, while cell 1 (green colored) is lightly loaded; a UE  $u$  is moving from cell 1 to 4. Central controller can estimate user  $u$  future cell using (5) and predict that user will be at cell 5 after time  $t + t'$ . Therefore, at time  $t$ , cell 1 can prefetch future contents  $\lambda_u$  of user  $u$  from the central controller and transfer into the local cache of  $u$  using Algorithm 1 before leaving the current cell.

Algorithm 2 provides detailed implementation procedure of the proposed PLB framework. Here is the step-by-step description of Algorithm 2. Firstly, central controller utilizes past handover history, ie, cell-ID and sojourn time data to build mobile user's mobility profile. Based on the mobility information, users' next cells are determined using (5). After that, a vector  $\tilde{u}_b$  of user IDs, which are currently residing in the lightly loaded cells at time  $t$  and their next destination at time  $t + t'$  is a congested cell  $b_{NC}$ , is determined. Then, central controller builds demand profile matrix  $\bar{D}$  for all users and future user association matrix  $\mathbf{X}$ ; if the next cell  $b_{CN}$  of user  $u$  is congested cell, then central controller guides lightly loaded cells, ie,  $l_b \leq L_{th}$ , to cache future demand files of  $u$  and push to  $u$ 's local cache using Algorithm 1. In Algorithm 1, assuming that initialization step complexity is  $O(1)$  and sorting step has  $O(F \log F)$ , ie, using *Timsort*, if  $B$  the total number of cells in the system, then complexity of caching algorithm becomes  $O(BF \log F)$ , which is depending upon number of cells  $B$  and number of files  $F$ .

## 4 | PERFORMANCE EVALUATION

We compared the performance of our proposed framework with Max-RSRP and CIO-induced offloading scheme in terms of cell load fairness, user satisfaction, system gain, and average DL rates. The details of the experimental procedure and results are provided in the next sections.

**Algorithm 2** Proactive load balancing implementation

---

```

1: Input: user handover time and cell-ID tuple  $\{T_{HO}, b_{NC}\}$ , No. of users  $U$ , No. of cells  $B$ , cell load threshold  $L_{th}$ 
2: for  $u \in U$  do
3:   Determine future cell using (5)
4:   Future user-Cell association  $X$  using (12)
5: end for
6: for  $b \in B$  do
7:   Get new cell loads,  $l_b(t + t')$  using (3)
8:   Get vector of lightly loaded cells  $\omega_b$  using (14)
9: end for
10: for  $b \in B$  do
11:   for  $u \in U$  do
12:     If  $\omega_b > L_{th}$ 
13:        $\check{y}_{b,u} \leftarrow 1$ ,   if cell load is greater than threshold then it is congested cell
14:     else
15:        $\check{y}_{b,u} \leftarrow 0$ ,   if cell load not is greater than threshold then it is lightly loaded cell
16:     end if
17:   end for
18: end for
19: for  $u \in U$  do
20:   Estimate  $\bar{D}$  through solving optimization problem (16)
21:   Get top files  $\lambda_u$ 
22: end for
23: for  $b \in B$  do
24:   Get caching matrix  $C_b$ 
25:   Invoke Algorithm 1 to cache files
26:   Calculate  $l_b = \sum_{b \in B} x_{bu} y_{bu}$ 
27: end for

```

---

## 4.1 | Simulation setup

We consider a typical HetNet consisting macro- and small-cell-based network with UE distributions using LTE 3GPP standard compliant<sup>44</sup> network topology simulator in MATLAB.<sup>45</sup> We exploited mobility management entity and eNodeBs integration as described in the 3GPP release.<sup>46</sup> In this release, the E-UTRAN architecture consists of a set of eNodeBs connected to the evolved packet core through the S1 interface. We used a LTE-based simulator that has been used in the previously published studies.<sup>8,47</sup> In simulations, the HetNet model has one MBS with 46 dBm transmit power, three SBS with 30 dBm transmit power each, and one small cell per sector. Moreover, UEs and SBS are uniformly placed. We consider  $|U| = 200$  UEs in the system. The distance dependent pathloss model is given by  $PL(d) = 128.1 + 37.6 \log_{10}(d)$  and  $PL(d) = 140.1 + 36.7 \log_{10}(d)$  for MBS and SBS, respectively, where  $d$  is distance between UE and BS in meters. Log-normal shadowing with standard deviation of 8 dB and noise power spectral density of 174 dBm/Hz is considered. System bandwidth 20 MHz is used. Moreover, a summary of the simulation parameters are listed in the Table 3. In all scenarios, UEs are video users with 1024 kbps desired throughput. Without loss of generality, prediction interval  $t'$  is set to 1 minute for all simulations. Any UE can request a file from the set of  $|F|$  library files. Moreover, backhaul links capacity  $v_b = 5$  Mbps is considered. We simulated the framework on a 64-bit Windows desktop machine with two 3.40 GHz CPUs and 16 GB RAM.

## 4.2 | Mobility and caching data

In order to realistically evaluate the performance of the proposed PLB framework, we utilized real-world datasets for mobility and caching models. We used a publicly available user mobility dataset<sup>48</sup> for the training of our mobility model. This data was collected using a Tower logger software operating at mobile terminals (ie, UEs). The data from 10 participants were collected from 3 to 6 weeks. Table 4 summaries important features of the mobility dataset.<sup>48</sup> The data contain

System parameters	Value
Number of MBS	1
Sectors per MBS	3
SBS per MBS sector	1
Transmission bandwidth	20 MHz
MBS Tx power	46 dBm
SBS Tx power	30 dBm
Cellular system standard	LTE
MBS height	25 m
SBS height	10 m
File library size, $ F $	1682

TABLE 3 Simulation parameters

Abbreviations: MBS, macro base station; SBS, small base station.

Name	Source	No. of users, $ U $	Acquisition duration	Important attributes
Mobility dataset	<sup>48</sup>	10	3 to 6 weeks	Time-stamp, cell-ID
Caching dataset	<sup>49</sup>	943	7 months	User-ID, movie-ID, rating

TABLE 4 Statistics of mobility and caching datasets used for performance evaluation

information such as the date, current time (hours, minutes, seconds), the cellular tower identity number (Cell-ID), and signal strength. Among these, we only considered the following information:

- Date
- Current time (hours, minutes, seconds);
- Cellular tower identity number (Cell-ID).

Before assessing the performance of our mobility model, necessary data preprocessing is performed. During the preprocessing stage, we performed two major steps. In the first step, ping-pong entries are removed because data contains more than one cellular tower coverage region for all users. Therefore, mobility logs of each participant were preprocessed to remove such entries as has been done in the work of Lee and Hou.<sup>50</sup> In the second step, we plotted and analyzed the most associated cell-IDs (BS) and selected only top six cell data for each user. After necessary preprocessing, we trained our mobility model on real traces of these 10 participants and map their mobility profiles to users defined in the simulations environment. Thus, each user  $u \in U$  belongs to only one real mobile trace-based profile.

To depict actual user-file rating phenomena, other than simulation based data, we used a real world sparse data-set *ML100K* provided by *MovieLens*, which is publicly available.<sup>49</sup> Previously, this dataset has been used for recommendation system studies.<sup>41</sup> Recently, dataset *ML100K* has been used for content caching in the cellular network.<sup>51,52</sup> This dataset contains 100-000 ratings between (1-5) from 943 user on 1682 movies. Here, each movie is considered as a proxy for the files  $f$  from  $|F|$  library. Moreover, each user has rated at least 20 files (movie files). These data were collected through the *MovieLens* website (*movielens.umn.edu*). Table 4 explains important aspects of caching dataset. We used 70% of this data to train the caching model and obtained the user-file relation matrix while 30% data is used for the user's future content predictions.

### 4.3 | Results and discussion

We presented the effectiveness of proposed PLB framework with state of the art user-association (ie, Max-RSRP) and 3GPP recommended CIO-based offloading scheme. Impact of mobility and content prediction accuracy are studied to analyze system sensitivity and performance gains.

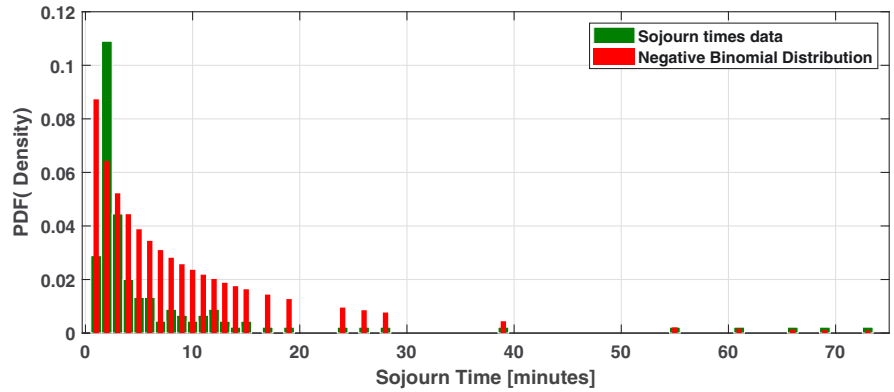
#### 4.3.1 | Mobility model prediction

For each user  $u \in U$ , its probability transition matrix  $P_u$  and sojourn time distribution matrix  $\Theta_u$  are estimated using Equations (10) and (11), respectively. Selected top six cells of the user10 are 516, 517, 521, 522, 528, and 599. Table 5 shows probability transition matrix  $P_u$  of user10 from the mobility data.<sup>48</sup> It is important to observe that as handover from cell to itself is not allowed, therefore, diagonal elements of the matrix  $P_u$  are all zero. User10 behavior can be studied from the Table 5 as it is clear that user10 has the highest probability of handover from cell 516 to 528, which is 0.6037, and from the cell 528 to 516, which is 0.6247. One of the cells can be either its home or office cell and vice versa. Similarly, the probability transition matrix  $P_u$  is computed for all the users  $U$ .

**TABLE 5** Transition probability matrix of the user 10. This matrix is estimated using real mobility traces dataset<sup>48</sup>

Cell-ID	516	517	521	522	528	599
516	0	0.0083	0.0062	0.3797	0.6037	0.0021
517	0.0833	0	0.4167	0.2500	0.2500	0.0000
521	0.1029	0.3529	0	0.1765	0.3676	0.0000
522	0.5223	0.0386	0.0326	0	0.4065	0.0000
528	0.6247	0.0405	0.0618	0.2708	0	0.0021
599	0.5000	0.0000	0.0000	0.0000	0.5000	0

**FIGURE 4** Mean sojourn times distribution of user 10 that depicts users' mobility profile in the cell 516 before moving to cell 528



In the next step, the mean sojourn time distribution matrix  $\Theta_u$  for each user is computed. Therefore, for each user  $u \in U$ , we computed its sojourn time distribution at cell  $i$  before moving to cell  $j$ . As a test case, Figure 4 shows user10 mean sojourn time distribution in the cell 516 before moving to cell 528. We used the Distribution Fitting Tool of MATLAB to get the best distribution fit which gives minimum Akaike Information Criterion (AIC).<sup>53</sup> In this case, best fitted distribution is negative binomial distribution with  $AIC_{nb} = 1.019e + 03$ . Next close distributions are gamma distribution with  $AIC_g = 1.019e + 03$  and Poisson distribution with  $AIC_p = 4.757e + 03$ . The mean sojourn time is 14.4126 minutes at cell 516 before moving to cell 528. Similarly, for all other cells, users' mean sojourn time distribution is computed. Finally, we computed the users' sojourn time distribution matrix  $\Theta_u$ .

The efficiency of mobility model is assessed through accuracy metric that is defined as follows:

$$Acc = \frac{N_c}{N_c + N_i} \quad (21)$$

where  $N_c$  and  $N_i$  are the number of correct and incorrect predictions, respectively. At each time interval  $t$ , when predicted future cell number for interval  $t'$  is same as actual future cell number, then prediction score is 1, otherwise 0. The larger the value of accuracy metric indicate better performance of the model.

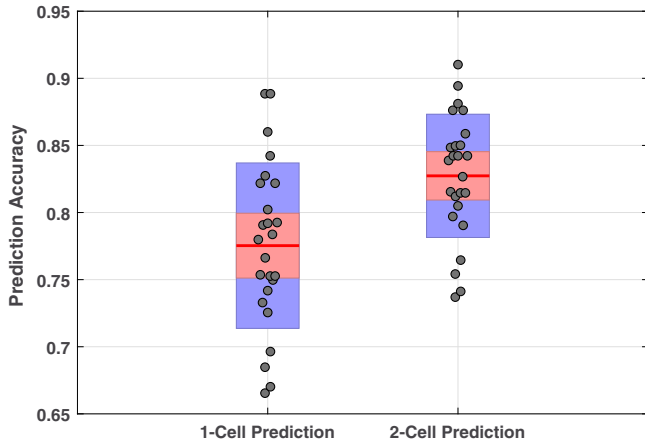
We calculated accuracy of mobility model using Equation (21). Figure 5 shows the next cell mobility prediction accuracy under (1) top-most one cell and (2) top-most two cells predictions. Under top 1-cell prediction, we achieved maximum 88.85% with mean of 77.53% accuracy. To improve the next cell prediction accuracy, we predicted next top 2-cells, and results are shown in the Figure 5. Under top 2-cell prediction, mobility prediction accuracy is improved with maximum 91.02% with mean of 82.73% accuracy. Reasonable accuracy gain in terms of the next cell prediction validates the performance of our mobility model.

### 4.3.2 | User satisfaction with proactive content downloading

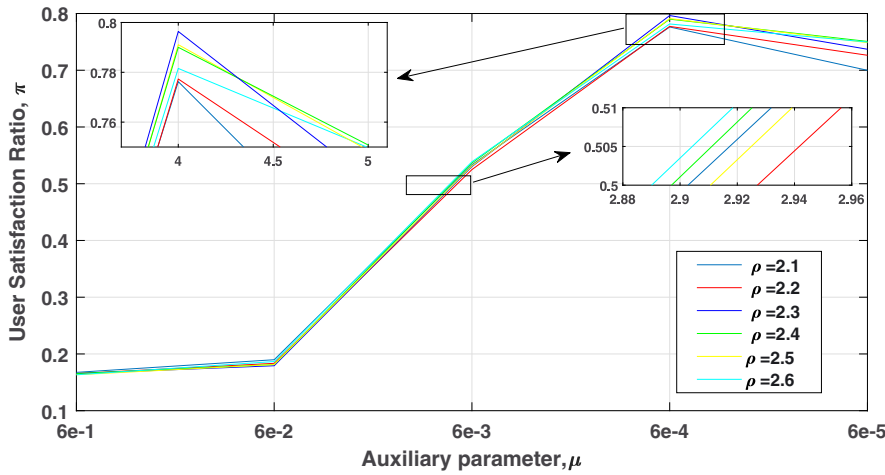
Caching model delivers users their future contents during their stay at lightly loaded cells. To evaluate the effectiveness of our caching model, there is a need to know the percentage of satisfied users at destination cells. Therefore, user satisfaction at congested cells with proactive content pushing, ie, *Top-N* files is defined as *User Satisfaction Ratio*,  $\pi$ , which is given as

$$\pi = \frac{\omega}{\phi}, \quad (22)$$

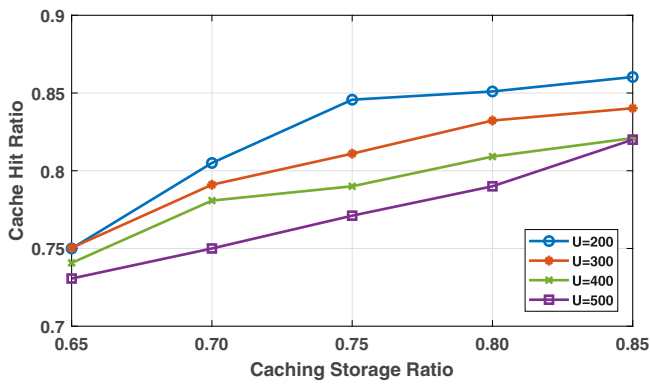
where  $\omega$  represents the number of users in the test set whose files lie in the *Top-N* and  $\phi$  is the total number of users in the test set. The algorithm defined in the work of Kang et al<sup>42</sup> depends upon two optimization parameters  $\rho$  and  $\mu$ .



**FIGURE 5** Next cell prediction accuracy under the proposed mobility model



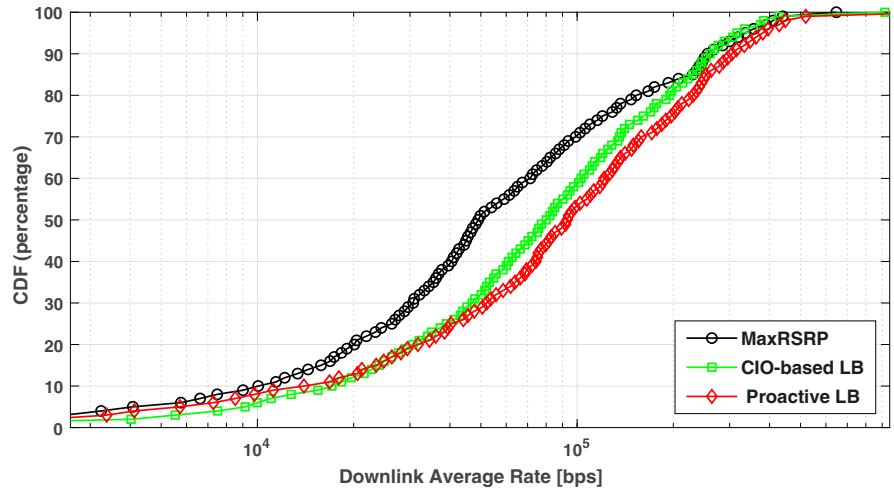
**FIGURE 6** User satisfaction ratio for proactive content demands at destination cells



**FIGURE 7** The overall cache hit ratio under varying the number of users in the system

Figure 6 shows user satisfaction with proactive content demands at the destination cells. With  $\rho = 2.3$  and  $\mu = 6 \times 10^{-4}$ , we achieved 0.80 user satisfaction ratio, which indicates that 80% of users in the test are satisfied with proactive contents, which are already in their cache.

The overall cache hit ratio is analyzed and presented in Figure 7. The number of users,  $|U|$ , are varied from 200 to 500 under different caching storage capacity ratio of [0.65, 0.70, 0.75, 0.80, 0.85]. With  $|U| = 200$ , we achieved 0.75 and 0.86 minimum and maximum cache hit ratio, respectively. Similarly, with  $|U| = 400$ , cache hit ratio follows its minimum at 0.737 and maximum at 0.821. From Figure 7, it can be observed that caching hit ratio improves with the increase in the caching storage size. However, the cache hit ratio is sensitive to the number of users in the system. With increase in the number of users, cache hit ratio decreases from its maximum value of 0.86 ( $|U| = 200$ ) to 0.821 ( $|U| = 400$ ).



**FIGURE 8** User quality of experience (QoE) gain under maximum reference signal received power (Max-RSRP), cell individual offset (CIO)-enabled load balancing (LB), and the proposed proactive load balancing (PLB) scheme

### 4.3.3 | User quality of experience gain

We used Equation (2) to calculate user  $u$ 's DL rate from the associated cell  $b$ . Our objective is to maximize the average DL of rates of all the users in the system. For this, we exploit rate distribution  $\Delta$ , which is the probability that the certain number of users receive a higher rate than a predefined threshold  $\chi$ , and given as

$$\Delta = Pr[R_{bu} \geq \chi | R_{bu} \geq 0]. \quad (23)$$

Figure 8 compares average DL rates (log-scale) under Max-RSRP, CIO-enabled, and PLB schemes. Under MaxRSRP, there is no offloading; therefore, cells 1, 2, and 3 are congested, while other cells are lightly loaded. Under CIO-based offloading with the implication of 10 CIO, some UEs from the cells 1, 2, and 3 are offloaded to cells 4, 5, and 6. Similarly, under the proposed PLB, 25, 11, and 13 UEs from cell number 1, 2, and 3 are proactively offloaded through providing their future contents at lightly loaded cells. Through this, these offloaded users are not further exerting load at associated cells, which are 1, 2, and 3. Eventually, average DL rates are improved under the proposed PLB framework. From Figure 8, Max-RSRP, CIO-enabled, and PLB are  $41.9 \times 10^3$ ,  $48.12 \times 10^3$ , and  $51.56 \times 10^3$  kbps, respectively. From these results, it is evident that DL rates per user are significantly improved under the proposed PLB scheme. This is due to the fact that the users that are coming to the congested cells have already offloaded with their future contents. Therefore, at congested cells, these users are not exerting extra load on the cell capacity; thus, all users (existing and arriving) are experiencing better DL throughputs.

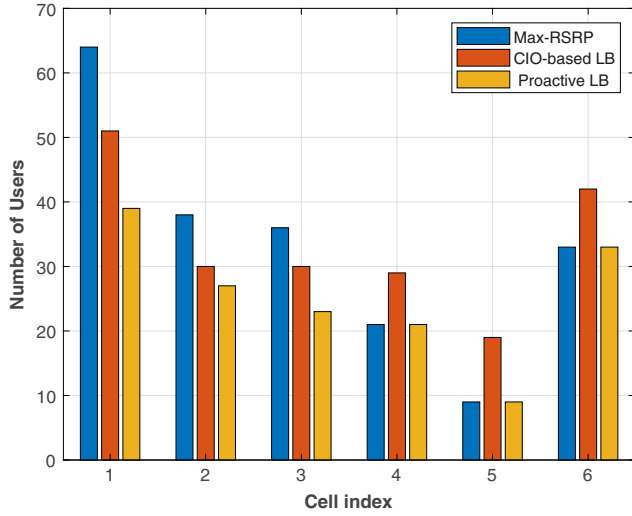
### 4.3.4 | Load balancing and average backhaul saved in the system

In this section, we study the performance of PLB framework in terms of cell load balancing and backhaul reduction in the system. Figure 9 shows cell loads Max-RSRP, CIO-based LB, and the proposed LB. Cell index 1, 2, and 3 are congested while cells 4, 5, and 6 are lightly loaded. Under Max-RSRP, there is no offloading, which under CIO-based LB offloading is reactive and under proposed LB offloading is proactive. From figure 9, it is clear that UE load distribution is uneven under Max-RSRP and CIO-enabled LB but nearly balancing number of UEs under proactive LB due to UEs profiling and data offloading at light cells rather than congested cells. Therefore, the proposed scheme not only offers LB but also creates smart capacity for upcoming traffic. Moreover, in order to measure cellular system load, we adopted Jain's fairness index.<sup>54</sup> Jain's fairness index has been used in many studies<sup>9,12</sup> to determine the fairness level among cell loads as LB indicator, and therefore, it is described as follows:

$$\zeta = \frac{(\sum_{b \in B} l_b)^2}{|B| \sum_{b \in B} (l_b^2)}, \quad (24)$$

where  $|B|$  is number of BSs and  $l_b$  is the load of BS  $b$ . Jain's fairness index ranges  $\zeta = [0, 1]$ , near to 1 means more fairness and vice versa. Fairness index  $\zeta$  for Max-RSRP, CIO-enabled, and proactive LB are 84.3, 91.05, and 98.7, respectively, which support our fairness LB argument under the proposed PLB scheme. Further, exploiting the proposed caching algorithm





**FIGURE 9** Cell loads distribution under maximum reference signal received power (Max-RSRP), cell individual offset (CIO)-enabled load balancing (LB), and the proposed proactive load balancing (PLB) scheme

No. of UEs in the system	Offloaded UEs	Backhaul saved, %
201	45	22.38
300	65	21.66
360	72	20.0
450	93	20.6
501	83	16.8

**TABLE 6** Percentage of backhaul load minimization through PLB framework

Abbreviations: UE, user entity; PLB, proactive load balancing.

for proactive LB, we saved a significant amount of backhaul load through proactively offloading of UEs. We extended our experiment to observe average backhaul load saving through a varying number of active users in the system. Table 6 shows simulation results through different cell loads and backhaul saving. It is evident from Table 6 that we saved about 20% average backhaul in the system.

#### 4.3.5 | Network gain under proactive content offloading

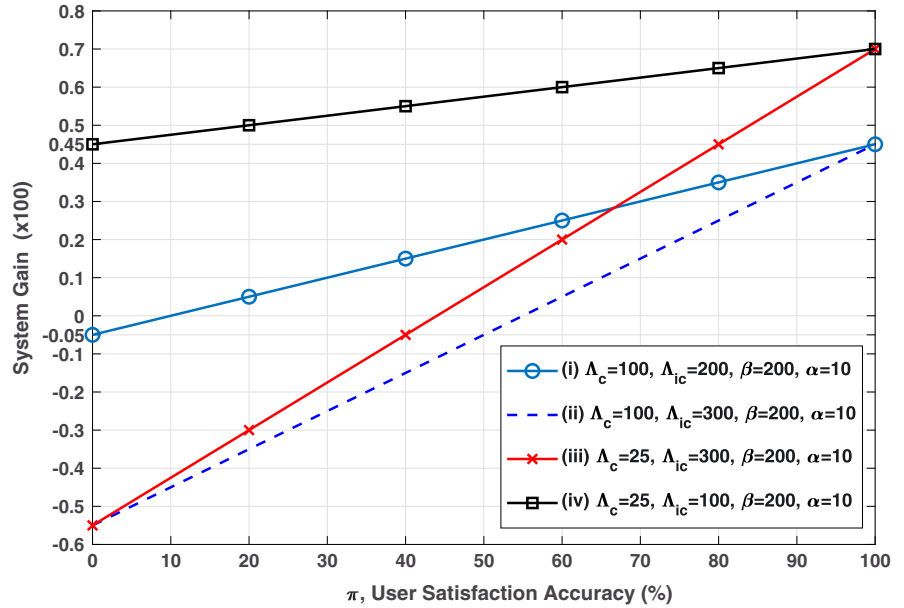
The parameter responsible for system gain under proactive offloading is user satisfaction ratio, ie, how accurately caching model is able to predict user future demands based on his file-demand profile. In order to measure the system gain under proactive offloading and reactive offloading, we used the metric defined in the work of Mohamed et al.<sup>55</sup> Let  $\varepsilon$  describe the system gain under proactive offloading

$$\varepsilon = \frac{\beta - \varphi_p}{\beta}, \quad (25)$$

where  $\beta$  represents resource utilization cost under reactive offloading and  $\varphi_p$  is expected resource utilization cost under proactive offloading, which is given as

$$\varphi_p = \pi(\Lambda_c) + (1 - \pi)(\Lambda_{ic}) + \alpha. \quad (26)$$

Here,  $\pi$  is user satisfaction ratio with proactively downloaded contents, and  $\alpha$  is the cost of copying the contents, while  $\Lambda_c$  and  $\Lambda_{ic}$  represent resource utilization cost under accurate and inaccurate future content predictions. Actually, an inaccurate future content prediction may lead to overall system degradation because it reserves resources in advance that might otherwise be used for other users. Figure 10 shows the overall system gain under reactive and proactive content LB. We considered four cases and cost under reactive;  $\beta = 200$  fixed value is considered. Moreover, a fixed value of  $\alpha = 10$  is considered as content copying cost. Here, we describe cases as (i) in which  $\Lambda_c = \beta/2$ ,  $\alpha = 10$ , and  $\Lambda_{ic} = \beta$ , system gain,  $\varepsilon$  is negative if  $\pi \leq 10\%$ . The system gain becomes positive if  $\pi > 10\%$  and lies between 0 to 45%. In (ii), when  $\Lambda_{ic} = 300$ , then gain can be negative, but at 50% accuracy, system behavior will be the same as under reactive LB (ie,  $\varepsilon = 0\%$ ). In (iii),  $\Lambda_c = 25$ , gain  $\varepsilon$  achieves its maximum value of 70%. In (iv),  $\Lambda_{ic} = \beta/2$ , then gain is always positive from 45% to 70%. In short, with 50% accurate content predictions, the gain is always positive if user satisfaction ratio,  $\pi$  is greater than 10%, and overall system achieves up to 45% gain as compared to reactive LB approach.



**FIGURE 10** System gain under reactive load balancing (LB) and proactive LB approaches

## 5 | ANALYSIS AND INSIGHTS

This section provides a comprehensive analysis and deep insights about the proposed load balancing framework. Our framework considers a central controller that leverages mobility prediction and content caching models to offload network traffic proactively. Each BS in our framework contains cache storage to cache users' future contents from the central controller. In fact, deploying caches at the BS level not only maximizes spectral efficiency but also improves energy efficiency in the network. Practically, cache deployment is also very cost effective due to the availability of power-efficient storage hardware such as high-speed solid state disks. Owing to these benefits, our framework considers content caching at the cell level to proactively cache a large amount of data and deliver to users before they are moving to congested cells.

Our ultimate objective is to reduce the overall network load by predownloading users' future files at lightly loaded cells. This enables the network to utilize the available bandwidth more efficiently and also reduces congestion at heavily loaded cells. The proposed framework, in its current shape, is suitable for proactive content caching and delivering the contents, which are delay-tolerant and completely downloadable files/jobs such as software updates, game applications, movies, music files, etc, because these applications consume an enormous amount of bandwidth during delivering to the end users.<sup>56</sup> The lightly loaded cells can download such a large amount of data and deliver to users before they are moving to the congested cells; this can improve users' service experience and reduce overall network load. Our framework lacks seamless handover support; however, video streaming contents can be proactively cached in the form of small chunks such as *YouTube* buffers videos into small segments. In the future, we aim to extend this work for live contents and video streaming applications with seamless handovers and coded caching support.

Further, in this section, we encapsulated the sensitivity analysis of the proposed proactive offloading framework. It is observed that mobility prediction of a mobile user is a challenging task due to random trajectory patterns. This challenge becomes more adverse in case of highly uncertain movements such as more randomness in the trajectory paths. In a case where users' next cell prediction accuracy is not substantial, we have to download multiple copies of future contents at more than one cell such as most likely next two cells. This improves mobility the prediction accuracy as evident from the results presented in Figure 5. However, this may lead to an increase in cell bandwidth consumption, ie, exerting more load on the system. In the proposed PLB framework, proactive caching is employed only at lightly loaded cells; therefore, this extra content copying load can be accommodated by these cells effectively without degradation of users' QoE.

The edge caching provides significant benefits such as proactive cell traffic offloading and overall system efficiency maximization. Under Max-RSRP- and CIO-based schemes, all content files have to be fetched from the core network, which leads to inauspicious network performance during congestion hours. In order to alleviate this problem, the proposed PLB framework shares the load of congested cells and caches users' future contents proactively when they are staying at lightly loaded cells. For this, we used a central controller to track and learn the demands of users for future content caching. At the central controller, the matrix  $D$  represents the demands of users in the network. Matrix  $D$  has a size of  $U \times F$ , where  $U$  and  $F$  represent the total number of users in the system and number of files respectively. In our simulations, by

considering a real scenario, we kept  $F$  size fixed with 1682 files because the system has only a limited number of contents which exert huge load and are delay-tolerant type, while we varied the number of users from 200 to 500. It took 1.598 seconds with 200 users, while it took 3.57 seconds when users are 500 to yield the caching results. This validates the time efficiency of our caching algorithm and provides insights into our realistic users' demand profiling approach through which we achieved 80% users' satisfaction ratio with their future contents (as evident from Figure 6).

This work doles out promising benefits of proactive load balancing framework over the reactive approaches, but it has some limitations. First, mobility model is unable to capture all type of mobility cases such as special events (uncertain gatherings or protests) in which a user's mobility is completely different from his/her daily routines. Second, it is not suitable for live contents such as live video broadcasting, which requires a minimum end-to-end delay. Further, we intend to improve system scalability with an increase in the number of users. With a large number of users, the collection of individual cell sojourn times and content demand statistics may cause high data processing and analysis overhead. To address this issue, one of the potential approaches can be a clustering of users based on the similarity between mobility trajectories and content demand patterns. As a result, user clusters based on social groups can be designed to improve caching policy that caches the contents for a cluster instead of user-level consideration.

## 6 | CONCLUSION AND FUTURE WORK

This article comprehensively studies current reactive load balancing approaches in the HetNets and highlights their limitations that confine their ability to deliver premium QoE to mobile users. In order to address the limitations of reactive load balancing, we propose an efficient PLB framework to offload traffic prior to congestion by proactively caching users' most expected future data during their stay at lightly loaded cells when their next probable cell is a congested cell. Our PLB framework exploits users' trajectories for future cells predictions and proactively caches future contents by modeling users' content demand profiling. In this way, most expected future contents are cached at the lightly loaded cells other than possibly congested cells to avoid congestion. Evaluation results show that the proposed framework outperforms reactive approaches in terms of user satisfaction, backhaul saving, cell load fairness, and average downlink rates. At congested cells, users are able to perceive higher downlink throughput, thanks to higher proactive content caching satisfaction. Specifically, the overall network achieved up to 98.7% cell load fairness along with nearly 20% backhaul load reduction.

The proposed PLB framework is the first attempt toward the proactive implementation of cells balancing through anticipation of mobile users' behavior. In this work, we successfully achieved significant gains as compared to reactive offloading schemes. These gains might be limited due to the dynamics of users' behavior in terms of mobility patterns and content demand statistics. This inference can be further improved through the exploitation of users' contextual information for mobility prediction rather than solely relying on the temporal information. Moreover, users' future demand prediction can also be improved with the availability of a large amount of user-file rating data. In the future, we plan to utilize context-aware mobility information to cache users' future demands for proactive traffic offloading.

## ACKNOWLEDGMENTS

This work was supported by Punjab Higher Education Commission (PHEC), Lahore, Pakistan, and the National Science Foundation under grants 1619346, 1559483, 1718956, and 1730650.

## ORCID

Sanaullah Manzoor  <https://orcid.org/0000-0002-4055-8469>

## REFERENCES

1. Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020. White paper. 2016.
2. Latif S, Qadir J, Farooq S, Imran MA. How 5G wireless (and concomitant technologies) will revolutionize healthcare? *Future Internet*. 2017;9(4):93.
3. Statista. <https://www.statista.com/chart/1620/top-10-traffic-hogs/>
4. Argyriou A, Poularakis K, Iosifidis G, Tassioulas L. Video delivery in dense 5G cellular networks. *IEEE Netw*. 2017;31:28-34.
5. Imran A, Zoha A, Abu-Dayya A. Challenges in 5G: how to empower son with big data for enabling 5G. *IEEE Network*. 2014;28(6):27-33.
6. Shah SWH, Mian AN, Mumtaz S, Crowcroft J. System capacity analysis for ultra-dense multi-tier future cellular networks. *IEEE Access*. 2019;7:50503-50512.

7. Andrews JG, Singh S, Ye Q, Lin X, Dhillon HS. An overview of load balancing in HetNets: old myths and open problems. *IEEE Wirel Commun.* 2014;21(2):18-25.
8. Asghar A, Farooq H, Imran A. Concurrent optimization of coverage, capacity, and load balance in HetNets through soft and hard cell association parameters. *IEEE Trans Veh Technol.* 2018;67:8781-8795.
9. Ye Q, Rong B, Chen Y, Al-Shalash M, Caramanis C, Andrews JG. User association for load balancing in heterogeneous cellular networks. *IEEE Trans Wirel Commun.* 2013;12(6):2706-2716.
10. Zhou T, Liu Z, Zhao J, Li C, Yang L. Joint user association and power control for load balancing in downlink heterogeneous cellular networks. *IEEE Trans Veh Technol.* 2018;67(3):2582-2593.
11. Ge X, Li X, Jin H, Cheng J, Leung VC. Joint user association and user scheduling for load balancing in heterogeneous networks. *IEEE Trans Wirel Commun.* 2018;17(5):3211-3225.
12. Zhou T, Huang Y, Fan L, Yang L. Load-aware user association with quality of service support in heterogeneous cellular networks. *IET Commun.* 2015;9(4):494-500.
13. Sun Y, Feng G, Qin S, Sun S. Cell association with user behavior awareness in heterogeneous cellular networks. *IEEE Trans Veh Technol.* 2018;67(5):4589-4601.
14. 3GPP. TS-36.331 radio resource control (RRC); protocol specification. 2009.
15. Franco CAS, de Marca JRB. Load balancing in self-organized heterogeneous LTE networks: a statistical learning approach. Paper presented at: 7th IEEE Latin-American Conference on Communications (LATINCOM); 2015; Arequipa, Peru.
16. Du H, Zhou Y, Tian L, Wang X, Pan Z, Shi J, et al. A load fairness aware cell association for centralized heterogeneous networks. Paper presented at: 2015 IEEE International Conference on Communications (ICC); 2015; London, UK.
17. Abbas ZH, Muhammad F, Lei J. Analysis of load balancing and interference management in heterogeneous cellular networks. *IEEE Access.* 2017;5:14690-14705.
18. Zhang T, Xu H, Liu D, Beaulieu NC, Zhu Y. User association for energy-load tradeoffs in HetNets with renewable energy supply. *IEEE Commun Lett.* 2015;19(12):2214-2217.
19. Boostanimehr H, Bhargava VK. Distributed and QoS-driven cell association in HetNets to minimize global outage probability. Paper presented at: 2014 IEEE Global Communications Conference (GLOBECOM); 2014; Austin, TX.
20. Coucheney P, Hyon E, Kelif J-M. Mobile association problem in heterogeneous wireless networks with mobility. Paper presented at: 24th Annual IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC); 2013; London, UK.
21. Sadr S, Adve RS. Handoff rate and coverage analysis in multi-tier heterogeneous networks. *IEEE Trans Wirel Commun.* 2015;14(5):2626-2638.
22. Pantisano F, Bennis M, Saad W, Debbah M. Cache-aware user association in backhaul-constrained small cell networks. Paper presented at: 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt); 2014; Hammamet, Tunisia.
23. Bastug E, Bennis M, Debbah M. Social and spatial proactive caching for mobile data offloading. Paper presented at: 2014 IEEE International Conference on Communications (ICC); 2014; Sydney, Australia.
24. Said A, Shah S, Farooq H, Mian A, Imran A, Crowcroft J. Proactive caching at the edge leveraging influential user detection in cellular D2D networks. *Future Internet.* 2018;10(10):93.
25. Fooladivanda D, Rosenberg C. Joint resource allocation and user association for heterogeneous wireless cellular networks. *IEEE Trans Wirel Commun.* 2013;12(1):248-257.
26. Asghar A, Farooq H, Imran A. A novel load-aware cell association for simultaneous network capacity and user QoS optimization in emerging HetNets. Paper presented at: 28th Annual IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC); 2017; Montréal, Canada.
27. Gholami MH, Azmi P, Mokari N, Forouzes M. Radio resource allocation in heterogeneous cellular networks based on effective capacity maximization: perspective mobile data offloading. *Trans Emerg Telecommunications Technol.* 2017;28(12):e3220.
28. Dhahri C, Ohtsuki T. Q-learning cell selection for femtocell networks: single-and multi-user case. Paper presented at: 2012 IEEE Global Communications Conference (GLOBECOM); 2012; Anaheim, CA.
29. Moharir S, Krishnasamy S, Shakkottai S. Scheduling in densified networks: algorithms and performance. *IEEE/ACM Trans Networking.* 2017;25(1):164-178.
30. Ma D, Ma M. Proactive load balancing with admission control for heterogeneous overlay networks. *Wirel Commun Mob Comput.* 2013;13(18):1671-1680.
31. Ichkov A, Atanasovski V, Gavrilovska L. Analysis of two-tier LTE network with randomized resource allocation and proactive offloading. *Mobile Netw Appl.* 2017;22(5):806-813.
32. Hu YC, Patel M, Sabella D, Sprecher N, Young V. Mobile edge computing—A key technology towards 5G. *ETSI White Paper.* 2015;11(11):1-16.
33. Liu D, Chen B, Yang C, Molisch AF. Caching at the wireless edge: design aspects, challenges, and future directions. *IEEE Commun Mag.* 2016;54(9):22-28.
34. Jin Y, Wen Y, Westphal C. Optimal transcoding and caching for adaptive streaming in media cloud: an analytical approach. *IEEE Trans Circuits Syst Video Technol.* 2015;25(12):1914-1925.
35. Hoang DT, Niyato D, Nguyen DN, Dutkiewicz E, Wang P, Han Z. A dynamic edge caching framework for mobile 5G networks. *IEEE Wirel Commun.* 2018;25(5):95-103.

36. Al Ridhawi I, Aloqaily M, Kotb Y, Al Ridhawi Y, Jararweh Y. A collaborative mobile edge computing and user solution for service composition in 5G systems. *Trans Emerg Telecommunications Technol.* 2018;29(11):e3446.
37. Tang S, Alnoman A, Anpalagan A, Woungang I. A user-centric cooperative edge caching scheme for minimizing delay in 5G content delivery networks. *Trans Emerg Telecommunications Technol.* 2018;29(8):e3461.
38. Park GS, Song H. Cooperative base station caching and X2 link traffic offloading system for video streaming over SDN-enabled 5G networks. *IEEE Trans Mob Comput.* 2018.
39. Farooq H, Imran A. Spatiotemporal mobility prediction in proactive self-organizing cellular networks. *IEEE Commun Lett.* 2017;21(2):370-373.
40. Hu Y, Zhang D, Ye J, Li X, He X. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Trans Pattern Anal Mach Intell.* 2012;35:2117-2130.
41. Fan J, Chow TW. Matrix completion by least-square, low-rank, and sparse self-representations. *Pattern Recognit.* 2017;71:290-305.
42. Kang Z, Peng C, Cheng Q. Top-N recommender system via matrix completion. Paper presented at: 30th AAAI Conference on Artificial Intelligence (AAAI-16); 2016; Phoenix, AZ.
43. Zoha A, Saeed A, Farooq H, Rizwan A, Imran A, Imran MA. Leveraging intelligence from network CDR data for interference aware energy consumption minimization. *IEEE Trans Mob Comput.* 2017;17:1569-1582.
44. 3rd Generation Partnership Project. TR 25.814 3rd Generation Partnership Project, Physical Layer Aspects for Evolved Universal Terrestrial Radio Access (E-UTRA). 2006.
45. MATLAB MathWorks. <https://www.mathworks.com/products/matlab.html>
46. 3rd Generation Partnership Project. 3GPP TS 36.410 V10.2.0, Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 general aspects and principles (Release 10). 2011.
47. Farooq H, Asghar A, Imran A. Mobility prediction based autonomous proactive energy saving (AURORA) framework for emerging ultra-dense networks. *IEEE Trans Green Commun Netw.* 2018;2:958-971.
48. Rahmati A, Zhong L. Context-for-wireless: context-sensitive energy-efficient wireless data transfer. In: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services; 2007; San Juan, Puerto Rico.
49. GroupLens. MovieLens Dataset. 2018. <http://grouplens.org/datasets/movielens/>
50. Lee J-K, Hou JC. Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application. In: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing; 2006; Florence, Italy.
51. Chatzieftheriou LE, Karaliopoulos M, Koutsopoulos I. Caching-aware recommendations: nudging user preferences towards better caching performance. Paper presented at: IEEE Conference on Computer Communications (INFOCOM); 2017; Atlanta, GA.
52. Jiang Y, Ma M, Bennis M, Zheng F-C, You X. User preference learning-based edge caching for fog radio access network. *IEEE Trans Commun.* 2019;67(2):1268-1283.
53. Akaike H. Information theory and an extension of the maximum likelihood principle. In: *Selected Papers of Hirotugu Akaike*. New York, NY: Springer; 1998:199-213.
54. Jain R, Chiu D-M, Hawe WR. A quantitative measure of fairness and discrimination for resource allocation in shared computer system, vol 38. Hudson, MA: Eastern Research Laboratory, Digital Equipment Corporation; 1984.
55. Mohamed A, Onireti O, Hoseinitabatabaei SA, Imran M, Imran A, Tafazolli R. Mobility prediction for handover management in cellular networks with control/data separation. Paper presented at: 2015 IEEE International Conference on Communications (ICC); 2015; London, UK.
56. Carbutar B, Potharaju R. A longitudinal study of the Google app market. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; 2015; Paris, France.

**How to cite this article:** Manzoor S, Mazhar S, Asghar A, Noor Mian A, Imran A, Crowcroft J. Leveraging mobility and content caching for proactive load balancing in heterogeneous cellular networks. *Trans Emerging Tel Tech.* 2020;31:e3739. <https://doi.org/10.1002/ett.3739>