Deep Learning-Driven Two-Stage Distributed Radio Resource Allocation in 6G Cell-Free Communication Systems

Iacovos Ioannou*, Christophoros Christophorou‡, Gusan Mufti†,
Charalambos Klitis†, Vasos Vassiliou*, Christos Verikoukis§
* Department of Computer Science, University of Cyprus and CYENS - Centre of Excellence, Cyprus
† eBOS Technologies Ltd, Nicosia, Cyprus
‡ University of Central Lancashire (UCLan), Larnaca, Cyprus
§ Industrial Systems Institute (ISI), Patras, Greece

***Abstract*—The densification of wireless networks in 6G introduces severe co-channel interference, making traditional centralised radio resource allocation (RRA) schemes ineffective. This issue is particularly challenging in cell-free architectures, where dense deployments of user equipment (UE) and distributed access points (APs) demand scalable, interference-aware, and latency-efficient solutions. To address this, we propose a novel two-stage distributed RRA framework that employs deep learning for dynamic frequency band (FB) assignment. In the first stage, a fully connected autoencoder extracts latent representations of the UEs' spatial distribution. These latent representations are then clustered using silhouette-optimised K-Means, which adaptively determines the number of clusters. The resulting cluster heads are selected to serve as UE-based virtual base stations (UE-VBSs). A novel "in–out" spectrum reuse policy is then applied to assign FBs based on cluster proximity to the central base station. In the second stage, extracted link and cluster features (such as SINR, data rate, and UE location) are used to train a generative adversarial network (GAN)-based classifier that predicts optimal FB allocations. This two-stage approach enables an efficient, scalable and interference-aware solution making it well-suited for real time RRA allocation in ultra-dense, cell-free 6G environments. Simulation results show that our K-Means–GAN pipeline achieves 89% FB assignment accuracy and minimizes residual interference to 1.10×10^{-7} W, outperforming deep clustering (DTC, DCN) and alternative classifiers (MLP, Wide & Deep, Random Forest) across multiple network sizes.**

***Index Terms*—6G networks, cell-free communication, distributed radio resource allocation, deep clustering, machine learning, classification models, scalable wireless networks.**

I. Introduction

Dense deployments push today's cellular infrastructure to its limits. As even more access points (APs) and user equipment (UEs) contend for the same narrow frequency slices, co-channel interference increases and legacy resource-allocation heuristics struggle to keep pace. The strain is most acute within the flagship 5G service classes. Massive Machine-Type Communications (mMTC) must simultaneously accommodate tens of thousands of battery-powered IoT devices per square kilometre, however the resulting signalling overhead and random-access collisions significantly increase latency and energy consumption [1]. Moreover, in enhanced Mobile Broadband (eMBB), the push for fibre-like data rates collides with aggressive spectrum reuse. As a result, neighbouring cells causing inter-cell interference (and not just thermal noise) become the primary throughput bottleneck once layouts turn ultra-dense [2]. To overcome the limitations of cell-bound architectures, the emerging 6G paradigms favours for *cell-free* (user-centric) networks in which swarms of distributed APs act collectively as a coherent single large-scale antenna array. By eliminating hard cell borders and orchestrating joint transmission and reception over a lightweight fronthaul mesh, cell-free massive-MIMO architectures promise near-uniform service quality, effective interference suppression, and significantly reduced power consumption on the user-side, even under extreme UE densities [3]. Realising these gains, however, demands intelligent spectrum-reuse strategies that sense spatial load fluctuations and adapt to spatial load dynamics, learn fast, and scale without the latency overhead of centralised handover procedures.

The present study addresses these demands through a **two-stage distributed RRA learning framework** that couples latent-space representation learning with *K*-means clustering in the first stage and a GAN-based supervised classifier in the second, enabling interference-aware frequency band (FB) allocation. In the first stage, a fully connected autoencoder compresses high-dimensional UE snapshots into a compact latent manifold. The latent vectors are then clustered using vanilla *k*-Means, which outperforms deep clustering alternatives (DTC, DCN) across power efficiency, throughput and cluster quality. The number of clusters is not fixed beforehand. Instead, it is determined adaptively by maximising clustering validity indices, including the Silhouette score, Davies–Bouldin index and Calinski–Harabasz index criteria.

Once the clusters are formed, an "in–out" spectrum reuse policy is employed, where exclusive frequency bands are first allocated to the cluster heads, and then those same bands are reused at the spatially most isolated edges of the network. This strategic reuse significantly reduces intra-cluster interference. In the second stage, the resulting descriptors (such as UE coordinates, distance to the AP, instantaneous SINR, estimated data rate and a cluster label) are fed into a grid-tuned **generative-adversarial-network (GAN) classifier**. This model attains **89%** FB assignment accuracy, a 4% gain over an MLP (achieving 85%) and 3% gain over a Wide & Deep (achieving 86%) network, while maintaining a training time below 85 s. The GAN also produces the lowest residual interference (1.10×10^{-7} W), confirming that higher assignment accuracy translates into a cleaner radio environment. Comprehensive simulations confirms that the revised K-Means → GAN pipeline boosts spectral efficiency, reduces aggregate transmit power and scales grace-

ISBN 978-3-903176-72-0 © 2025 IFIP

fully with device density, which are capabilities essential for the distributed, fronthaul-lean operation envisioned in next-generation cell-free 6G Networks.

The main contributions of this work are as follows:

- Proposes a novel two-stage framework, combining clustering and classification for radio resource allocation in 6G cell-free networks.
- Introduces an adaptive interference-aware clustering approach, using K-Means and deep clustering methods to group UEs and select cluster heads (i.e, the UE-VBSs) based on spatial characteristics.
- Trains a GAN-based classifier that outperforms other baseline models in predicting optimal frequency bands allocations for UEs.
- Designs a novel in–out frequency reuse strategy to minimize interference within clusters and enhance spectral efficiency.

The remainder of the paper is organized as follows. Section II provides background information regarding the RRA and the ML approaches employed in this study. Section III describes the system architecture of our proposed 6G cell-free network, outlining the roles of Free Cells Access Points (APs) and UE. In Section IV, we present our integrated methodology, including the generation of synthetic datasets, clustering techniques, feature extraction, and the design and training of the classification models used for predicting optimal FB allocations. Section V evaluates and compares the efficiency of the investigated approaches under scenarios with varying device densities. Finally, Section VI concludes the paper by summarizing our key findings and discussing potential directions for future research.

II. Background Work

Cell-free communication eliminates traditional cell boundaries by deploying a large number of distributed access points (APs) that coherently serve all user equipment (UEs) without cell-centric handovers. Unlike conventional cellular networks, where each UE is typically linked to a single base station, in cell-free systems each UE can be served cooperatively by multiple APs, thereby providing uniform service quality while mitigating inter-cell interference [4], [5].

To further enhance flexibility and resource management, the concept of User Equipment as Virtual Base Stations (UE-VBS) [6] leverages the UEs of the end-users to act as local coordinating points—providing caching, relaying, and resource-allocation functions, thereby reducing latency and backhaul load [7]. After clustering the UEs based on their spatial characteristics, one UE per cluster is selected as the cluster head and designated as a UE-VBS. Notably, the UE-VBS concept extends the role of UEs from passive terminals to active network nodes, enabling scalable and dynamic fronthaul-less operation in 6G deployments.

Below, we provide background information regarding the RRA and the ML approaches employed in this study. RRA refers to the process of assigning available radio spectrum, time-slots, and transmission power among UEs to optimize network throughput, fairness, and energy efficiency [8]. An interesting work regarding DRRA that is using the DAI framework along with Federated Learning in 6G networks, which also achieves high accuracy, is at [9].

A. Background on the Cluster Approaches

Effective resource allocation is paramount in modern wireless communications, especially in dense networks such as those found in 5G/6G mMTC and eMBB scenarios and emerging 6G cell-free architectures. This section reviews the primary methods employed in our framework, describing each approach in dedicated subsections. We detail the various clustering techniques, their formulations, and the associated FB assignment strategies that are used to mitigate interference and optimize system performance.

1) Enhanced K-Means (vanilla K-Means)

K-means iteratively refines cluster assignments and centroids to minimise within-cluster variance. Its simplicity and linear scalability make it well suited for partitioning static feature vectors, though susceptibility to local minima and sensitivity to initialisation remain practical concerns [10], [11]. Given a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ and a prescribed number of clusters K, the K-means objective is

$$\min_{\{\mu_k\}_{k=1}^{K}, \{r_{ik}\}} J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2,$$

where $r_{ik} \in \{0, 1\}$ encodes the assignment of point i to centroid k with $\sum_k r_{ik} = 1$, and $\mu_k \in \mathbb{R}^d$ is the k-th centroid. Lloyd's algorithm alternates between an *assignment step*, $r_{ik} \leftarrow \mathbf{1}\big(k = \arg\min_j \|\mathbf{x}_i - \mu_j\|_2^2\big)$, and a *centroid-update step*, $\mu_k \leftarrow \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_i r_{ik}}$, monotonically decreasing J until local convergence, although finding the global optimum is NP-hard [10].

After projecting each input $\mathbf{x}_i$ into its d-dimensional latent representation $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$, we cluster the set $\{\mathbf{z}_i\}_{i=1}^{N}$ with vanilla K-means. Centroids are initialised with the K-means++ heuristic to reduce the probability of poor local optima [12]. The procedure iterates assignment and update steps until the relative decrease in objective falls below 10^{-4}. The number of clusters K is chosen by the elbow criterion [13] and corroborated with silhouette analysis [14]. In the proposed appoarach, the clustering procedure is repeated ten times with different random seeds, and the solution yielding the lowest objective value is retained.

2) Deep Temporal Clustering (DTC)

DTC integrates a temporal autoencoder with a differentiable clustering layer. Joint optimization of reconstruction and KL-based clustering losses preserves temporal structure in the latent space while encouraging separation among sequence-level centroids, producing coherent and discriminative clusters for multivariate time-series data [15]. Consider n length-T multivariate time series $\{\mathbf{x}_{1:T}^{(i)}\}_{i=1}^{n}$. A temporal autoencoder (E_θ, D_ϕ) yields latent sequences $\mathbf{z}_{1:T}^{(i)} = E_\theta(\mathbf{x}_{1:T}^{(i)})$. With learnable centroids $\mathbf{c}_k \in \mathbb{R}^{m \times T}$ and a temporal distance $d(\cdot, \cdot)$, soft assignments are computed by a Student kernel

$$q_{ik} = \frac{\big(1 + d(\mathbf{z}_{1:T}^{(i)}, \mathbf{c}_k)/\alpha\big)^{-\frac{\alpha+1}{2}}}{\sum_{j=1}^{K} \big(1 + d(\mathbf{z}_{1:T}^{(i)}, \mathbf{c}_j)/\alpha\big)^{-\frac{\alpha+1}{2}}},$$

then sharpened via $p_{ik} = q_{ik}^2 \big/ \sum_{j=1}^{K} q_{ij}^2$. The end-to-end loss

$$\mathcal{L} = \sum_{i,t} \|\mathbf{x}_t^{(i)} - D_\phi(\mathbf{z}_t^{(i)})\|_2^2 + \lambda \sum_{i,k} p_{ik} \log \frac{p_{ik}}{q_{ik}}$$

balances reconstruction fidelity and KL-divergence clustering, letting gradient descent jointly tune network weights and centroids [15].

3) Deep Clustering Network (DCN)

DCN extends basic autoencoding by embedding an explicit K-means distortion term in the training objective. The alternating scheme simultaneously reconstructs inputs and sculpts the latent manifold into well-separated, centroid-friendly regions, giving more stable assignments and efficient inference compared with post-hoc clustering [16]. DCN learns an encoder–decoder pair (f_θ, g_ϕ) that maps $\mathbf{x}_i \mapsto \mathbf{z}_i = f_\theta(\mathbf{x}_i) \in \mathbb{R}^m$ while forcing the codes to form K tight spherical clusters. With latent centroids μ_k and hard assignment $s_i = \arg\min_k \|\mathbf{z}_i - \mu_k\|_2^2$, the objective

$$\min_{\theta,\phi,\{\mu_k\}} \sum_{i=1}^{n} \|\mathbf{x}_i - g_\phi(\mathbf{z}_i)\|_2^2 + \gamma \sum_{i=1}^{n} \|\mathbf{z}_i - \mu_{s_i}\|_2^2$$

alternates between stochastic-gradient updates of (θ, ϕ) and closed-form centroid updates, yielding an embedding space explicitly shaped for K-means [16].

B. Background on the Classification Approaches

Beyond clustering, our framework relies on supervised models to label devices or resource blocks once coarse structure has been inferred. The methods span generative-adversarial learning, hybrid linear–nonlinear predictors, classical bagging ensembles, and a plain multilayer perceptron used as a baseline. Each technique is summarized below with its mathematical formulation and key training objective.

1) GAN Dense Classifier

A GAN-based dense classifier augments a feed-forward discriminator $D_{\theta_d} : \mathbb{R}^d \to [0,1]^{C+1}$ with a generator $G_{\theta_g} : \mathbb{R}^\ell \to \mathbb{R}^d$ that synthesizes pseudo-samples. Let $y \in \{1, \dots, C\}$ denote the true class and $C+1$ the *fake* label. The discriminator minimises

$$\mathcal{L}_D = -\mathbb{E}_{(\mathbf{x},y)\sim p_{\text{data}}}\big[\log D_y(\mathbf{x})\big] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\big[\log D_{C+1}\big(G_{\theta_g}(\mathbf{z})\big)\big],$$

while the generator minimises the feature-matching objective

$$\mathcal{L}_G = \big\|\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\psi(\mathbf{x}) - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}\psi\big(G_{\theta_g}(\mathbf{z})\big)\big\|_2^2,$$

where $\psi(\cdot)$ is the penultimate dense layer of D_{θ_d} that embeds inputs into $\mathbb{R}^m$. Alternating stochastic-gradient updates of (θ_d, θ_g) enlarge the effective training set and regularise D_{θ_d}, giving improved generalisation in data-sparse wireless scenarios [17].

2) Wide & Deep Neural Network

The Wide & Deep model combines a linear "memorisation" component with a multilayer perceptron that captures higher-order interactions [18]. Given raw features $\mathbf{x} \in \mathbb{R}^d$ and learned nonlinear features $\boldsymbol{\phi}(\mathbf{x}) = f_\theta(\mathbf{x}) \in \mathbb{R}^m$ produced by L hidden layers,

$$p(y = c \mid \mathbf{x}) = \sigma\big(\mathbf{w}_c^\top \mathbf{x} + \mathbf{v}_c^\top \boldsymbol{\phi}(\mathbf{x}) + b_c\big),$$

with class-specific weights $(\mathbf{w}_c, \mathbf{v}_c, b_c)$. The parameters of both branches are trained jointly by minimising the cross-entropy

$$\mathcal{L} = -\sum_{i=1}^{n} \sum_{c=1}^{C} \mathbf{1}(y_i = c) \log p(y_i = c \mid \mathbf{x}_i).$$

The wide term memorises sparse manual crosses (e.g., CQI × location bin), whereas the deep term generalises to unseen feature combinations, an asset in non-stationary radio environments.

3) Baseline Multilayer Perceptron

The MLP is a standard feed-forward network with L hidden layers. For input $\mathbf{x}$, activations are

$$\mathbf{h}^{(0)} = \mathbf{x}, \qquad \mathbf{h}^{(\ell)} = \sigma\big(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\big),\ \ell = 1, \dots, L,$$

and the softmax output

$$p(y = c \mid \mathbf{x}) = \frac{\exp\big(\mathbf{u}_c^\top \mathbf{h}^{(L)} + a_c\big)}{\sum_{j=1}^{C} \exp\big(\mathbf{u}_j^\top \mathbf{h}^{(L)} + a_j\big)}.$$

Training minimises the same cross-entropy as above by back-propagation [19]. Although lacking the specialised structure of the other models, this MLP provides a competitive baseline and a sanity check for gains brought by more elaborate architectures.

4) Random Forest Ensemble

A random forest constructs T decision trees $\{h_t\}_{t=1}^{T}$, each trained on a bootstrap sample and split using a random subset of features [20]. The ensemble prediction is the majority vote

$$\hat{y} = \arg\max_c \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\big(h_t(\mathbf{x}) = c\big).$$

Bagging reduces variance, and random feature selection decorrelates trees, yielding strong out-of-the-box accuracy and robustness to heterogeneous feature scales commonly encountered in radio-access measurements.

III. System Description

Figure 1 depicts a flexible, cell-free network in which a distributed layer of Free-Cells access points (APs) cooperates under a central unit. Each UE that is selected to become AP can deliver extended services and even behave as user-equipment virtual base stations (UE-VBSs [21])—for example, by caching content—so the same hardware both relays data and, when needed, serves end users' storage and computation demands. Before frequency-band (FB) assignment, the system clusters UEs to clustered UEs and UE-VBSs according to their locations and connectivity characteristics, revealing natural groupings and allowing the network to choose the most suitable subset of APs for every cluster. These localized clusters enable targeted interference control and resource management.

Next, a machine-learning classifier, whose hyperparameters are exhaustively tuned, uses features from the clustering stage to predict the optimal FB for each UE-VBS, thereby further suppressing interference and boosting spectral efficiency. By fusing AP selection through clustering with ML-driven FB allocation, the framework delivers the dynamic, distributed resource management that emerging cell-free and 6G deployments require, sustaining high data rates and low power consumption while remaining inherently scalable [5], [22], [23].

IV. Methodology

This section details the experimental methodology used in this study. It outlines the data generation process, the construction of feature sets, the clustering and classification

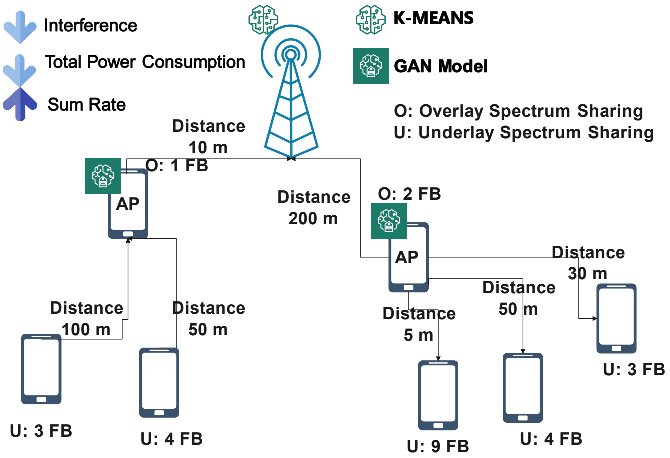

Fig. 1: The Distributed Radio Resource Allocation System.

model architectures, and the procedures followed for training, optimization, and performance evaluation.

A. Data Generation for the Dataset

A simulation algorithm is employed to generate the dataset. UEs with integrated agent functionalities are randomly positioned within a defined area representing a cell-free 6G network. Our methodology uses a clustering process to group them based on their spatial distributions after the UEs are deployed. The device closest to the cluster centroid is selected from each resulting cluster to serve as a Free Cells Access Point (AP) or User Equipment as Virtual Base Station (UE-VBS[1]). These selected UE-VBS devices function as local coordinators for resource management within their clusters.

A dynamic FB allocation algorithm is executed once the clusters and their corresponding UE-VBS (Free Cells APs) are determined. This algorithm leverages the spatial information by calculating the distances between each UE and its assigned UE-VBS, as well as the distances between the UE-VBS and a centralised coordinating unit (e.g., WiFi AP, Base Station) acting as gateway. The key objective is to allocate FBs in such a way that each cluster receives a unique set of frequency bands, which minimizes interference and maximizes resource utilisation across the network. The pseudocode in Algorithm 1 summarizes the RRA procedure:

In summary, data generation in our methodology involves creating a realistic simulation of a distributed cell-free 6G network by: i) Randomly positioning UEs within a defined area; ii) Employing clustering algorithms to group UEs; iii) Selecting as Free Cells APs (UE-VBS) the devices closest to the cluster centroids; and iv) Dynamically allocating frequency bands based on distance criteria to ensure efficient resource utilisation and minimal interference.

B. Features & Descriptions

Table I presents the principal attributes extracted during simulation and clustering that feed the machine-learning models that will assign to UE a frequency band (FB). Each feature stems either from the UE's spatial coordinates or from network-performance metrics gathered while assigning frequency bands. Specifically, the dataset records the UE's location, its distance to the chosen cluster head (i.e., the device nearest the cluster centroid acting as the Free Cells Access Point, or UE-VBS), the calculated SINR (in dB), the predicted data rate, and the cluster label assigned in the clustering stage.

Algorithm 1 Radio Resource Allocation Algorithm

```
Require: List of UEs with integrated agent functionalities,
    Clustering results (including cluster centroids and selected UE-VBS), Distance information
Ensure: Allocation of underlay FBs (reused) to each UE and
    overlay FB (no to be reused) to UE-VBS
Select UE-VBS: Based on clustering outcomes, select as UE-VBS the device closest to the centroid in each cluster and assign an overlay FB.
Compute Distances: Calculate distances between each UE and its respective UE-VBS, and between each UE-VBS and the centralised coordinating unit.
Initialize FB List: Create a list of available (reused) underlay FBs.
for each UE in the network do
    if there are available (reused) underlay FBs then
        Directly assign an unused underlay FB to the UE.
    else
        Compute the distance from the UE to its corresponding UE-VBS.
        Assign a FB based on this distance and the priority of the UE-VBS.
    end if
end for
Output: FB allocation for each UE.
```

TABLE I: Features and descriptions for model training.

Feature	Description
X Position	Denotes the horizontal spatial coordinate of the UE within the simulated deployment area. This value helps model spatial distribution and user mobility patterns.
Y Position	Represents the vertical spatial coordinate of the UE, used alongside the X position to calculate relative distances and coverage zones in the 2D simulation space.
Distance to Cluster Head	Measures the straight-line (Euclidean) distance from the UE to its respective cluster head, which is determined as the node closest to the geometric centre of the cluster. This metric is critical for evaluating connectivity efficiency and signal degradation.
SINR (dB)	The Signal-to-Interference-plus-Noise Ratio in decibels. It quantifies the quality of the received signal by comparing the power of the desired signal to the combined power of interference (from other UEs using the same frequency) and background noise.
Data Rate (Mbps)	Refers to the estimated transmission rate achievable by the UE, typically computed based on the allocated bandwidth and the corresponding SINR value using standard channel capacity formulas.
Cluster	Indicates the cluster identifier assigned to the UE during the clustering phase. This label signifies which group the UE belongs to for localized resource coordination and allocation.

C. Architecture of Clustering and Classification Approaches

1) Architecture of the Clustering Approaches

The first stage of the framework is tasked with uncovering latent topological structure in the UE deployment so that radio resources can be coordinated on a per-cluster basis. Three algorithms of escalating model complexity are explored—namely, the centroid-based K-Means, the sequence-aware Deep Temporal Clustering (DTC), and the representation-learning Deep Clustering Network (DCN). Together these cover the continuum from purely geometric grouping to fully data-driven, end-to-end training.

a) K-Means Clustering Architecture

K-Means remains a competitive baseline for spatial partitioning because it is parameter-free, deterministic under k-means++, and scales linearly with the number of UEs [10], [24]. In each iteration the following are repeated:

[1] UE-VBS is an extension of the AP, it provides additional services such as caching and content sharing

- **Initialisation:** K centroids are seeded to maximise inter-centroid distance, mitigating the local-minimum problem inherent in random starts.
- **Assignment:** A UE $\mathbf{x}_i$ is labelled with the index of its nearest centroid in Euclidean distance, $c_i = \arg\min_k \|\mathbf{x}_i - \mu_k\|_2$.
- **Update:** The centroid is relocated to the arithmetic mean of its members, $\mu_k = \frac{1}{|C_k|}\sum_{i\in C_k}\mathbf{x}_i$, thereby minimising the within-cluster sum of squares (WCSS).

The cycle is repeated until the fractional reduction in WCSS falls below 10^{-4}. To balance granularity against interference over-segmentation, K is selected from the range $1 \dots 10$ by maximising the silhouette coefficient, ensuring well-separated yet cohesive UE groups.

b) Deep Temporal Clustering (DTC)

DTC targets deployments in which the spatial footprint alone is insufficient to reveal meaningful clusters, for example when UEs are quasi-static but experience heterogeneous temporal fading profiles [25]. Its pipeline is as follows:

- **Input representation:** Each UE contributes a short SINR trace of length $T = 64$ samples, capturing fast fading and intermittent blockage events in sub-6-GHz bands.
- **Encoder:** Two bidirectional LSTM layers (128 and 64 hidden units) digest the sequence into a compact code $\mathbf{z}_i \in \mathbb{R}^{64}$. The use of recurrent gates enables the model to retain long-term interference context while remaining parameter-efficient.
- **Clustering Layer:** A set of trainable centroids $\{\mu_j\}$ is maintained; soft assignment is realised by a Student's-t kernel $q_{ij} \propto (1 + \|\mathbf{z}_i - \mu_j\|_2^2)^{-1}$, which is heavy-tailed and thus more tolerant to outliers than a Gaussian kernel.
- **Optimisation:** Every five epochs the target distribution $p_{ij} = q_{ij}^2 / \sum_l q_{il}$ is recomputed, and the encoder is updated by minimising $\mathrm{KL}(P\|Q)$. A secondary reconstruction loss on the input sequence enforces temporal fidelity. The dual objective steers the latent space to be both information-preserving and cluster-friendly.

Model selection is again conducted with a silhouette sweep over K, yielding a data-adaptive number of clusters without manual tuning.

c) Deep Clustering Network (DCN)

DCN generalises the concept of fitting a Gaussian mixture model in an unknown feature space by learning that feature space jointly with the cluster centroids [26]. Its architecture is composed of:

- **Encoder–Decoder Backbone:** Five dense layers progressively compress the 2-D coordinates into a 10-D manifold, while a mirrored decoder reconstructs the input. Layer sizes follow a geometric progression (1024 $\to$ 512 $\to$ 256 $\to$ 128 $\to$ 10) to maintain a smooth information bottleneck.
- **Latent K-Means Block:** After each mini-batch a closed -form centroid update is performed in the latent space, $\mu_k \leftarrow \frac{1}{|C_k|}\sum_{i\in C_k}\mathbf{z}_i$. The additional loss term $\lambda \cdot \|\mathbf{z}_i - \mu_{c_i}\|_2^2$ ($\lambda = 0.1$) explicitly shapes the embedding to favour spherical, well-separated clusters.
- **Training Regime:** Fifty epochs of auto-encoder pre-training align the reconstruction and latent objectives, followed by fifty epochs of joint optimisation with momentum SGD (learning rate 0.01), resulting in a stable convergence profile even under highly non-Gaussian UE layouts.

The trio, K-Means, DTC, and DCN, provides a flexible toolkit: K-Means for lightweight, geometry-dominated scenes; DTC for temporal diversity; and DCN for sophisticated, non-linear spatial manifolds. roids can be interchanged without adjusting the downstream FB allocator.

2) Architecture of the Classification Approaches

Given the cluster index and local link attributes, the second stage predicts a discrete frequency band (FB) allocation. The input vector **f** concatenates coordinates, distance to the cluster head, instantaneous SINR, Shannon spectral efficiency, and one-hot cluster membership, resulting in a 70-dimensional feature space. Four classifiers encapsulate different modelling philosophies.

a) GAN Dense Classifier

Borrowing the discriminator topology of the Auxiliary-Classifier GAN [27], the network comprises:

- **Two to three** fully connected layers of 256–512 ReLU neurons equipped with batch normalisation and 20–30 % dropout for regularisation.
- **A softmax head** producing posterior probabilities over the N_{FB} frequency bands.
- **Training with sparse cross-entropy** under the Adam optimiser ($lr = 10^{-3}, \beta_1 = 0.9$) for up to 20 epochs; the model is early-stopped when the validation loss fails to improve for three consecutive epochs.

Although no generator is used at inference, the discriminator-style design retains the GAN's capacity for class-specific feature separation.

b) Wide & Deep Neural Network

The Wide–Deep paradigm jointly learns low-degree feature interactions and higher-order patterns [28].

- **Wide component:** A single linear layer acts on raw spatial coordinates and SINR, directly memorising rare but predictive feature crosses.
- **Deep component:** A three-layer tower (256–128–64 units, ReLU activations, 30 % dropout) captures non-linear interactions among cluster dummies and rate estimates.
- **Fusion and output:** The logits from both streams are concatenated and passed to a final softmax layer. The model is optimised with AdaGrad ($lr = 0.01$), which suits the sparse–dense hybrid input.

c) Baseline Multilayer Perceptron

A standard multilayer perceptron serves as a lower-bound reference [29].

- **Architecture:** Two or three hidden layers, each with 256–512 ReLU units followed by 25 % dropout.
- **Optimisation:** Trained with Adam using a learning rate of 10^{-3}.
- **Performance:** Despite its simplicity, a well-tuned MLP can achieve competitive accuracy when rich feature engineering is available.

d) Random Forest Ensemble

Random Forest supplies a non-parametric, interpretable baseline [30]:

- **An ensemble** of 200 decision trees is grown to full depth on bootstrap samples, each split considering a random $\sqrt{d}$ feature subset.
- **Class probability** estimates are obtained by averaging the soft-class counts across trees, with optional class weighting to mitigate label imbalance.

Collectively, these classifiers expose a rich accuracy–interpretability–latency design space, ensuring that the FB allocation layer can be tuned to the operational constraints of any target network.

D. Overall Methodology, Optimization and Validation

Our methodology integrates two main stages, that is, clustering and classification, to achieve dynamic and interference-aware radio resource allocation in dense, cell-free 6G networks. The process involves generating synthetic UE positions, clustering them to select optimal cell-free Access Points (APs) (to become, in a later stage, UE-VBS), extracting features from clustering and interference analysis, and finally training classification models to predict optimal FB allocations. This section details the overall workflow, including the optimization and validation procedures.

Algorithm 2 Clustering Evaluation & Best Method Selection

```
Require: Synthetic UE positions for various network sizes
    (e.g., 100, 200, ..., 1000 UEs)
for each network size in {100, 200, ..., 1000} do
    for each clustering method in {K-Means, DTC, DCN} do
        Apply the clustering method to obtain cluster labels and centroids.
        Identify the UEs nearest to each centroid and designate them as Access Points (APs) or UE-VBS.
        Process clusters and assign frequency bands (FBs) and compute performance metrics:
          • Assign non-reused FBs (overlay FB) to the APs (one per cluster).
          • For each remaining UE in the cluster:
              – Identify the furthest AP from the current cluster's AP, based on inter-AP distance.
              – From the furthest cluster, select the UE that is furthest from its AP.
              – If this UE has an assigned FB, reuse that FB for the current UE (underlay FB).
              – If not, assign an available underlay FB that has not yet been reused.
          • After FB assignments, compute:
              – Clustering quality metrics: Silhouette Score, Davies-Bouldin Index, Calinski-Harabasz Index.
              – Network performance metrics: Average data rate, Total power consumption.
    end for
    Select the clustering method with the highest average data rate and lowest total power consumption.
end for
return The best clustering method and corresponding metrics for each network size.
```

a) Stage 1: Clustering and Free Cells AP Selection

Synthetic data is generated to simulate the spatial positions of UE for varying network sizes (ranging, for example, from 100 to 1000 UEs). Multiple clustering algorithms are then applied to this data:

1) **Clustering Methods:**
   - **K-Means:** Clusters (partitions) UE positions by minimizing within-cluster variance.
   - **DEC:** Uses a fully-connected autoencoder to learn latent representations and then refines clustering via a dedicated clustering layer.
   - **DTC (Deep Temporal Clustering):** Extracts temporal features and clusters in latent space using a temporal autoencoder.
   - **DCN (Deep Clustering Network):** Jointly minimizes reconstruction and clustering loss.
2) **Optimization:** For each clustering algorithm, the number of clusters is varied (e.g., in a specified range), and performance metrics, such as the silhouette score, Davies-Bouldin index, and Calinski-Harabasz index, are computed to evaluate clustering quality. These metrics, along with interference measurements (e.g., SINR and data rates), are used to select the best clustering method for each network size.

Algorithm 2 benchmarks three clustering schemes, K-Means, DTC, and DCN, across growing network sizes while embedding an interference-aware frequency-reuse policy. For each scenario it quantifies clustering quality and network-level KPIs, then selects the scheme that maximises average data rate and minimises total power consumption, delivering the most spectrally- and energy-efficient topology.

b) Stage 2: Classification Model Training and Selection

Features are then extracted from the clustering outcomes and interference analysis. The extracted features include:

- **Spatial Coordinates:** The X and Y positions of each device.
- **Distance to the Cluster Head:** The Euclidean distance from each device to its corresponding cluster head (or Free Cells AP), which reflects the proximity of the device to the cluster's representative.
- **SINR (in dB):** The Signal-to-Interference-plus-Noise Ratio computed from the interference measurements is critical for assessing the link quality.
- **Estimated Data Rate (in Mbps):** The data rate estimated based on the computed SINR and available system bandwidth.
- **Raw Cluster Label:** The discrete cluster identifier obtained directly from the clustering algorithm.
- **One-Hot Encoded Cluster Labels:** A binary (one-hot) encoding of the cluster labels to capture categorical membership information in a format suitable for classification.

These features serve as input to the classification models that predict the optimal FB allocation for each UE. The classification module includes several models:

- **GAN Dense Classifier:** A discriminator-style feed-forward network with two–three Dense (ReLU) layers, batch normalisation, 20–30% dropout, and a softmax head that outputs multi-class FB probabilities.

- **Wide & Deep Neural Network:** A hybrid architecture that concatenates a linear "wide" layer (memorising sparse feature crosses) with a three-layer deep tower (256–128–64 ReLU units, 30% dropout) before a softmax FB output.
- **MLP:** A standard multilayer perceptron comprising two–three Dense (ReLU) layers, 25% dropout for regularisation, and a softmax output for multi-class FB prediction.
- **RF:** A random-forest ensemble of 200 fully grown decision trees, each trained on bootstrap samples and random feature subsets, whose averaged votes provide FB class probabilities.

Algorithm 3 summarizes the classification model training and selection process.

Algorithm 3 Classification Model Training and Selection

```
Require: Feature set extracted from clustering and interference analysis, with corresponding FB labels.
Split the dataset into training (80%) and testing (20%) sets.
for each classification model in {GAN, Wide & Deep NN, MLP, and RF} do
    Perform hyperparameter tuning using GridSearchCV with k-fold cross-validation.
    Evaluate each candidate model on the validation set.
end for
Select the model with the best validation performance (lowest loss or highest accuracy).
return The best classification model and its optimised hyperparameters.
```

For the four candidate predictors, such as GAN Dense classifier, Wide & Deep network, vanilla MLP and Random-Forest (RF), we carry out a GridSearchCV sweep with two-fold cross-validation on an 80:20 train–test split, using z-score–normalised synthetic features. Tuned hyperparameters include the number of hidden layers, neurons, dropout rate, batch size, and learning rate for the three neural networks, and tree count, maximum depth, and class weighting for RF. The grid search selects the configuration that maximises fold-level validation accuracy; that model is then retrained on the full training fold and evaluated on the hold-out set. Test-set accuracy (after merging single-sample classes), together with auxiliary metrics, is compared across the four approaches to identify the most reliable frequency-band allocator.

1) Overall Evaluation and Model Selection

The evaluation process integrates performance outcomes from both the clustering and classification stages, ensuring optimal FB allocation and resource management. The process is organized as follows:

1) **Clustering Evaluation:** For each network size and clustering method, clustering performance is quantified using the silhouette score, Davies-Bouldin index, Calinski-Harabasz index, and interference-related metrics (such as average interference, sum rate, total power consumption), as well as the execution time. The best clustering method is selected based on its overall performance—typically the one that yields the highest average data rate and lowest power consumption while maintaining high cluster quality. The centroids from the best clustering method are used to identify Free Cells APs (i.e., the devices closest to the cluster centroids).
2) **Frequency Band Assignment and Metrics Computation:** The selected clustering output is processed to assign FBs to the UEs. An interference-aware FB allocation algorithm is applied, computing distance-based interference metrics, SINR, and estimated data rates for the UEs.
3) **Feature Extraction:** A comprehensive feature set is extracted from the clustering and interference analysis stages. These features include spatial coordinates, distance to the selected cluster head (Free Cells AP), SINR (in dB), estimated data rate (in Mbps), and encoded cluster labels. This feature set forms the dataset for the classification stage.
4) **Classification Model Training:** Multiple classification models are trained using the extracted feature set. With GridSearchCV and k-fold cross-validation, hyperparameter optimization is performed to rigorously explore the parameter space and ensure robust model training.
5) **Model Evaluation and Selection:** The performance of each classifier is evaluated based on validation accuracy. The classifier with the highest validation accuracy is selected to predict optimal FB allocation, using the outputs from the best clustering approach as input data.
6) **Integrated Model Selection:** The final system is chosen based on the best clustering method (for grouping UE based on their positions and selecting APs/ UE-VBS) and the best-performing classifier (for predicting FB allocation). This combined approach ensures optimised resource allocation across various network sizes and operating conditions.

The integrated evaluation framework enables dynamic selection of the best clustering method and, subsequently, the best classification model. By rigorously optimizing and validating each stage using appropriate metrics, the framework ensures optimal FB allocation, maximizing aggregate data rates while minimizing interference in dense wireless networks. In our experiments, as network sizes scale from 100 to 1000 UEs, the framework dynamically adapts to select the clustering method that yields the best interference and data rate performance (K-Means). Concurrently, the most accurate classifier (which in this case is GAN) is optimised to predict FB allocation with high accuracy, ensuring robust resource management in dense wireless environments. This integrated approach—comprising clustering, feature extraction, model optimization, and rigorous validation—provides a comprehensive solution for dynamic FB allocation in 6G cell-free networks, maximizing data rates while minimizing interference.

V. Performance analysis

In our framework, we evaluate both the clustering stage and the classification stage using a variety of performance metrics. These metrics not only allow us to compare different clustering methods but also to assess the predictive accuracy of the classification models for FB allocation.

A. Simulation Parameters

The simulation parameters used to evaluate our framework are designed to emulate realistic network conditions. Specifically, the training dataset comprises **1,000 samples**. The system operates over **500 frequency bands** at a carrier frequency of **2.4 GHz**, with each band having a **20 MHz** bandwidth. The number of user equipments (UEs) ranges from **100 to 1000**, enabling evaluation under varying load scenarios.

B. Evaluation of Clustering Approaches in a Cell-Free Scenario

This section presents the evaluation of different clustering techniques within a simulated cell-free 6G network. It introduces the metrics used to assess both clustering quality and network performance, and compares the effectiveness of several algorithms, namely K-Means, DTC, and DCN, in terms of energy efficiency, data throughput, and clustering quality.

1) Evaluation Metrics for Clustering

Clustering quality is evaluated using both network-level and statistical metrics to reflect spatial structure and communication performance.

- **Network Metrics:** These include the *total power consumption*, calculated as $\sum_{i=1}^{N} P_i$, where N is the number of user equipments (UEs) and P_i is the power used by the i-th UE. Additionally, the *sum rate* is computed as $\sum_{i=1}^{N} R_i$, where R_i denotes the data rate of UE i. These metrics assess energy efficiency and overall network throughput [22].
- **Calinski-Harabasz Index (CH):** This index measures the ratio of between-cluster to within-cluster variance using $\text{CH} = \frac{\text{trace}(B_K)}{\text{trace}(W_K)} \cdot \frac{N-K}{K-1}$, where B_K and W_K are the between- and within-cluster dispersion matrices, N is the total number of data points, and K is the number of clusters. Higher values indicate better cluster separation [31].
- **Davies-Bouldin Index (DB):** This evaluates average cluster similarity as $\text{DB} = \frac{1}{K}\sum_{i=1}^{K} \max_{j\neq i}\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right)$, where σ_i is the average distance of members in cluster i from its centroid c_i, and $d(c_i, c_j)$ is the distance between the centroids of clusters i and j. Lower DB values indicate more compact and distinct clusters [32].
- **Silhouette Score:** This score quantifies how well a data point fits within its assigned cluster. It is defined as $s = \frac{b-a}{\max(a,b)}$, where a is the average distance to other points in the same cluster (intra-cluster), and b is the minimum average distance to points in the nearest neighboring cluster (inter-cluster). A score closer to 1 suggests better-defined clusters [33].

2) Clustering and AP Selection Results

Table II presents the performance comparison of the evaluated clustering approaches for a network comprising 1000 UEs. The metrics are organized as follows: *total_power_consumption* and *total_sum_rate*; followed by *calinski_harabasz_index*, *davies_bouldin_index*, and *silhouette_score*, which collectively assess clustering quality in terms of compactness, separation, and cohesion.

A detailed analysis of Table II reveals that K-Means delivers superior performance across all key metrics when compared to deep learning-based clustering approaches such as DTC and DCN in the cell-free scenario with 1000 UEs. Starting with *total_power_consumption*, K-Means shows a remarkably low value of 9.81×10^9 W, which is several orders of magnitude lower than the 6.50×10^{11} W observed for both DTC and DCN. This indicates a significant advantage in terms of energy efficiency. In terms of *total_sum_rate*, K-Means again leads with a throughput of 68390.78 bps, which is substantially higher than the rates achieved by DTC (18073.40 bps) and DCN (19990.10 bps). This demonstrates K-Means' ability to support more efficient data delivery in dense wireless environments.

Examining the clustering quality metrics, K-Means achieves the highest *calinski_harabasz_index* value at 1050.48, indicating well-separated and dense clusters. Its *davies_bouldin_index* is also the lowest at 0.7554, suggesting that the clusters are distinct with minimal overlap. Lastly, K-Means secures the highest *silhouette_score* of 0.4193, reflecting strong intra-cluster cohesion and inter-cluster separation. Taken together, these results demonstrate that K-Means not only minimizes power usage and maximizes data throughput but also forms high-quality clusters. These advantages make it highly suitable for cell-free network deployments. Similar findings are reported in recent studies, such as [34], which highlight the continued relevance and effectiveness of conventional clustering techniques when properly adapted to wireless network settings.

C. Evaluation of Classification Approaches

In this section, we evaluate the performance of the classifiers in terms of classification accuracy and execution time for hyperparameter tuning.

1) Evaluation Metrics for Classification

To assess the effectiveness of the classifiers used in predicting frequency band (FB) allocations from UE-based features, we adopt two primary evaluation metrics:

- **Accuracy:** This metric quantifies the proportion of correctly predicted FB labels out of all predictions, computed as
$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}.$$
Accuracy serves as the main performance metric, especially when class distribution is relatively balanced. For comprehensive evaluation, additional metrics such as precision, recall, and F1-score can be incorporated in future extensions [35], [36].
- **Execution Time:** This refers to the duration required to perform the complete training and hyper-parameter optimization using `GridSearchCV`. It is calculated as $T_{\text{exec}} = t_{\text{end}} - t_{\text{start}}$, where t_{start} and t_{end} denote the time markers before and after model tuning, respectively. Execution time is particularly critical in real-time or latency-sensitive environments where deployment speed and computational cost are important [37].
- **Average Residual Interference:** After each classifier completes frequency-band assignment, the remaining aggregate interference power observed in the system is sampled and averaged:
$$I_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} I_i,$$

TABLE II: Performance Comparison of Clustering Approaches for 1000 UEs

num_ues	approach	total_power_consumption (W)	total_sum_rate (bps)	calinski_harabasz_index	davies_bouldin_index	silhouette_score
1000	K-Means	9.81×10^{9}	68390.78	1050.48	0.7554	0.4193
1000	DTC	6.50×10^{11}	18073.40	546.87	1.2362	0.3374
1000	DCN	6.50×10^{11}	19990.10	434.05	1.4564	0.2502

where I_i is the measured interference during the i^{th} evaluation period and N is the total number of periods. Lower I_{avg} indicates better spatial–spectral separation and a cleaner radio environment, which directly improves link SINR and overall network capacity. Including this metric enables a holistic assessment that couples classification accuracy with its tangible impact on radio-frequency performance.

These metrics are reported for the examined classification algorithms. All classifiers undergo hyperparameter tuning with 2-fold cross-validation to ensure robust performance across varying class distributions.

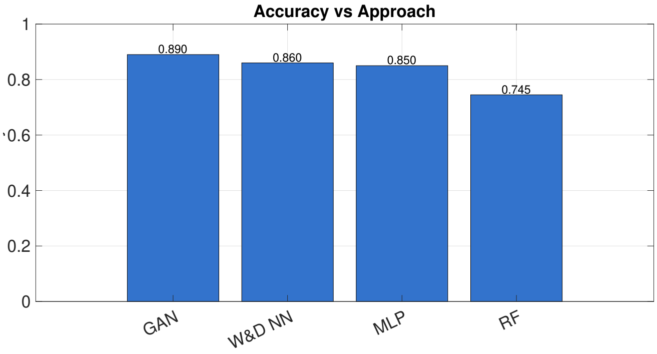

Fig. 2: Accuracy vs. Approach.

2) *Classification Results*

The experimental results clearly show that the **GAN** attains the highest classification accuracy at **89 %**, closely followed by the **Wide & Deep NN** (86 %) and the **MLP** (85 %). The **Random Forest** model records the lowest accuracy (74.5 %), rendering it less attractive for accuracy-critical use cases but still acceptable for scenarios with relaxed performance requirements (Fig. 2).

Figure 3 highlights the trade-off between runtime and accuracy. The **Random Forest** finishes hyper-parameter tuning in only $\sim$ 9 s, making it the fastest model by an order of magnitude. In contrast, the **MLP**, **Wide & Deep NN**, and **GAN Classifier** require between 73 s and 83 s, a consequence of multiple back-propagation passes and larger parameter counts. Thus, deployment choice should weigh the latency budget against the accuracy target: Random Forest for ultra-fast, low-resource settings; GAN or Wide & Deep NN when maximum accuracy justifies longer processing time.

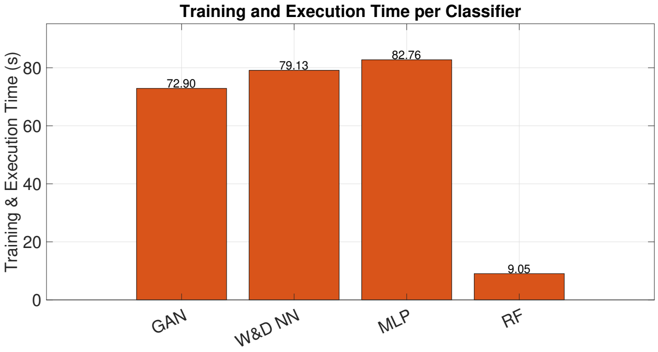

Fig. 3: Training and execution time for each classification approach.

After frequency-band allocation, the *GAN Classifier* leaves the lowest average residual interference at 1.10×10^{-7} W, closely followed by the *Wide & Deep NN* with 1.20×10^{-7} W. The *MLP* exhibits a slightly higher value of 1.25×10^{-7} W. Despite its speed, the *Random Forest* model produces the highest residual interference at 1.45×10^{-7} W, which could degrade overall network SINR in dense deployments. These figures confirm that classifiers with higher accuracy not only select the correct frequency band more often but also translate directly into a cleaner radio environment.

Overall, the GAN Classifier offers the best joint *(accuracy, interference)* performance, whereas Random Forest is the clear choice when execution speed outweighs stringent accuracy and radio cleanliness requirements.

VI. Conclusion and Future Work

This paper introduced a novel two-stage distributed radio resource allocation framework for 6G cell-free communication networks, integrating unsupervised clustering with deep-learning-based frequency band (FB) prediction. The first stage leveraged K-Means, DEC, DCN, and DTC clustering methods to spatially partition user equipment (UE) into interference-aware clusters, with cluster heads (i.e., selected UE-VBSs) acting as local coordinators. K-Means consistently outperformed deep clustering approaches in terms of energy efficiency, sum rate, and cluster quality metrics, especially at high UE densities. In the second stage, we employed supervised classifiers—including GAN-based, Wide & Deep, MLP, and Random Forest models—to predict optimal FB assignments based on features extracted from clustering and network conditions. The GAN classifier achieved the highest FB-assignment accuracy (89%) and lowest residual interference (1.10×10^{-7} W), demonstrating its suitability for latency-sensitive and interference-constrained 6G environments. Overall, our integrated K-Means $\rightarrow$ GAN pipeline provided scalable, energy-efficient, and interference-resilient resource allocation in ultra-dense deployments, validated by extensive simulations over varying UE densities.

Future work will extend this framework in several directions. First, we plan to incorporate mobility models and time-varying channel conditions to assess real-time adaptability under dynamic network scenarios. Second, we aim to explore reinforcement learning techniques to continuously refine FB allocation policies based on observed network feedback. Third, federated learning will be investigated to preserve data privacy while enabling distributed model training across UE-VBS nodes. Finally, we will prototype the framework in a testbed environment using software-defined radios (SDRs) and edge-computing platforms to assess practical feasibility and deployment overhead in real-world 6G settings.

VII. Acknowledgment

This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 739578, the ADROIT6G project of the SNS-JU under Grant Agreement No. 101095363, and the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

References

[1] S. Akiishi, A. Ali, and E. Esenogho, "Interference challenges on 5g networks: A review," in *Proceedings of IEEE AFRICON*, (Gaborone, Botswana), pp. 1–6, IEEE, 2023.

[2] M. Kamal, N. Dey, J. Li, M. R. Akhter, and A. Alqahtani, "Resource allocation schemes for 5g network: A systematic review," *Sensors*, vol. 21, no. 19, p. 6588, 2021.

[3] H. Ngo, G. Interdonato, E. Larsson, G. Caire, and J. Andrews, "Ultra-dense cell-free massive mimo for 6g: Technical overview and open questions." arXiv preprint arXiv:2401.03898, Mar. 2024. Version 2.

[4] H. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, 2017.

[5] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5g," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[6] C. Christophorou, A. Pitsillides, and I. Akyildiz, "Celec framework for reconfigurable small cells as part of 5g ultra-dense networks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, 2017.

[7] I. Ioannou, P. Nagaradjane, Ala' Khalifeh, C. Christophorou, Vasos Vassiliou, G. S. Surya Sashank, Charu Jain, and Andreas Pitsillides, "Ml-aided dynamic clustering and classification of ues as vbs in d2d communication networks," *WiSPNET 2023 – International Conference on Wireless Communications, Signal Processing and Networking*, pp. 1–8, 2023.

[8] Y. Liu, M. Chen, and X. Li, "Radio resource allocation for 5g and beyond: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1665–1708, 2020.

[9] I. Ioannou, C. Christophorou, P. Nagaradjane, and V. Vassiliou, "Distributed radio resource allocation using deep and federated learning in 6g networks," in *2024 Asian Conference on Communication and Networks (ASIANComNet)*, pp. 1–7, 2024.

[10] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, University of California Press, 1967.

[11] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[12] D. Arthur and S. Vassilvitskii, "*k*-means++: The advantages of careful seeding," in *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, (New Orleans, LA), pp. 1027–1035, SIAM, 2007.

[13] R. L. Thorndike, "Who belongs in the family?," *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.

[14] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[15] N. S. Madiraju, S. M. Sadat, Dennis Fisher, and Homayoun Karimabadi, "Deep temporal clustering: Fully unsupervised learning of time-domain features," *arXiv preprint arXiv:1802.01059*, 2018.

[16] B. Yang, X. Fu, Nicholas D. Sidiropoulos, and Mingyi Hong, "Towards *k*-means–friendly spaces: Simultaneous deep learning and clustering," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 3861–3870, 2017.

[17] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems 29*, pp. 2234–2242, 2016.

[18] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS '16)*, pp. 7–10, 2016.

[19] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[20] Leo Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] Iacovos Ioannou, Prabagarane Nagaradjane, Ala' Khalifeh, Christophoros Christophorou, Vasos Vassiliou, Gundepudi V. S. Surya Sashank, Charu Jain, and Andreas Pitsillides, "Ml-aided dynamic clustering and classification of ues as vbs in d2d communication networks," in *2023 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 1–8, 2023.

[22] Iacovos Ioannou, Christophoros Christophorou, Prabagarane Nagaradjane, and Vasos Vassiliou, "Distributed radio resource allocation using deep and federated learning in 6g networks," in *2024 Asian Conference on Communication and Networks (ASIANComNet)*, pp. 1–7, 2024.

[23] "Study on scenarios and requirements for next generation access technologies," Tech. Rep. TR 38.913 V14.0.0, 3rd Generation Partnership Project (3GPP), 2015. Release 14.

[24] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[25] Julien Franceschi, Aurélien Dieuleveut, and Matthijs Jégou, "Unsupervised scalable representation learning for multivariate time series," in *Advances in Neural Information Processing Systems 32*, pp. 3795–3805, 2019.

[26] B. Yang, X. Fu, N. Sidiropoulos, and M. Hong, "Towards *k*-means–friendly spaces: Simultaneous deep learning and clustering," in *International Conference on Machine Learning*, pp. 3861–3870, 2017.

[27] Augustus Odena, Christopher Olah, and Jonathon Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 2642–2651, 2017.

[28] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, and T. *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 7–10, 2016.

[29] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[30] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[31] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.

[32] D. Davies and D. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[33] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[34] Iacovos Ioannou, Christophoros Christophorou, Prabagarane Nagaradjane, and Vasos Vassiliou, "Performance evaluation of machine learning cluster metrics for mobile network augmentation," in *2024 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 1–7, 2024.

[35] Marina Sokolova and Guy Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[36] Christopher D. Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[37] Divya Sambasivan, Joel Sommers, and Paul Barford, "Modeling the performance of machine learning workloads in edge and cloud environments," in *IEEE INFOCOM 2020*, pp. 1716–1725, 2020.