Clustering-based Redundancy Minimization for Edge Computing in Future Core Networks

Abida Perveen[1], Raouf Abozariba[1], Mohammad Patwary[2], Adel Aneiba[1], and Anish Jindal[3]

[1]School of Computing and Digital Technology, Birmingham City University, United Kingdom.
[2]School of Mathematics and Computer Science, University of Wolverhampton, United Kingdom.
[3]School of Computer Science and Electronic Engineering, University of Essex, United Kingdom.
{abida.perveen, raouf.abozariba, adel.aneiba}@bcu.ac.uk, patwary@wlv.ac.uk, a.jindal@essex.ac.uk

***Abstract*—Evolving mobile edge computing has greatly improved cellular network performance and mobile user experience. However, the ever-increasing demand for new and heterogeneous services generates redundant signalling, leading to communication overheads and congestion in the network's core. We propose a novel AI-enabled edge architecture to support future networks with minimizing signalling redundancy at its heart. In this domain, we deploy a cluster-based signal and admission control framework to maximize the efficiency of link (or bandwidth resources) between the edge and core networks. We minimize the redundant signalling by employing two popular unsupervised machine learning algorithms, i.e., K-mean- and Ranking-based clustering. We evaluate the proposed framework through comparisons with recent studies in the literature. Our results show that the proposed framework provides substantial latency reduction while maximizing resource utilization. The proposed approach is 35% superior in reducing the redundant signalling compared to the current work.**

***Index Terms*—5G/6G, mobile edge computing, control signalling, resource management, and admission control.**

I. Introduction

Multi-access mobile edge computing (MEC) enables latency-sensitive services such as autonomous vehicle control and real-time health monitoring systems. The technology which was first introduced by the *European Telecommunications Standard Institute* (ETSI) in 2015 [1], extends the capabilities of cloud computing by moving resources closer to the network's edge. MEC along with the support of Artificial Intelligence (AI) techniques is an essential part of future network architecture for improving overall network performance. For instance, AI-enabled edge networks would fulfil the visions of 6G (e.g., seamless connectivity, ultra-low latency, ultra-high data rates and reliability) [2].

However, MEC faces various challenges in traffic flow management due to the ever-increasing demand for new and heterogeneous services [3]. One example is the need to manage the MEC network in such a way that it maximizes end-to-end (E2E) link efficiency and minimises latency in the Data Plane (DP) and Control Plane (CP). Recent research into MEC shows the concept of data offloading at cloud nodes to accommodate a massive number of users while keeping latency within acceptable bounds [4], [5]. In contrast, the current MEC provides limited or no access to the core control and management functions of a cellular network. This limited access reduces overall edge network performance. For example, if device density increases beyond edge network capacity, a large amount of traffic generated from the connected devices flows towards the core cellular network. The serving edge network may collaborate with other edge networks in the area and offloads a certain amount of load to neighbouring edge networks. However, these networks might have similar signalling for service and resource demand from their associated traffic loads. This inefficient resources utilization create congestion in the network that induces delays in service provisioning. However, higher delays in the edge network are unacceptable for latency-critical communication [6].

Similarly, the imbalance between traffic flow and management in the edge network causes congestion in the core network. Whenever a user request for connectivity is received, the edge node accesses the user's profile, which is stored and managed by a centralized *Unified Data Management* (UDM) in the core network. UDM offers various services, including subscriber data management, authentication and event exposure, with the help of service operations. A subscriber data management service is offered by *Get*, *Subscription*, *Unsubscription*, *Modify*, and *Notification* service operations [7]. These operations go through several rounds of signalling between the edge and core network functions (NFs) for the provisioning of UDM services. The core NFs also share the user's information in the case of modification of policies or privileges, notification of the user subscribing or unsubscribing from a particular application [8]. This exchange of information among core NFs is to limit the users' access to the network when needed. The massive number of user requests could have similar signalling to and response from the core, which would create communication overheads on the link capacity of the core network. Such overhead reduces overall network performance by inducing substantial latency and congestion, which is potentially intolerable for real-time applications [9]. These issues in the cellular network have attracted significant attention from the research community. For example, a novel solution for signalling optimization, *Diameter Protocol*, proposed carrying out CP signalling of the LTE network [10]. The authors in [11] proposed an E2E connectivity model to handle CP signalling redundancy generated by massive IoT devices in 5G networks. Similarly, Wang et al. [12] proposed an intelligent edge management and optimization framework for 5G networks for latency-critical applications. Hund et. al., [13] proposed a hash-based grouping scheme for flow management in the MEC system. Furthermore, Cao et al. proposed a fast-authentication and data transfer scheme to reduce signalling and communication overheads in 5G mIoT networks [14]. In MEC, the existing research emphasised reducing latency via data offloading and traffic flow management between access and edge node. However, reducing latency and congestion induced between edge and core control NFs in the cellular network is still an open issue.

The proposed framework aims to prevent the entry of a control signalling storm into the core network to ensure efficient traffic flow management. This is achieved by moving the essential core NFs to the network edge. For the sake of limited edge resources, two very popular unsupervised machine learning (ML) algorithms are employed in the proposed framework for efficient admission control and resource allocation to manage the massive devices connectivity demand. The main contributions of this work are as follows: (1) a novel AI-enabled edge architecture is proposed for the core network to support heterogeneous applications and massive connectivity demand. (2) Two K-mean- and Ranking-based clustering and optimization algorithms are established for signalling optimization and efficient admission control. (3) Performance of the proposed framework is evaluated in terms of latency, link utilization efficiency, admission control, and resource allocation fairness. The outcomes are also compared with the existing schemes found in the literature.

The remainder of the paper is organized as follows: Section II presents the system model. Section III introduces the proposed framework and evaluation schemes. Results are shown in Section IV. The conclusion is presented in Section V.

II. System Model

In this section, a novel AI-enabled edge architecture has been presented for CP signalling optimization and efficient admission control in future core networks, as explained in detail in the following subsection.

A. Proposed Edge Architecture

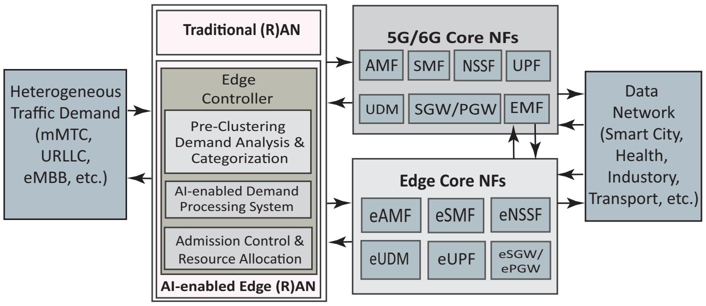
Traditional (R)AN
Heterogeneous Traffic Demand (mMTC, URLLC, eMBB, etc.)
Edge Controller
Pre-Clustering Demand Analysis & Categorization
AI-enabled Demand Processing System
Admission Control & Resource Allocation
AI-enabled Edge (R)AN
5G/6G Core NFs
AMF
SMF
NSSF
UPF
UDM
SGW/PGW
EMF
Edge Core NFs
eAMF
eSMF
eNSSF
eUDM
eUPF
eSGW/ ePGW
Data Network (Smart City, Health, Industory, Transport, etc.)

Fig. 1. Proposed AI-enabled Edge Architecture for Future Core Networks

For application latency sensitivity, the service signalling and resource demand are sent to the core network through the traditional RAN or AI-enabled edge RAN in this work, as illustrated in Fig. 1. In the AI-enabled edge RAN, the edge controller is considered as a crucial entity, which analyses and centralizes the incoming demands from various applications to ensure optimal network management in massive device connectivity and latency-sensitive situations. The controller consists of three major components: pre-clustering demand analysis and categorization, AI-enabled demand processing (clustering) system, and admission control and resource allocation. The demand analyser, as the name suggests, analyses and categorizes the application-specific services and resource demand for clustering at the edge. In this work, the processing system is employing two popular unsupervised ML algorithms i.e., K-mean- and Ranking-based clustering. The processing system processes the incoming services and resources demands based on the homogeneous characteristics by adopting these algorithms and cluster them for signalling optimization and admission control. The clustered signals for the services and resources demands are sent to the core network to fetch the user's service-specific profile from the core UDM to the edge node for efficient admission control and resource allocation. Users' service-specific profiles in the core UDM will also be clustered based on homogeneous responses and control information. This can help reduce control signalling redundancy among the core NFs generated due to similar demands and responses, which may otherwise lead to congestion and resource inefficiency in the core network. The essential core NFs are configured at the network edge to accomplish a massive number of latency-sensitive applications with faster authentication, admission control and resource allocation. These are called as *Edge Access and Mobility Management Function* (eAMF), *Edge Session Management Function* (eSMF), *Edge Network Slice Selection Function* (eNSSF), *Edge User Data Management Function* (eUDM), *Edge User Plan Function* (eUPF), and *Edge Serving Gateway/Edge Packet Gateway* (eSGW/ePGW). These core edge functions are managed by the proposed *Edge Management Function* (EMF) in the core network. To ensure network security, the edge NFs acquire only limited privileges through EMF from the core *Policy Control Function* (PCF) [3].

B. Key Notations and Description

Let us consider the deployment of an MEC network in an urban area. We consider that U total number of users, represented as $\mathcal{U} = \{1, 2, \cdots, U\}$ are associated with this network. Each user has a set of services and resources demands, represented as $\mathcal{M} = \{1, 2, \cdots, M\}$. The service demand represents the service signalling requests of the UDM offered services, denoted as a set $\mathcal{S} = \{1, 2, \cdots, S\}$, and $\mathcal{S} \subset \mathcal{M}$. A set of service operations for the service s is $\mathcal{P} = \{1, 2, \cdots, P\}$, where $s \in \mathcal{S}$. Similarly, the resource demand represents the resources requests of a particular application, denoted as a set $\mathcal{L} = \{1, 2, \cdots, L\}$, and $\mathcal{L} \subset \mathcal{M}$. We also assumed that each user can support up to K number of heterogeneous applications simultaneously, denoted as a set $\Lambda = \{1, 2, \cdots, K\}$.

III. Signalling and Admission Control

In this section, we propose an *Edge Redundancy Minimization and Admission Control* (E-RMA) framework in future core networks. The systematic diagram of the proposed framework is illustrated in Fig. 2 and discussed in detail in the following subsections.

A. Pre-clustering Demand Analysis and Categorization

The edge node continually assesses the demand of the users associated with the serving edge base stations for each application of set Λ for optimal network management in a massive-device-connectivity and latency-sensitive situation. When the users of the κth application are known to the edge controller, a demand matrix $\mathbf{V}_{e(U\times M)}$ is constructed over set $\mathcal{U}$ and $\mathcal{M}$. Each element, v_{um}, of $\mathbf{V}_e$ represents the user-application-specific service or resource demand from set $\mathcal{M}$ of κth application, where $\kappa \in \Lambda$. The pre-clustering system isolates the UDM service signalling and resource demand of application κ for clustering, as shown in Fig. 2. For service s signalling, the service operations of the edge users are populated as a row entry in the service matrix $\mathbf{A}_{s(U\times P)}$, where $s_{up} \in \mathbf{A}_s$ represents user-application-specific service demand. Similarly, resource demand for application κ contains a set of user-application-specific resource demands. These demand characteristics of the associated edge users are populated as a row entry in the resource demand matrix $\mathbf{A}_{r(U\times L)}$, where $r_{ul} \in \mathbf{A}_r$ represents the user-application-specific resource demand. Isolated services and resource demand are processed by the processing system for reducing redundancy in signalling.

B. Demand Processing (Clustering) System for Signalling

The proposed demand processing system efficiently serves massive device connectivity by minimizing the impact of constraints with the help of the optimization and clustering approach, as discussed in the following subsections.

1) Clustering for Capacity Optimization: In the network, uplink capacity, C_{up}, is the sum of the total capacity reserved for signalling and data transmission of κth application. Such as:

$$C_{up} = C_{up(sig)} + C_{up(data)}. \tag{1}$$

In the case of massive demand for κth application, the reserved capacity, C_{up}, should be greater than or equal to the observed capacity. The observed capacity, $C_{up(obs)}$, is the total capacity consumed by signalling and data transmission by κth application in the network, as shown below:

$$C_{up(obs)} = C_{up(obs\_sig)} + C_{up(obs\_data)} \leq C_{up}. \tag{2}$$

$C_{up(obs\_sig)}$ for s service and r resources demand over the set $\mathcal{U}$ can be determined as follows:

$$C_{up(obs\_sig)} = \sum_{u\in\mathcal{U}} \alpha_{(u,s)} s_{Sig(u)} + \sum_{u\in\mathcal{U}} \alpha_{(u,r)} r_{Sig(u)}\ , \tag{3}$$

where, $\alpha = 1$ only if service s or resource r demand is granted, otherwise 0. $s_{Sig(u)}$ and $r_{Sig(u)}$ are the uth user desired service and resource demand signalling. The observed capacity increases exponentially with an increase in the application demand on the edge, which creates inefficiency in resource utilization. This problem can be modelled as an optimization problem. The objective is to meet signalling demand from set $\mathcal{U}$ in such a way that efficiently utilizes the overall uplink capacity, symbolized as $C_{up(sig)}$. Mathematically, it can be described as follows:

$$\begin{aligned} \min \quad & \textstyle\sum_{u=1}^{U} C_{up(obs\_sig)}, \\ \text{s.t.} \quad & \sum_{u=1}^{U} C_{up(obs\_sig)} \leq C_{up(sig)}\ , \\ & \sum_{u=1}^{U} \alpha_{(u)} \leq 1\ , \end{aligned} \tag{4}$$

where, $u \in \mathcal{U}$. The observed signalling capacity, denoted as $C_{up(obs\_sig)}$, should not exceed the overall reserved uplink signalling capacity over set $\mathcal{U}$. For signalling, all users from set $\mathcal{U}$ should be admitted by the edge controller. In this work, a ranking-based clustering technique has been applied to service and resource demand signalling. The isolated services and resource signalling from the pre-clustering system are grouped into clusters based on their homogeneous demand. This is to fetch the user profile to the edge. Homogeneous signalling respective coefficients are populated in the ranking matrix as a single row entry for the particular cluster demand. The updated ranking matrix for s service and r resource demand will be represented as $\mathbf{A}_{\mathcal{R}_s(R\times P)}$ and $\mathbf{A}_{\mathcal{R}_r(R\times L)}$, respectively. $\mathcal{R} = \{1, 2, 3, \cdots, R\}$ is the ranking set with R possible individual clusters for service or resource signalling. The proposed clustering mechanism reduces the complexity from $\mathcal{O}(U)$ to $\mathcal{O}(R)$. Hence, after clustering, the clustered signalling reduces $C_{up(obs\_sig)}$ to make it approximately equal to or less than $C_{up(sig)}$, as shown below:

$$C_{up(obs\_sig)} = Rank(\mathbf{A}_{\mathcal{R}_s}) s_{Sig(u)} + Rank(\mathbf{A}_{\mathcal{R}_r}) r_{Sig(u)}\ , \tag{5}$$

where, $Rank(\mathbf{A}_{\mathcal{R}_s})$ and $Rank(\mathbf{A}_{\mathcal{R}_r})$ are the possible individual cluster signals for s service and r resource demand by κth application after clustering. Now, the ranking-based service signalling will be sent to the core UDM to provide the particular service. Similarly, the ranking-based resource signalling will be sent to the edge core NFs for admission control and resource allocation. Once service signalling is

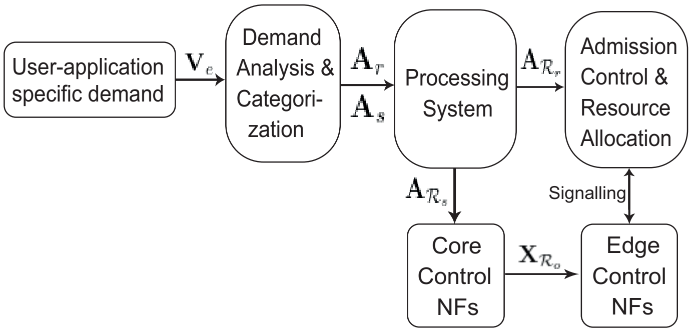
User-application specific demand
$\mathbf{V}_e$
Demand Analysis & Categori-zation
$\mathbf{A}_r$
$\mathbf{A}_s$
Processing System
$\mathbf{A}_{\mathcal{R}_r}$
Admission Control & Resource Allocation
$\mathbf{A}_{\mathcal{R}_s}$
Signalling
Core Control NFs
$\mathbf{X}_{\mathcal{R}_o}$
Edge Control NFs

Fig. 2. Systematic diagram of the proposed E-RMA framework

received, the core UDM reclusters user service responses into a response matrix, $\mathbf{X}_{\mathcal{R}_o(R\times P)}$, based on the homogeneous service-signalling response. The matrix $\mathbf{X}$ rank will be less than $\mathbf{V}$ but greater than or equal to $\mathbf{A}_{\mathcal{R}_s}$. Significantly, a clustered user service profile will be built over set $\mathcal{S}$ for signalling and information exchange in the core network. The service processing by UDM over user service signalling is illustrated in Algorithm 1. Each user service operation privilege will be accessed against the user profile in the edge UDM. $\mathcal{S}\_List_\kappa$ is the list of service s clusters maintained by the edge, where $s_\tau \in \mathcal{S}\_List_\kappa$ and $\tau \in \mathcal{R}$. If all users of cluster s_τ have the same access, then the clustered users will be added with a unique group ID, G_id, to the service list, $Service\_List_\kappa$. Otherwise, users of s_τ will be subclustered into $\mathcal{SS}\_List_\kappa$ w.r.t. possible response and added to the $Service\_List_\kappa$ or $Reject\_List_\kappa$ for admission control.

Algorithm 1: Service Signalling over Clustering

```
Input: Chose $s_\tau$ clustered service demand,
  $s_\tau \in \mathcal{S}\_List_\kappa = \{s_1, s_2, s_3, \cdots, s_{R_s}\}$,
  $Service\_List_\kappa = Reject\_List_\kappa = \emptyset$.
Output: $Service\_List_\kappa \neq \emptyset$ & $|Reject\_List_\kappa| \geq 0$
for $(i = 0,\ i < \mathcal{S}\_List_\kappa.length,\ i++)$ do
    $s_\tau \longleftarrow \mathcal{S}\_List_\kappa[i]$
    Assign a $G\_id$ to $\mathcal{S}\_List_\kappa[i]$ users
    if (Check user privileges w.r.t. $s_\tau$ matches) then
        Add $G\_id$ of the cluster users of $s_\tau$ demand in
          $Service\_List_\kappa[i]$ of edge.
    else
        sub cluster $s_\tau$ users into $\mathcal{SS}\_List_\kappa$ w.r.t.
          possible cluster service response.
        for $(j = 1,\ j < \mathcal{SS}\_List_\kappa.length,\ j++)$ do
            $ss_\tau \longleftarrow \mathcal{SS}\_List_\kappa[j]$
            Assign a $G\_id$ to $\mathcal{SS}\_List_\kappa[j]$ users
            if (check $ss_\tau$ of each associated user) then
                Add $G\_id$ of the cluster users of $ss_\tau$
                  demand in $Service\_List_\kappa[i]$ of edge.
            else
                Add $G\_id$ of the cluster users of $ss_\tau$
                  demand in $Reject\_List_\kappa[i]$ of edge.
Send $Service\_List_\kappa$ & $Reject\_List_\kappa$ towards edge
  core NFs for admission control.
```

2) Clustering for Latency Optimization: In latency-sensitive scenarios along with the massive connectivity demands, resource allocation becomes challenging for the edge operators with limited network capacity [3]. The presented latency minimization problem can be modelled as an optimization problem. The prime objective is to minimizes the mean latency, $T(\mathcal{N})$, of N optimal clusters, represented as a set $\mathcal{N} = \{1, 2, 3, \cdots, N\}$. Mathematically, it can be written as:

$$\begin{aligned} \min \quad & T(\mathcal{N}), \\ \text{s.t.} \quad & \sum_{u=1}^{U} \beta_{(u,r)} \gamma_{(u,r)} \leq \Upsilon_{(r)} \, , \\ & \sum_{u=1}^{U} \sum_{n=1}^{N} U_{(u,n)} = U \, , \end{aligned} \tag{6}$$

where, $u \in \mathcal{U}$, $n \in \mathcal{N}$, $r \in \mathbf{A}_r$. $\beta_{(u,l)} = 1$, only if demanded resource r is allocated to the user u, otherwise 0. γ is the quantity of rth resource demanded by the uth user. The aggregate resources allocated to the set $\mathcal{U}$ should not exceed the total available resources, Υ, of the particular resource r. Each user from set $\mathcal{U}$ should belong to a particular cluster with regards to the homogeneous resource demand. Thus, *K-mean* and *Ranking-based* clustering algorithms have been applied in this work for clustering the users' resource demand. It is essential to acquire an optimal solution that ensure minimum latency and efficient link utilization in the deployed network. However, this is an NP-hard problem. Thus, *Nondominated Sorting Genetic Algorithm II* (NSGA-II) has been adopted as a basic optimization method. The crucial step in NSGA-II is to define an appropriate genetic representation of set $\mathcal{N}$ over their mean resource demand r. U users are distributed into N clusters over each cluster K-mean resource demand. The similarity index, δ, among users of a cluster is obtained via the ranking-based approach. As, the objective function is to minimize mean latency, $T(\mathcal{N})$, over N clusters w.r.t. rth resource allocation, where $N \leq U$. This can be obtained by computing the latency of each cluster over its associated users r, such as:

$$\Delta_n = \sum_{u=1}^{U_n} \frac{\gamma_{(u,r)}}{\Upsilon_{(r)}} - \delta_n \sum_{u=1}^{U_n} \frac{\gamma_{(u,r)}}{\Upsilon_{(r)}} \, , \tag{7}$$

where, $\delta_n = [0, 1]$. Now, the mean latency $T(\mathcal{N})$ is:

$$T(\mathcal{N}) = \frac{1}{N} \sum_{n=1}^{N} \Delta_n \, . \tag{8}$$

Now, the MAC layer will multiplex the nth cluster resources demand into an aggregate demand, γ_n, and send a frame with a brief header for associated user identification to the physical layer for transmission. In the core network, the optimal number of clusters, along with their aggregate demand, are placed into the resource list R_List_κ for admission control and resource allocation.

C. Admission Control and Resource Allocation

In the core network, the nth cluster γ_n from R_List_κ will be assessed against available edge capacity (i.e. C_e) by the eAMF, as illustrated in Algorithm 2. If demand is within the guaranteed edge QoE bounds, users belonging to γ_n will be added to the admission queue, represented as $Admit\_List_\kappa$. In the case of $\gamma_n > C_e$, nth cluster users will go through subclustering. Their aggregate demand, $\gamma\gamma_n$, will be populated into the $\mathcal{RR}\_List_\kappa$, concerning the edge available capacity. Now subcluster aggregate demand will be assessed, and the user will be admitted into $Admit\_List_\kappa$ for resource allocation. Otherwise, the clustered users placed in $Offload\_List$ would be offload to the neighbouring edges.

In the network, each cluster demand has to be executed either locally or offloaded to a neighbouring edge node. Thus, the unidirectional E2E latency, symbolized as T_{ee}, would be computed as a sum of transmission time ($T_{(tx)} = \frac{\beta_\gamma}{\mathcal{B}}$), queuing

```
Algorithm 2: Admission Control
Input: Build $\mathcal{R}\_List_{\kappa} = \{\gamma_1, \gamma_2, \dots, \gamma_N\}$ via K-mean
  and Ranking-based clustering. Chose $\gamma_n$ as a $kth$
  application $n$th cluster demand ($\gamma_n \in \mathcal{R}\_List_{\kappa}$),
  $C_e \neq 0$, $Admit\_List_{\kappa} = 0$.
Output: $Admit\_List_{\kappa} \neq \emptyset$, $|Offload\_List| \geq 0$.
for ($i = 0,\ i < \mathcal{R}\_List_{\kappa}.length,\ i++$) do
    $\gamma_n \longleftarrow \mathcal{R}\_List_{\kappa}[i]$
    Assign a group ID to $\mathcal{R}\_List_{\kappa}[i]$ users
    if ($\gamma_n \leq C_e$) then
        Add the cluster users of $\gamma_n$ demand in
          $Admit\_List_{\kappa}[i]$ of edge.
        Update $C_e$.
        Compute $\Delta$ of cluster $n$
        Compute $C_{up(obs\_sig)}$ with cluster $n$
    else
        sub cluster $\gamma_n$ users into $\mathcal{RR}\_List_{\kappa}$ w.r.t. $C_e$.
        for ($j = 1,\ j < \mathcal{RR}\_List_{\kappa}.length,\ j++$) do
            $\gamma\gamma_n \longleftarrow \mathcal{RR}\_List_{\kappa}[j]$
            Assign a sub-group ID to $\mathcal{RR}\_List_{\kappa}[i]$
              users
            if ($\gamma\gamma_n \leq C_e$) then
                Add the sub-cluster users of $\gamma\gamma_n$
                  demand in $Admit\_List_{\kappa}[i]$ of edge.
                Update $C_e$.
                Compute $\Delta$ of cluster $n$
                Compute $C_{up(obs\_sig)}$ with cluster $n$
            else
                Add the sub-cluster users of $\gamma\gamma_n$
                  demand in $Offload\_List[i]$ for
                  offloading to the neighbouring edge
                  network.
Find opt. $T(\mathcal{N})$ and $C_{up(obs\_sig)}$ via NSGA II.
Send $Admit\_List_{\kappa}$ towards edge core NFs.
Send offload demands to the neighbouring edges from
  $Offload\_List$ .
Compute $T_{ee}$ and $\mathcal{F}_{\eta}$ over $\mathcal{U}$.
```

time ($T_{(qu)} = \Delta$), and execution time ($T_{(ex)} = \frac{c_{\gamma_n}}{C_{(exe)}}$) at each edge node. Δ denotes the user's average queuing latency by clustering. c_{γ_n} is the required computation capacity of γ_n demand. $C_{(exe)}$ is the computation capacity (i.e. CPU cycles per second) of the edge node. β_{γ} and $\mathcal{B}$ are the corresponding data size in bits and available data rate in bits per second.

$$T_{ee} = \omega_e \big(T^e_{(qu)} + T^e_{(ex)} + T^e_{(tx)}\big)\,, \tag{9}$$

where, the admission index $\omega = 1$, if request is admitted to the edge node, otherwise zero. The uth user acquired throughput, $\eta(u)$, is a product of the resource allocation probability p_r and the tolerable latency probability, $p_{T_{ee}}$. Thus, the fairness of resource allocation among users of set $\mathcal{U}$ are:

$$\mathcal{F}_{\eta} = \frac{\big(\sum_{u \in \mathcal{U}} \eta(u)\big)^2}{U \times \sum_{u \in \mathcal{U}} (\eta(u))^2}\,. \tag{10}$$

IV. Results and Discussion

To evaluate the proposed framework, a set of analytical results are presented in this section. The parameters used for the numerical analysis in MatLab are: (1) $U = [500, 10^4]$, (2) $C_e = 32$ GB, (3) $\Upsilon = 200$, (4) $C_{up} = 500$ MHz, (5) $r_{MTC} = [0.064, 1]$ Mb/s, $r_{URL} = [1, 5]$ Mb/s and $r_{MBB} = [25, 100]$ Mb/s.

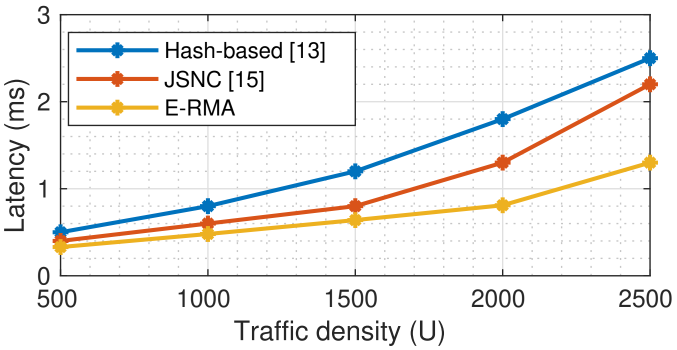

Fig. 3. latency measurements on varying traffic density, $r_{MTC} = [0.064, 1]$ Mb/s, $r_{URL} = [1, 5]$ Mb/s and $r_{MBB} = [25, 100]$ Mb/s.

A. Impact of Clustering on Latency

Fig. 3 shows mean uplink latency measurements for various traffic densities. The latency is obtained using mathematical analysis and compared to Hash-based [13], and JSNC [15]. At high demand $\mathcal{U}_e = 2500$, the achieved uplink latency is 1.3 ms, significantly lower compared to JSNC, 2.2 ms and Hash-based, 2.5 ms. The difference in performance compared to the existing schemes is by the proposed K-mean– and ranking-based clustering approach for users admission into the network, which reduces capacity overheads and results in relatively low latency in the edge network.

B. Impact of Clustering on Capacity

Fig. 4 illustrates the achieved link utilization efficiency for the given traffic density. We can observe that the achieved link utilization efficiency over the proposed framework is approximately 95% at full load (i.e. $U = 10^4$), markedly higher compared to the existing scheme in [11]. The achieved gain of the proposed framework over 5GS is 19% at $U = 10^3$ and 24% at $U = 10^4$. The rise in gain is because of the lower flow of redundant signalling to the core of the dense network. Moreover, in the case of heavy traffic load, the proposed clustering approach would save a significant amount of resources via efficient resources utilization in the network.

C. Impact of Clustering on Admission Control

Fig. 5 illustrates the user admission with and without clustering over varying traffic load. Clustered users are admitted into the network in order concerning the tolerable application latency. We can observe that the admission efficiency of clustered users is 100% in every case on the entire range of U (i.e. $U = [1000, 3000]$). This admission efficiency is gained by the optimal resource utilization, which reduces latency and congestion on the network to admit 10% more users. However, user admission without clustering will be lower in every case, as a result of signalling redundancy and congestion. This results in users being offloaded onto the neighbouring edge network. Hence, in the case of massive traffic demand, the clustering approach guarantees zero or fewer user clusters being offloaded from the edge, subject to the availability of resources and latency consideration.

D. Impact of Clustering on Resource Utilization

Fig. 6 indicates the resource allocation fairness vs time. A significant difference can be seen in the resource allocation of the proposed framework compared to its counterparts (i.e. Bankruptcy Game (BG), Equal Ratio (EQ), and Traffic Proportion (TP) [16]). The resource allocation index is approximately 1 compared to that of BG, EQ and TP, with their fairness indexes hovering around 0.99, 0.92 and 0.91. The achieved fairness is by the admission of the edge's users in form of clusters. Thus, efficient resource allocation on the arrival of the clustered request leads to maximum resource utilization and fairer resource allocation among clustered users. To summarise, the k-mean and ranking based clustering approach along with optimisation not only reduce the signalling redundancy in the access and core network but can also enhance the admission gain and resources utilisation in the future network.

V. Conclusion

In this paper, we presented a novel AI-enabled edge architecture, a clustering-based signalling control procedure and an admission control framework. We have employed two popular unsupervised ML-based K-mean and Ranking-based clustering approaches to reduce the communication overheads on the edge by reducing signalling redundancy, providing low latency and efficient resource utilization. The proposed clustering mechanism reduces the complexity from $\mathcal{O}(U)$ to $\mathcal{O}(R)$ for service signalling and $\mathcal{O}(N)$ for resource signalling. This represents a significant saving in the uplink control plane signalling and link capacity compared to the results found in the literature. Future work is to enhance the proposed framework via adopting slice elasticity in both uplink-downlink traffic flow to efficiently support the multi-edge network environment.

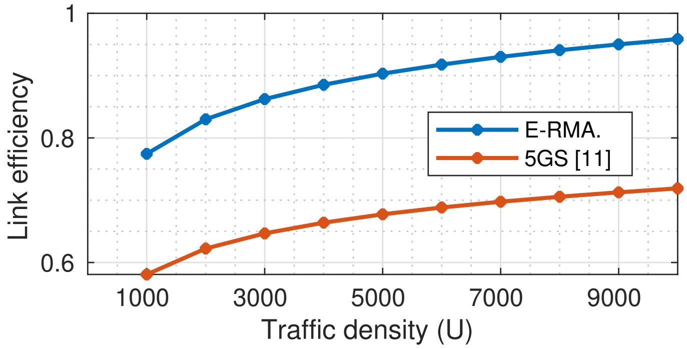

Fig. 4. Computation of link utilization efficiency due to varying traffic density, $C_{up(sig)} = 100$ MHz, $s_{sig} = 0.002$ MB/s, and $r_{sig} = 0.006$ MB/s.

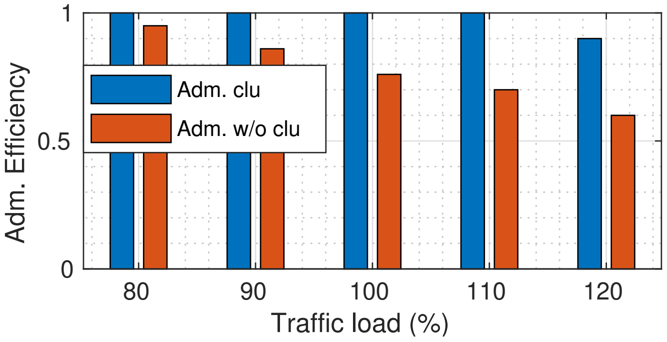

Fig. 5. Admission efficiency on varying traffic density, $U = [1000, 3000]$.

References

[1] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[2] Y. Qu, C. Dong, J. Zheng, Q. Wu, Y. Shen, F. Wu, and A. Anpalagan, "Empowering the edge intelligence by air-ground integrated federated learning in 6G networks," *arXiv preprint arXiv:2007.13054*, 2020.

[3] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[4] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *Ieee Access*, vol. 6, pp. 12 825–12 837, 2018.

[5] M. Sun, X. Xu, X. Tao, and P. Zhang, "Large-scale user-assisted multi-task online offloading for latency reduction in D2D-enabled heterogeneous networks," *IEEE Transactions on Network Science and Engineering*, 2020.

[6] H. Ullah, N. G. Nair, A. Moore, C. Nugent, P. Muschamp, and M. Cuevas, "5G communication: an overview of vehicle-to-everything, drones, and healthcare use-cases," *IEEE Access*, vol. 7, pp. 37 251–37 268, 2019.

[7] 3GPP, "Technical specification group core network and terminals; 5G system; unified data management services," *3GPP TS 29.503 Release 17*, 2020.

[8] S. Behrad, E. Bertin, S. Tuffin, and N. Crespi, "A new scalable authentication and access control mechanism for 5G-based IoT," *Future Generation Computer Systems*, vol. 108, pp. 46–61, 2020.

[9] M. Emara, M. C. Filippou, and D. Sabella, "MEC-assisted end-to-end latency evaluations for C-V2X communications," in *2018 European Conference on Networks and Communications (EuCNC)*. IEEE, 2018, pp. 1–9.

[10] J. Ewert, L. Norell, and S. Yamen, "Diameter signaling controller in next-generation signaling networks," *Ericsson Review*, vol. 284, pp. 23–31 761, 2012.

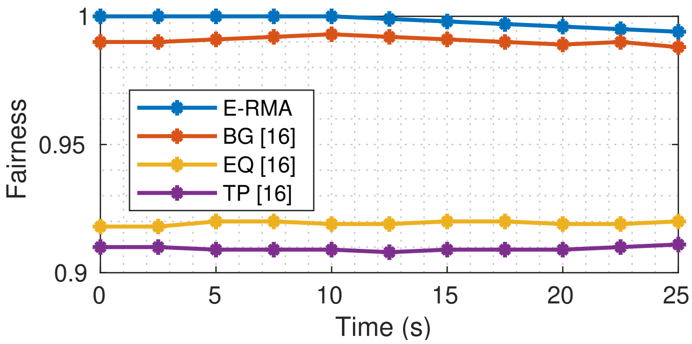

Fig. 6. Resource allocation fairness on varying traffic density, $U = [500, 2500]$.

[11] R. Trivisonno, M. Condoluci, X. An, and T. Mahmoodi, “mIoT slice for 5G systems: Design and performance evaluation,” *Sensors*, vol. 18, no. 2, p. 635, 2018.

[12] Z. Wang and Y. Cai, “Management optimization of mobile edge computing (MEC) in 5G networks,” in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[13] C.-H. Hung, Y.-C. Hsieh, and L.-C. Wang, “Control plane latency reduction for service chaining in mobile edge computing system,” in *2017 13th International Conference on Network and Service Management (CNSM)*. IEEE, 2017, pp. 1–5.

[14] J. Cao, P. Yu, X. Xiang, M. Ma, and H. Li, “Anti-quantum fast authentication and data transmission scheme for massive devices in 5G NB-IoT system,” *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9794–9805, 2019.

[15] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, “Joint network slicing and mobile edge computing in 5G networks,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[16] Y. Jia, H. Tian, S. Fan, P. Zhao, and K. Zhao, “Bankruptcy game based resource allocation algorithm for 5G Cloud-RAN slicing,” in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.