An optimized Handover management scheme tailored for Heavy Hitters in a disaggregated 5G O-RAN architecture

Franci Gjeci, Ilario Filippini, Antonio Capone
name.surname@polimi.it - Politecnico di Milano, Milan, Italy

***Abstract*—In this paper we propose a tailored handover scheme for Heavy Hitters (HHs) in Next Generation Self-Organizing Network (NG SON). The conventional handover scheme takes into consideration only the signal strength of source and target cell on handover decision, without considering the traffic specifics of the User Equipments (UEs). We believe that customizing the handover decision on individual UEs' needs gives higher Quality of Experience (QoE). Particularly, in this work we focus on optimizing the QoE of HH UEs in the 5th generation (5G) network by controlling and customizing the handover mechanism specifically for these users. We test our approach in Network Simulator 3 (ns-3) and use the emerging Open RAN (O-RAN) framework as a supporting architecture to proactively monitor the presence of HHs in the network and react upon their appearance. The contribution in this paper is twofold: We present an offline detection scheme for HHs based on Deep-Neural Network (DNN) with an accuracy of 94% and a Greedy Handover algorithm that improves the overall average throughput of HHs.**

***Index Terms*—Open-RAN, SON, 5G, handover optimization, heavy hitters**

I. Introduction

Heavy Hitters (HHs) have long been studied in traffic engineering. These are flows having the peculiarity of being few in numbers, but occupying a relevant portion of the overall traffic present in the internet. On the opposite, the other flows, which we name them as regular flows in this paper, are many but with low demands for traffic. Efforts have been made to detect Heavy Hitters (HHs) in cellular networks [18], [20], however, little has been done to improve their Quality of Experience (QoE). Meanwhile, services generating HHs are proliferating in 5th generation (5G), such as high resolution Video-On-Demand, Augmented Reality etc. In this paper we propose in the context of Self-Organizing Network (SON) a HHs detection strategy in 5G networks as well as a tailored handover scheme with the aim to ameliorate their perceived QoE.

The concept ofSON has emerged as a transformative paradigm with a vision was to create an autonomous cellular network, which could dynamically adapt to the environment, by giving to the network the capability of self-configuring, self-optimizing and self-healing. New technological facets have been introduced in 5G: the addition of mmWave spectrum, besides the conventional sub-6 Ghz spectrum; the New Radio (NR) air interface; the heterogeneity of services and devices it must support. All the above imposes new proliferating network management challenges for 5G, which has made SON become an even more appealing feature.

Handover management is one of those management challenges. Conventionally, the standard handover algorithms have been centered around predefined events, such as the A3 or A2 event [4]. Even in literature, the primary objective has been the optimization of parameters related to these events, including Hysteresis, Time-to-Trigger, and Cell Individual Offset. Additionally, Machine Learning (ML) approaches have been suggested to dynamically adapt these typical SON parameters [14]. In [22], the authors have proposed a Reinforcement Learning (RL) paradigm, whose an agent takes handover actions to maximizes the long-term utility of the chosen cell. In [6], the authors propose a Recurrent Neural Networks (RNN) model and take handover decisions based upon the expected QoE. Similarly, the primary emphasis of this paper lies in the realm of handover management designed specifically for maximizing the expected QoE of HHs.

To test our model, we have used Network Simulator 3 (ns-3) [17] as a simulation environment and the Open RAN (O-RAN) architecture [1], [16]. ns-3 is an open-source network simulator deploying a non stand-alone 5G full protocol stack. It deploys 3rd Generation Partnership Project (3GPP) communication protocol standards. Whereas O-RAN is an emerging architecture having the capability to control the Radio Access Network (RAN) components of 5G network via standardized interfaces. Our solution leverages the integration among the two entities introduced in [12].

Contributions In this paper, we propose an end-to-end solution for detecting HHs, by designing an offline Deep-Neural Network (DNN) algorithm, and enhancing their QoE, by building a tailored handover scheme. The solutions are custom designed to guarantee minimal

978-3-903176-63-8/24/$31.00 ©2024 IEEE

overhead in 5G networks. They are implemented in O-RAN [16] architecture, which is a open framework with control capabilities over 5G networks.

The results yielded by the simulations we have conducted in ns-3 demonstrate improvements in throughput and better resource utilization with respect to the classical handover scheme.

Organization The rest of the paper is organized as follows. Section II gives details of the implemented O-RAN architecture. Section III describes the offline detection algorithm using DNN, which is exploited in Sect. IV as a pre-stage to filter the set of HHs running in the system, to latter apply the designed handover algorithm. Finally, Sect. VI summarizes and concludes the work.

II. O-RAN based architecture

O-RAN [1], [16] is an emerging architectural framework for mobile radio networks focusing on disaggregating the components of the Base Station (BS) connecting them via open interfaces. O-RAN features the closed-loop control of these components by means of intelligent controllers. Near Real-time RAN Intelligent Controller (near-RT RIC) is the intelligent controller with a role to directly control the RAN nodes through xApps. These are micro-services running on top of near-RT RIC with an operation granularity of 10ms-1s connected with the Next Generation Node B (gNB) via the open E2 interface;

Recently, the authors in [12] have integrated O-RAN in ns-3. The last enables primarily the following: 1) exposes data collection from the gNB, which are sent to the RAN Intelligent Controller (RIC) to be processed by the xApp. 2) Thereafter the xApp generates control commands, which are send them back to the gNB to be executed by the late. Fig.1 portrays the architecture as explained above, delineating the gNB on the left and a simplified representation of the O-RAN architecture on the right. These two entities are connected via the E2 open interface, visually depicted as a straight orange line, linking the two endpoints, E2 termination and E2 RIC termination. The former is the endpoint of the E2 interface on the 5G network side, terminating in gNB, while the latter is the endpoint of the E2 interface terminating in the O-RAN architecture.

Within our scenario, we define two independent for the HH Detection and Handover Management Scheme, visually illustrated by the dashed blue and green line in Fig.1. For each microservice, we have created a corresponding micro-service / xApp.

The Detection microservice, as extensively explained in section III, has the responsibility of detecting HHs within the 5G network relaying the information to Handover microservice. The later, as detailed in section IV, orchestrates the handover algorithm and, ultimately, makes informed decisions regarding the necessity of handovers for the detected HHs.

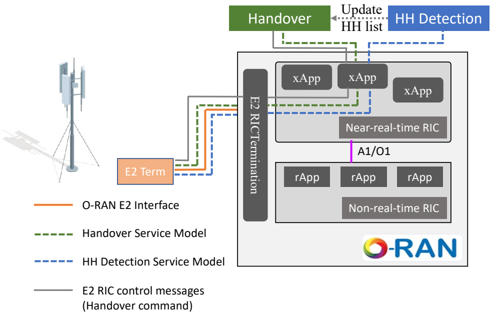

Figure 1: O-RAN enabled architecture

III. Heavy Hitters Detection Microservice

In this section we describe the characteristics of the HH detection algorithm we have implemented as well as its strength compared to alternative solutions with regard to the O-RAN architecture. Authors in [9], [15], [18], [21] propose a detection scheme based on a flow level. Similar works have tackled the problem of identifying HHs in the programmable switches of the core network [8], [11], [19], [20].

Authors in [9], [11], [19], [21] deploy a top-k flow detection. Besides their high performance, choosing the static parameter k in not trivial. The number of HHs in the system change in time and from cell to cell. Thus, these solutions are impractical for algorithms which needs to know the real number of HHs to offer a good optimization performance. Authors in [15], [18] deploy an alternative solution based on ML, though the detection time they claim, 900 milliseconds in [18] and 143 milliseconds seconds having an accuracy of 75%, is restricting and reducing the ability to promptly react and being able to improve HH performance.

Our solution, instead, has two main advantages: 1) it has a lower data granularity, to the extent that we determine if a user has a service which generates a HH flow, thus having an aggregated view of all the single flows of an user; 2) as explained below, it is designed to be implemented in a disaggregated architecture, distributing the detection task into multiple computing units in the 5G network, also taking advantage of the edge-computing capabilities of the 5G network.

The features used for training the DNN model are reported in Table I alongside the applied aggregated function on the data. These feature are collected from the Physical (PHY), Medium Access Control (MAC) and Radio Link Control (RLC) layer of the Distributed Unit (DU) [16]. Practically speaking, in the real implementation we have deployed in ns-3, we fetch the indicated data in Table I in real time from the traces of the aforementioned layers and temporary store them into arrays. Thereafter, they are aggregated per frame level applying the aggregation function specified in the Table I. The frame aggregation level gives the least overfitting and highest accuracy. It is O-RAN vision to enable data collection and application of ML models upon the data [10], [16], enabling the implementation of the proposed solution.

We employ the finest granularity of data reports generation supported by near-RT RIC, that is 10 ms. Hence, every 10 ms, every BS sends a data report composed of the aggregated data of Table I for all its attached User Equipments (UEs) to the xApp. Next, the xApp for each active UE in the 5G network, compiles time series of values per listed feature in Table I spanning over the last 5 frames. They shall serve as the input data needed by the xApp to detect an HH. The aggregation granularity based on an user level, rather than on a flow level, reduces the efforts and time consumed to discover these users.

Moreover, when viewed in the perspective of a disaggregated BS, as indicated in the O-RAN architecture as well, the detection module could be placed in the DUs [16], at the edge of the network. This further subdivides the efforts of detection into multiple units, making it highly scalable and fast. This is further endorsed by the fact that the data for the detection is extracted in the DU only, thus by default, this scheme minimizes the message exchange overhead for detection.

The selected features are the smallest set with the least correlation among all the extractable features, as well as provide significant information. Table. III represents the detection accuracy achieved through variations in the time series depth of features' data, reported in number of frames. It can be seen that the accuracy saturation is reached when using features data over the last 5 frames, which is the why we have chosen it.

The data is flatten into a single data array, which is then ready to be input into the model. The DNN model's layout is a dense one with 30 input nodes (6 features per 5 frames), two intermediate layers with 16 and 8 intermediate nodes with Rectified Linear Unit (ReLU) activation function and a single output node with a Sigmoid activation function. This proved to be the lightest layout for the considered dataset providing an accuracy higher than 90%. We have trained an offline model using the keras module in python with data samples we have extracted from different simulation campaigns by ranging: the number of active users, be it regular – i.e., not a HH user – (20, 30, 40, 50, 60) or HHs (1 - 7) each with a simulation duration of 10 sec considering random user movements.

Thereupon, we deployed the trained DNN model inside the xApp and use it in real time for HH detection. Whenever the xApp receives new data reports coming from a BS, it categorizes the users into HHs and regular ones, then updates the list of HHs when necessary. As explained in Sect. IV, this list shall be used as a base by the handover algorithm.

Table I: Features of DNN HH detection model

Features
Mean RLC current queue size
Mean RLC head of line packet delay
Number of scheduled UEs
Total used symbols by all users
User used symbols
Sum of transport block size

IV. Greedy Handover Microservice

Current handover schemes rely solely on the signal strength as the primary factor for determining the destination cell [3]. In general terms, it aims to avoid the occurrence of Radio Link Failures (RLFs). Nonetheless, it lacks a mechanism to prioritize and guarantee a superior perceived QoE for bandwidth-intensive services, as it does not account for the cell load when making handover decisions.

With the advent of ML, greater emphasis is being given to the optimization of the perceived QoE. Authors in [5], [6] propose an handover management scheme based on the experienced QoE of UEs. However, the scheme they have deployed is generic and not applicable to HH users. On the contrary, we focus specifically on HHs and propose an optimization scheme tailored for HHs in 5G RAN. To the best of our knowledge, this is the first work proposing such an approach.

In this context, we introduce an handover management scheme that focuses on improving the throughput, and thus the QoE perceived by HHs. Our handover strategy is to distribute the HHs among the active cells in the system by accounting their estimated throughput and maximizing the overall HHs' throughput. This strategy is supported by the fairness of the scheduler upon two main premises: 1) it guarantees service to all the users. Hence, the presence of HHs in the cell shall not dry out the resources, thus shall not substantially impact regular users. 2) In presence of multiple HHs competing for the same resources, the scheduler ensures the fair distribution of these resources among them. Both of these premises are guaranteed by all type of schedulers.

The greedy algorithm in principle aims at quantifying a goodness factor γ as in Eq. 4, which assesses the benefit of making an user handover in a reachable BS. Simplifying the reasoning to a single handover, the overall benefit/cost of taking an handover decision is defined by the impact of three groups of UEs, which are influenced by that decision: 1) The UE itself for which an handover is being triggered due to changes of channel conditions and available resources in the new cell, an impact factor we denote as α; 2) the remaining UEs in the origin cell, due to more resources becoming available, they experience an overall positive impact factor β; 3) the UEs present in the destination cell, experiencing a negative impact δ from the addition of a new user, due to division of the available resources across more users. In practical terms, the overall factor γ consists of the positive impact $\alpha + \beta$ and the negative impact δ, of which we consider the ratio: $\gamma = \frac{\alpha+\beta}{1+\delta}$.

Referring back to the previous discussion on scheduler fairness, at the beginning of the scheduling, the scheduler of a cell will assign the same priority to all the active users connected to it. Even if HHs have much higher demands than regular flows, the scheduler effectively serves the regular users without being adversely affected by the presence of HHs. Whereas, the HHs have a filling effect on resource utilization, as they shall occupy all the unused resources from the regular users. Whenever more than one HH is attached to a cell, the scheduler distributes the resources with equal priority among them.

$$\alpha^l_{i,j} = RSRP^{j,i} * \Delta prb^{j,i} - RSRP^{l,i} * prb^{l,i}$$
$$, where\ \Delta prb^{j,i} = prb^j(\frac{1}{|H_j|} - \frac{1}{|H_j|+1}) \quad (1)$$

$$\beta^l_{i,j} = \sum_{k \in H_l, k \neq i} RSRP^{l,k} * \Delta prb^{l,k}$$
$$, where\ \Delta prb^{l,k} = prb^l(\frac{1}{|H_l|-1} - \frac{1}{|H_l|}) \quad (2)$$

$$\delta_{i,j} = \sum_{k \in H_j} RSRP^{j,k} * \Delta prb^{j,k}$$
$$, where\ \Delta prb^{j,k} = prb^j(\frac{1}{|H_j|} - \frac{1}{|H_j|+1}) \quad (3)$$

$$\gamma^l_{i,j} = \frac{\alpha^l_{i,j} + \beta^l_{i,j}}{1 + \delta_{i,j}} \quad (4)$$

We define as $\beta^l_{i,j}$ in Eq. 2 all the peers gain which are connected to the source cell l when moving user i to cell j, due to more resources becoming available for them. All the HHs attached to the destination cell are impacted by a new HH coming, for the available resources shall be shared among the new HH and previously avaialable HHs. We define as $\delta_{i,j}$ in Eq. 3 the gain (loss) of all users in the destination cell j when transferring user i to cell j. Finally, the last term is the gain perceived by user i, $\alpha^l_{i,j}$ in Eq. 1, due to changes in channel conditions as well as in available resources between the source cell l and destination j when an handover is triggered on it.

To combine all the gain terms into a single quantity, we use Eq. 4, which gives a goodness value for transferring user i to cell j.

To translate the gain into a measurable quantity, we have defined it as the product of the signal strength of the Reference Signal Received Power (RSRP) [dBm] and Physical Resource Blocks (PRBs). In the ns-3 implementation, we use the last updated value of received RSRP. The prb^j denotes the allocatable PRBs for HHs on a given cell j, whereas $prb^{j,i}$ denotes the used PRBs of HH i in cell j. Practically, it is calculated as the difference of the available PRBs and the PRBs which have been used by regular users. In our implementation, in a 10ms optimization round, it corresponds to the difference of the available PRBs across 10 frames and the used PRBs by regular users in the last 10 frames. Let S be the set of candidate cells and N the set of all HHs. Let H_j be the set of HHs available in cell $j \in S$ and D_i the set of destination cells which can be reached by user i.

Δprb in Eq. 1, 2 and 3 refer to the difference in the number of PRBs before and after the change of the number of HHs in a cell. In Eq. 1 and 3 a new HH is being added to the destination cell j. Hence, if before the change the number of available PRBs for HHs in cell j (prb^j) were shared among $|H_j|$ HHs, with a new HH added, they shall be shared among $|H_j|+1$ HHs. In Eq. 2 a HH is removed from the source cell l, thus more resources are available for the remaining HHs in cell l. If before the PRBs are redistributed among $|H_j|$ HHs, after one HH has been moved from the cell, the resources shall be divided among $|H_j|-1$ HHs.

The algorithm Alg. 1 starts by considering all the HHs in the network. The Eq. 5a picks the HH with the smallest throughput and extracts the cell $\hat{l}$ this user is assigned to. From here we can follow two handover strategies: 1) moving this user to a new cell with more resources to provide higher throughput, or 2) move another HH from the same cell $\hat{l}$ to another cell to free resources from cell $\hat{l}$. Regardless of which HH is moved to a different cell, the impact is the same on the remaining HHs in the source cell. Each HH shall have more available resources, thus a higher throughput. We have chosen the second strategy. Upon this logic, from reference cell $\hat{l}$, we pick the user $\hat{i}$ and the destination cell $\hat{j}$ yielding the overall highest goodness according to Eq. 5b. We consider it to be a feasible solution and eventually insert it in the handover list when the gain $\gamma_{\hat{i},\hat{j}}$ is positive. After that, we remove user $\hat{i}$ from the set of candidate HHs to transfer.

Same as for the HH detection, we have defined a microservice with the finest time granularity of near-RT RIC, which is 10 ms, to generate the data needed by the handover algorithm in the xApp. Practically speaking, in ns-3 is generated a data report every 10 ms for each BS containing the following information per UE referring to the last completed 10 frames: 1) the last updated value of received RSRP per ue for the attached and neighbor BSs 2) the number of used PRBs per UE in the referring last 10 frames. By default, the execution periodicity of the algorithm is 10 ms.

Algorithm 1 Handover Greedy Algorithm

```
$I = N$
$R' = \{\}$
while $I \neq \emptyset$ do
    $\hat{l} \leftarrow \underset{l}{\text{argmin}}\, RSRP^{l,i} * prb^{l,i}, \quad \forall\, i \in I$   (5a)
    $\hat{i}, \hat{j} \leftarrow \underset{i,j}{\text{argmax}}\, \gamma_{i,j}, \quad where\ i \in H_{\hat{l}}$   (5b)
    if $\gamma_{\hat{i},\hat{j}} > 0$ then
        $R' = R' + (\hat{i}, \hat{j})$
    end if
    $I = I - \{\hat{i}\}$
end while
return $R'$
```

The Alg. 1 aims at maximizing the minimum throughput of HHs in the network. To increase its performance, an additional alternative solution is computed as in Eq. 6.

$$\hat{l} \leftarrow \underset{l}{\text{argmax}}\, RSRP^{l,i} * prb^{l,i} \qquad \forall\, i \in I \tag{6}$$

$$R \leftarrow \underset{R \in \{R', R''\}}{\text{argmax}} \sum_{(i,j) \in R} RSRP^{j,i} * prb^{j,i} \tag{7}$$

In essence, the algorithm is re-executed by substituting Eq. 5a in Alg. 1 with Eq. 6 and obtain the solution R''. The final handover solution among R' and R'' is the solution yielding the highest system throughput as in Eq. 7.

V. Simulations

The xApp has been implemented in Python and it has been deployed on a containerized near-RT RIC that is based on O-RAN's reference implementation[1].

The propagation parameters used in this scenario are set by the recommendations of 3GPP [2]. The gNBs are positioned in an hexagonal topology, 1 in the center of the hexagon and 6 at the edges. The Intersite distance (ISD) between BSs is set to 200m. Also the number of deployed UEs is fixed to 87, 7 of which are configured to generate a traffic profile of a HH when active. They are initially randomly positioned within the are of the hexagon and move around in a Random Walk fashion with speed up to 1.8m/s. In the simulation scenarios we have taken into consideration the number of active users, whether regular or HHs users, ranging between 1, 3, 5 and 7 HHs and 20, 40 and 60 regular users, are randomly selected among 87 deployed. The traffic profile for HHs is that of Adaptive Video Streaming [13], whereas the traffic profile of regular users is a Poisson Pareto Burst Process (PPBP) [7], whose parameters are randomly generated and set for each UE.

Table II: Simulation variables

Simulation parameters	
Parameter	Value
Central Frequency	28 GHz
Bandwidth	200 MHz
Number of UEs	80
Number of BSs	7
OFDM Numerology μ	3
gNB Transmit power	30 dBm
UE Transmit power	23 dBm
UE antenna size	2×2
BS antenna size	8×8
Scheduler	Round Robin
Regular UEs traffic profile	Poisson
Heavy UEs traffic profile	Video Streaming
Simulation time	10 s
Pathloss model	3GPP Umi

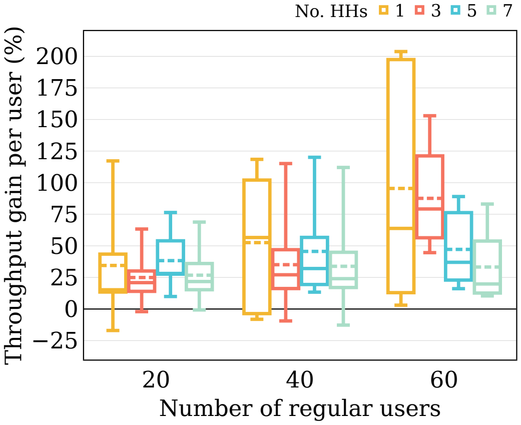

Figure 2: Boxplot distribution of throughput gain per user in percentage for simulated scenarios with 25%, 50% and 75% active users out of a total of 80 regular users. The dashed line within each boxplot represents the average, whilst the straight line corresponds to the median value.

A. Greedy algorithm performance analysis

The performance of the handover management scheme enabled by the xApp is compared to the standard handover scheme, that is A2-RSRP based handover algorithm. We compare the two by taking into consideration the average relative throughput gain per HH user. Fig.2 shows the boxplot of the throughput gain per single user. Three different percentages of active users are simulated: 25%, 50% and 75% over a total of 80 regular users and 1, 3, 5 and 7 HHs.

Referring to Fig.2, it can be seen that only in very few instances the algorithm deteriorates the HHs conditions

[1] https://gerrit.o-ran-sc.org

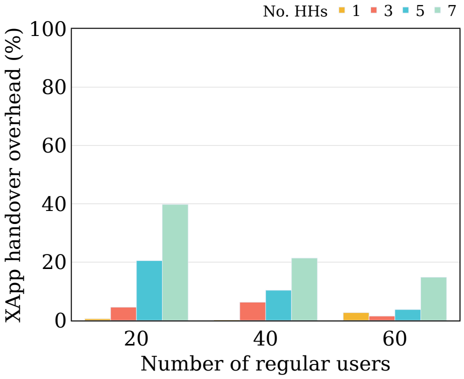

Figure 3: XApp handover overhead

Table III: Detection accuracy per number of consecutive frames

# frames	2	3	4	5	6	7	8	9	10
%	87	89	89	94	93	89	91	90	89

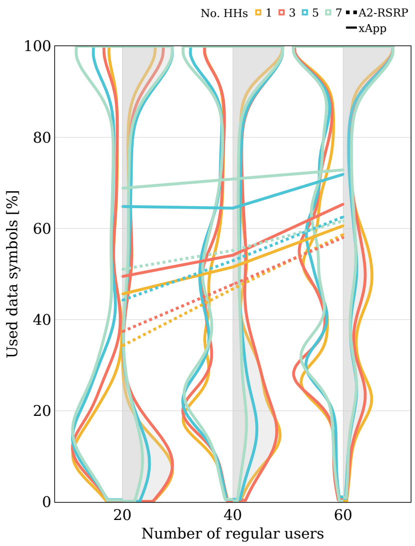

Figure 4: Used data symbols statistics per scenario in all cells

yielding a negative throughput gain. This is more likely to happen in unloaded cells with few HHs in the service area. Indeed, from the load perspective, in those scenarios cells are similarly and lightly loaded, thus there is no unbalance among them for the algorithm to take advantage.

On loaded scenarios, where the cell unbalance is manifested more, the loss of moving a HH to a cell with lower signal strength from the cell with the best signal strength is compensated by the abundance of available resources of the first. Indeed, the relative throughput gain when moving a HH is more substantial in loaded scenarios. In broad terms, the increased number of HHs tends to diminish the average throughput gain per HH user, as they contend for the available resources.

On average the xApp algorithm demonstrates superior performance when evaluated against the A2-RSRP-based handover algorithm in all the cases.

The implementation of the greedy handover algorithm within the xApp may result in increased overhead, attributed to more frequent handovers. Fig.3 provides insight into the percentage representing the average additional overhead per user across various reference scenarios. In a first observation, the presence of more HHs in the system generates unavoidably more handover overhead. On the other hand, more regular users diminish the impact of the overhead added by the HHs.

Lastly, the xApp handover algorithm balances the load of HHs among the available cells in the system, having an impact into their overall load. Fig. 4 reports the statistics of cell load by accounting the occurrence of used data symbols in a subframe expressed in percentage for all the cells and for the same reference scenarios. The data are grouped and scaled in relation to the number of regular users in the scenario. The right side of each violin plot, which is filled in grey, reports the statistics of the used data symbols for the xApp handover scheme and the left side the statistics of the standard A2-RSRP handover scheme. The dashed line indicates the average used data symbols per A2-RSRP handover scenario, whereas the straight line the equivalent average of used data symbols per xApp handover scenario.

We generally see a transfer of the peak in the violin plot from low percentages of used data symbols for the standard A2-RSRP handover scheme towards higher percentages for the xApp handover algorithm in all the cases.

In higher load scenarios, that is with with more regular users, we see a shift of the violin peak towards higher percentages of used data symbols. We attribute this effect to the presence of the load term of traffic coming from regular users.

It can be observed that the percentage of time the cells are in low load is considerably lower when applying the xApp handover algorithm.

Another important observation to account for is the average percentage of used data symbols for the xApp handover scheme and 7 HHs that is pretty constant when the number of regular users changes. This shows that the xApp handover scheme efficiently balances the load of HHs across different cells, thus making HHs' traffic component the main one to define the average load of the system.

B. Detection analysis

The overall accuracy of the detection model is reported at 94%, although it changes dependent on the network density, from 98% for 20 active regular users to 93% for 60. This observed phenomenon can be attributed to the distinctive traffic patterns between HHs and regular users. The discrepancy is particularly noticeable in scenarios with fewer regular users, where their resource utilization is low, creating a significant differentiation factor between HHs and regular users traffic pattern. In cases with more regular users, the resource utilization by the regular users becomes significant, increasing the scarcity of resources remaining for the HHs. This yields into a situation that HHs and regular users using a more similar pattern of resource utilization, thus making it more challenging for the model to differentiate between the two groups.

The detection time of the offline model in the near-RT RIC xApp adds other 10 milliseconds delay, making the minimum detection of HHs in 5G network 60 milliseconds.

VI. Concluding Remarks

In this paper, we have presented a DNN detection model to detect the presence of HH users in 5G networks and an Handover Greedy Algorithm to react upon their discovery with the aim of improving their perceived QoE. The proposed detection approach has a high accuracy and offers the advantage of being a scalable solution in the disagraggated 5G architecture as well as an opportunity to exploit the fine monitor& control possibilities provided by O-RAN. The Handover Greedy algorithm we have presented has shown to improve the QoE of HH users. Considering the current A2-RSRP event-based Handover algorithm as a benchmark, the results proved that the xApp Greedy Algorithm cab remarkably outperforms the benchmark in terms of average HH user throughput. This works underlines the potentials of implementing a tailored Handover Scheme for HHs. Nevertheless, we envision that even better performance can be achieved using more specialised algorithms, such as optimization algorithms or ML modules to take better handover decisions based on more context information.

References

[1] 0-RAN alliance. https://www.o-ran.org/.

[2] 3GPP. Study on channel model for frequencies from 0.5 to 100 ghz. Technical Report TR38.901, 3rd Generation Partnership Project (3GPP), Dec. 2018.

[3] 3GPP TS 36.331 V8.21.0. LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (3GPP TS 36.331 version 8.21.0 Release 8), 2014.

[4] 3GPP TS 38.331 V16.1.0. 5G; NR; Radio Resource Control (RRC); Protocol specification (3GPP TS 38.331 version 16.1.0 Release 16), 2020.

[5] Zoraze Ali, Lorenza Giupponi, Marco Miozzo, and Paolo Dini. Multi-task learning for efficient management of beyond 5g radio access network architectures. *IEEE Access*, 9:158892–158907, 2021.

[6] Zoraze Ali, Marco Miozzo, Lorenza Giupponi, Paolo Dini, Stojan Denic, and Stavroula Vassaki. Recurrent neural networks for handover management in next-generation self-organized networks. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–6, 2020.

[7] Doreid Ammar, Thomas Begin, and Isabelle Guerin-Lassous. A new tool for generating realistic internet traffic in ns-3. In *4th international ICST Conference on Simulation Tools and Techniques*, 2012.

[8] Ran Ben Basat, Xiaoqi Chen, Gil Einziger, and Ori Rottenstreich. Designing heavy-hitter detection algorithms for programmable switches. *IEEE/ACM Transactions on Networking*, 28(3):1172–1185, 2020.

[9] Ran Ben Basat, Gil Einziger, Roy Friedman, and Yaron Kassner. Optimal elephant flow detection. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, 2017.

[10] Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. OpenRAN Gym: AI/ML Development, Data Collection, and Testing for O-RAN on PAWR Platforms. *Computer Networks*, 220:1–11, January 2023.

[11] Rob Harrison, Qizhe Cai, Arpit Gupta, and Jennifer Rexford. Network-wide heavy hitter detection with commodity switches. In *Proceedings of the Symposium on SDN Research*, pages 1–7, 2018.

[12] Andrea Lacava, Matteo Bordin, Michele Polese, Rajarajan Sivaraj, Tommaso Zugno, Francesca Cuomo, and Tommaso Melodia. ns-o-ran: Simulating o-ran 5g systems in ns-3. *arXiv preprint arXiv:2305.06906*, 2023.

[13] Guoxi Liu and Liren Kong. Simulation of video streaming over wireless networks with ns-3. *arXiv preprint arXiv:2302.14196*, 2023.

[14] Sriram Parameswaran, Tanmoy Bag, Sharva Garg, and Andreas Mitschele-Thiel. Cognitive network function for mobility robustness optimization in cellular networks. In *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2035–2040, 2022.

[15] Adrian Pekar, Alejandra Duque-Torres, Winston K.G. Seah, and Oscar M. Caicedo Rendon. Towards threshold-agnostic heavy-hitter classification. *International Journal of Network Management*, 32(3):1–22, 2022.

[16] Michele Polese, Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Communications Surveys & Tutorials*, pages 1–1, 2023.

[17] George F Riley and Thomas R Henderson. The ns-3 network simulator. In *Modeling and tools for network simulation*, pages 15–34. Springer, 2010.

[18] Davide Sanvito, Giuseppe Siracusano, Roberto Gonzalez, and Roberto Bifulco. Prediphant: Short term heavy user prediction. In *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pages 704–709. IEEE, 2022.

[19] Vibhaalakshmi Sivaraman, Srinivas Narayana, Ori Rottenstreich, Shan Muthukrishnan, and Jennifer Rexford. Heavy-hitter detection entirely in the data plane. In *Proceedings of the Symposium on SDN Research*, pages 164–176, 2017.

[20] Pedro Rodrigues Torres, Alberto Garcia-Martinez, Marcelo Bagnulo, and Eduardo Parente Ribeiro. An Elephant in the Room: Using Sampling for Detecting Heavy-Hitters in Programmable Switches. *IEEE Access*, 9:94122–94131, 2021.

[21] Yanshu Wang, Dan Li, and Jianping Wu. Fastkeeper: A fast algorithm for identifying top-k real-time large flows. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 01–07. IEEE, 2021.

[22] Vijaya Yajnanarayana, Henrik Ryden, and Laszlo Hevizi. 5G Handover using Reinforcement Learning. *2020 IEEE 3rd 5G World Forum, 5GWF 2020 - Conference Proceedings*, pages 349–354, 2020.