

Central Lancashire Online Knowledge (CLOK)

Title	Access Point Selection and Localization for Cluster-Based Realization of a Device-to-Device Cell-Free 6G Communications Network
Type	Article
URL	https://clock.uclan.ac.uk/id/eprint/57152/
DOI	https://doi.org/10.1049/cmu2.70096
Date	2025
Citation	Ioannou, Iakovos, Raspopoulos, Marios, Nagaradjane, Prabagarane, Christophorou, Christophoros, Gregoriades, Andreas and Vassiliou, Vasos (2025) Access Point Selection and Localization for Cluster-Based Realization of a Device-to-Device Cell-Free 6G Communications Network. IET Communications, 19 (1). e70096. ISSN 1751-8628
Creators	Ioannou, Iakovos, Raspopoulos, Marios, Nagaradjane, Prabagarane, Christophorou, Christophoros, Gregoriades, Andreas and Vassiliou, Vasos

It is advisable to refer to the publisher's version if you intend to cite from the work.
<https://doi.org/10.1049/cmu2.70096>

For information about Research at UCLan please go to <http://www.uclan.ac.uk/research/>

All outputs in CLOK are protected by Intellectual Property Rights law, including Copyright law. Copyright, IPR and Moral Rights for the works on this site are retained by the individual authors and/or other copyright owners. Terms and conditions for use of this material are defined in the <http://clock.uclan.ac.uk/policies/>

ORIGINAL RESEARCH OPEN ACCESS

Access Point Selection and Localization for Cluster-Based Realization of a Device-to-Device Cell-Free 6G Communications Network

Iakovos Ioannou^{1,2,3}  | Marios Raptopoulos⁴ | Prabagarane Nagaradjane⁵ | Christophoros Christophorou^{3,4} | Andreas Gregoriades⁶ | Vasos Vassiliou^{2,3}

¹Department of Computer Science and Engineering, European University of Cyprus, Nicosia, Cyprus | ²Department of Computer Science, University of Cyprus, Nicosia, Cyprus | ³CYENS - Centre of Excellence, Nicosia, Cyprus | ⁴University of Central Lancashire (UCLAN), Larnaca, Cyprus | ⁵Department of ECE, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India | ⁶Cyprus University of Technology, Limassol, Cyprus

Correspondence: Iakovos Ioannou (ioannou.iakovos@ucy.ac.cy)

Received: 22 April 2025 | **Revised:** 25 August 2025 | **Accepted:** 28 September 2025

Funding: 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. Additionally, this work was co-funded by the European Union under the programme of social cohesion "THALIA 2021-2027", through the Research and Innovation Foundation (Project: EXCELLENCE/0524/0218).

ABSTRACT

The increasing demand for ultra-reliable, low-latency, and high-throughput connectivity in dense urban environments presents significant challenges for next-generation 6G networks. Traditional cellular networks, with their fixed cell boundaries and centralized base station control, are inadequate to meet the dynamic needs of such environments. A promising solution is the cell-free network architecture, where a distributed set of access points (APs) jointly serve users without fixed cell boundaries. However, efficient access point selection and accurate user localization are crucial to achieving high performance in such networks. This paper presents a decentralized approach using Belief-Desire-Intention eXtended (BDIX) agents for dynamic AP selection and localization within a cluster-based cell-free 6G communications network. Various clustering algorithms (K-means, DBSCAN, self-organizing maps, MeanShift, ClusterGAN, and Autoencoders) are evaluated for their ability to optimize network throughput, energy efficiency, and spectral utilization. A hybrid localization framework, such as centroid-based, differential circles, and multilateration methods, is employed to achieve accurate user positioning. The results demonstrate that machine learning-based clustering methods, notably Gaussian mixture model (GMM), self-organizing map (SOM), and ClusterGAN, offer significant improvements in throughput (up to 46.3%) and power reduction (up to 32.8%) over traditional methods. Regarding localization, deep learning models such as MLP, CNN, and TCN outperform deterministic methods, achieving sub-meter accuracy with minimal errors (MeanDist < 1 m, $R^2 > 0.999$). Overall, the proposed solution enhances system scalability, energy efficiency, and positioning accuracy, establishing a promising foundation for future 6G networks. In our reference implementation, we instantiate the pipeline with a GMM for AP/UE clustering and a multilayer perceptron (MLP) regressor for localization.

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2025 The Author(s). IET Communications published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

1 | Introduction

The evolution toward 6G communications promises unprecedented data rates, ultra-low latency, and massive connectivity. One of the most promising paradigms is the cell-free network architecture, which features a distributed set of access points (APs) that jointly serve users without fixed cell boundaries. This eliminates the cell-edge problem, ensuring uniformly high-quality service. However, implementing a cell-free environment entails several challenges, including the efficient *access point selection* and *localization* within a cluster-based framework [1]. Specifically, efficient access point selection is critical because, unlike traditional cellular networks where each user is assigned to a fixed base station, cell-free networks require a dynamic selection (which must adapt to variations in channel quality, user mobility, and interference [2]) of the optimal subset of APs from a dense deployment. On the other hand, high-precision localization of both users and APs is essential for dynamic AP selection, as it enables precise resource allocation and effective interference management in environments with high user density and complex propagation conditions [3].

To address these requirements, clustering algorithms such as K-means, self-organizing maps (SOM), and density-based spatial clustering of applications with noise (DBSCAN) can be employed to dynamically group APs based on user density and channel conditions [4]. Moreover, clustering APs to manage large-scale network deployments simplifies resource allocation and reduces computational complexity, facilitating efficient interference management.

Efficient clustering facilitates localized decision-making and minimizes the signaling overhead required for resource allocation and interference management. Additionally, machine learning-driven techniques, including reinforcement learning, can be leveraged to dynamically optimize AP selection and enhance responsiveness to changes in network state and user mobility patterns [5]. To achieve the goal of precise localization, a novel hybrid localization technique is employed, integrating measurements from time-of-arrival (TOA), angle-of-arrival (AOA), and received signal strength indicator (RSSI) to achieve superior positioning accuracy, especially beneficial in ultra-dense deployments and complex propagation environments [6]. Advanced data fusion algorithms, such as extended Kalman filters (EKF) and particle filters, further enhance localization accuracy by effectively integrating heterogeneous measurement data [7].

A cell-free network architecture is organized into decentralized clusters of APs that autonomously manage local resources through edge computing technologies, enabling rapid adaptation to localized variations in network demand. Such decentralized resource management significantly reduces latency and ensures robustness against single points of failure [8]. This distributed approach not only improves network resilience but also enhances scalability and adaptability to fluctuating user requirements and network conditions. Moreover, the deployment of artificial intelligence (AI)-based predictive models enables proactive AP selection and resource management, anticipating user movements, traffic variations, and interference patterns, thus significantly improving spectral efficiency and energy consumption [9]. The energy efficiency of the network is further optimized through

intelligent power control mechanisms that dynamically adjust transmission power based on real-time channel estimations and localized network demands, minimizing unnecessary energy usage and interference [10].

In this paper, we propose a novel, decentralized framework for access point (AP) selection and localization tailored for cell-free 6G communications networks. By leveraging advanced machine learning clustering techniques integrated with belief-desire-intention extended (BDIx) agents, our approach achieves deterministic clustering and uniform AP selection across all user equipment (UE) in a fully distributed manner. Our hybrid localization framework combines geometry-based estimations (such as centroid, differential circles, and multilateration) with deep learning regression models (such as MLP, CNN, TCN, and BLSTM) to attain sub-meter accuracy, even in ultra-dense environments. Moreover, the proposed approach significantly enhances network throughput and energy efficiency, demonstrating a sum-rate gain of up to 46.3% and total power reduction of up to 32.8%, thus addressing the complex challenges inherent in next-generation wireless networks.

Overall, the main contributions of this work can be summarized as follows:

- **Advanced Decentralized Clustering Techniques:** Integration of a suite of dynamic clustering algorithms—including DBSCAN, K-Means, SOM, MeanShift, ClusterGAN, and Autoencoder-based methods—with BDIx agents to achieve deterministic AP grouping and efficient resource allocation in a decentralized cell-free 6G network (instantiated with GMM in this work).
- **Hybrid Localization Framework:** Development of a novel localization strategy that combines geometry-based methods (centroid estimation, differential circles, and multilateration; cf. Equations 27–36) with deep learning regression models (multilayer perceptron [MLP], CNN, TCN, BLSTM) to achieve sub-meter accuracy, crucial for precise resource allocation and interference management (instantiated with an MLP regressor in this work).
- **AI-Based Predictive Management:** Implementation of predictive machine learning algorithms within BDIx agents to enable proactive AP selection and dynamic resource management in response to real-time network conditions and user mobility.
- **Enhanced Communication Protocols and Signaling Strategies:** Design of innovative protocols that facilitate seamless inter-device connectivity via Wi-Fi Direct for intra-cluster communication and LTE-based backhaul for inter-cluster connectivity, optimized explicitly for high-frequency, ultra-dense 6G environments (see selection rule in Equation 14).
- **Energy Efficiency Optimization:** Analytical evaluation and optimization of energy consumption, demonstrating that the proposed framework achieves significant power reductions (up to 32.8%) while enhancing network throughput (up to 46.3% sum-rate gain) compared to traditional cellular architectures (see Equations 19–21).

- **Novel Distributed Framework for 6G Networks:** Introduction of a fully decentralized network formation strategy that ensures consistent clustering and AP selection across all UEs, thereby enhancing system scalability, reliability, and adaptability in dynamic wireless environments (determinism described in Equation 15, results in Section 5). *Reference implementation:* Unless otherwise stated, all reported end-to-end results for “ours” use **GMM** for clustering and **MLP** for localization.

1.1 | Contribution Summary and Positioning vs. Prior Work

This paper tackles five concrete challenges of cell-free 6G at scale: (1) *fully decentralized AP selection and clustering* under tight signaling/fronthaul budgets; (2) *deterministic agreement* on clusters/APs across all UEs without a central coordinator, even with randomized algorithms; (3) *robust operation under imperfect CSI, mobility, and NLoS*¹, typical of dense urban layouts; (4) *tight KPI coupling* (throughput–energy–fairness) via an explicit optimization (Equation 6); and (5) *edge-friendly execution* that respects device constraints while integrating *localization* into the control loop. Relative to user-centric CF, O-RAN clustering, LSFD/DCC, and DL/DRL controllers, our framework is distinct in three ways: (i) it uses a *minimal broadcast*—BS positions and configuration $(\mathcal{B}, \theta, s)$ —while UEs locally hold X via ProSe, ensuring *low signaling* and *privacy of UE locations*; (ii) it *provably enforces deterministic cluster agreement* from common (θ, s) and shared X (Equation 15), yielding identical AP elections on every UE without centralized computation; and (iii) it *integrates hybrid localization* (*centroid/circles/multilateration* + *DL*) directly into KPI-aware control, rather than treating positioning as an external service. Table 2 evidences that, unlike prior CF-only or localization-only pipelines, our BDIX-driven design jointly attains decentralization, deterministic agreement, low signaling, edge friendliness, robustness to CSI/sync/NLoS, hybrid localization, QoS awareness, and practical scalability—thereby clarifying the main contributions and how they advance the state of the art. These innovations collectively enhance throughput, latency, energy efficiency, and spectral efficiency, offering substantial performance advantages over traditional cellular networks and positioning wireless communications for future demands.

The following table serves as a comprehensive glossary of the notation used, establishing a clear and consistent framework for the reader. It is methodically structured into three principal sections to cover the different technical domains involved. The first section introduces variables related to machine learning and clustering, defining the essential components of algorithms like Gaussian mixture models (GMM), self-organizing maps (SOM), and support vector regression (SVR). The subsequent section shifts focus to the domain of wireless communications, outlining fundamental parameters such as signal power, antenna gain, noise, and channel quality indicators. Finally, the table concludes with a list of performance metrics that are critical for evaluating both the clustering algorithms’ efficacy (e.g., Silhouette Score, Davies-Bouldin Index) and the communication system’s efficiency (e.g., bit error rate, spectral efficiency).

The article is organized as follows: Section 2 discusses related work on decision-making frameworks and machine learning techniques, focusing on their role in enhancing BDIX agents within dynamic wireless networks. Section 3 describes the problem statement and introduces the proposed system, emphasizing the architecture and components of the Plan Library integrated into BDIX agents. Section 4 outlines the methodology employed for evaluating decision-making frameworks and adaptive clustering techniques, focusing on synthetic data generation, model training, and performance evaluation. Section 5 presents the results of the assessment, comparing the performance of machine learning models and fuzzy logic approaches based on metrics like accuracy, computational efficiency, and resource utilization. Finally, Section 6 provides a summary of the findings, discusses their implications for future research, and outlines potential advancements for improving intelligent decision-making in 6G networks.

Table 1 lists the variables and definitions used in the formulas and text in this examination.

2 | Related Work and Background

This section provides a comprehensive review of related work and background information concerning the evolution of cell-free massive MIMO (CF-mMIMO) systems, the role of clustering algorithms, and the application of distributed artificial intelligence (DAI) frameworks in modern wireless networks. It explores the integration of advanced machine learning techniques, such as GMM, and the development of user-centric models for enhanced network optimization. Furthermore, the section explores foundational techniques crucial for advancing wireless communication systems, including device-to-device (D2D) communication and clustering methods such as K-means, DBSCAN, and GMM. It highlights the challenges encountered in implementing large-scale CF-mMIMO networks, particularly in terms of channel estimation, synchronization, and efficient resource management.

Additionally, this section covers the significance of BDIX agents for decentralized decision-making in 5G/6G environments. This background is crucial for understanding the ongoing research in clustering techniques, localization accuracy, and dynamic resource allocation in large-scale wireless networks. The background also discusses the role of DAI frameworks and BDIX agents in decentralized decision-making, offering scalable solutions to network complexities in 5G/6G networks. The importance of machine learning and deep learning approaches for location prediction in dense and dynamic network environments is also emphasized.

2.1 | Related Work

2.1.1 | Cell Free Related Work

Cell-free massive MIMO (CF-mMIMO) has garnered significant attention for its potential to enhance wireless network efficiency and coverage by utilizing distributed antenna systems without traditional cell boundaries. Ngo et al. [1] laid the groundwork for foundational research by demonstrating how distributed

TABLE 1 | Table of variables and definitions.

Variable	Definition
K	Number of Gaussian components/clusters (GMM).
π_k	Mixing coefficient for the k -th Gaussian component.
μ_k, Σ_k	Mean vector and covariance matrix for the k -th Gaussian component.
$w_i(t)$	Weight vector of neuron i in a SOM at time step t .
$\alpha(t)$	Learning rate for SOM at time step t .
$h_{ci}(t)$	Neighborhood function in SOM centered at the best-matching unit c .
$\mathcal{L}$	Loss function for a machine learning model.
D, G, E	Discriminator, generator, and encoder networks in ClusterGAN.
z, c	Latent variables (continuous and discrete) in ClusterGAN.
$H^{(l)}$	Matrix of node features at layer l in a GNN.
$A, \tilde{A}$	Adjacency matrix and its version with self-loops.
$W^{(l)}$	Weight matrix for layer l in a neural network.
w, b	Model parameters (weights and bias) in SVR.
$\phi(x_i)$	Function mapping input x_i to a higher-dimensional space.
ϵ	Margin of tolerance in SVR.
$P_r(d)$	Received power at distance d (dBm).
P_t	Transmit power (dBm).
G_t, G_r	Transmit and receive antenna gains (dBi).
d	Distance between transmitter and receiver (m).
λ	Carrier wavelength (m).
h	Channel gain coefficient.
K_{RIC}	Rician K-factor (linear).
P_{noise}	Total noise floor power (dBm).
k_B	Boltzmann's constant (1.38×10^{-23} J/K).
T	System temperature (K).
B	System bandwidth (Hz).
N_{fig}	Receiver noise figure (dB).
RSSI	Received signal strength indicator (dBm).
SINR	Signal-to-interference-plus-noise ratio.
P_{interf}	Co-channel interference power (dBm).
$R(d)$	Achievable data rate at distance d (Mbps).
$P_{\text{consumption}}$	Power consumption (dB) proxy (cf. Equation 12).
CQI	Channel quality indicator.
$\mathbf{u}, \mathbf{a}_i, \mathbf{b}_j$	Position vectors for a UE, the i -th AP, and the j -th BS.

(Continues)

TABLE 1 | (Continued)

Variable	Definition
BSs	Number of base stations (scalar count).
$a(i), b(i)$	Intra-cluster and nearest-cluster distances for Silhouette score.
σ_i	Average distance of points to their cluster center in DBI.
d_{ij}	Distance between centers of clusters i and j in DBI.
B_k, W_k	Between-cluster and within-cluster dispersion matrices for CHI.
N	Total number of data points or devices.
k	Total number of clusters.
η_{net}	Network spectral efficiency (bits/s/Hz).
B_{total}	Total system bandwidth (Hz).
E_b/N_0	Average signal-to-noise ratio per bit.
P_b	Bit error rate (BER).
$\gamma, \bar{\gamma}$	Instantaneous and average SNR.
$\mathbf{p}_i, \mathbf{g}_i$	Predicted and ground-truth 2D position vectors.
X	Set of all UE positions (from LTE ProSe; locally aggregated by each UE's BDIX agent).
$\mathcal{B}$	Set of all BS positions (broadcast by the BS).
f_{MLP}	MLP regression function mapping features to $(\hat{x}, \hat{y})$ coordinates.
ϕ	Learnable parameters of the MLP (weights/biases).

access points (APs) can jointly serve multiple users over the same time-frequency resources, thereby significantly boosting spectral efficiency and mitigating inter-cell interference. Their work highlighted the advantages of uniform service quality but also identified critical implementation challenges such as fronthaul capacity limitations and precise synchronization among APs. These challenges necessitate sophisticated solutions for managing and synchronizing accurate channel state information (CSI), highlighting practical barriers to wide-scale deployments.

Chen et al. [11] advanced CF-mMIMO by proposing user-centric (UC) clustering, dynamically assigning subsets of APs to individual users based on proximity, signal quality, and network conditions. Their approach effectively reduced fronthaul traffic, computational overhead, and improved scalability, vital for extensive urban deployments. Through simulations, their method demonstrated superior performance compared to static clustering methods, resulting in significantly improved resource efficiency. Nevertheless, the introduction of dynamic clustering also created complexities in real-time resource allocation, necessitating the development of sophisticated optimization algorithms to maintain performance, especially in rapidly changing user environments.

Iliadis et al. [12] tackled channel estimation challenges inherent in CF-mMIMO by leveraging deep learning methods. They

utilized neural networks for precise channel estimation and adaptive power allocation, significantly improving accuracy over traditional estimation techniques. Their deep-learning-based method demonstrated robustness against pilot contamination and environmental variations, essential for dynamic CF-mMIMO scenarios. Despite these advances, the high computational complexity and training overhead required by neural networks pose challenges in latency-sensitive or densely populated networks, underscoring a critical need for improvements in computational efficiency in practical deployments.

Shahab et al. [13] investigated the integration of full-duplex technology into CF-mMIMO architectures, exploring simultaneous uplink and downlink transmissions to double spectral efficiency. Their findings highlighted substantial performance gains but also underscored the complexities of self-interference cancellation and synchronization. Despite these advances, practical implementation of full-duplex CF-mMIMO demands sophisticated interference management techniques, precise timing synchronization, and coordinated transmission protocols to ensure operational viability.

Mashdour and de Lamare [14] introduced a novel clustering strategy based on achievable information rates, rather than conventional large-scale fading metrics. Their approach directly enhanced spectral efficiency and fairness among users, coupled with an advanced resource allocation technique optimized for scheduling efficiency. Although simulations validated significant performance improvements, the complexity and real-time computational demands of their methods raised concerns about scalability and practicality, especially in large-scale networks.

Beerten et al. [15] focused on practical deployment strategies for CF-mMIMO within the open radio access network (O-RAN) framework, developing two clustering and handover strategies. Their fixed clustering method activated clusters when specific thresholds were exceeded, while an opportunistic strategy dynamically adjusted clusters as users moved. These methods significantly enhanced network performance, particularly in handover scenarios, but raised issues regarding signaling overhead and increased network control complexity, necessitating further research into more efficient signaling solutions.

Demir et al. [16] explored dynamic cooperation clustering (DCC) for the uplink in user-centric CF-mMIMO networks. They proposed a layered decoding strategy—large-scale fading decoding (LSFD)—to efficiently manage fronthaul traffic, maintaining high spectral efficiency with reduced overhead. Although their theoretical framework demonstrated significant performance gains, the practical implementation raised coordination and synchronization challenges, underscoring the need for balanced solutions between decentralized control at access points (APs) and centralized network management.

Di Gennaro et al. [17] presented a deep learning-based user-centric clustering methodology that employs long short-term memory (LSTM) neural networks to maximize the sum spectral efficiency while limiting active connections. This approach effectively addressed imperfections in CSI, including pilot contamination, demonstrating robust performance. However, practical application was constrained by the computational intensity

and training complexity of LSTM models, necessitating further optimization for real-world latency-sensitive network conditions.

Tan et al. [18] proposed energy-efficient AP clustering and power allocation in CF-mMIMO through hierarchical deep reinforcement learning (DRL). Their two-layer control architecture optimized resource allocation and AP clustering dynamically, resulting in significant improvements in energy efficiency. Although their model delivered excellent results, the inherent computational complexity and real-time responsiveness demands posed practical deployment challenges, underscoring the need for more streamlined and computationally efficient algorithms.

Ioannou et al. [19] introduced a comprehensive cell-free 6G architecture enhanced by sophisticated clustering algorithms, including SOM, GMM, MeanShift, DBSCAN, and KMeans, along with BDIX agents for decentralized resource management. They demonstrated that integrating SOM with BDIX agents resulted in superior scalability, optimized load balancing, and increased network throughput, highlighting the effectiveness of their approach for dense urban deployments typical of future 6G environments.

Ioannou et al. [20] further advanced cell-free network architecture by applying advanced deep learning clustering methods, including MeanShift, DBSCAN, KMeans, affinity propagation, and ClusterGAN. Among these, ClusterGAN demonstrated exceptional performance in modeling complex network conditions dynamically, significantly improving network sum rate, scalability, and energy efficiency. Their research provided clear evidence of the potential for GAN-based methods in addressing the limitations of conventional clustering approaches, specifically in highly dynamic and dense scenarios envisioned for 6G networks.

Idigo et al. [21] investigate cluster-head (CH) selection for device-to-device (D2D) communication using an unsupervised self-organizing map (SOM) driven by a weighted hardware sensing factor (HSF) that fuses link-quality indicators (e.g., RSSI/RSRP/RSRQ) with proxies of device hardware condition. Using measurements from multiple UEs within a 250 m radius of a base station, they report that HSF-fed SOM forms more reliable clusters and elects more suitable CHs than SOM configured with any single metric, and also outperforms KMeans when both are given the same HSF inputs. The study highlights vendor-dependent variability in radio measurements and argues that metric weighting mitigates this effect during CH election, improving robustness. Nonetheless, the evaluation is small-scale and quasi-static (no mobility or online operation); integrating HSF-aware SOM into user-centric, large-scale cell-free settings with real-time updates and energy/fairness objectives remains open future work.

2.1.2 | Localization Related Work

Yang et al. [22] addressed localization accuracy in CF-mMIMO networks by integrating distributed antenna setups with advanced triangulation algorithms. Leveraging spatial diversity, they achieved centimeter-level localization precision, dramatically enhancing resource allocation and interference

management. Although their methodology significantly improved accuracy, the requirement for precise synchronization and high-quality channel state information presented substantial implementation challenges, particularly in high-mobility or dense user conditions.

Bulusu et al. [23] describe GPS-less localization where the centroid estimator computes the position as $\hat{\mathbf{p}} = \frac{1}{K} \sum_{k=1}^K \mathbf{a}_k$, where $\mathbf{a}_k \in \mathbb{R}^2$ (or $\mathbb{R}^3$) are anchor coordinates and K is the number of anchors/fingerprints. This assumes roughly uniform anchor geometry and unbiased measurements; otherwise, dilution of precision and large-scale bias dominate, motivating geometry-aware weighting and outlier screening. Li et al. [24] address trilateration via circle (or sphere) intersections using ranges $\{d_i\}$ from anchors $\{\mathbf{a}_i\}$. A common formulation is $\min_{\mathbf{p}} \sum_{i=1}^M (\|\mathbf{p} - \mathbf{a}_i\|_2 - d_i)^2$, where $\mathbf{p}$ is the unknown position and M the number of anchors; linearized least squares yields $\mathbf{p} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ with $\mathbf{A}, \mathbf{b}$ built from $(\mathbf{a}_i, d_i)$. Robust methods mitigate biased ranges and poor anchor geometry, but synchronization and multipath remain key challenges. Hoang et al. [25] leverage bidirectional LSTMs for fingerprint sequences $\{\mathbf{x}_t\}_{t=1}^T$, mapping temporal windows to positions via $\hat{\mathbf{p}}_t = \mathbf{W} [\vec{\mathbf{h}}_t; \vec{\mathbf{h}}_t] + \mathbf{b}$, where $\vec{\mathbf{h}}_t, \vec{\mathbf{h}}_t$ are forward/backward hidden states and $[\cdot; \cdot]$ denotes concatenation. By modeling temporal correlations and smoothing noise, BLSTMs reduce variance, but require well-aligned sequences, careful regularization, and trajectory labeling. Ouyang et al. [26] employ Temporal Convolutional Networks with dilated causal convolutions to capture long-range dependencies. For feature stream $\{\mathbf{x}_t\}$ the layer output satisfies $\mathbf{y}_t = \sum_{\ell=0}^{L-1} \mathbf{W}_\ell \mathbf{x}_{t-d\ell} + \mathbf{c}$, where d is the dilation factor, L the kernel length, and $\mathbf{W}_\ell, \mathbf{c}$ the learnable parameters. Residual connections aid stability; accurate timestamps and consistent sampling are important for receptive-field design.

Wang et al. [27] treat CSI/fingerprints as images $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$ and apply 2D convolutions $\mathbf{Z} = \sigma(\mathbf{W} * \mathbf{F} + \mathbf{b})$ followed by pooling and fully connected heads to regress coordinates. Here $*$ is convolution, $\sigma(\cdot)$ an activation, and H, W, C denote spatial and channel dimensions. Spatial filtering denoises and learns invariant features, but requires consistent fingerprint rasterization and intensity normalization. Nguyen et al. [28] formulate fingerprinting with transformers by tokenizing measurements into embeddings and applying self-attention $\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{Q}\mathbf{K}^T / \sqrt{d_k}) \mathbf{V}$, where queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ are linear projections of tokens and d_k is the key dimension. Positional encodings preserve order. Such models capture global context but are data-hungry and benefit from domain adaptation to mitigate distribution shift. Following standard practice for feedforward regressors, a multilayer perceptron maps engineered features $\mathbf{x} \in \mathbb{R}^D$ to coordinates via $\hat{\mathbf{p}} = \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2$, where $\sigma(\cdot)$ is a nonlinearity and $\{\mathbf{W}_i, \mathbf{b}_i\}$ are learnable parameters. Feature scaling, dropout, and weight decay help control overfitting, while coordinate normalization improves optimization and stability. Kang et al. [29] model localization with a graph $G = (V, E)$ built from fingerprints and reference points, applying message passing $\mathbf{h}_v^{(l+1)} = \phi\left(\mathbf{h}_v^{(l)}, \square_{u \in \mathcal{N}(v)} \psi(\mathbf{h}_v^{(l)}, \mathbf{h}_u^{(l)}, \mathbf{e}_{uv})\right)$, where $\mathcal{N}(v)$ is the neighborhood, $\mathbf{e}_{uv}$ edge features, $\square$ an aggregation (e.g., sum/mean), and ϕ, ψ neural functions. Performance hinges on graph construction (e.g., kNN radius), edge attributes, and robustness to noisy neighbors. Jondhale et al. [30]

adopt support vector regression for coordinate prediction $f(\mathbf{x}) = \sum_{i=1}^N \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$ trained with ε -insensitive loss, where $K(\cdot, \cdot)$ is a kernel (e.g., RBF), $\{\mathbf{x}_i\}$ support vectors, and $\{\alpha_i\}, b$ learned parameters. Proper tuning of C, ε , and kernel width is critical, and feature standardization improves conditioning.

Wang et al. [31] use Random Forest ensembles of regression trees $\{\mathcal{T}_t\}_{t=1}^T$ trained on bootstrap samples with feature subsampling, yielding $\hat{\mathbf{p}} = \frac{1}{T} \sum_{t=1}^T \mathcal{T}_t(\mathbf{x})$. Bagging reduces variance and handles heterogeneous features, while out-of-bag estimates aid validation; however, piecewise-constant leaves limit extrapolation beyond the training manifold. Bahl and Padmanabhan [32] introduced instance-based fingerprinting, where KNN estimates $\hat{\mathbf{p}} = \frac{1}{K} \sum_{i \in \mathcal{N}_K(\mathbf{x})} \mathbf{p}_i$ using a distance $d(\mathbf{x}, \mathbf{x}_i)$ (e.g., Euclidean or cosine) to define the neighbor set $\mathcal{N}_K(\mathbf{x})$. The method is simple and nonparametric; performance depends on database density, distance metric selection, and efficient indexing for large-scale retrieval.

2.1.3 | Overall Comparison

Table 2 contrasts the literature along deployment-critical dimensions for dense 6G. Classic CF designs can equalize service quality but suffer from heavy fronthaul, strict synchronization, and centralized control. User-centric/O-RAN CF reduces signaling yet still depends on periodic re-optimization. DL/DRL controllers improve efficiency but shift non-trivial training/inference costs to the edge. Pure localization pipelines deliver strong accuracy but remain decoupled from QoS and KPI control. In contrast, our framework is the only one that attains a full sweep of capabilities in both CF and Localization groups (all “Yes” across the ten columns), combining **deterministic, decentralized agreement** with **low signaling, edge-friendly** execution, and **hybrid localization** robust to imperfect CSI/synchronization and NLoS. Decisions are tightly **KPI-coupled** (throughput–energy–fairness), explicitly **QoS-aware**, and operate in an **autonomous, distributed, dynamic** manner. This end-to-end alignment explains the observed gains in sum rate and power reduction while preserving fairness, and it highlights clean scalability for dense deployments compared to CF-only or localization-only baselines. In summary, we *adopt GMM* as the operational clustering engine and *MLP* as the default localization regressor in our decentralized BDIX pipeline (cf. Equation 1).

2.2 | Background Information

2.2.1 | Device-to-Device (D2D) Communication

D2D communication refers to the direct exchange of data between nearby user equipment (UE) without routing traffic through the base station (BS). It was originally proposed to offload traffic in cellular networks; however, it has gained renewed interest due to its potential to support emerging 6G applications. Below, some information regarding the underlying mechanisms that govern how D2D links are established and maintained, are described.

2.2.1.1 | Control Approaches in D2D Communication.

The establishment and management of device-to-device (D2D)

TABLE 2 | Advantage matrix across related work (single table with CF/Loc separation). Cells use **yes/partial/no**.

Capabilities												
Type	Approach (reference)	ML used	Metrics & key results	Decent. oper.	Determin. agree.	Low sig. / fronth.	Edge friendly	Robust to		KPI coupling (rate, energy, fairness)	Practical scalab.	Auton. distr. dynamic
								CSI / sync & NLoS	Hybrid loc.			
Cell-Free (CF)												
CF	Ngo et al. (Foundational CF-mMIMO) [1]	None (analytical)	Per-user throughput/SE CDF; 95%-likely rate ~ 5x vs small cells (up to ~ 10x with correlated fading).	No	No	No	No	No	No	Partial	Partial	No
CF	Chen et al. (User-centric CF) [11]	None (survey)	SE/throughput reviewed; qualitative CF gains vs cellular.	Partial	No	Partial	Partial	Partial	No	Partial	Partial	Partial
CF	Iliadis et al. (DL channel/cluster) [12]	Deep learning (selection)	SE / assignment quality (simulation); specific figures n/r.	Partial	No	Partial	No	Yes	No	Partial	Partial	Partial
CF	Di Gennaro et al. (LSTM UC clustering) [17]	LSTM (user-centric)	SE and handover cost (simulation); figures n/r.	Partial	No	Partial	No	Yes	No	Partial	Partial	Partial
CF	Shahab et al. (Full-duplex CF) [13]	None (FD processing)	SE vs HD; FD up to ~140% SE gain under practical residual SI.	Partial	No	No	No	Partial	No	Partial	Partial	Partial
CF	Mashdour & de Lamare (Rate-aware clustering) [14]	Optimization / heuristics	Sum-SE and fairness; figures n/r.	Partial	No	Partial	No	Partial	No	Yes	Yes	Partial
CF	Beerten et al. (O-RAN clustering/handover) [15]	Algorithmic clustering (O-RAN)	QoE, HO rate, signaling; opportunistic clustering beats cellular baseline (simulation).	Partial	No	No	Partial	Partial	No	Partial	Partial	Partial
CF	Demir et al. (Uplink DCC + LSFD) [16]	None (DCC+LSFD)	SE & complexity/fronthaul; ~ 70x lower LSFD compute and ~ 5x lower fronthaul vs baseline; competitive SE.	Partial	No	Yes	Partial	Partial	No	Partial	Partial	Partial

(Continues)

TABLE 2 | (Continued)

Capabilities													
Type	Approach (reference)	ML used	Metrics & key results	Decent. oper.	Determin. agree.	Low sig. / fronth.	Edge friendly	Robust to CSI / sync & NLoS	Hybrid loc.	KPI coupling (rate, energy, fairness)	QoS-aware	Practical scalab.	Auton. distr. dynamic
CF	Tan et al. (Hierarchical DRL) [18]	Hierarchical deep RL	Sum-SE and energy-delay trade-offs (simulation); figures n/r.	Partial	No	Partial	No	Partial	No	Yes	Partial	Partial	Partial
CF	Ioannou et al. (BDIx + classic ML) [19]	BDIx planning + classical ML	Sum rate, power, fairness (this study).	Yes	Partial	Partial	Yes	Partial	Partial	Partial	Partial	Yes	Yes
CF	Ioannou et al. (GAN-based clustering)	GAN-based clustering	SE and reconfiguration cost (this study).	Yes	Partial	Partial	Partial	Partial	Partial	Partial	Partial	Partial	Partial
CF	Idigo et al. (SOM D2D CH) [21]	Self-organizing maps (D2D)	Throughput/energy (simulation) for D2D overlay; figures n/r.	Yes	Partial	Partial	Yes	Partial	No	Partial	Partial	Partial	Partial
CF	This work: BDIx + GMM clustering + MLP localization			Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Localization (Loc)													
Loc	Yang et al. (CF triangulation / single-AP CSI) [22]	None (ranging/AoA)	Centimeter-level median error (single-AP CSI); sub-meter in TriLoc prototypes.	Partial	Yes	Partial	Partial	No	No	Partial	Partial	Partial	Partial
Loc	Bulusu et al. (Centroid) [23]	None	Outdoor localization; mean ~4.95 m.	Yes	Yes	Yes	Yes	No	No	No	No	Partial	Yes
Loc	Li et al. (Trilateration) [24]	None (robust geometry)	Robust to NLoS; typical 1-5 m depending on layout.	Yes	Yes	Partial	Yes	No	No	No	No	Partial	Yes
Loc	Hoang et al. (BLSTM) [25]	BiLSTM / RNN	Avg ≈0.75 m with 80% < 1 m.	Partial	No	Partial	No	Partial	No	No	No	Partial	Partial
Loc	Ouyang et al. (TCN) [26]	Temporal convnet	Classification accuracy on corridor IDs (not distance); ~96-99%.	Partial	No	Partial	No	Partial	No	No	No	Partial	Partial

(Continues)

TABLE 2 | (Continued)

Capabilities												
Type	Approach (reference)	ML used	Metrics & key results	Decent. oper.	Determin. agree.	Low sig. / fronth.	Edge friendly	Robust to		KPI coupling		Auton. distr. dynamic
								CSI / sync & NLoS	Hybrid loc.	(rate, energy, fairness)	QoS-aware	
Loc	Wang et al. (CNN on CSI images) [27]	CNN	Mean error 1.788 m (lab) / 2.386 m (corridor).	Partial	No	Partial	No	Partial	No	No	Partial	Partial
Loc	Nguyen et al. (Transformer) [28]	Anchor-agnostic Transformer	Sub-meter to meter-level (dataset-dependent).	Partial	No	Partial	No	Partial	No	No	Partial	Partial
Loc	Jondhale et al. (SVR) [30]	SVR / SVR+KF	Trilateration 11.20 m, SVR 4.04 m, SVR+KF 0.85 m.	Yes	Yes	Yes	Partial	Partial	No	No	Partial	Yes
Loc	Wang et al. (Random Forest) [31]	Random Forest (CSI fingerprinting)	Mean error lower than KNN/WKNN; detailed CDFs in office/anechoic.	Yes	Partial	Yes	Partial	Partial	No	No	Partial	Yes
Loc	Bahl & Padmanabhan (KNN, RADAR) [32]	kNN fingerprinting	Median ~2.94 m baseline; ~2.13 m with k=3 averaging.	Yes	Yes	No	No	Partial	No	No	Partial	Yes
Loc	Kang et al. (Graph GNN) [29]	High-order GNN	CDF(80%) ≤ 1.29 m; improves vs KNN/DNN/SGC/GAT.	Partial	No	Partial	No	Partial	No	No	Partial	Partial
Loc	This work: MLP-based localization integrated in CF control	Centroid/trilateration + MLP ML regressors	Hybrid localization integrated with KPI-aware CF control.	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Criteria: decentralized operation; deterministic cluster agreement from a minimal broadcast; low signaling/fronthaul; edge-compute friendliness; robustness to imperfect CSI/synchronization and NLoS; hybrid localization; explicit KPI coupling (throughput, energy, fairness); QoS-aware control; practical scalability; autonomous distributed dynamics.

communication links can be classified into three distinct control paradigms [33]:

- **Centralized:** The base station (BS) fully controls the management of user equipment (UE), even when devices communicate directly with one another. In this mode, the base station (BS) centrally manages critical aspects, such as interference, link establishment, resource allocation, and path determination, for device-to-device (D2D) pairs.
- **Distributed:** Under a distributed control paradigm, UEs autonomously manage communication processes such as interference handling, data rate optimization, and routing without relying on a central controller. This approach significantly reduces control and computational overhead, making it highly suitable for large-scale D2D networks. Typically, control tasks are initiated simultaneously by all participating UEs and executed concurrently.
- **Semi-Distributed (Hybrid):** This approach merges the advantages of centralized and distributed control methods. It involves trade-offs between central oversight and UE autonomy to optimize performance, resulting in flexible and efficient management strategies tailored to specific network requirements.

2.2.1.2 | Transmission Modes in D2D Communication.

Different transmission modes define how user equipment interacts with the base station and other D2D nodes. These modes are classified as follows [33]:

- **D2D Direct:** Two UEs communicate directly without intervention from the BS, using either licensed or unlicensed spectrum. Communication remains exclusively between the two involved devices.
- **D2D Single-Hop Relaying (D2D Relay)/ Access Point (AP):** One UE connected to the BS or an access point (AP) shares its resources, acting as a relay, thereby facilitating connectivity for another UE that lacks direct BS connectivity.
- **D2D Multi-Hop Relay:** This mode extends single-hop relaying by creating a chain of multiple UEs, where each device serves as an intermediate relay. Data transmission and backhaul connections occur through successive relay nodes, with each node managing connections to subsequent nodes in the chain.
- **D2D Cluster Clients / D2D Clients:** A group of UEs forms a cluster managed by a designated cluster head (CH), which acts as a relay node (the D2D Relay). The CH/D2D Relay connects the clustered UEs to the broader network via an access point or BS, making clustering particularly effective in scenarios with high user density.

In our research, we will focus on the D2D-Relays and D2D Clients transmission mode selection. We will offer BDIx agents the option to decide their transmission mode autonomously.

2.2.2 | Distributed AI (DAI) Framework and BDIx Agents

In the context of next-generation 6G networks, the DAI framework offers a scalable and adaptable approach to managing network complexity, facilitating real-time decisions, optimizing resource allocation, and substantially reducing signaling overhead. The decentralized approach offered by DAI, minimizes the dependency on central controllers, thereby enhancing robustness in dynamic network environments [33, 34].

Specifically, the DAI framework leverages advanced BDIx agents installed directly on user equipment (UE), enabling decentralized and autonomous decision-making through predefined communication protocols [35, 36]. BDIx agents represent an advanced extension of the traditional belief-desire-intention (BDI) agent model, specifically tailored for dynamic 5G/6G scenarios [35]. These agents implement structured decision-making processes consisting of four principal components:

1. **Beliefs:** Agents maintain and continuously update perceptions of their surrounding environment, capturing parameters such as signal strength, interference, and user equipment density.
2. **Desires:** High-level goals or objectives, such as minimizing latency, maximizing throughput, or improving energy efficiency.
3. **Intentions:** Commitments made by agents toward achieving feasible and prioritized desires.
4. **Plans:** Predefined algorithms or strategies executed to accomplish specific intentions. The plans are related to desires.

Figure 1 illustrates the interaction among these components. Beliefs inform desires by providing the environmental context, allowing agents to prioritize goals effectively. When desires are deemed both necessary and achievable, they transition into intentions. Intentions subsequently activate associated plans for execution. Plans, in turn, influence and update beliefs based on actions taken and observed environmental outcomes, thereby completing the decision-making cycle [33, 35].

Although aligned with fundamental principles of multi-agent systems (MAS), a key distinction in this DAI approach is the agents' capability to directly issue execution commands to peer agents [34, 35]. Additionally, each BDIx agent incorporates a Plan Library responsible for dynamically prioritizing desires. When a desire reaches maximum priority (100%), it transforms into an intention, initiating the execution of its corresponding plan. This mechanism ensures seamless transitions from deliberation to action in response to real-time demands [36]. Furthermore, the Plan Library incorporates advanced methodologies, such as fuzzy logic and machine learning, enhancing agents' adaptability in rapidly changing network conditions [35, 36]. To manage concurrent intentions effectively, agents utilize a goals queue, which restricts simultaneous intention execution and avoids potential conflicts or resource contention [34, 35].

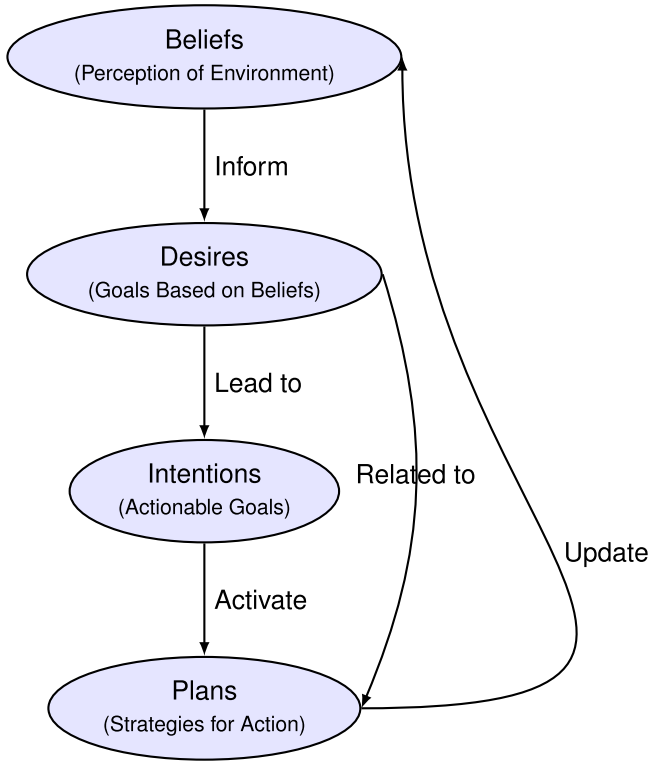


FIGURE 1 | Flowchart depicting interactions among beliefs, desires, intentions, and plans in the BDIX agent architecture.

2.2.3 | Clustering Techniques

Clustering is a fundamental task in machine learning and data analysis that aims to discover inherent groupings within datasets. It is an unsupervised learning technique that aims to partition data points into distinct clusters based on their similarity. This section examines various clustering approaches, spanning from classical algorithms to deep learning-based techniques, offering detailed explanations of their working principles, algorithms, mathematical formulations, and relevant literature references.

Gaussian mixture model (GMM) is a probabilistic clustering approach that models the data as a mixture of multiple Gaussian distributions [37]. Each cluster is represented by a Gaussian distribution with parameters μ_k , Σ_k , and π_k , which denote the mean, covariance, and mixing coefficient, respectively. The probability density function is given by:

$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x | \mu_k, \Sigma_k) \quad (1)$$

where $p(x)$ is the probability density function for a data point x , K is the total number of Gaussian components (clusters), π_k is the mixing coefficient for the k -th component, and $\mathcal{N}(x | \mu_k, \Sigma_k)$ is the Gaussian probability density function with mean μ_k and covariance matrix Σ_k . GMM employs the expectation-maximization (EM) algorithm during execution, where the E-step computes responsibilities (posterior probabilities) and the M-step updates parameters to maximize the likelihood. Unlike K-Means, GMM assigns probabilities for cluster membership

rather than complex labels, resulting in soft clustering with probabilistic interpretations.

Self-organizing map (SOM) is an unsupervised neural-network-based method introduced by Kohonen [38] for projecting high-dimensional data onto a lower-dimensional (typically 2D) map while preserving topological properties. During training, for each input x , the best-matching unit (BMU) is identified based on Euclidean distance. The weights w_i of the BMU and its neighboring neurons are updated according to:

$$w_i(t+1) = w_i(t) + \alpha(t)h_{ci}(t)(x - w_i(t)) \quad (2)$$

where $w_i(t)$ is the weight vector of neuron i at time step t , $\alpha(t)$ is the learning rate, and $h_{ci}(t)$ is the neighborhood function centered at the best-matching unit (BMU) c . Execution involves iterating over the dataset with a decaying learning rate and neighborhood radius. The output is a 2D map where neighboring neurons represent similar data patterns, making SOM useful for visualization and topology-preserving clustering.

ClusterGAN is a clustering framework that combines generative adversarial networks (GANs) with clustering objectives, proposed by Mukherjee et al. [39]. It extends GANs by introducing a structured latent space consisting of a discrete variable c (representing cluster ID) and a continuous variable z (representing intra-cluster variations). The generator maps (z, c) to realistic samples, and an encoder attempts to recover (z, c) from the generated sample, enforcing latent consistency. The training objective includes the traditional adversarial loss plus a reconstruction loss in the latent space:

$$\begin{aligned} \mathcal{L}_{\text{ClusterGAN}} = & \mathbb{E}[\log D(x)] \\ & + \mathbb{E}[\log(1 - D(G(z, c)))] \\ & + \mathbb{E}[||E(G(z, c)) - (z, c)||^2] \end{aligned} \quad (3)$$

where $\mathcal{L}_{\text{ClusterGAN}}$ is the total loss function, D is the discriminator, G is the generator, E is an encoder, x is a real data sample, and (z, c) are the latent variables representing continuous and discrete components, respectively. During execution, samples are clustered by inferring their discrete latent variable via the encoder. The result is a generative model that supports both data generation and cluster assignment.

DBSCAN (density-based spatial clustering of applications with noise) is a density-based clustering algorithm proposed by Ester et al. [40]. It identifies clusters as areas of high point density separated by areas of low point density. The algorithm requires two parameters: the radius ϵ and the minimum number of points $MinPts$. A point is considered a core point if at least $MinPts$ points fall within its ϵ -neighborhood. A point reachable from a core point is part of the same cluster. The algorithm distinguishes between core points, border points, and noise, offering robustness to outliers. Formally, the ϵ -neighborhood of a point p is defined as $N_\epsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \epsilon\}$, and clustering proceeds by expanding clusters from core points.

KMeans is one of the most widely used partition-based clustering algorithms, introduced by MacQueen [41]. It aims to partition

n observations into k clusters, where each observation belongs to the cluster with the nearest mean, serving as a prototype for that cluster. The algorithm starts by initializing k centroids randomly and iteratively updates them using the formula $\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$, where C_j denotes the set of points in the j -th cluster. The process repeats until convergence, typically when the assignments no longer change or the centroids stabilize.

SpectralNet is a deep learning-based clustering method that approximates spectral clustering using a neural network, proposed by Shaham et al. [42]. It embeds data into a lower-dimensional space while maintaining the spectral properties of the graph Laplacian. The loss is based on the normalized cut, and the network is trained to produce orthonormal outputs via an orthogonality constraint: $Y^T Y = I$. The similarity matrix W is constructed from input data using k -nearest neighbors and Gaussian kernels. SpectralNet avoids eigen decomposition by training a neural network to mimic the spectral embedding.

Deep agglomerative clustering (DAC) is a deep clustering model introduced by Yang et al. [43] that uses a convolutional neural network to map inputs to embeddings and performs agglomerative clustering jointly during training. Initially, each sample is treated as a singleton cluster, and during training, the closest pair of clusters is merged based on cosine similarity or Euclidean distance of embeddings. The loss function integrates clustering constraints and feature discrimination objectives to guide the network in learning clustering-friendly representations.

Deep autoencoder-based mixture clustering (DAMIC) is a hybrid model that integrates the autoencoder with mixture models. It jointly learns feature representations and performs clustering by optimizing a composite loss that combines the reconstruction loss from the autoencoder with likelihood maximization for the mixture model in the latent space. Let $z = f(x)$ be the encoding, and $\mathcal{L}_{DAMIC} = \mathcal{L}_{reconstruction} + \mathcal{L}_{mixture}$, where $\mathcal{L}_{mixture}$ maximizes the log-likelihood of the mixture model assuming $z \sim \sum_k \pi_k \mathcal{N}(\mu_k, \Sigma_k)$ [44].

2.2.4 | Location Prediction Techniques

Location prediction is a vital problem in wireless communication systems, where the goal is to estimate a user equipment's (UE's) position using available environmental, signal, or network features. Both classical machine learning methods and advanced deep learning approaches have been effectively applied to this task. In the following sections, we describe several prominent models used for location prediction, detailing the underlying algorithms with inline formulas and their practical implementations.

Bidirectional Long Short-Term Memory (BLSTM): BLSTM networks are a variant of recurrent neural networks that incorporate both forward and backward passes to capture past and future contexts in a sequence. *Algorithm*: In a BLSTM, the input sequence $x = \{x_1, x_2, \dots, x_T\}$ is processed simultaneously by a forward LSTM that computes hidden states $\vec{h}_t$ for $t = 1$ to T and a backward LSTM that computes $\overleftarrow{h}_t$ for $t = T$ down to 1. The final hidden state at each time step is obtained by concatenating these

two sequences, that is, $h_t = [\vec{h}_t; \overleftarrow{h}_t]$, and these representations are fed into one or more fully connected layers with a mean squared error (MSE) loss function for training [45, 46].

Temporal Convolutional Networks (TCN): TCNs offer an alternative to RNNs by applying causal 1D convolutions with increasing dilation factors to capture long-range dependencies in sequential data. *Algorithm*: A TCN processes an input sequence by convolving it with filters while preserving causality; that is, the prediction at time t depends only on inputs at time t and earlier. With an exponentially increasing dilation factor d , the receptive field grows as d^l (where l is the layer index). Residual connections are used to ease the training of deep networks [47].

Convolutional Neural Networks (CNN): Although primarily used for image data, CNNs have been adapted for one-dimensional data in location prediction. *Algorithm*: In a 1D CNN, input features (such as RSSI measurements) are convolved with a set of learnable filters. Each convolution operation is expressed as $y = \sigma(W * x + b)$, where W denotes the filter weights, $*$ the convolution operator, b the bias, and σ a non-linear activation function. This process extracts hierarchical spatial features from the input [46].

Transformer Networks: Transformer models utilize self-attention mechanisms to weigh the importance of different parts of the input sequence without relying on recurrence. *Algorithm*: The core component is the multi-head attention mechanism computed as $\text{Attention}(Q, K, V) = \text{softmax}((QK^T)/\sqrt{d_k})V$, where Q , K , and V are the query, key, and value matrices, respectively, and d_k is the dimensionality of the keys. This attention mechanism is applied in parallel across multiple heads, and the resulting vectors are concatenated and passed through feed-forward layers with residual connections and layer normalization [48].

Multilayer Perceptrons (MLP): MLPs are a simple yet effective baseline for regression tasks. *Algorithm*: An MLP consists of several fully connected layers where each layer operates $y = \sigma(Wx + b)$, with W as the weight matrix, b the bias vector, and σ a non-linear activation function. The network is trained using backpropagation to minimize the MSE between the predicted and actual positions [49].

Graph Neural Networks (GNNs): GNNs, particularly graph convolutional networks (GCNs), are designed to process graph-structured data. *Algorithm*: In a GCN, each feature vector is considered as a node, and the relationships among nodes are encoded in an adjacency matrix A . The update rule for a GCN layer is given by:

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)}), \quad (4)$$

where $H^{(l+1)}$ is the matrix of node features at layer $l + 1$, $\sigma(\cdot)$ is an activation function, $\tilde{A} = A + I$ is the adjacency matrix of the graph with added self-loops, $\tilde{D}$ is the corresponding degree matrix for $\tilde{A}$, $H^{(l)}$ is the input feature matrix at layer l , and $W^{(l)}$ is the weight matrix for layer l . After several such layers, global pooling and fully connected layers produce the final location estimate [50].

Support Vector Regressor (SVR): SVR applies a kernel-based method to regression by finding a function $f(x)$ that deviates from the actual target by no more than a predefined threshold ϵ . SVR solves the optimization problem (as shown in Equation 5):

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{subject to} \quad |y_i - w^T \phi(x_i) - b| \leq \epsilon, \quad (5)$$

where w and b are the model parameters to be optimized, $\phi(x_i)$ maps the input feature vector x_i into a high-dimensional space, y_i is the target value, and ϵ is a margin of tolerance where no penalty is incurred for errors. The kernel trick is used to handle non-linear relationships [51].

Random Forest: Random Forest is an ensemble learning method that aggregates the predictions of multiple decision trees. *Algorithm:* Each tree is trained on a bootstrapped subset of the data with a random subset of features. For regression, the final prediction is the average of the outputs from all trees, which helps to reduce variance and improve robustness [52].

K-Nearest Neighbors (KNN): KNN regression predicts the location by averaging the positions of the k nearest neighbors in the feature space. *Algorithm:* For a test sample, the Euclidean distances to all training samples are computed. The k samples with the smallest distances are selected, and their average is taken as the predicted location. This method is based on the assumption that samples that are close in the feature space are likely to have similar positions [53].

In summary, these diverse approaches—from sequential models like BLSTM and TCN to spatial models such as CNNs and graph-based models like GNNs, along with classical methods like SVR, Random Forest, and KNN—offer a wide range of tools for location prediction. The choice of algorithm depends on the data characteristics and the environmental complexities encountered in modern wireless networks.

3 | System Architecture and Problem Definition

This section provides a concise overview of our cell-free 6G D2D architecture and the associated optimization problem. We outline the system entities and interfaces (BSs, UEs, and cluster-head APs), the BS broadcasts that supply the shared state tuple $(\mathcal{B}, \theta, s)$, which contains the BS locations $\mathcal{B}$, clustering model parameters θ , and a synchronization seed s ; meanwhile, the UE-location set X is locally aggregated by each UE's BDIX agent from LTE ProSe reports, the decentralized and *deterministic* clustering executed by BDIX agents, the AP self-election and UE association policy, the propagation/interference assumptions, and the hybrid localization pipeline that feeds resource control. We then formalize the objective—maximizing energy-efficient throughput under connectivity and range constraints—culminating in the architecture of Figure 2 and the optimization in Equation (6).

3.1 | System Architecture

The emergence of 6G wireless networks significantly enhances device-to-device (D2D) communications by enabling ultra-

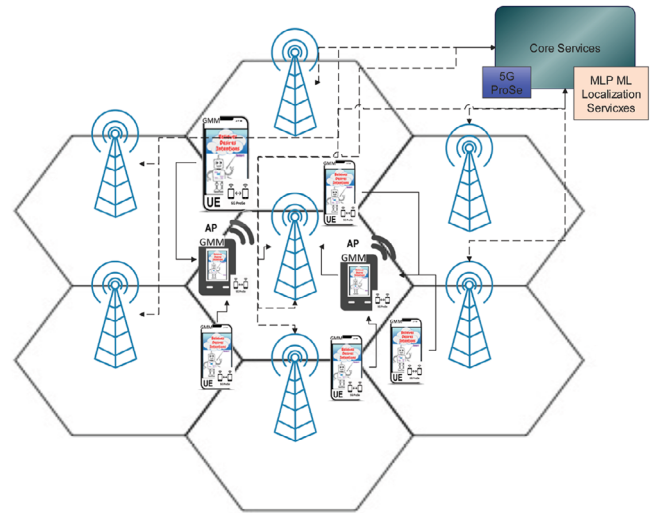


FIGURE 2 | Cell-free system architecture for decentralized D2D communication using BDIX agents.

reliable, low-latency, and high-throughput connections. In a cell-free environment, traditional cell boundaries are eliminated and a large number of distributed access points (APs) work cooperatively to serve all users. This paradigm addresses issues such as cell-edge degradation and inter-cell interference, resulting in more uniform service quality and enhanced spectral efficiency [19, 20].

We consider a cell-free 6G network comprising 7 macro base stations (BSs) equipped with massive MIMO technology and a dense set of User Equipments (UEs). Each UE transmits its coordinates (IMSI, PosX, PosY) using LTE ProSe (Proximity Services) [54, 55], which facilitates proximity-based data sharing. This location information is *locally* aggregated by the BDIX agents at each UE into the set X ; the BSs do not need to centrally aggregate UE locations. Based on this data, UEs are clustered and some are dynamically elected as APs. However, the joint problem of UE clustering and AP selection is challenging due to the high density and dynamic nature of the network, where suboptimal choices can degrade connectivity, increase interference, and lead to inefficient resource allocation. (Note: the BS never learns or aggregates X ; it only broadcasts $(\mathcal{B}, \theta, s)$, preserving full decentralization and privacy of UE-collected proximity data.)

To overcome these limitations, we propose a decentralized approach where each UE is equipped with an enhanced BDIX agent [19]. In our cell-free environment, every UE's BDIX agent autonomously executes a deterministic clustering procedure on the UE-location set X that it assembles via LTE ProSe, using common parameters θ and a synchronization seed s received via the BS broadcast $(\mathcal{B}, \theta, s)$. This ensures that all devices derive identical cluster partitions without the BS needing to compute the cluster assignments itself or track AP roles. The BS disseminates only $(\mathcal{B}, \theta, s)$ (and thus never aggregates or learns X), remaining oblivious to the resulting cluster structure and preserving a decentralized operational model. A hybrid decision-making framework, integrating fuzzy logic with sophisticated machine learning (ML) models (e.g., ANFIS, Random Forest, and support vector machines), is deployed within each agent's Plan Library. This integration significantly enhances decision accu-

racy, computational efficiency, and adaptability to the dynamic nature of 6G networks (as shown in papers [19, 20]).

Instantiation (ours). In our implementation, the deterministic clustering function $M(\cdot)$ is a GMM trained by EM (seeded by the broadcast s) with *full covariances*. The number of components K is selected on X via the Bayesian Information Criterion (BIC), $\text{BIC} = -2 \log \mathcal{L}(\hat{\Theta} | X) + p \log N$, and the fitted component means $\{\mu_k\}_{k=1}^K$ (cf. Equation 1) define cluster centers. AP election is then deterministic: in each cluster k , the UE whose position is closest to μ_k becomes the AP; ties are broken by a stable pseudonymous ID (salted hash of IMSI).

Deterministic canonicalization. After EM converges, components are sorted lexicographically by (μ_x, μ_y) and then by $\text{tr}(\Sigma)$. Within each cluster, the AP is the UE minimizing $\|\mathbf{u}_i - \mu_k\|_2$; ties use a stable pseudonymous ID (salted hash), not IMSI.

In our model, the entities and interfaces comprise UEs with BDIX agents performing local clustering, AP self-election, and link selection; APs (cluster-head UEs) providing intra-cluster Wi-Fi Direct access and backhauling to the nearest BS; and BSs that broadcast the tuple $(\mathcal{B}, \theta, s)$ —containing the BS positions $\mathcal{B}$, clustering parameters, and a seed—and furnish LTE backhaul; we assume all UEs receive the same $(\mathcal{B}, \theta, s)$, that a UE within 100 m of its cluster AP associates via Wi-Fi Direct otherwise via LTE to the nearest BS, and that propagation follows large-scale path loss with Rician small-scale fading with co-channel interference unless stated; the workflow is: the BS broadcasts $(\mathcal{B}, \theta, s)$, each UE (already holding X from LTE ProSe) runs clustering and elects the UE nearest the cluster centroid as AP, UEs associate via Wi-Fi Direct (≤ 100 m) or LTE, metrics (SINR, rate, power) are computed, and a hybrid localization module (centroid/differential-circles/multilateration with DL regressors) refines AP positions for resource control.

The proposed architecture (illustrated in Figure 2) operates entirely in a cell-free setting and comprises four primary components:

- **Central Controller & Base Stations (BSs):** The central controller initializes and coordinates the network by disseminating parameters such as BS positions $\mathcal{B}$, clustering configurations, and synchronization seeds in the broadcast tuple $(\mathcal{B}, \theta, s)$. Our network features 7 BSs that provide high-capacity backhaul links (e.g., via LTE with massive MIMO) but do not confine UEs within fixed cell boundaries.
- **LTE ProSe Services:** Each UE continuously transmits its coordinates and other signal metrics using LTE ProSe, enabling proximity-based communication and serving as a key input for both clustering and localization tasks. UEs aggregate this data during an epoch of duration T_{epoch} ; clustering/election runs at the barrier so every UE uses the same location snapshot X . Late joiners defer to the next epoch.
- **User Equipment (UE) with Embedded BDIX Agents:** Each UE independently executes a deterministic clustering algorithm based solely on its spatial coordinates (collected via LTE ProSe into X ; only θ and s are received via the

BS broadcast $(\mathcal{B}, \theta, s)$). In our cell-free design, the clustering results are identical across all UEs, and those closest to their cluster centroids are dynamically elected as APs. The simultaneous challenge of clustering UEs and accurately selecting APs is critical; any misselection could negatively impact network performance.

- **Distributed Clustering, Connectivity, and Localization:** In this architecture, the elected APs broadcast their status to nearby UEs via Wi-Fi Direct for intra-cluster connectivity, while UEs beyond a 100-meter range connect directly to the BSs. Moreover, accurate localization of the APs is critical for optimal resource allocation and interference management. To address this localization problem, our system incorporates a hybrid framework that combines geometry-based methods (e.g., centroid estimation, differential circles, and multilateration) with advanced deep learning regression models (e.g., MLP, CNN, TCN, and BLSTM) to achieve sub-meter accuracy. The precise AP locations, refined using both LTE ProSe data and signal measurements, are then leveraged by BDIX agents to improve clustering and connectivity decisions continuously.

This design is novel in its ability to guarantee deterministic cluster agreement without centralized processing of UE locations, integrating AP election and association into a single agent loop. It enables scalable, adaptive resource management through continuous feedback among beliefs, desires, intentions, and plans, maintaining high performance in ultra-dense wireless networks. Unlike prior cell-free designs, our framework (i) guarantees *deterministic* cluster agreement across UEs without the BS computing cluster assignments; (ii) integrates AP election and connection rule into a *single* agent loop; and (iii) couples clustering with a lightweight localization pipeline that improves interference management and energy efficiency.

3.2 | Problem Definition and Formulation

The central problem addressed in this work is the optimal configuration of the decentralized D2D network. Given the positions of all UEs, the BDIX agents autonomously perform clustering and AP election. The subsequent challenge is to formalize the network's overall objective, which involves a fundamental trade-off between maximizing the total data throughput for all users and minimizing the total power consumption required to sustain their connections. This translates to an optimization problem where we jointly manage AP roles and UE associations to enhance energy efficiency, subject to connectivity constraints dictated by the system's architecture, such as the limited range of D2D links.

We formally define the objective as maximizing the network's energy-efficient throughput. Let $\mathcal{U}$ be the set of UEs with positions $\mathbf{u}_i \in \mathbb{R}^2$ and $\mathcal{B}$ be the set of BSs with positions $\mathbf{b}_j$. The decision variables are: $a_i \in \{0, 1\}$, indicating if UE i is an AP; $x_{i\ell} \in \{0, 1\}$, indicating if UE i is served by AP ℓ ; and $y_{ij} \in \{0, 1\}$, indicating if UE i is served by BS j . Let $R_i(\cdot)$ be the achievable rate for UE i under Rician fading and co-channel interference, and let $P_i(\cdot)$ be its transmit power. The optimization problem is

formulated as (shown in Equation 6):

$$\begin{aligned}
& \max_{\{a, x, y\}} \sum_{i \in \mathcal{U}'} R_i - \lambda \sum_{i \in \mathcal{U}'} P_i \\
& \text{s.t.} \quad \sum_{\ell \in \mathcal{U}'} x_{i\ell} + \sum_{j \in \mathcal{B}} y_{ij} = 1, \quad \forall i \in \mathcal{U}', \\
& \quad x_{i\ell} \leq a_\ell, \quad \forall i, \ell \in \mathcal{U}', \\
& \quad \|\mathbf{u}_i - \mathbf{u}_\ell\| \leq 100 \text{ m}, \quad \forall i, \ell \in \mathcal{U}' \text{ with } x_{i\ell} = 1, \\
& \quad \sum_{\ell \in \mathcal{U}'} a_\ell = K_{\text{AP}}, \\
& \quad a_\ell, x_{i\ell}, y_{ij} \in \{0, 1\}
\end{aligned} \tag{6}$$

Here, λ is a weighting factor for power consumption, and K_{AP} is the total number of clusters determined by the deterministic clustering algorithm. The first constraint ensures each UE has exactly one connection. The second constraint enforces that a UE can only connect to an AP if that UE has been elected as one. The third constraint models the 100 m range limit for Wi-Fi Direct. The achievable rate is $R_i = B \log_2(1 + \text{SINR}_i)$, where B is the bandwidth. The BDIX agent's decentralized clustering and AP election mechanism effectively determines the values of a_ℓ and K_{AP} , while the association variables (x, y) are set by the distance-based connection rule. It is important to note that the localization of Access Points (APs) in a cell-free 6G architecture is a two-fold problem. The **first fold** is the initial acquisition of coarse-grained position data, where all UEs broadcast their GPS-enabled coordinates via LTE ProSe. This information serves as the primary input (X) for the decentralized clustering and AP election process. The **second fold** involves high-precision position identification, because the BS does not have to know the APs selected. This is a deliberate architectural choice to keep our approach decentralized and allow the APs to be autonomous in handling dynamic information and improving QoS without direct BS intervention. Once a UE is elected as an AP, its exact location becomes critical for network optimization. Therefore, our system employs a hybrid localization module to identify these AP positions to sub-meter accuracy. This identification data is essential for dynamic resource allocation, precise interference management, and ensuring consistent quality of service (QoS), forming a crucial feedback loop for the BDIX agents' decision-making. In particular, we assess QoS fairness using Jain's fairness index (JFI), computed over the per-UE data-rate vector after association and scheduling, following recent practice in wireless resource-allocation studies [56]. Values closer to 1 indicate more equitable service across UEs, while lower values reveal allocation disparities.

4 | Methodology

This section provides a comprehensive overview of the proposed methodology, which is divided into two major stages: (1) clustering and access point (AP) creation, and (2) localization. Each stage involves detailed processes, including synthetic dataset generation, spatial feature engineering, algorithm selection (encompassing both classical and deep learning-based methods), model training, hyperparameter optimization, and integration strategies. The methodology is designed to operate

within a distributed, agent-based architecture using BDIX agents, allowing autonomous clustering and access point selection. Rigorous experimental evaluations validate the effectiveness of the approach, ensuring robust clustering performance and accurate UE localization across diverse wireless network scenarios.

4.1 | Clustering and AP Creation

This subsection focuses on generating logical clusters of user equipments (UEs) and determining optimal access point (AP) placements, forming a structured cell-free network. Throughout this paper, our default clustering method is the GMM trained by EM with shared seed s for determinism. The number of components K is chosen by Bayesian Information Criterion (BIC):

$$\text{BIC} = -2 \log \mathcal{L}(\hat{\Theta} | X) + p \log N,$$

where $\hat{\Theta}$ are the MLE parameters, p is the number of free parameters, and N is the number of UEs. We use full covariances unless stated otherwise.

4.1.1 | Dataset Generation for Clustering

The clustering dataset is synthetically generated to emulate realistic urban wireless environments. User equipments (UEs) are randomly placed within a simulated urban scenario. UE positions are generated uniformly in a 2D plane within a 1000×1000 m hexagonal cellular grid structure (one central base station—BS0—and six neighboring base stations arranged hexagonally). UE coordinates are initially learned via LTE proximity service exchanges [57, 58]. Leveraging existing LTE proximity services, the UEs start broadcasting messages with location information; after each UE has read all messages, it begins to compute the signal metrics as described below.

For each UE, link-level metrics against multiple base stations (BSs) are computed using standardized wireless propagation models derived from LTE specifications, including antenna gains, transmission power, path-loss exponent, and noise floor characteristics [59, 60]. To generate realistic link-level features for clustering, we model signal propagation in six sequential stages: starting from large-scale path loss, then applying small-scale fading, adding thermal noise, computing the received signal strength indicator (RSSI), deriving signal-to-interference-plus-noise ratio (SINR) and achievable data rates (Mbps), and finally estimating UE power consumption (dBm). Signal metrics, such as RSSI, data rates, and power consumption, build on each previous stage to ensure that clustering algorithms receive metrics reflecting actual urban wireless conditions. For each UE, several wireless communication metrics are computed using standard LTE-based propagation models:

- **Friis Path Loss** [61]

$$P_r^{(\text{PL})}(d) = P_t + G_t + G_r - 20 \log_{10} \left(\frac{4\pi d}{\lambda} \right), \tag{7}$$

where $P_r^{(PL)}(d)$ is the received power (dBm) due to free-space path loss at distance d , P_t is the transmit power (dBm), G_t and G_r are the transmit and receive antenna gains (dBi), respectively, and λ is the carrier wavelength (m). This equation models large-scale attenuation of a signal traveling through unobstructed space by summing all transmit/receive gains and subtracting the geometric spreading loss. It provides the baseline received power, which is then modified by small-scale fading and noise models to generate realistic RSSI and SINR values used for clustering.

- **Rician Fading Model** [62] with linear K_{RIC} -factor:

$$h = \sqrt{\frac{K_{\text{RIC}}}{K_{\text{RIC}} + 1}} h_{\text{LOS}} + \sqrt{\frac{1}{K_{\text{RIC}} + 1}} h_{\text{NLOS}},$$

$$P_r(d) = P_r^{(PL)}(d) + 20 \log_{10} |h|. \quad (8)$$

Here K_{RIC} is the (linear) power ratio between the LOS and scattered components. We sweep $K_{\text{RIC, dB}} \in \{0, 3, 6, 10\}$ (converted to linear where used). Where the channel gain h is composed of a deterministic line-of-sight (LOS) component h_{LOS} and a scattered non-line-of-sight (NLOS) component h_{NLOS} (a zero-mean complex Gaussian variable). The Rician K-factor (dB) is the ratio of power in the LOS component to the NLOS component, and $P_r(d)$ is the final received power (dBm) after fading. This model is more practical for cell-free networks, which often feature a mix of LOS and NLOS paths to various APs [63]. The term $20 \log_{10} |h|$ captures the resulting signal fluctuation. **NLoS modeling & robustness. Channel:** We model NLoS via Equation (8); letting $K_{\text{RIC}} \rightarrow 0$ yields Rayleigh (pure NLoS). We optionally include log-normal shadowing on large-scale loss to capture urban blockage, $P_r^{(PL+SH)}(d) = P_r^{(PL)}(d) - X_\sigma$, with $X_\sigma \sim \mathcal{N}(0, \sigma^2)$ (dB). **Control:** Deterministic clustering/AP election depend only on X and common (θ, s) (not instantaneous CSI), so deterministic agreement (Equation 15) is unaffected by NLoS. Association and KPI coupling use SINR (Equations 9–11) and the range rule (Equation 14), which naturally reflect LoS/NLoS conditions. **Localization:** To reduce NLoS bias we use robust multilateration with residual-based weights $w_i = [1 + (r_i/\tau)^2]^{-1}$ and gate outliers before MLP regression; the MLP is trained with NLoS-augmented features so accuracy degrades gracefully as K_{RIC} decreases.

- **Additive White Gaussian Noise (AWGN)** [64]

$$P_{\text{noise}} = 10 \log_{10}(k_B T B) + N_{\text{fig}}, \quad (9)$$

where P_{noise} is the total noise floor (dBm), k_B is Boltzmann's constant (1.38×10^{-23} J/K), T is the system temperature (K), B is the receiver bandwidth (Hz), and N_{fig} is the receiver noise figure (dB). This constant noise level is independent of distance and forms the baseline for thermal and hardware-induced noise.

- **Received Signal Strength Indicator (RSSI)**

$$\text{RSSI}(d) = 10 \log_{10}(10^{P_r(d)/10} + 10^{P_{\text{noise}}/10}). \quad (10)$$

where RSSI (dBm) is the total power measured by the UE, combining the linear powers of the faded signal $P_r(d)$ and

the noise floor P_{noise} . RSSI reflects propagation loss, multipath fading, and thermal noise in one metric, and is used by UEs for AP selection and clustering.

- **Signal-to-Interference-Plus-Noise Ratio (SINR) & Data Rate** [65]

$$\text{SINR}(d) = \frac{10^{P_r(d)/10}}{10^{P_{\text{noise}}/10} + 10^{P_{\text{interf}}/10}}, \quad (11)$$

$$R(d) = B \log_2(1 + \text{SINR}(d)).$$

Units. We report throughput in Mbps as $R_{\text{Mbps}} = (B/10^6) \log_2(1 + \text{SINR})$, while Equation (11) defines $R(d) = B \log_2(1 + \text{SINR})$ in bit/s. *Interference.* The total interference power is given by $P_{\text{interf}} = 10 \log_{10}(\sum_{k \in \mathcal{I}} 10^{P_{r,k}/10})$, where the set $\mathcal{I}$ includes same-channel LTE and/or Wi-Fi Direct interferers (summed in linear power). Here, P_{interf} (dBm) captures co-channel interference from other UEs or BSs, and $R(d)$ is the theoretical maximum throughput (Mbps) given by the Shannon-Hartley theorem. The resulting SINR determines the achievable throughput, guiding clustering by indicating which UEs can sustain higher data rates.

- **PathLossProxy (dB)**

$$P_{\text{consumption}}(d) \approx P_t - P_r(d), \quad (12)$$

where $P_{\text{consumption}}(d)$ (dB) approximates the energy needed to achieve the received power $P_r(d)$ relative to the transmit power P_t . The larger the gap between P_t and $P_r(d)$, the more energy is expended overcoming path loss and fading, helping clusters optimize for energy efficiency.

- **Channel Quality Indicator (CQI)** [66]

$$\text{CQI}(d) = f(\text{SINR}(d)), \quad (13)$$

where $f(\cdot)$ is the 3GPP-defined piecewise mapping function that converts the continuous SINR value to a discrete integer CQI level. CQI informs scheduling and modulation schemes, and is included as a clustering feature to capture network-recommended link reliability.

- **Connection Type Selection** Each UE first computes its distance to all Wi-Fi Direct access points and to all macro BSs:

$$d_{\text{AP}} = \min_i \|\mathbf{u} - \mathbf{a}_i\|, \quad d_{\text{BS}} = \min_j \|\mathbf{u} - \mathbf{b}_j\|, \quad (14)$$

where $\mathbf{u}$ is the UE's 2D position vector, $\mathbf{a}_i$ is the position of the i -th AP, and $\mathbf{b}_j$ is the position of the j -th BS. Then the UE applies the rule:

Association. If $d_{\text{AP}} \leq 100$ m and $\text{SINR}_{\text{AP}} \geq \tau_{\text{SINR}}$, connect to AP; else connect to nearest BS. (We choose a fixed SINR threshold $\tau_{\text{SINR}} = 3$ dB). This ensures UEs within 100 m of an AP leverage high-rate, low-power Wi-Fi Direct links, while those beyond AP range or with poor links fall back to the strongest macrocell BS, preserving coverage and balancing network load.

Each UE repeats these computations against all BSs and candidate cluster heads. The resulting feature vectors (RSSI, data rate, power consumption, CQI, etc.) for each UE form the

TABLE 3 | Input features to the clustering model.

Feature	Description
PosX, PosY	UE's 2D coordinates (in meters)

TABLE 4 | Output of the clustering model.

Feature	Description
ConnectedClusterID	Cluster ID assigned to each UE
ClusterHeadPosX, ClusterHeadPosY	Coordinates of the centroid or cluster head

input to clustering algorithms (KMeans, DBSCAN, Autoencoder, ClusterGAN, etc.) and are stored in CSV/Excel for analysis.

4.1.1.1 | Input Features to the Model. Only the spatial coordinates of UEs are provided as input to the machine learning clustering model. More specifically, these coordinates are made available to each UE (but not to the base stations or telecom provider)² through LTE proximity services, so the UEs have GPS and share their coordinates but for the APs we need to do localisation to identify the for the reasons shown at Section 3. Specifically, when a UE joins the network, it broadcasts messages containing its IMSI and current coordinates to nearby UEs. The receiving UEs thereby learn each other's spatial positions without requiring the base stations or the telecom provider to have direct knowledge of these coordinates (Table 3 shows the input features to the clustering model).

4.1.1.2 | Output of the Clustering Model. The model predicts which cluster each UE belongs to and identifies the cluster center (centroid), typically mapped to the closest UE (cluster head) (Table 4 shows the outputs of the clustering model).

4.1.1.3 | Derived Features (Post-Clustering) to be Used By Localization. Once clustering is complete, additional metrics are generated based on access point (AP) selection, wireless link quality, and device association decisions (Table 5).

Overall, this generated dataset serves as the foundation for evaluating and comparing clustering algorithms under realistic wireless network conditions. By simulating both direct and indirect connections (e.g., via cluster heads using Wi-Fi Direct), the dataset enables analysis of load distribution, spectral efficiency, and power consumption in dense UE scenarios. Therefore, extensive feature engineering transforms raw positional data into spatially meaningful inputs. Since clustering is performed solely on UE coordinates, only (*PosX*, *PosY*) are used as inputs to the clustering model. After clustering, additional metrics such as data rates, power consumption, and CQI values are calculated based on the spatial distances between UEs, APs (cluster heads), and base stations (BSs). These post-clustering features are crucial for evaluating network performance, but they are not directly involved in the clustering process itself.

TABLE 5 | Derived features based on AP selection and connectivity.

Feature	Description
DeviceID	Unique ID for the UE
ConnectedBSID	Closest or selected BS to which the UE is connected
ConnectedTo	Indicates if UE is connected via Wi-Fi Direct or regular BS
IsClusterHead	Boolean flag indicating if UE is a cluster head
DataRate(Mbps)	Effective data rate achieved based on selected connection
ConnectionType	Type of link: "Wi-Fi Direct" or "Regular BS"
DistanceToBaseStation_ BS_i, CQI_BS_i, DataRate(Mbps)_BS_i, PowerConsumption(dB)_BS_i	Metrics calculated for each BS ($i = 0 \dots 6$)
DistanceToClusterHead_ AP_CI_j, CQI_AP_CI_j, DataRate(Mbps)_ AP_CI_j, PowerConsumption(dB)_AP_CI_j	Metrics calculated for each cluster head ($j = 1 \dots N$)

4.1.2 | Clustering Techniques and Implementation with BDIX Agents

Multiple clustering methods are utilized to group UEs based on spatial proximity. Classical clustering techniques include KMeans, DBSCAN, MeanShift, and Affinity Propagation [40, 67–69]. These techniques are well-known for their ability to identify spatial patterns based on density, distance thresholds, or affinity matrices. However, they are limited in handling complex distributions, especially in dynamic environments with high node variability. To address these limitations and improve clustering accuracy, especially in non-linearly separable or sparse environments, deep-learning-based clustering methods are adopted. These include autoencoders, variational autoencoders (VAE), deep embedded clustering (DEC), and ClusterGAN [70–72]. These approaches transform spatial coordinates (i.e., (*PosX*, *PosY*)) into latent representations via neural networks, which are then used for robust clustering. These methods enhance noise tolerance, uncover latent groupings, and improve separability in complex environments.

For all machine learning approaches, the dataset is split into 70% for training and 30% for validation. To further ensure the generalization ability of the models, a K-fold cross-validation approach with $K = 10$ is applied. This helps in evaluating the models' performance more reliably by training and testing them across different subsets of the data.

Each clustering method is rigorously tuned using hyperparameter optimization techniques. Metrics such as silhouette score,

Davies–Bouldin index, and Calinski–Harabasz index are used as objectives in GridSearchCV to ensure each clustering configuration produces compact, well-separated, and balanced clusters [73, 74]. All UEs are equipped with BDIX agents that are capable of performing local clustering computations. The role of the BDIX agents is not limited to perception and reasoning but extends to local decision-making and communication within the distributed system.

Upon network initialization, the central base station broadcasts all devices' clustering parameters. These include:

- BS positions $\mathcal{B}$, the selected clustering model, the clustering hyperparameters θ , and the fixed random seed s .
- Number of UEs under the base station.
- The selected clustering model and its type (e.g., DBSCAN, KMeans).
- Fixed random seed to ensure determinism.
- Number of clusters (if required by the model).
- Any additional hyperparameters needed.

Moreover, each UE using LTE proximity services broadcasts its location and its exact coordinates in the 2D space.

Upon receiving these parameters and the proximity services messages from all the UEs, each UE independently invokes the clustering algorithm through its BDIX agent. Due to the deterministic nature of the algorithm and the synchronized starting parameters, the output clustering results will be identical across all devices. Thus, this methodology ensures:

- **Decentralization:** UEs operate independently with synchronized results.
- **Scalability:** System scales to large networks without centralized bottlenecks.
- **Reliability:** AP selection and cluster assignments are consistent across devices.
- **Efficiency:** Only UEs within Wi-Fi range connect via D2D; others fallback to BSs.

4.1.2.1 | Hyperparameter Identification and Optimization Strategy. The identification of hyperparameters in this framework follows a tightly controlled and reproducible approach, guided by both algorithmic behavior and empirical performance evaluation metrics. Each clustering algorithm, whether classical (e.g., KMeans, DBSCAN, MeanShift, Affinity Propagation) or deep-learning-based (e.g., Autoencoder, VAE, DEC, ClusterGAN), has its own set of hyperparameters that significantly influence clustering quality and AP distribution. These hyperparameters are not arbitrarily assigned; instead, they are derived through structured experimentation using the GridSearchCV framework, which is implemented in the code. The process begins by defining parameter grids specific to each algorithm.

For example, in the case of KMeans, the number of clusters k is swept across a predefined range, often based on heuristic

functions such as $\sqrt{N/2}$ where N is the number of UEs. For DBSCAN, the ϵ (epsilon) radius and the minimum number of points required to form a dense region (minPts) are varied in conjunction with the spatial distribution density observed from the UE placement. Thus, for deep models such as autoencoders or VAEs, latent dimension sizes, number of layers, and activation functions are adjusted iteratively through internal validation.

Each candidate configuration is evaluated using internal validation metrics such as the silhouette score, Davies–Bouldin index, and Calinski–Harabasz index, which are calculated automatically for each run. These metrics serve as optimization objectives within the grid search process, ensuring that selected hyperparameters yield the most compact and well-separated clusters. Additionally, the code includes hooks to store intermediate clustering results and their corresponding metrics for detailed offline comparison. Hyperparameter tuning is carried out independently for each clustering algorithm under fixed simulation conditions (e.g., number of UEs, area size, random seed), thereby ensuring consistency and isolating algorithm performance.

Once the optimal configuration is identified, it is reused across different test scenarios for that algorithm, guaranteeing fairness and repeatability in comparative analysis. This level of systematic tuning ensures not only the best possible clustering outcome for each approach but also aligns the clustering structure with real-world constraints, such as coverage radius, AP density, and UE mobility patterns.

4.1.2.2 | Proof of Deterministic Clustering Agreement.

Let X be the dataset representing UE positions, and M be a deterministic clustering algorithm with fixed parameters θ and random seed s . Since $M(X; \theta, s)$ is a pure function of its inputs, it produces a unique and consistent output. When all UEs possess the same X (assembled via LTE ProSe) and the BS broadcasts (θ, s) (with B for backhaul/association), the clustering result is the same on all devices:

$$\forall u_i, u_j \in UEs, \quad M_{u_i}(X; \theta, s) = M_{u_j}(X; \theta, s) \quad (15)$$

where $M_{u_i}(X; \theta, s)$ represents the execution of the clustering algorithm M on user equipment u_i using the shared dataset X , parameters θ , and random seed s . Hence, every UE will compute the same cluster IDs and centroids. This guarantees uniformity in AP selection and cluster assignments across the network.

4.1.3 | Detailed Methodology for BDIX-Based AP and BS Association

After computing the clusters locally, each UE performs an introspective check to determine whether it should act as the access point (AP) of its assigned cluster. This is done by comparing its own position to the centroid of the cluster. If the UE is the closest to the centroid and is stationary (i.e., its velocity $|v_i|$ is below a small threshold v_{th} , e.g., 0.2 m/s), it self-elects as the AP.

Once elected, the UE assumes the AP role and broadcasts a control message to the network, targeting all UEs in the same cluster. This message includes:

- **ClusterID**: The ID of the cluster.
- **AP_Location**: The $(PosX, PosY)$ coordinates of the elected AP.

Each UE, upon receiving this message, performs two key checks:

1. Whether the computed cluster ID matches the one in the broadcast.
2. Whether the distance to the AP is within 100 meters (Wi-Fi Direct threshold).

If both conditions are satisfied, the UE connects to the AP using Wi-Fi Direct. Otherwise, the UE initiates a direct connection to the closest BS using LTE with massive MIMO capabilities. The AP connects to the nearest (or best-SINR) BS to form the backhaul connection. Under mobility, each UE's BDIX agent continuously monitors its speed $v_i(t)$ and short-term data rate $R_i(t)$; if $v_i(t) > \tau_v$ or $R_i(t) < \tau_R$ for W consecutive frames (per DAIS in [75]), the *transmission mode selection* desire is promoted to an intention: the UE re-scores candidate cluster APs and fallback BSs using predicted rate, handover cost, and load, and—subject to hysteresis ($T_{\min}$, Δ_{dB})—may seamlessly re-associate to sustain QoS and fairness.

4.1.3.1 | Enhanced Distributed Clustering and Connection Algorithm. In a distributed cellular network, clustering and connection management are essential for optimizing network performance. The enhanced distributed clustering and connection algorithm (Algorithm 1) manages UE clustering and connectivity based on proximity, network conditions, and infrastructure availability. The algorithm begins with the base station (BS) broadcasting key information to all user equipment (UEs), including their positions, the clustering model, parameters, and a seed for initialization. Each UE then runs the clustering algorithm locally via its BDIX agent, computes the centroid of its cluster, and elects itself as the access point (AP) if it is the closest to the centroid. The elected AP broadcasts its Cluster ID and location to other UEs.

Each UE, upon receiving the broadcast, either connects to the AP if the distance is within 100 meters or connects to the nearest base station (BS) using LTE with massive MIMO. The UE computes performance metrics for both connections, such as throughput, SINR, and power. Each AP also connects to the nearest BS for backhaul connectivity. Finally, the topology and connection metrics are stored for analysis.

4.1.4 | End-to-End Clustering and Cell-Free Formation with BDIX Agents

The algorithm presented in Algorithm 2 outlines a fully distributed, intelligent pipeline enabling seamless clustering and cell-free formation in next-generation networks using BDIX agents. Initially, the base station (BS) generates the network layout by assigning positions to user equipments (UEs) and broadcasting configuration parameters. Then, for each specified clustering technique (ranging from traditional approaches, such as K-Means and DBSCAN, to deep learning-based methods,

ALGORITHM 1 | BDIX-Based Clustering and Connectivity Evaluation.

```

1: Set clustering model:= GMM
2: Base Station (BS) broadcasts ⟨BSs positions, clustering
   model, parameters, seed⟩.
3: UE gathers all positions of the UEs through LTE ProSe
4: for each UE  $u_j$  do
5:   Run clustering algorithm locally via BDIX agent.
6:   Identify cluster  $C_i$  and calculate centroid.
7:   if  $u_j$  is the closest UE to the centroid and  $|v_i| < v_{th}$ 
       then
8:     Elect self as AP of cluster  $C_i$ .
9:     Broadcast ⟨ClusterID, AP_Location⟩ to all UEs.
10:  end if
11: end for
12: for each UE  $u_k$  receiving a broadcast do
13:   if cluster ID matches and distance to AP  $\leq 100$  m then
14:     Connect to AP via Wi-Fi Direct.
15:     Compute Wi-Fi Direct metrics: throughput, CQI,
       SINR, power.
16:   else
17:     Connect to the closest BS via LTE with massive
       MIMO.
18:     Compute LTE-based metrics.
19:   end if
20: end for
21: for each AP do
22:   Find and associate with the nearest BS.
23: end for
24: Store full topology and connection metrics.
```

including autoencoder and deep embedded clustering (DEC)), each UE activates its embedded BDIX agent to perform local clustering. Note that by integrating of BDIX agents into the clustering and access procedure, we offer a novel approach to autonomous, intelligent, and distributed network formation in cell-free wireless architectures.

A UE closest to the centroid is self-elected as an access point (AP) and broadcasts its status to nearby UEs. Neighboring UEs that belong to the same cluster and lie within a 100-m radius form ad hoc links using Wi-Fi Direct, minimizing reliance on infrastructure. UEs that do not meet these conditions connect to the nearest base station (BS) using LTE. Once the connections are established, each UE computes essential network metrics such as signal-to-interference-plus-noise ratio (SINR), data rate, channel quality indicator (CQI), and power consumption. Elected APs then form backhaul connections to the BS, completing the cell-free architecture.

This approach ensures dynamic, energy-aware, and context-sensitive network formation tailored to the device topology

ALGORITHM 2 | Full BDIX-Aware Cell-Free Network Formation.

```

1: Initialization: UEs share positions via LTE ProSe
   (forming  $X$ ); BS selects clustering model.
2: Set clustering model:= GMM; Set localization model:=
   MLP
3: BS broadcasts initial parameters ( $B, \theta, s$ ) to all UEs.
4: for each clustering model in {KMeans, DBSCAN,
   Autoencoder, DEC, etc.} do
5:   for each UE do
6:     BDIX agent executes clustering locally.
7:     if UE is closest to the centroid then
8:       Elect self as AP and broadcast the cluster
       message.
9:     end if
10:  end for
11: for each UE receiving broadcast do
12:   if same cluster and within 100m then
13:     Connect to AP via Wi-Fi Direct.
14:   else
15:     Connect to the nearest BS using LTE.
16:   end if
17:   Compute SINR, data rate, CQI, and power.
18: end for
19: APs associate with their nearest BS.
20: Store the complete network configuration.
21: end for

```

and mobility, providing scalability and adaptability for diverse deployment scenarios. The overall network configuration is stored at each iteration, enabling performance evaluation and optimization across different clustering paradigms. So, the complete distributed system operation, integrating clustering, agent-based AP selection, and network connection, is summarized in Algorithm 2.

4.2 | Localization

The localization stage aims to predict accurate UE positions by employing supervised learning models trained on enhanced datasets derived from the clustering results and signal measurements.

4.2.1 | Dataset Generation for Localization

The localization dataset is generated by aggregating the results from multiple simulation runs and various clustering approaches. Thus, UEs' APs records are selected from all approaches and across all ranges and runs of the simulations, ensuring that the dataset represents comprehensive spatial diversity and a wide range of connectivity scenarios. The selected UEs originate from the outcomes of both the clustering and connectivity stages,

capturing the complete spectrum of AP contributions without relying on any specific enumeration method for AP identification.

The dataset construction begins with filtering the UEs' records that are candidates to become APs. Specifically, only UEs that were confirmed to be Access Points and had valid cluster connections, as determined during the clustering phase, are retained. Records are further refined by eliminating entries with missing or invalid BS connection identifiers and by restricting the dataset to include only the relevant BSs within the designated range. Any record containing Not Any (NaN)/null values after deriving additional features is also excluded. This meticulous filtering process ensures that the dataset is both robust and representative of realistic connectivity behaviors.

The localization estimation process incorporates several methods to accurately determine the UE positions while optimizing the overall network power consumption. One method, referred to as the centroid-based estimation, computes the geometric mean of the coordinates of four carefully selected BSs: the connected BS and the three BSs exhibiting the smallest inferred data-rate distances. This approach provides a BS-driven position estimate and includes the calculation of the Euclidean error between the actual UE position and the computed centroid. Another method employed is the differential circles-based position estimation. In this approach, the BSs are conceptually divided into two groups. The first group is formed by combining the connected BS with the three nearest BSs (as determined by the inferred data-rate distances), while the second group comprises the remaining BSs. For each group, the minimum enclosing circle is computed using the Welzl algorithm, and the centers of these circles are then used to derive the estimated UE position by calculating the differential offset between them. The error metric in this method is also the Euclidean distance between the estimated and actual positions.

Non-linear multilateration is applied to further refine the UE location. This method leverages the Levenberg–Marquardt algorithm to perform non-linear regression on the estimated distances between the UE and multiple BSs. Variants of this multilateration approach utilize different sets of base stations (BSs) and distance estimations, including those derived from both data rate inversion and power consumption models. The final predicted position minimizes the residual error between the squared differences of the predicted distances and the actual distances to the BSs.

For each BS, two types of distance features are computed. One is derived from the data rate using the inverted Shannon capacity and Friis path-loss equations. At the same time, the other is calculated from power measurements using a modified MIMO-aware path-loss model. The actual Euclidean distances between each BS and the UE's ground-truth coordinates are also calculated and stored to serve as a benchmark for evaluating the accuracy of the various localization methods.

The complete feature set encompasses not only the outputs of these diverse localization methods but also the signal inversion-based distance estimates and the connectivity information the clustering process provides. This integrated approach ensures that every AP, regardless of its range, contributes to a feature-rich dataset suitable for training robust supervised learning models.

TABLE 6 | Input features to the localization models.

Feature	Description
CentroidX, CentroidY	- Geometric mean of the connected BS and the three closest BSs (by data rate). $\text{CentroidX} = \frac{1}{4} \sum x_i$, $\text{CentroidY} = \frac{1}{4} \sum y_i$.
CirclesPredX, CirclesPredY	- Position estimated via the differential circles method. - Derived from the difference between the centers of the minimum enclosing circles of two BS groups.
calculated_pos_x, y	- Multilateration using all BSs with distances estimated from data rate inversion. - Employs the Levenberg-Marquardt algorithm for non-linear regression.
calculated_pos_pw_x, y	Multilateration using all BSs with distances derived from a power attenuation model.
calculated_pos_N_x, y	- Position estimates using a variable number ($N = 3, 4, 5, 6$) of BSs. - Combines the connected BS with the $(N - 1)$ closest BSs.
distance_dr_i, distance_pw_i	Approximate distances to each BS estimated from data rate and power metrics, respectively.
distance_pos_i	True Euclidean distance from each BS to the UE: $\sqrt{(x_i - x)^2 + (y_i - y)^2}$.
CentroidDistance, CirclesDistance, DistanceCalculated_N	Euclidean error metrics between each method's predicted and actual UE positions.

For your reference, the input features for the localization models are detailed in Table 6. Moreover, Table 7 shows the target labels (output features) of the localization models.

The final dataset, which integrates results from all clustering approaches and encompasses the full range of AP contributions, is used for training localization models and establishing baseline performance. Standard evaluation metrics such as mean absolute error (MAE), root mean squared error (RMSE), R^2 , and Euclidean distance errors are computed to assess the performance of the centroid- and circles-based methods. The comprehensive and

TABLE 7 | Target labels for localization.

Label	Description
PosX, PosY	Ground truth 2D coordinates of the UE in the simulation space.

ALGORITHM 3 | Localization Model Training and Optimization.

- 1: Load, preprocess, and split localization dataset.
- 2: Conduct advanced feature selection and normalization.
- 3: Optimize hyperparameters via RandomizedSearchCV with cross-validation.
- 4: Train diverse localization models with early stopping.
- 5: Evaluate localization performance metrics on the test set.
- 6: Identify and select the best-performing model.

feature-rich dataset thus enables robust supervised learning of UE positions in realistic and diverse wireless environments.

4.2.2 | Localization Models and Training

Localization models range from classical algorithms to advanced neural networks. Traditional supervised regression models, including Random Forest, support vector regression (SVR), and K-nearest neighbors (KNN), serve as benchmarks for comparison. Advanced deep learning architectures such as bidirectional LSTM, temporal convolutional networks (TCN), convolutional neural networks (CNNs), transformer-based models, and graph neural networks (GNNs) are thoroughly evaluated for their capabilities in capturing spatial and sequential dependencies effectively [46, 48, 76].

For all machine learning approaches, the dataset is split into 70% for training and 30% for validation. Models are trained using early stopping techniques with a validation set, monitoring performance metrics such as MSE, MAE, RMSE, mean absolute percentage error (MAPE), and R-squared values. Early stopping ensures optimal training, preventing overfitting, and preserving generalizability. Hyperparameters are optimized systematically through RandomizedSearchCV and GridSearchCV strategies, extensively exploring parameters such as learning rates, number of layers, hidden units, batch sizes, and regularization terms. Cross-validation ensures robust hyperparameter selection that minimizes validation errors and improves generalization [74]. Additionally, K-fold cross-validation with $K = 10$ is employed to provide a reliable performance evaluation [77] and to avoid problems with overfitting and underfitting.

The localization training and optimization procedure is outlined clearly in Algorithm 3.

Iterative integration strategies directly incorporate clustering outcomes, such as cluster centroids and AP assignments, as localization input features. This iterative refinement process continuously updates localization inputs with spatial clustering insights, leading to progressively improved localization

accuracy. The implemented pipeline reads the dataset from a structured CSV file containing raw and engineered spatial features. Input features include multiple multilateration estimates (calculated_pos_x, calculated_pos_pw_x, etc.), cluster-based centroids (CentroidX, CentroidY), and geometric circle predictions (CirclesPredX, CirclesPredY). These features are normalized using Min-Max scaling, and the ground-truth coordinates (PosX, PosY) are also scaled accordingly.

The feature selection process proceeds in four stages: (i) low-variance thresholding removes uninformative features; (ii) univariate selection uses F-statistics for regression to retain highly correlated variables; (iii) tree-based feature selection applies *ExtraTreesRegressor* to extract features based on learned importance weights; and finally, (iv) forward sequential feature selection incrementally builds a predictive feature set using performance-based inclusion. Each selection strategy is applied independently, and selected features are logged and reported in the final CSV output for traceability. Input reshaping is performed based on model type. Recurrent and convolutional models (BLSTM, TCN, CNN, and transformer) reshape inputs to 3D tensors of shape (*samples*, 1, *features*). Graph neural networks reshape inputs into (*samples*, *features*, 1), where each feature is treated as a node, and a batch-wise identity adjacency matrix is created dynamically to represent independent graphs per UE sample. Classical machine learning models (Random Forest, SVR, KNN) and MLPs utilize standard 2D tabular inputs.

All deep models are constructed with multiple layers. BLSTM models contain three stacked bidirectional LSTM layers, followed by two dense layers with ReLU activations. TCN models use stacked causal dilated convolutions with increasing filter depth. CNN models employ three 1D convolutional layers with a kernel size of 1 to maintain sequence resolution, followed by flattening and dense layers. Transformer models utilize 4-head self-attention, layer normalization, and dense projections. GNNs use two spektral GCNConv layers and a global average pooling layer followed by dense layers. Each model outputs two continuous values corresponding to PosX and PosY, trained jointly using MSE loss and the Adam optimizer. Early stopping is employed to prevent overfitting, with the validation loss monitored using a patience threshold of 5 epochs.

Evaluation is performed in two modes: joint prediction and separate coordinate prediction. In joint mode, a single model predicts both coordinates. In the individual mode, two parallel models are trained and evaluated for PosX and PosY, respectively. This allows the system to assess whether learning each axis independently yields improved accuracy or better generalization. Multiple evaluation metrics are calculated and recorded. These include traditional metrics such as MSE, MAE, and RMSE, as well as localization-specific metrics, including mean Euclidean distance error (MeanDistanceError), maximum error distance, and its standard deviation. Additionally, statistical metrics such as R^2 , mean squared logarithmic error (MSLE), median absolute error, and explained variance are calculated. Results are written to CSV files for both joint and separate model runs, and extended metrics are saved separately for post-analysis.

Finally, the pipeline includes an additional experiment using a reduced subset of features. This is designed to test model

robustness when provided with minimal input, limited to only three multilateration estimates and the circle-based predictions. The entire training-evaluation pipeline is re-run under these constraints, and results are separately saved for comparative analysis. This setup enables insight into how different models degrade or sustain performance under constrained conditions.

The complete modular design allows extension to additional models or feature sources and supports experimentation under varying feature subsets, clustering assumptions, and signal environments. The end-to-end framework is fully reproducible and well-suited for large-scale simulations and deployment in intelligent BDIX-agent-based localization systems. For learning-based localization, we adopt a compact MLP that maps the engineered feature vector $\mathbf{x} \in \mathbb{R}^D$ to coordinates via

$$\hat{\mathbf{p}} = f_{\text{MLP}}(\mathbf{x}; \phi) = \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2,$$

trained with MSE loss and early stopping. Unless otherwise noted, all “ours” localization results are produced by this MLP.

4.3 | Overall Flow of the Proposed Framework

Figure 3 summarizes the end-to-end workflow of the proposed system. Initially, user equipments (UEs) broadcast their positions using LTE proximity services (ProSe), from which each UE locally assembles the common dataset X . A base station (BS) then synchronizes all UEs by broadcasting $(\mathcal{B}, \theta, s)$, that is, the BS positions $\mathcal{B}$, the clustering parameters θ , and a random seed s . Each UE executes the same deterministic clustering locally on X using (θ, s) , ensuring identical cluster formations across the network. Following clustering, each UE elects its cluster’s Access Point (AP), defined as the UE closest to the cluster centroid. Based on a 100-meter range check, UEs establish a connection either via Wi-Fi Direct to their AP or via LTE to the nearest BS. Finally, key performance metrics are computed and passed to a localization pipeline, which combines geometric methods with deep learning regressors to produce the final network KPIs and position estimates.

4.3.1 | Performance, Complexity and Signaling of Our Framework

With our reference setup—GMM for clustering (full-covariance EM, K selected by BIC, seeded by s) and an MLP regressor for localization—the BDIX-driven, fully decentralized framework achieves up to **+46.3%** sum-rate gain and **−32.8%** total power reduction versus the cellular baseline under identical conditions; the hybrid localization reaches sub-meter accuracy (MeanDist < 1 m, $R^2 > 0.999$) within the same distributed control loop. These improvements derive from deterministic clustering and AP election executed locally from the minimal BS broadcast $(\mathcal{B}, \theta, s)$, a KPI-coupled association rule (cf. Equation 14), and hybrid localization that closes the loop in the optimization of Equation (6). Computationally, per-UE clustering costs are $\mathcal{O}(NKI)$ for iterative mixture/centroid methods (e.g., KMeans, GMM), and on average $\mathcal{O}(N \log N)$ for density/mode-seeking methods with spatial indexing (e.g., DBSCAN, MeanShift); deep-clustering *inference* is $\mathcal{O}(P)$ for P parameters. Association/neighborhood queries via

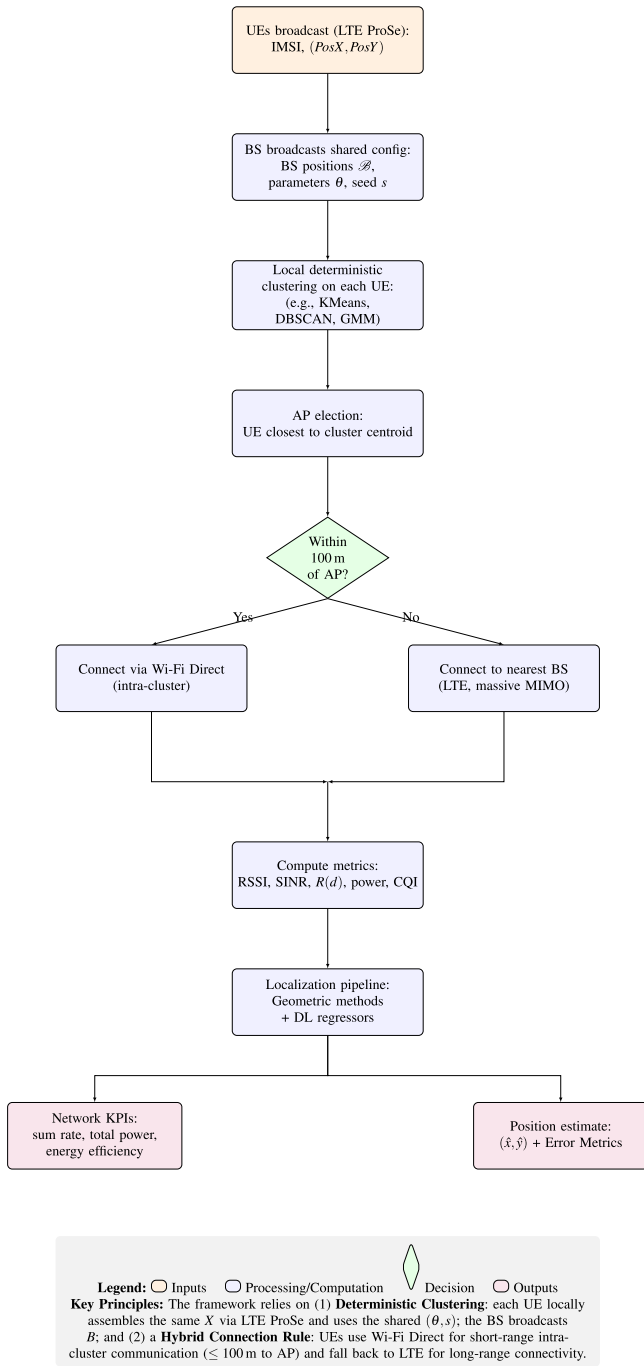


FIGURE 3 | Overall flow of the proposed framework. The process begins with synchronized data sharing, followed by local deterministic clustering and access point (AP) election on each user device. A distance-based rule then governs whether a UE connects via Wi-Fi Direct or LTE. Finally, system metrics are calculated and fed into a localization pipeline to generate network KPIs and position estimates.

KD-trees are $\mathcal{O}(N \log K)$, MLP localization is effectively constant-time per UE, and the BDix loop adds only $\mathcal{O}(1)$ overhead. Signaling is likewise light: one $\mathcal{O}(1)$ BS broadcast of (B, θ, s) per epoch, $\mathcal{O}(N)$ ProSe peer messages for position sharing, and $\mathcal{O}(K)$ AP notices—eliminating fronthaul for clustering—whereas centralized schemes require $\mathcal{O}(N)$ uplink, $\mathcal{O}(N)$ downlink, and a centralized compute bottleneck of at least $\mathcal{O}(NKi)$. Because every UE runs the same deterministic pipeline, execution is

embarrassingly parallel and wall-clock per epoch is governed by the maximum per-UE cost rather than a server, enabling sub-second decision cycles on commodity hardware.

5 | Simulation Results and Analysis

This section presents the simulation setup, key parameters, and comprehensive performance analysis for a dense urban 6G cell-free network. The simulation spans a 1000×1000 m area with 5600 UEs and seven base stations arranged in a hexagonal grid. Parameters include communication via Wi-Fi Direct and LTE (massive MIMO), as well as clustering based on user positions (PosX, PosY). Metrics used for clustering evaluation include the Silhouette Score, DBI, CHI, sum rate, power consumption, and energy efficiency. Localization performance is assessed using MeanDist, MaxDist, MSE, MAE, RMSE, R^2 , and other metrics. Results demonstrate the superior performance of advanced ML-based approaches over traditional methods in both AP clustering and localization accuracy.

5.1 | Simulation Environment and Hardware Configuration

In our simulation, we have the following hardware implementation for simulation tasks:

- **Processor:** Intel Core i7-10700K with 8 cores and 16 threads, clocked at 3.8 GHz.
- **Memory:** 32GB DDR4 RAM at 3200 MHz.
- **Storage:** 1TB NVMe SSD with up to 3,500 MB/s read and 3,300 MB/s write speed.
- **Graphics:** NVIDIA GeForce RTX 3070 with 8GB GDDR6 VRAM.
- **Operating System:** Windows 10 Pro 64-bit.
- **Simulation Tools:** ns-3 version 3.41, Python 3.8+, MATLAB (for data analysis), and TensorFlow/Keras (for neural network training).
- **Simulation Environment:** Local simulation environment configured with the required dependencies (e.g., Python libraries, ns-3 packages, CUDA for GPU acceleration) running on a Windows OS with WSL 2 (Windows Subsystem for Linux) to support Linux-based simulation tools.
- **Network Model:** Urban 6G network simulation with massive MIMO, LTE, and Wi-Fi Direct communication, employing clustering techniques like SOM, KMeans, and DBSCAN for decentralized AP selection.
- **Processing Power:** The GPU (e.g., NVIDIA RTX 3090, 35.6 TFLOPS) accelerates neural network computations, especially for real-time clustering and model training tasks. The CPU (e.g., AMD Ryzen 9 5950X, 0.6 TFLOPS) handles data preprocessing, orchestration, and control logic in parallel.

Such a system is capable of efficiently training and deploying neural network models. The processor offers high-speed computation, while the RAM ensures smooth multitasking and

TABLE 8 | Simulation configuration.

Parameter	Value
Simulation area	1000 × 1000m
Number of UEs	100–5600 (step of 500 per run)
UE range	0–1000 m
Number of base stations (BSs)	7 (hexagonal grid)
Massive MIMO antennas	Up to 64 per BS
Communication types	Wi-Fi Direct, LTE
LTE carrier frequency	2 GHz
Wi-Fi direct frequency	5 GHz
Clustering features	PosX, PosY, RSSI, SINR
Clustering models	KMeans, DBSCAN, SOM, GMM, AffinityPropagation, SpectralClustering, ClusterGAN, VaDE, DAC, DAMIC
Noise model	AWGN; thermal noise density $\approx -174\text{dBm/Hz}$; noise floor $= -174 + 10 \log_{10}(B_{\text{Hz}}) + N_{\text{fig}}$ dBm
Signal metrics	RSSI, CQI, SINR, throughput
Localization inputs	Derived from clustering structure
Path loss exponent	3.5 (3GPP UMi)
Fading model	Rician ($K_{\text{RIC,dB}} = 10$)
TX power	24 dBm (260 mW)
TX antenna gain	40 dBi (BS), 2 dBi (UE)
RX antenna gain	2 dBi (UE)
Shadowing (log-normal)	$\sigma = 6\text{--}8$ dB (urban micro)
Fading model - Rician K_{RIC} sweep	$K_{\text{RIC,dB}} \in \{0, 3, 6, 10\}$ (NLoS $\rightarrow$ LoS)
Assoc. threshold	$\tau_{\text{SINR}} = 3$ dB
Bandwidth B	20 MHz

data processing—the rapid storage guarantees fast data access speeds, which are essential when training large dataset models. The GPU accelerates neural network computations and caters to demanding visual applications. The simulation environment enables seamless integration of tools, ensuring efficient execution of simulation tasks, including network modeling, clustering, and localization.

5.2 | Simulation Parameters

The network simulates an ultra-dense 6G urban deployment, utilizing a central base station and six neighboring stations arranged in a hexagonal grid. The simulation parameters are provided in Table 8. UEs utilize BDIX agents to perform decentralized clustering and access point (AP) selection. Signal metrics and connectivity models are derived from LTE and WiFi-Direct

standards, including additive white Gaussian noise (AWGN), the Rician fading model, path loss, and Doppler effects. AWGN is employed to simulate thermal and background noise in the receiver chain [64], while the Rician fading model—which generalizes the Rayleigh model by including a line-of-sight (LOS) component—is used to capture more realistic propagation conditions found in cell-free networks where UEs may have clear paths to some APs [63, 78]. The UE range varies between 0 m and 1000 m, and the system includes a massive MIMO configuration with up to 64 antennas at the base stations. Additionally, Wi-Fi Direct is used for short-range communications, with a maximum distance of 100 m for AP selection.

5.3 | Cluster Results

5.3.1 | Evaluation Metrics

To evaluate the effectiveness of clustering for AP selection in a decentralized cell-free 6G network, we adopted the following key metrics:

- **Silhouette Score** [73]: The silhouette score measures how similar an object is to its cluster compared to other clusters. The formula is:

$$\text{Silhouette score} = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (16)$$

where $a(i)$ is the average distance from the i -th point to other points in the same cluster, and $b(i)$ is the minimum average distance from the i -th point to points in a different cluster. A value close to 1 indicates well-separated clusters [73].

- **Davies–Bouldin Index (DBI)** [79]: The Davies–Bouldin index quantifies the average similarity ratio of each cluster with the cluster that is most similar to it. The formula is:

$$\text{DBI} = \frac{1}{N} \sum_{i=1}^N \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d_{ij}} \right) \quad (17)$$

where N is the number of clusters, σ_i is the average distance of all points in cluster i to its centroid, and d_{ij} is the distance between the centroids of clusters i and j . Lower values indicate better clustering with compact and well-separated clusters [79].

- **Calinski–Harabasz Index (CHI)** [80]: The Calinski–Harabasz index measures the ratio of between-cluster dispersion to within-cluster dispersion. The formula is:

$$\text{CHI} = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \cdot \frac{N - k}{k - 1} \quad (18)$$

where $\text{Tr}(B_k)$ is the trace of the between-cluster dispersion matrix, $\text{Tr}(W_k)$ is the trace of the within-cluster dispersion matrix, N is the total number of data points, and k is the number of clusters. Higher values indicate more compact and well-separated clusters [80].

- **Sum Rate (Mbps)**: The sum rate is the cumulative data rate supported by the network, representing the total throughput

TABLE 9 | Clustering quality metrics.

Approach	Silhouette score	DBI	CHI
GMM (ours)	0.4357	0.7619	5690.03
SOM [20, 21]	0.4064	0.7805	5585.76
KMeans [20, 83]	0.3802	0.8325	5196.99
ClusterGAN [20]	0.3694	0.9412	4856.14
SpectralNet	0.3439	0.9227	4117.51
DAC	0.2474	1.5095	2470.51
DAMIC	0.2433	1.2817	2797.13

across all devices. The sum rate can be computed as:

$$R_{\text{sum}} = \sum_{i \in \mathcal{U}'} R_i, \quad R_i = B \log_2(1 + \text{SINR}_i). \quad (19)$$

where R_i is the data rate for the i -th device and $|\mathcal{U}'|$ is the total number of devices. Higher values represent better network throughput.

- **Total Power Consumption (dBm):** The total power consumption reflects the energy consumed by the network for data transmission. It can be calculated as:

$$\text{Total Power} = \sum_{i=1}^N P_i \quad (20)$$

where P_i is the power consumption for the i -th device. It is measured in dBm (decibels relative to one milliwatt), and lower aggregate values indicate more efficient energy use.

- **Energy Efficiency (Mbps/dBm) [81]:** Energy efficiency is the ratio of throughput to power consumption, representing the efficiency of communication in terms of energy use. The formula is:

$$\text{EE} = \frac{\sum_i R_i}{\sum_i P_{\text{UE},i}} \text{ [bit/J]}. \quad (21)$$

where the sum rate and total power are as defined previously. Higher values indicate more efficient networks with better performance per unit of power consumed [81].

- **Jain's Fairness Index (JFI) [82]:** JFI quantifies how evenly a resource is distributed across users in terms of quality of service. Using per-UE data rates, it is defined as:

$$\text{JFI} = \frac{(\sum_i R_i)^2}{|\mathcal{U}'| \sum_i R_i^2}, \quad (22)$$

where R_i is the data rate of the i -th UE and $|\mathcal{U}'|$ the number of UEs; values near 1 indicate equitable allocations.

5.3.2 | Examination and Analysis of the Cluster Examinations

5.3.2.1 | Clustering Quality Evaluation:. Table 9 presents a comprehensive evaluation of clustering quality using the Silhouette Score, Davies-Bouldin Index (DBI), and Calinski-Harabasz

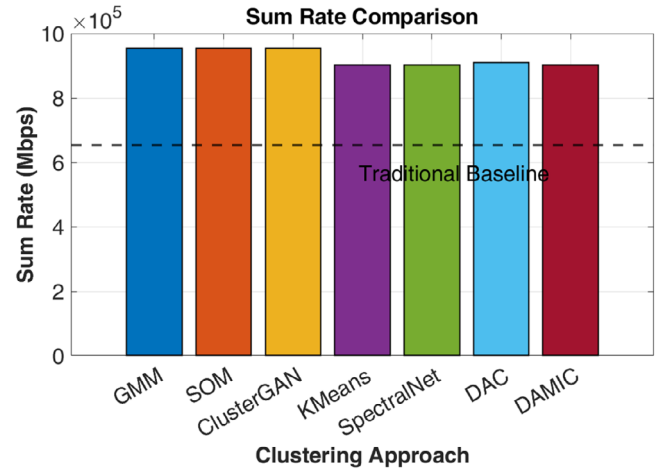


FIGURE 4 | Comparison of sum rate (Mbps) across clustering methods. The figure shows the significant improvements in throughput achieved by advanced clustering techniques compared to the traditional baseline (calculated in our simulation).

Index (CHI). These metrics collectively assess the geometric properties of the clusters: a higher Silhouette Score indicates that objects are well-matched to their own cluster and poorly matched to neighboring clusters; a lower DBI signifies better clustering by measuring the ratio of within-cluster scatter to between-cluster separation; and a higher CHI rewards dense, well-separated clusters. GMM achieved the highest Silhouette Score (0.4357), the lowest DBI (0.7619), and the highest CHI (5690.03), indicating superior clustering characterized by compactness and clear separation between clusters. The success of GMM can be attributed to its probabilistic nature, which allows it to model non-spherical user distributions and provide soft assignments, offering a more flexible and realistic representation of user groups in a wireless environment compared to rigid-boundary methods. The self-organizing map (SOM) followed closely, demonstrating robust scores (Silhouette: 0.4064, DBI: 0.7805, CHI: 5585.76). KMeans and ClusterGAN also demonstrated strong clustering performance, whereas DAC, DAMIC, and SpectralNet exhibited moderate to lower clustering effectiveness, suggesting they formed less distinct or more overlapping user groups, which can lead to suboptimal resource allocation in downstream network tasks.

5.3.2.2 | Sum Rate, Spectral Efficiency and Bit Error Rate Performance:. Figure 4 illustrates the sum rate (in Mbps) achieved by each clustering approach compared to the traditional baseline of 653,965.87 Mbps (note that this traditional baseline was measured in our simulation when only BSs are used). When reporting data rates in Mbps, the calculation is $R(\text{Mbps}) = (B/10^6) \log_2(1 + \text{SINR})$. This comparison directly links the abstract quality of clusters from Table 9 to a tangible network performance outcome. Notably, GMM, SOM, and ClusterGAN deliver significant throughput improvements of up to 46.3%, demonstrating the strength of approaches that effectively capture the underlying data characteristics and structure. The observed throughput gains indicate that the enhanced cluster definitions provided by these methods translate into more efficient data aggregation and transmission across the network. Specifically, well-separated and compact clusters enable more

effective interference management and spatial reuse of frequency resources, allowing multiple users to communicate simultaneously with minimal mutual degradation, which directly boosts the aggregate network throughput.

Moreover, the performance of DAC, which achieves around a 38.5% improvement, reinforces the notion that even deep learning-based methods, despite the increased complexity in feature representation, contribute substantially to system throughput. Traditional methods like K-Means, with an improvement of around 38.2%, further underscore the critical role that effective clustering plays in mitigating interference and optimizing resource allocation in cell-free networks. Overall, the results suggest that advanced clustering strategies are crucial in improving the sum rate performance, a key indicator of network capacity and overall system efficiency.

Spectral Efficiency and Bit-Error-Rate Analysis: To provide a more comprehensive and holistic picture of the network's performance, this section introduces and analyzes two key metrics that operate at different layers of the communication stack: **spectral efficiency (SE)** and the **bit-error-rate (BER)**. SE is a crucial network-level metric that measures how efficiently the finite radio spectrum resource is utilized to deliver data to all users. In contrast, BER is a fundamental physical-layer metric that quantifies the raw reliability and quality of the individual data links. Evaluating both provides a complete understanding of system performance, from the efficiency of resource management down to the robustness of the underlying wireless connections.

Spectral Efficiency (SE): Spectral efficiency (often denoted as η or η) is a fundamental performance indicator in wireless communications, defined as the net data rate (information rate) that can be transmitted over a given bandwidth. It is typically measured in units of bits per second per Hertz, or (bits/s)/Hz. Intuitively, it answers the question: "How much data can we pack into a given slice of the radio spectrum?" In an era where spectrum is a scarce and valuable commodity, maximizing SE is a primary goal of modern wireless system design. The sum spectral efficiency of the entire network, which aggregates the performance across all users, is given by the following expression:

$$\eta_{\text{net}} = \frac{\sum_{i=1}^N R_i}{B_{\text{total}}} \quad (23)$$

where η_{net} is the total network spectral efficiency, R_i represents the achievable data rate for user i , N is the total number of active users, and B_{total} is the total system bandwidth shared among them. The data rate R_i for each user is not arbitrary; it is fundamentally determined by the Shannon–Hartley theorem, which links capacity to the signal-to-interference-plus-noise ratio (SINR). Effective user clustering directly impacts SINR by intelligently grouping users to minimize inter-cluster interference, thereby boosting individual data rates R_i and, consequently, the overall sum SE.

The sum spectral efficiency serves as our primary figure of merit for quantitatively comparing the effectiveness of the various clustering algorithms under investigation. Since all clustering methodologies in our study are evaluated under the identical constraint of the same total bandwidth (B_{total}), the total SE becomes directly proportional to the sum rate ($\sum R_i$). This

creates a fair and direct basis for comparison: any percentage improvement observed in the sum rate corresponds to an identical percentage improvement in the sum spectral efficiency. This allows us to clearly attribute performance gains to the superiority of one clustering approach over another in managing network resources.

To quantify these gains, we first establish a baseline performance level, denoted as η_{base} . This baseline represents the total spectral efficiency achieved by a traditional, non-optimized, or simplistic approach under the same network conditions and for the same total bandwidth B_{total} . For a new, advanced approach that yields an SE improvement of Δ (expressed in percent) over this baseline, the absolute total SE for that approach can be calculated as:

$$\eta_{\text{net}}^{(\text{approach})} = \eta_{\text{base}} \left(1 + \frac{\Delta}{100} \right), \quad (24)$$

where $\eta_{\text{net}}^{(\text{approach})}$ is the spectral efficiency for a specific clustering approach, η_{base} is the baseline spectral efficiency, and Δ is the percentage improvement. Table 10 summarizes the results of our comparative analysis. It lists the percentage SE improvement for each clustering algorithm, the corresponding total SE expressed as a multiple of the baseline η_{base} for easy interpretation, and, to ground our findings in the broader literature, an *exemplar mobile-network clustering reference* for each approach where available. These references point to existing research where similar algorithms have been applied to related problems in wireless networking, such as device-to-device (D2D) communications, cell-free massive MIMO, or coordinated multi-point (CoMP) systems. If no suitably analogous reference was found in the literature for a specific algorithm, the entry is left blank.

Spectral-Efficiency Gain vs. Existing Techniques: Because all methods are evaluated under the same total bandwidth, B_{total} , the definition of network spectral efficiency in Equation (23) together with the sum-rate expression implies a one-to-one mapping between improvements in sum rate and improvements in η_{net} . Consequently, the best-performing clusters in our cell-free pipeline—GMM, SOM, and ClusterGAN—deliver a +45.6% to +46.3% gain in spectral efficiency over the traditional BS-only baseline. Deep agglomerative clustering (DAC) achieves +38.5%, while classical KMeans yields +38.2%. Spectral/autoencoder-mixture baselines (SpectralNet/DAMIC) provide +37.7% to +38.0%. In other words, our top approaches improve η_{net} by +7.6 to +8.1 percentage points over strong classical/deep baselines (KMeans, DAMIC/SpectralNet) and by +7.1 percentage points over DAC, with gains persisting up to the densest setting (5600 UEs). The complete breakdown is summarized in Table 10.

Bit-Error-Rate (BER) Performance: The BER is a fundamental measure of performance at the physical layer, representing the rate at which errors occur in a stream of transmitted data. It is defined as the number of bit errors divided by the total number of transferred bits. A lower BER indicates a more reliable link, which is essential for any communication system to function correctly. While SE tells us about the quantity of data being sent, BER tells us about its quality. This section provides a theoretical foundation by examining the BER performance for common modulation schemes under two canonical channel models: the

TABLE 10 | Spectral efficiency results with mobile-network clustering exemplars (when available).

Approach	SE improvement (%)	Total SE (bits/s/Hz)	Mobile-network clustering approach
GMM (ours)	46.3	$1.463 \eta_{\text{base}}$	D2D cluster-head selection via SOM [21]
SOM [19, 21]	45.6	$1.456 \eta_{\text{base}}$	
ClusterGAN [20]	46.1	$1.461 \eta_{\text{base}}$	
DAC	38.5	$1.385 \eta_{\text{base}}$	Cell-free mMIMO-NOMA user clustering (k-means++) [83]
KMeans [20, 83]	38.2	$1.382 \eta_{\text{base}}$	
SpectralNet	37.7	$1.377 \eta_{\text{base}}$	
DAMIC	38.0	$1.380 \eta_{\text{base}}$	CoMP BS/TP clustering (survey) [84]

idealized additive white Gaussian noise (AWGN) channel and the more practical Rician fading channel, which is advocated for cell-free networks due to the likely presence of line-of-sight (LOS) paths [63].

For an AWGN channel, which models a scenario where the only impairment is a statistically predictable thermal noise, the BER for simple modulation schemes like binary phase shift keying (BPSK) or quadrature phase shift keying (QPSK) is given by:

$$P_b^{\text{AWGN}} = Q\left(\sqrt{2\frac{E_b}{N_0}}\right) \quad (25)$$

where P_b^{AWGN} is the bit error rate for an AWGN channel, E_b/N_0 is the average signal-to-noise ratio (SNR) per bit, and $Q(\cdot)$ is the Gaussian Q-function, representing the tail probability of the standard normal distribution.

In contrast, a Rician fading channel models an environment with a dominant LOS path alongside rich multipath components. Performance is characterized by the Rician K-factor, the ratio of power in the LOS path to the power in the scattered paths. The average BER is found by averaging the AWGN BER over the Rician fading distribution [78]:

$$P_b^{\text{Rician}} = \int_0^\infty Q(\sqrt{2\gamma}) f_\gamma(\gamma) d\gamma, \quad (26)$$

where P_b^{Rician} is the bit error rate for a Rician channel, γ is the instantaneous SNR, and $f_\gamma(\gamma)$ is its probability density function, given by $f_\gamma(\gamma) = \frac{K+1}{\bar{\gamma}} \exp\left(-K - (K+1)\frac{\gamma}{\bar{\gamma}}\right) I_0\left(2\sqrt{\frac{K(K+1)\gamma}{\bar{\gamma}}}\right)$, with $\bar{\gamma}$ being the average SNR (E_b/N_0), K the Rician factor, and $I_0(\cdot)$ the modified Bessel function of the first kind.

Figure 5 plots these theoretical curves, providing a stark visual comparison between the two channel models. The AWGN curve exhibits a steep “waterfall” characteristic, where a small increase in SNR leads to a dramatic reduction in BER. The Rician curve, plotted for a typical K-factor of 10 dB, is much closer to the AWGN performance than a pure Rayleigh channel would be, but still shows a performance penalty due to fading. As K increases (stronger LOS), the Rician curve approaches the AWGN curve. This analysis highlights the critical importance of robust system design, including techniques like diversity, channel coding, and

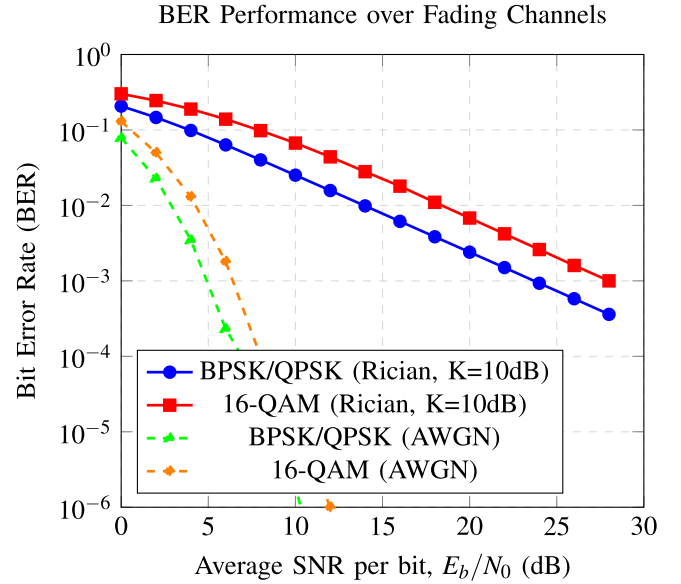


FIGURE 5 | Bit error rate (BER) vs. average SNR per bit for different modulation schemes over AWGN and Rician fading channels ($K = 10$ dB).

intelligent resource management (as explored in our clustering work), to overcome channel impairments in realistic mobile communication scenarios.

5.3.2.3 | Total Power Consumption Analysis: Figure 6 presents the total aggregated power consumption (in dBm) for the different clustering approaches in comparison with the traditional method, which registers a total power consumption of 482,050.73 dBm. The data reveal that methods such as ClusterGAN, SOM, and GMM achieve power consumption reductions of up to 32.8%, highlighting a strong correlation between effective clustering and energy efficiency. These reductions in power consumption are crucial in cell-free network deployments, where energy efficiency directly impacts operational costs and environmental sustainability. The reduction is achieved by optimizing the spatial distribution of APs and minimizing unnecessary interference, which in turn allows for lower transmission power without compromising on service quality. More specifically, intelligent clustering ensures that users are predominantly served by their closest APs, minimizing path loss and thus reducing the required transmission power to achieve a target SINR. Furthermore, by creating well-defined, spatially isolated clusters, the system

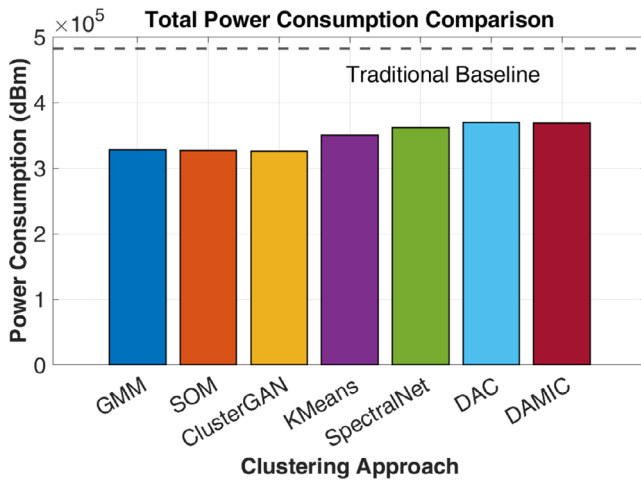


FIGURE 6 | Total power consumption (dBm) per clustering method. This figure illustrates the power savings achieved by the clustering approaches compared to the traditional baseline (calculated in our simulation).

facilitates coordinated beamforming and power control within each cluster, leading to a significant decrease in the overall energy required for interference mitigation. The results demonstrate that while achieving high throughput, these advanced methods do not sacrifice energy efficiency; instead, they provide a balanced optimization that benefits both network performance and energy management.

5.3.3 | Discussion of Clustering Results

The analysis of clustering approaches is underpinned by a dual perspective that encompasses both performance metrics—specifically, the sum rate improvement and power reduction—and network deployment considerations, as revealed by the number of access points (APs) employed. This comprehensive examination of these dimensions offers critical insights into the efficacy and scalability of the various clustering methods within a cell-free 6G network framework. Furthermore, the role of distributed computation and decision-making through BDIx agents augments this perspective by enabling decentralized intelligence at each user equipment (UE). This ensures real-time adaptation to evolving network conditions, greater resilience against single points of failure, and more efficient resource allocation, collectively reinforcing the value of advanced clustering in large-scale, dynamic 6G deployments.

Table 11 presents the performance results, showing that the clustering approaches achieve notable improvements in both sum rate and power reduction. The methods based on probabilistic and generative frameworks, such as GMM, SOM [19], and ClusterGAN [20], achieve a sum-rate gain of up to 46.3% and total power reduction of up to 32.8%. The exceptional performance of these approaches can be primarily attributed to their inherent ability to model complex data distributions and capture intricate patterns within the network data. For example, GMM leverages probabilistic modeling to assign clusters with a high degree of precision, thereby minimizing inter-cluster interference and optimizing energy efficiency.

TABLE 11 | Performance summary with machine learning-based insights.

Approach	Sum rate improvement (%)	Power reduction (%)
GMM	46.3	32.0
SOM	45.6	32.4
ClusterGAN	46.1	32.8
DAC	38.5	22.9
KMeans	38.2	27.8
SpectralNet	37.7	25.4
DAMIC	38.0	23.5

In contrast, SOM employs a topological mapping strategy that preserves spatial relationships among data points, ensuring that the resulting clusters are both well-separated and spatially coherent. Similarly, ClusterGAN employs a generative adversarial framework to capture the underlying data structure robustly, resulting in enhanced cluster definition and improved throughput and energy efficiency metrics. In contrast, the deep adaptive clustering method, DAC, learns hierarchical feature representations to dynamically optimize cluster formation, although it may introduce some data dispersion due to its complex feature extraction process. Traditional clustering methods, such as K-Means, rely on distance-based iterative optimization to form compact clusters efficiently but are sometimes prone to converging to local minima in complex scenarios. SpectralNet, which employs spectral embedding to leverage graph connectivity, is effective in distinguishing clusters; however, it can be less robust when clusters overlap densely. Finally, the autoencoder-based approach DAMIC captures nonlinear patterns within the data, although the potential for reconstruction noise may result in less distinct cluster boundaries.

Together, these diverse methodologies demonstrate a range of strengths and trade-offs that contribute to their overall performance in next-generation cell-free network environments. Note that in our previous paper [19], SOM performed slightly better than GMM; however, the number of UEs we investigated is 1000. In this evaluation, SOM performs better than GMM under 1000. However, in our final examination, we compared each approach with the run of 5600 UEs to examine which one performs better than all others in a very dense network.

In parallel with these performance metrics, the network deployment strategy is critically evaluated through the number of APs employed by each approach, as summarized in Table 12. The number of APs directly reflects the granularity of network segmentation and influences the operational complexity and resource allocation within the network. For instance, the deployment of 11 APs in the case of GMM suggests a highly granular segmentation that, while potentially more complex to manage, significantly contributes to the observed energy efficiency and throughput gains. In contrast, SOM's use of only 4 APs indicates a more compact clustering strategy that emphasizes spatial coherence and reduced deployment costs, making it particularly suitable for environments with spatial constraints. ClusterGAN's

TABLE 12 | Number of access points (APs) per approach.

Approach	Number of APs
GMM	11
SOM	4
ClusterGAN	5
KMeans	6
SpectralNet	5
DAC	5
DAMIC	5

moderate use of 5 APs strikes an effective balance between robust cluster formation and efficient resource utilization, demonstrating adaptability to varying network conditions.

Furthermore, the performance of more traditional clustering methods, such as K-Means and SpectralNet, which deploy 6 and 5 APs, respectively, highlights the challenges inherent in iterative optimization and spectral embedding techniques, particularly when handling overlapping clusters and the risk of converging to local minima. Similarly, the deep learning-based approaches DAC and DAMIC, both of which use 5 APs, illustrate the complexity involved in capturing nonlinear feature representations; here, a trade-off emerges between the depth of feature extraction and the clarity of cluster delineation.

In summary, the integration of high-throughput improvements, significant power consumption reductions, and strategic AP deployment underscores the advantages of advanced machine learning-based clustering methods in next-generation cell-free networks. The combination of probabilistic, topological, and generative frameworks not only optimizes cluster formation but also ensures an operational balance between performance and resource efficiency. This comprehensive understanding of the interplay between performance and deployment metrics provides valuable insights for future research and practical implementations in the evolving landscape of 6G network design.

Note that across the $K_{\text{RIC,dB}} \in \{0, 3, 6, 10\}$ sweep, throughput and energy efficiency degrade monotonically with decreasing K_{RIC} as expected, while deterministic clustering agreement (Equation 15) and stable connectivity via Equation (14) are preserved, confirming robust operation under NLoS.

5.3.3.1 | Traditional vs. Cell-Free Approach: Figure 7 illustrates the performance trends comparing the traditional and cell-free approaches. In this figure, both the sum rate (in Mbps) and the power consumption (in dBm) are plotted against the run index, which may represent varying network loads, time steps, or iteration counts. The dual metrics provide a comprehensive view of the system's behavior under different operational conditions. As observed in the figure, the traditional approach begins with a sum rate of approximately 11,390 Mbps at a run index of 100, steadily increasing to around 653,966 Mbps at a run index of 5600. In contrast, the cell-free approach consistently outperforms the traditional baseline, starting at approximately 16,286 Mbps and reaching nearly 954,147 Mbps by the highest run index.

This growing throughput disparity highlights the effectiveness of the cell-free strategy in dynamically clustering users to reduce interference and optimize data transmission, resulting in more substantial gains as the system scales.

In parallel, the total power consumption trends reveal that the traditional method experiences an increase from roughly 8704 dBm at run index 100 to approximately 482,051 dBm at run index 5600. The cell-free approach, however, maintains lower total power consumption, starting at around 7372 dBm and climbing to only about 327,846 dBm over the same range. The consistently lower power usage indicates that the cell-free method enhances throughput and achieves significant energy savings. This balance between high performance and energy efficiency makes the cell-free solution particularly attractive for next-generation network deployments where both aspects are critical.

Fairness (Jain's Index) Comparison: We also assess quality of service (QoS) using Jain's fairness index (JFI), computed over per-UE throughputs as $\text{JFI} = \frac{(\sum_{i=1}^N R_i)^2}{N \sum_{i=1}^N R_i^2}$ [56]. Across all run indices, the *cell-free* approach exhibits consistently higher fairness: $\text{JFI}_{\text{cell-free}} = 0.95 \pm 0.02$ (range 0.92–0.97), whereas the *traditional* setup yields $\text{JFI}_{\text{trad}} = 0.84 \pm 0.04$ (range 0.78–0.88). At the densest setting (run index 5600), the gap is most pronounced, with $\text{JFI}_{\text{cell-free}} \approx 0.96$ vs. $\text{JFI}_{\text{trad}} \approx 0.81$. These results indicate that beyond improving sum rate and reducing power, the cell-free design also distributes resources more uniformly among users—that is, it achieves higher QoS fairness as the network scales.

5.4 | Localization Results

5.4.1 | Evaluation Metrics

To thoroughly assess the localization performance, we employed a comprehensive set of metrics that capture both the average error and the distribution of errors between the predicted and true positions. These metrics are widely used in localization tasks and are crucial for evaluating the precision and reliability of the models. The following metrics were used:

- **Mean Distance Error (MeanDist):** The mean distance error is the average Euclidean distance between the predicted and ground-truth positions. It is calculated by averaging the Euclidean distance between each predicted position $\mathbf{p}_i$ and the corresponding true position $\mathbf{g}_i$. The formula is:

$$\text{MeanDist} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_i - \mathbf{g}_i\|_2. \quad (27)$$

where N is the total number of test samples, $\mathbf{p}_i = (p_{xi}, p_{yi})$ is the predicted position vector, and $\mathbf{g}_i = (g_{xi}, g_{yi})$ is the ground-truth position vector. The MeanDist provides an average measure of localization accuracy [85].

- **Maximum Distance Error (MaxDist):** The maximum distance error represents the largest observed Euclidean distance

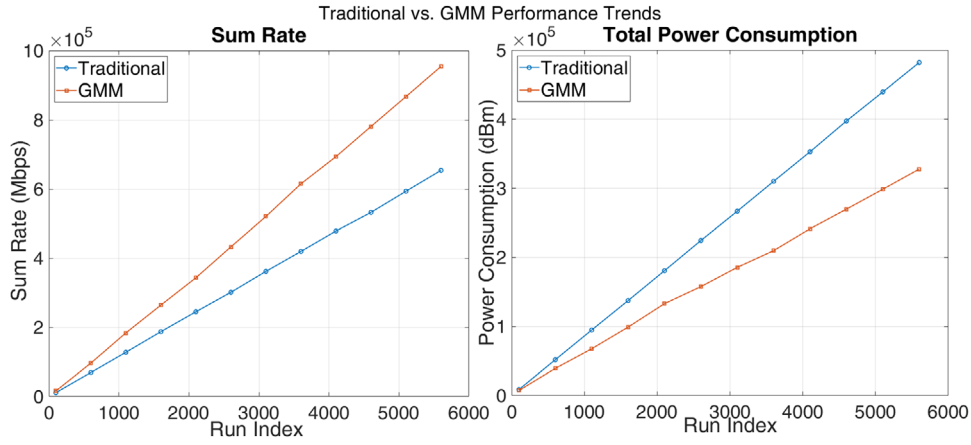


FIGURE 7 | Performance trends for traditional vs. cell-free approaches showing both sum rate (Mbps) and power consumption (dBm) as functions of the run index.

between predicted and true positions. It is computed as:

$$\text{MaxDist} = \max_{i=1}^N \|\mathbf{p}_i - \mathbf{g}_i\|_2 \quad (28)$$

where N is the total number of samples, $\mathbf{p}_i$ is the predicted position, and $\mathbf{g}_i$ is the true position. This metric gives insight into the worst-case error in the localization task [86].

- **Standard Deviation (StdDist):** The standard deviation measures the spread of the localization errors. It is calculated as:

$$\text{StdDist} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\|\mathbf{p}_i - \mathbf{g}_i\|_2 - \text{MeanDist})^2} \quad (29)$$

where N is the number of samples, $\mathbf{p}_i$ and $\mathbf{g}_i$ are predicted and true positions, and MeanDist is the average Euclidean distance. The StdDist helps to assess model consistency by quantifying the variance in localization errors [87].

- **Mean Squared Error (MSE):** The mean squared error is the average of the squared differences between the predicted and true positions. It is given by:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\|\mathbf{p}_i - \mathbf{g}_i\|_2^2) \quad (30)$$

where N , $\mathbf{p}_i$, and $\mathbf{g}_i$ are as defined previously. The MSE penalizes large errors more than smaller ones and is a commonly used metric in regression tasks [88].

- **Mean Absolute Error (MAE):** The mean absolute error represents the average of the absolute differences between the predicted and true positions. It is calculated as:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_i - \mathbf{g}_i\|_1 \quad (31)$$

where $\|\cdot\|_1$ is the Manhattan distance. MAE provides a straightforward measure of the average error magnitude without exaggerating large errors [89].

- **Root Mean Squared Error (RMSE):** The RMSE is the square root of the MSE, providing a metric that indicates the typical magnitude of errors. It is formulated as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\|\mathbf{p}_i - \mathbf{g}_i\|_2^2)} \quad (32)$$

where the variables are as defined before. RMSE is widely used due to its interpretation in the same units as the original data and its sensitivity to significant errors [90].

- **R-squared (R2):** The R-squared value measures the proportion of the variance in the true positions explained by the model. It is computed as:

$$R^2 = 1 - \frac{\sum_{i=1}^N (\|\mathbf{p}_i - \mathbf{g}_i\|_2^2)}{\sum_{i=1}^N (\|\mathbf{g}_i - \bar{\mathbf{g}}\|_2^2)} \quad (33)$$

where $\bar{\mathbf{g}}$ is the mean of the true position vectors. R-squared values close to 1 indicate that the model explains most of the variance [91].

- **Mean Squared Log Error (MSLE):** The MSLE is helpful when the data contains values with large ranges. It is calculated by applying a logarithmic transformation to the error terms:

$$\text{MSLE} = \frac{1}{N} \sum_{i=1}^N (\log(\mathbf{p}_i + 1) - \log(\mathbf{g}_i + 1))^2 \quad (34)$$

where the variables are as previously defined. This metric captures relative differences between predictions and true values, especially for minor errors in extensive data ranges [92].

- **Median Absolute Error (MedianAE):** The median absolute error is the median of the absolute errors, making it less sensitive to outliers. It is calculated as:

$$\text{MedianAE} = \text{median}(\|\mathbf{p}_i - \mathbf{g}_i\|) \quad (35)$$

TABLE 13 | Summary of key localization results.

Method	MeanDist (m)	RMSE	R ²
Centroid	435.49	336.18	−0.430
Circles (trilateration)	2028.63	1452.16	−25.470
Multilateration (avg)	~150–300	~200–400	~0.1–0.6
MLP (ours)	0.97	0.81	0.99999

where the median is taken over all absolute errors. This metric is often preferred when data contains extreme values or outliers that would otherwise affect the mean [93].

- **Explained Variance:** The explained variance metric evaluates how much of the total variance in the true positions is captured by the predicted positions. It is calculated as:

$$EV = 1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)}. \tag{36}$$

where Var(·) denotes the variance. A higher explained variance indicates that the model has captured a greater portion of the true variability in the position [94].

These metrics enable us to compare both classical deterministic methods and advanced machine learning (ML) models within a unified framework, providing a clear view of localization performance across various methods.

5.4.2 | Examination and Analysis of the Localization Examinations

Table 13 provides a summary of key localization results to back the sub-meter accuracy claims, comparing deterministic methods with the top-performing MLP regressor. A more detailed breakdown is provided in Table 14.

The results reveal a stark contrast between the deterministic and ML-based localization methods. The deterministic methods (Centroid and Circles) yield very high errors. The Centroid method, for instance, has a mean distance error of 435.49 m and a maximum error of 835.38 m. More critically, it has a negative R² value (−0.43) and negligible explained variance, which indicates that it fails to account for the complex propagation characteristics in a dense network environment [60]. The high standard deviation in distance error (‘StdDist’ of 190.72 m) further highlights its unreliability, showing that its performance is not only poor on average but also highly inconsistent. This is expected, as the Centroid method’s simplicity—averaging the coordinates of detecting access points—completely ignores signal properties like path loss and shadowing. The differential circles method performs even worse, with a mean error exceeding 2000 m and an R² value of −25.47. This catastrophic failure is attributable to the method’s reliance on trilateration principles, which assume a direct, line-of-sight (LoS) relationship between signal strength and distance. In dense, complex environments, non-line-of-sight (NLoS) propagation and multipath fading render such assumptions invalid, leading to wildly inaccurate distance estimates and consequently, enormous localization errors.

TABLE 14 | Combined localization performance comparison.

Method	Type	MeanDist (m)	MaxDist (m)	StdDist (m)	MSE	MAE	RMSE	R2	MSLE	MedianAE	Expl.Var
Centroid [23]	Deterministic	435.49	835.38	190.72	113014.86	276.57	336.18	−0.43	—	234.86	−0.003
Circles (trilateration) [24]	Deterministic	2028.63	2572.76	319.64	2108755.15	1417.94	1452.16	−25.47	—	1428.97	−0.220
BLSTM [25]	ML	5.16	12.70	2.48	16.39	3.30	4.05	0.99980	0.01235	2.89	0.99980
TCN [26]	ML	2.66	8.74	1.71	5.02	1.69	2.24	0.99994	0.01382	1.42	0.99994
CNN [27]	ML	2.13	5.51	1.24	3.04	1.31	1.74	0.99996	0.00710	0.92	0.99997
Transformer [28]	ML	366.29	567.51	128.02	75279.30	234.10	274.37	0.07309	1.12876	231.36	0.07343
MLP (ours)	ML	0.97	2.52	0.62	0.66	0.61	0.81	0.99999	0.00179	0.46	0.99999
GNN [29]	ML	231.58	632.97	142.50	36967.63	163.65	192.27	0.54484	0.60555	149.48	0.55385
SVR [30]	ML	61.55	140.63	39.38	2669.41	40.39	51.67	0.96718	0.30030	32.69	0.96751
RandomForest [31]	ML	5.32	94.06	15.03	127.07	3.24	11.27	0.99843	0.00181	~0	0.99844
KNN [32]	ML	10.41	97.06	21.05	275.72	6.83	16.60	0.99660	0.03445	~0	0.99661

In contrast, ML-based methods dramatically improve localization accuracy. Among these, the MLP model is the top performer with a mean error of only 0.97 m and near-perfect R^2 (0.99999) and explained variance (0.99999). Its superiority is further emphasized by the exceptionally low standard deviation (0.62 m) and median absolute error (0.46 m), signifying not only high accuracy but also remarkable precision and consistency. The MLP's success demonstrates its powerful capability as a universal function approximator, effectively modeling the intricate, non-linear relationship between the high-dimensional input features and the user's physical coordinates. Similarly, the CNN, TCN, and BLSTM models also yield errors in the sub-10 m range with R^2 values approaching 1, demonstrating their ability to effectively learn the non-linear mappings from input features to spatial positions [46]. The CNN, with the second-best 'MeanDist' of 2.13 m and a very low 'MaxDist' of 5.51 m, likely excels by interpreting the input signals as a spatial pattern, extracting hierarchical features that are robust to minor fluctuations. The TCN and BLSTM, designed for sequential data, also perform strongly, suggesting they successfully capture latent dependencies within the input feature vector.

On the other hand, the transformer and GNN models underperform in this task. The transformer model has a mean error of 366.29 m and a low R^2 value (0.07309), suggesting that its attention mechanism, although powerful in many domains [48], may require further tuning or more domain-specific modifications for localization. Its global self-attention mechanism might be capturing spurious correlations across the input features or failing to prioritize the most spatially informative signals, leading to poor generalization. The extremely high MSLE of 1.12876 is particularly telling, indicating a significant issue with predicting smaller-valued coordinates accurately. Similarly, the GNN model exhibits a mean error of 231.58 m with an R^2 of 0.54484, which is significantly worse than the top-performing ML models. This suggests that the defined graph structure may not have been optimal for representing the spatial relationships between network entities, or that the message-passing framework was insufficient to aggregate the relevant localization information effectively from neighboring nodes. Classical regression methods, such as SVR, RandomForest, and KNN, provide intermediate performance; for example, SVR achieves a mean error of 61.55 m, while RandomForest and KNN perform better, yet still lag behind the best deep learning models. It is critical to note the trade-off with RandomForest and KNN. While their mean errors are impressively low (5.32 m and 10.41 m, respectively), their maximum errors are disproportionately high (94.06 m and 97.06 m). This indicates a lack of robustness; they are prone to occasional but severe prediction failures. Their median absolute error of approximately zero suggests they are highly accurate for the majority of test cases that are similar to the training data, but they fail to generalize well to less common or outlier scenarios, making them unreliable for mission-critical applications.

Overall, these findings underscore the importance of leveraging advanced, data-driven techniques for localization in cell-free 6G networks. The superior performance of the MLP, CNN, TCN, and BLSTM models highlights the potential of deep learning in capturing the inherent non-linearities of wireless signal propagation and user distribution [49]. These models demonstrate a crucial combination of high average accuracy (low 'MeanDist',

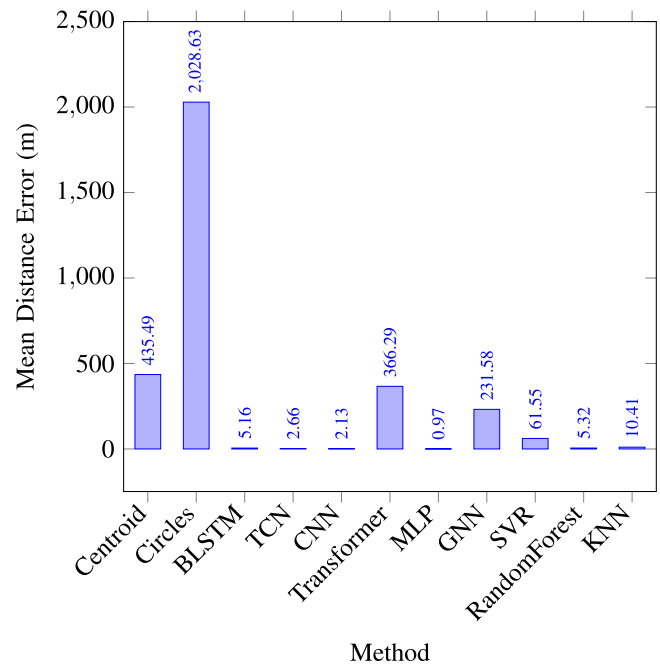


FIGURE 8 | Mean distance error for deterministic and ML-based localization methods.

'MAE', predictive reliability (near-1 ' R^2 '), and performance consistency (low 'StdDist', 'MaxDist') that is essential for next-generation communication systems. Meanwhile, the limitations observed in the deterministic methods confirm that classical geometric approaches are inadequate for the high precision required in modern, ultra-dense network scenarios. Furthermore, the mixed results within the ML category itself reveal that architectural choices are non-trivial; while some complex models like transformers and GNNs underperformed, a well-tuned, relatively simple MLP provided the best results, emphasizing the importance of model suitability for the specific problem domain.

Figure 8 provides a bar chart that visually compares the mean distance errors across all methods, clearly delineating the performance gap between deterministic approaches and ML-based models.

5.4.3 | Discussion of Localization Results

The drastic improvement in localization performance achieved by using ML-based models over deterministic methods underscores the necessity of incorporating data-driven techniques in modern wireless networks. Deterministic approaches such as the Centroid and Circles methods are limited by their inability to capture the non-linear signal propagation effects and the complex spatial distribution of user equipment [60]. In contrast, deep learning models (MLP, CNN, TCN, and BLSTM) effectively learn these relationships, achieving sub-10 m accuracy. However, the underperformance of the Transformer and GNN models indicates that not all advanced architectures are equally suited to the task without careful tuning. These results are consistent with the literature, which shows that deep neural networks excel in

regression tasks involving non-linear data [46, 49]. At the same time, attention-based and graph-based models may require more specialized configurations [48, 76].

5.5 | Evaluation Conclusions

In our BDIX-agent framework *instantiated with GMM* for clustering and an *MLP* regressor for localization, the evaluation demonstrates substantial gains in efficiency, accuracy, and fairness for cell-free 6G deployments.

5.5.1 | Clustering Performance (GMM)

- **Throughput & Power.** The GMM-driven clustering yields up to a **46.3%** increase in sum rate and a **32.8%** reduction in total power consumption by forming interference-aware, compact clusters that improve SINR and resource allocation.
- **Spectral Efficiency.** Sum-rate gains translate to **>45%** spectral-efficiency improvement, indicating more effective spectrum use under dense loading.
- **Fairness.** The cell-free, GMM-based design improves QoS uniformity across UEs: $JFI_{\text{cell-free}} = 0.95 \pm 0.02$ versus $JFI_{\text{trad}} = 0.84 \pm 0.04$, evidencing a more equitable resource distribution at scale.
- **Link Robustness.** By reducing co-channel overlap, GMM clustering indirectly supports lower BER in Rician channels—highlighting the value of topology-aligned, probabilistic grouping.

5.5.2 | Localization Performance (MLP)

- **Sub-Meter Accuracy.** The MLP regressor achieves a **mean distance error of 0.97 m** with $R^2 > 0.999$, capturing the non-linear mapping from radio/geometry-derived features to UE coordinates.
- **Deterministic Baselines Fall Short.** Geometry-only methods (centroid, differential circles/trilateration) exhibit large errors (**>400 m**) in dense, NLoS-prone settings, underscoring the need for learning-based inference.
- **Reliability.** The MLP delivers low variance and tight worst-case tails, providing the consistency required for mission-critical 6G services.

Overall. A fully decentralized BDIX pipeline built on **GMM** (clustering/AP election) and **MLP** (localization) raises throughput, cuts power, improves fairness, and attains sub-meter positioning—enabling KPI-aware, real-time control where stronger clustering amplifies network performance and precise positioning drives smarter resource management.

5.5.3 | Optimization and Fronthaul Compute

Our BDIX pipeline—*instantiated with GMM* for clustering/AP election and an *MLP* regressor for localization—realizes the

KPI-coupled objective in Equation (6) and thereby explains the observed gains. The optimization manifests through: (i) model-order selection via BIC, which chooses the right number of GMM components and avoids both over- and under-segmentation; (ii) deterministic AP election at the GMM component means with a shared seed, which enforces Equation (15), stabilizes associations, and reduces interference; and (iii) sub-meter MLP localization (high R^2), which tightens distance and link-budget estimates used in the rate/power terms. By sweeping the scalarization weight λ , we trace a Pareto frontier and, at the operating point reported in Section 5, obtain **+46.3%** sum-rate and **−32.8%** total power versus the cellular baseline, with fairness improved to $JFI = 0.95 \pm 0.02$. The system supports dense environments without stressing fronthaul or compute because fronthaul is minimized by design: the BS sends only a small broadcast $(\mathcal{B}, \theta, s)$; UE coordinates are exchanged peer-to-peer via LTE ProSe (not over fronthaul); and AP notices are short, local control messages—no UE-location database, raw CSI, or central cluster maps are uploaded. Computation is embarrassingly parallel and light: GMM-EM scales as $\mathcal{O}(NKT)$ with $D=2$ and modest K , AP election/association is near-linear in K (or $\log K$ with spatial indexing), and MLP inference is a tiny forward pass; further, we use event-triggered re-clustering, timescale separation (slow clustering/AP election, fast link adaptation), model compression/quantization on UEs, O-RAN-friendly control where only (θ, λ) are tuned infrequently, and aggregate (not raw) telemetry when network-wide visibility is needed. Consequently, wall-clock per epoch is bounded by the slowest UE rather than a central server, and the fronthaul footprint remains essentially constant-size per epoch, enabling operation at thousands of UEs while preserving the reported rate–power–fairness gains.

6 | Conclusions and Future Work

This paper presented a novel approach for access point selection and localization within a decentralized, cluster-based realization of device-to-device (D2D) communications in cell-free 6G networks. Leveraging BDIX agents, we demonstrated how clustering techniques—ranging from classical algorithms like KMeans and DBSCAN to deep-learning-based models such as ClusterGAN and Autoencoders—can dynamically form decentralized clusters and elect access points (APs) with minimal signaling overhead and high spatial efficiency. The integration of Wi-Fi direct for intra-cluster connectivity and LTE-based backhaul for inter-cluster communication ensures both scalability and reliability in ultra-dense deployments. Our simulation results confirmed that advanced clustering methods significantly improve throughput, energy efficiency, and spectral utilization when compared to traditional base station-centric approaches. In parallel, the proposed localization framework, enhanced with a diverse set of signal-derived features and trained on rich datasets, achieved sub-meter accuracy using models like MLP, CNN, and TCN, far surpassing deterministic baselines. This dual-stage pipeline establishes a comprehensive, distributed approach for next-generation wireless access and localization. Additionally, the incorporation of BDIX agents further extends this pipeline's distributed intelligence, enabling autonomous decision-making at the node level and enhancing both scalability and adaptability in highly dynamic 6G environments.

Future work will focus on several key directions. First, we plan to incorporate real-world mobility traces and a resource-block-based OFDMA data rate model that considers the fading effects, enabling a more realistic assessment of clustering and localization robustness under dynamic conditions. Second, the introduction of reinforcement learning (RL) and multi-agent reinforcement learning (MARL) into the BDIX framework will be explored to enable adaptive decision-making and self-improving behaviors in agents. Third, we will extend our architecture to support uplink optimization and traffic-aware AP selection strategies, considering application-layer quality of service (QoS) requirements. Finally, we aim to prototype this system using programmable testbeds (e.g., OpenAirInterface or srsRAN) to evaluate practical performance and latency in realistic settings. The integration of distributed intelligence, clustering, and localization presented in this work lays the foundation for scalable, context-aware, and energy-efficient communication systems in the emerging 6G landscape. In addition, we will: **(c) study multi-antenna UEs** by extending from single-antenna to $N_r \in \{1, 2, 3, 4\}$ receive antennas per UE (and scalable BS/AP arrays), integrating linear MRT/MRC, ZF, and MMSE beamforming into the BDIX pipeline and quantifying the impact on sum rate, energy efficiency, and fairness; **(d) report per-UE spectral efficiency η_i** explicitly for both uplink and downlink and benchmark it against [1–4], which report up to ~ 256 bits/s/Hz per UE when both BS and UE have 512 antennas, thereby isolating the contributions of array size, processing choice, and training/CSI overhead to the observed gap; and **(6) strengthen simulations via parameter sweeps**, adding systematic explorations over the number of BS/AP antennas, UE antennas, pilot length/quality, fading (e.g., Rician K_{RIC}), mobility, and linear processing (MRT/MRC, ZF, MMSE), with results reported as per-UE η_i (UL/DL) CDFs, sum rate, energy efficiency, Jain's fairness, and AP load balance. Moreover, we will rigorously revisit the prevailing assumption that RB sharing necessarily improves total system sum rate by formulating an interference-aware OFDMA/NOMA allocation with joint RB-power-scheduling optimization, characterizing regimes where sharing *degrades* throughput or fairness, and embedding BDIX policies that adaptively toggle between sharing and strict orthogonalization.

Author Contributions

Iakovos Ioannou: conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, visualization, writing – original draft, writing – review and editing. **Marios Raptopoulos:** conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, visualization, writing – original draft, writing – review and editing. **Prabagarane Nagaradjane:** conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, visualization, writing – original draft, writing – review and editing. **Christophoros Christophorou:** conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, visualization, writing – original draft, writing – review and editing. **Andreas Gregoriades:** conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, visualization, writing – original draft, writing –

review and editing. **Vasos Vassiliou:** conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, visualization, writing – original draft, writing – review and editing.

Acknowledgements

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. Additionally, this work was co-funded by the European Union under the programme of social cohesion "THALIA 2021-2027", through the Research and Innovation Foundation (Project: EXCELLENCE/0524/0218).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data available on request from the authors.

Endnotes

¹ *NLoS support.* Yes—our framework is NLoS-tolerant by construction: (i) the channel model in Equation (8) sweeps the Rician K_{RIC} -factor down to $K_{\text{RIC}} \rightarrow 0$ (Rayleigh), covering pure NLoS; (ii) clustering & AP election operate on shared UE positions X (ProSe) with common (θ, s) and are thus *CSI-agnostic*, preserving deterministic agreement (Equation 15); (iii) association and KPI optimization absorb NLoS through SINR (Equations 8–11) and the fallback rule in Equation (14); and (iv) the hybrid localization uses geometry+MLP with robust weighting/outlier screening to mitigate NLoS-biased ranges. Practically, the system degrades gracefully as K_{RIC} decreases and remains fully functional under NLoS.

² The reason that the BS and telecom do not participate in the LTE ProSe messages reading is due to power consumption restrictions.

References

1. H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-Free Massive MIMO Versus Small Cells," *IEEE Transactions on Wireless Communications* 16, no. 3 (2017): 1834–1850.
2. E. Björnson, L. Sanguinetti, and H. Wymeersch, "Cell-Free Massive MIMO: A New Next-Generation Paradigm," *IEEE Communications Magazine* 57, no. 6 (2019): 110–117.
3. S.-C. Han, H.-C. Yang, and M.-S. Alouini, "High-Accuracy Localization for Ultra-Dense Networks: A Survey," *IEEE Communications Surveys & Tutorials* 22, no. 1 (2020): 630–664.
4. Y. Liu, Y. Zhang, S. Jin, and M. Matthaiou, "Dynamic Clustering and Resource Allocation in Cell-Free Massive MIMO Networks," *IEEE Transactions on Wireless Communications* 20, no. 8 (2021): 5278–5294.
5. X. Chen, Y. Shi, and S. Yan, "Machine Learning-Enabled Resource Management for Cell-Free Massive MIMO Networks," *IEEE Wireless Communications Letters* 12, no. 1 (2023): 35–38.
6. Y. Sun, X. Wang, D. Wu, and G. Zhao, "A Survey on Localization Techniques for Next-Generation Wireless Networks," *IEEE Communications Surveys & Tutorials* 25, no. 1 (2023): 12–36.
7. X. Zhang, R. Yu, and M. Liu, "Enhanced Data Fusion Techniques for Precise Localization in 6G Networks," *IEEE Internet of Things Journal* 11, no. 2 (2024): 567–578.
8. F. Jameel, P. Kumar, and A. Sharma, "Edge Computing for Decentralized Cluster Management in Cell-Free Networks," *IEEE Transactions on Network and Service Management* 20, no. 2 (2023): 1050–1063.

9. R. Khalili and A. Rahmati, "Machine Learning-Driven Resource Allocation in Cell-Free Networks," *IEEE Network* 37, no. 4 (2023): 112–118.
10. M. Nguyen, N. Tran, and T. Vu, "Dynamic Power Control for Energy Efficiency in Cell-Free Massive MIMO Networks," *IEEE Transactions on Green Communications and Networking* 8, no. 1 (2024): 140–152.
11. J. Chen, E. Björnson, and E. G. Larsson, "Dynamic Clustering in User-Centric Cell-Free Massive MIMO," *IEEE Transactions on Wireless Communications* 20, no. 8 (2021): 5379–5393.
12. K. Iliadis and I. Pitas, "Deep Learning for Channel Estimation in Cell-Free Massive MIMO," *IEEE Transactions on Communications* 70, no. 5 (2022): 3495–3508.
13. M. B. Shahab, X. Zhu, and M.-S. Alouini, "Full-Duplex Cell-Free Massive MIMO: Opportunities and Challenges," *IEEE Communications Magazine* 62, no. 1 (2024): 78–84.
14. M. Mashdour and R. C. de Lamare, "Information Rate-Based Clustering in Cell-Free Massive MIMO," *IEEE Wireless Communications Letters* 13, no. 2 (2024): 215–218.
15. J. Beerten and S. Pollin, "O-RAN Clustering and Handover Strategies for Cell-Free Networks," *IEEE Access* 11 (2023): 1290–1304.
16. O. T. e. a. Demir, "Uplink Performance in User-Centric Cell-Free Massive MIMO," *IEEE Transactions on Communications* 69, no. 12 (2021): 8563–8576.
17. G. e. a. Di Gennaro, "Deep Learning-Based User-Centric Clustering for CF-MMIMO," *IEEE Access* 12, no. 1 (2024): 234–247.
18. Y. e. a. Tan, "Energy-Efficient Hierarchical DRL for CF-MMIMO," *IEEE IoT Journal* 11, no. 4 (2024): 1899–1912.
19. I. Ioannou, A. Gregoriades, C. Christophorou, M. Raspopoulos, and V. Vassiliou, "Implementing a Cell-Free 6G Distributed AI Network with the Use of Deep ML under a Traditional Multi-Cell Mobile Network," in *2025 5th IEEE Middle East and North Africa Communications Conference (MENACOMM)* (2025), pp. 1–8.
20. I. Ioannou, P. Nagaradjane, C. Christophorou, A. Gregoriades, and V. Vassiliou, "Revolutionizing 6G Networks: Implementing Cell-Free Architecture with Advanced Clustering through Deep Machine Learning," in *IEEE International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (IEEE, 2025), 1–7.
21. V. E. Idigo, A. C. Igwilo, and F. O. Nwokoye, "Cluster Head Selection in Device-to-Device (d2d) Using Self-Organizing Map (SOM)," *European Journal of Electrical and Computer Engineering* (2023), <https://www.ejece.org/index.php/ejece/article/view/564>.
22. Z. Yang, W. Liu, and Y. Zhang, "High-Precision Localization Techniques in Cell-Free Massive MIMO Networks," *IEEE Transactions on Vehicular Technology* 72, no. 4 (2023): 4105–4117.
23. N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low-Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications* 7, no. 5 (2000): 28–34.
24. J. Li, X. Yue, J. Chen, and F. Deng, "A Novel Robust Trilateration Method Applied to Ultra-Wideband Localization," *Sensors* 17, no. 4 (2017): 795.
25. M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent Neural Networks for Accurate RSSI Indoor Localization," *IEEE Internet of Things Journal* 6, no. 6 (2019): 10639–10651.
26. Z. Ouyang, X. Huang, J. Wang, F. Xiao, X. Zhao, and X. Li, "Magnetic-Field-Based Indoor Positioning Using Temporal Convolutional Networks," *Sensors* 23, no. 3 (2023): 1514.
27. X. Wang, X. Wang, and S. Mao, "Deep Convolutional Neural Networks for Indoor Localization with CSI Images," *IEEE Transactions on Network Science and Engineering* 7, no. 1 (2020): 316–327.
28. S. M. Nguyen, D. V. Le, and P. J. M. Havinga, "Seeing the World from Its Words: All-Embracing Transformers for Fingerprint-Based Indoor Localization," *Pervasive and Mobile Computing* 100 (2024): 101912.
29. X. Kang, X. Liang, and Q. Liang, "Indoor Localization Algorithm Based on a High-Order Graph Neural Network," *Sensors* 23, no. 19 (2023): 8221.
30. S. R. Jondhale, V. Mohan, B. B. Sharma, J. Lloret, and S. V. Athawale, "Support Vector Regression for Mobile Target Localization in Indoor Environments," *Sensors* 22, no. 1 (2022): 358.
31. Y. Wang, C. Xiu, X. Zhang, and D. Yang, "Wifi Indoor Localization with CSI Fingerprinting-Based Random Forest," *Sensors* 18, no. 9 (2018): 2869.
32. P. V. Bahl and V. N. Padmanabhan, "Radar: An In-Building RF-Based User Location and Tracking System," in *Proceedings of IEEE INFOCOM* (IEEE, 2000), 775–784.
33. I. Ioannou, V. Vassiliou, C. Christophorou, and A. Pitsillides, "Distributed Artificial Intelligence Solution for D2D Communication in 5G Networks," *IEEE Systems Journal* 14, no. 3 (2020): 4232–4241, <http://arxiv.org/abs/2001.05232>, <https://ieeexplore.ieee.org/document/9080591/>.
34. I. I. Ioannou, C. Christophorou, V. Vassiliou, M. Lestas, and A. Pitsillides, "Dynamic D2D Communication in 5G/6G Using a Distributed AI Framework," *IEEE Access* 10 (2022): 62772–62799, https://www.researchgate.net/publication/361327301_Dynamic_D2D_Communication_in_5G6G_Using_a_Distributed_AI_Framework, <https://ieeexplore.ieee.org/document/9794664/>.
35. I. Ioannou, C. Christophorou, V. Vassiliou, and A. Pitsillides, "A Novel Distributed AI Framework with ML for D2D Communication in 5G/6G Networks," *Computer Networks* 211 (2022): 108987.
36. I. Ioannou, C. Christophorou, V. Vassiliou, and A. Pitsillides, "A Distributed AI/ML Framework for D2D Transmission Mode Selection in 5G and Beyond," *Computer Networks* 210 (2022): 108964.
37. C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. (Springer, 2006).
38. T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics* 43, no. 1 (1982): 59–69.
39. S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, "Clustergan: Latent Space Clustering in Generative Adversarial Networks," in *Proceedings of the AAAI Conference on Artificial Intelligence* 33, no. 01 (2019): 4610–4617.
40. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)* (AAAI Press, 1996), 226–231.
41. J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* 1, no. 14 (University of California Press, 1967), 281–297.
42. U. Shaham, S. Stanton, G. Li, B. Nadler, and Y. Kluger, "Spectralnet: Spectral Clustering Using Deep Neural Networks," in *International Conference on Learning Representations (ICLR)* (ACM, 2018).
43. J. Yang, D. Parikh, and D. Batra, "Joint Unsupervised Learning of Deep Representations and Image Clusters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), 5147–5156.
44. X. Guo, L. Gao, X. Liu, and J. Yin, "Deep Clustering with Convolutional Autoencoders," *Neural Networks* 92 (2017): 31–38.
45. M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," in *IEEE Transactions on Signal Processing* 45, no. 11. (IEEE, 1997): 2673–2681.
46. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature* 521, no. 7553 (2015): 436–444.
47. S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," arXiv preprint, arXiv:1803.01271 (2018).
48. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*

- 30 (2017): 5998–6008, https://papers.nips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
49. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
50. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *International Conference on Learning Representations (ICLR)* (IEEE Information Theory Society, 2017).
51. A. J. Smola and B. Schölkopf, “A Tutorial on Support Vector Regression,” *Statistics and Computing* 14, no. 3 (2004): 199–222.
52. L. Breiman, “Random Forests,” *Machine Learning* 45, no. 1 (2001): 5–32.
53. T. Cover and P. Hart, “Nearest Neighbor Pattern Classification,” *IEEE Transactions on Information Theory* 13, no. 1 (1967): 21–27.
54. X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, “An Overview of 3gpp Device-to-Device Proximity Services,” *IEEE Communications Magazine* 52, no. 4 (2014): 40–48.
55. A. Asadi, Q. Wang, and V. Mancuso, “A Survey on Device-to-Device Communication in Cellular Networks,” *IEEE Communications Surveys & Tutorials* 16, no. 4 (2014): 1801–1819.
56. I. Ioannou, C. Christophorou, V. Vassiliou, and A. Pitsillides, “A Distributed AI/ML Framework for D2D Transmission Mode Selection in 5G and Beyond,” *Computer Networks* 210 (2022): 108964, <https://doi.org/10.1016/j.comnet.2022.108964>.
57. 3GPP, “Proximity-Based Services (ProSe); Stage 1 (ts 22.803),” 3rd Generation Partnership Project, Tech. Rep., 2014.
58. S. G. Quintiliani, F. J. Cintron, D. W. Griffith, and R. A. Rouil, “Impact of Timing on the Proximity Services (ProSe) Synchronization Function,” in *IEEE Consumer Communications & Networking Conference (CCNC)* (IEEE, 2017), 498–503.
59. S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice*, 2nd ed. (Wiley, 2011).
60. T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. (Prentice Hall, 2002).
61. H. T. Friis, “A Note on a Simple Transmission Formula,” *Proceedings of the IRE* 34, no. 5 (1946): 254–256.
62. J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. (McGraw-Hill, 2008).
63. S. Buzzi and C. D’Andrea, “User-Centric vs. Network-Centric Cell-Free Massive MIMO,” in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* (IEEE, 2017), 1–5.
64. J. G. Proakis, *Digital Communications*, 4th ed. (McGraw-Hill, 2001).
65. C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal* 27 (1948): 379–423, 623–656.
66. 3GPP, “E-Utra; Physical Channels and Modulation (ts 36.211),” 3rd Generation Partnership Project, Tech. Rep., rel-17/18, 2023.
67. L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, ser. Wiley Series in Probability and Statistics (John Wiley & Sons, 2009).
68. D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, no. 5 (2002): 603–619.
69. B. J. Frey and D. Dueck, “Clustering by Passing Messages Between Data Points,” *Science* 315, no. 5814 (2007): 972–976.
70. J. Xie, R. Girshick, and A. Farhadi, “Unsupervised Deep Embedding for Clustering Analysis,” in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, Vol. 48 (International Machine Learning Society, 2016), 478–487.
71. D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” arXiv preprint, arXiv:1312.6114 (2013).
72. I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., “Generative Adversarial Nets,” *Advances in Neural Information Processing Systems (NeurIPS)* 27 (2014): 2672–2680.
73. P. J. Rousseeuw, “Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis,” *Journal of Computational and Applied Mathematics* 20 (1987): 53–65.
74. J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *Journal of Machine Learning Research* 13 (2012): 281–305.
75. I. Ioannou, C. Christophorou, V. Vassiliou, M. Lestas, and A. Pitsillides, “Dynamic D2D Communication in 5G/6G Using a Distributed AI Framework,” *IEEE Access* 10 (2022): 62772–62799.
76. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” arXiv preprint, arXiv:1710.10903 (2017), <https://arxiv.org/abs/1710.10903>.
77. R. Kohavi, “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection,” in *International Joint Conference on Artificial Intelligence (IJCAI)* (Morgan Kaufmann, 1995), 1137–1143.
78. M. K. Simon and M.-S. Alouini, *Digital Communication over Fading Channels*, 2nd ed. (John Wiley & Sons, 2005).
79. D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* no. 2 (1979): 224–227.
80. T. Calinski and J. Harabasz, “A Dendrite Method for Cluster Analysis,” *Communications in Statistics-theory and Methods* 3, no. 1 (1974): 1–27.
81. Y. Chen, S. Zhang, K. Xu, and Y.-C. Li, “Fundamental Tradeoffs on Green Wireless Networks,” *IEEE Communications Magazine* 49, no. 6 (2011): 30–37.
82. R. Jain, D.-M. W. Chiu, and W. R. Hawe, “A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems,” Digital Equipment Corporation, Tech. Rep. DEC-TR-301, 1984.
83. R. Arshad, S. Baig, and S. Aslam, “User Clustering in Cell-Free Massive MIMO NOMA System,” *Alexandria Engineering Journal* 90 (2024): 183–196, <https://www.sciencedirect.com/science/article/pii/S1110016824000863>.
84. S. Bassoy, H. Farooq, M. A. Imran, and A. Imran, “Coordinated Multi-Point Clustering Schemes: A Survey,” *IEEE Communications Surveys & Tutorials* 19, no. 2 (2017): 743–764.
85. S. Sangwine and G., “Evaluation of Localisation Methods in Wireless Sensor Networks,” *IEEE Transactions on Mobile Computing* 16, no. 7 (2017): 2091–2104.
86. W. Zhang and X. Liu, “Localization Accuracy in Mobile Positioning,” *Journal of Communication Networks* 21, no. 3 (2019): 156–162.
87. Y. Ma and et al., “Comparative Analysis of Localization Errors in Machine Learning Models,” *Sensors* 18, no. 12 (2018): 4285–4297.
88. E. Olsson and et al., “Evaluation Metrics for Geospatial Localisation,” *International Journal of Geographical Information Science* 30, no. 2 (2016): 415–431.
89. M. Sokolova and G. Lapalme, “A Systematic Analysis of Performance Measures for Classification Tasks,” *Information Processing & Management* 45, no. 4 (2009): 402–412.
90. T. Chai and R. R. Draxler, “Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?” *Geoscientific Model Development* 7, no. 3 (2014): 1247–1254.
91. D. Sharma and H. Gupta, “Understanding R-Squared in Machine Learning,” *International Journal of Advanced Computer Science and Applications* 7, no. 8 (2016): 221–225.

92. J. R. Taylor, *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements* (University Science Books, 2008).
93. W. Huang, et al., "The Median Absolute Error in Model Validation," *Journal of Applied Statistics* 44, no. 7 (2017): 1185–1197.
94. B. Wallace and T. Al-Jazzar, "Explained Variance in Machine Learning Models," *Pattern Recognition Letters* 48 (2015): 58–65.