Journal of Inventive Engineering and Technology (JIET)
ISSN: 2705-3865
Volume-7, Issue-1, pp-1-11
www.jiengtech.com
Research Paper Open Access

User Centric Based Clustering for Mitigating Intercell Interference for 5G Network using Machine Learning Approach

[1]Ogene, Somto E., Department of Electronic & Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria
[2]Oguejiofor, Obinna, S., Department of Electronic and Computer Engineering , Nnamdi Azikiwe University, Awka, Anambra State, Nigeria

*[1]**Corresponding Author**: ogenesomto13@gmail.com*

ABSTRACT : *The introduction of fifth-generation (5G) mobile communication networks has brought about revolutionary improvements in user experience, latency reduction, and data speeds. However, issues like frequency reuse-induced inter-cell interference continue to exist and impede the best possible network performance. In order to reduce inter-cell interference in dense 5G networks, this study investigates a novel user-centric clustering strategy that makes use of machine learning techniques. The suggested approach optimizes scheduling and resource allocation by dynamically classifying users according to interference levels and proximity, guaranteeing increased spectrum efficiency and user satisfaction. Through the use of a variety of clustering algorithms, including KM-Means clustering, the model successfully handles network variability. The outcomes show notable increases in network efficiency, decreased interference, and improved quality of service, providing a scalable answer to the needs of contemporary communications. In particular, after KM-means clustering, the user's interference from the various base stations (BS1=1.9dBm, BS2=2.6dBm, BS3=2.55dBm, BS4=1.91dBm, BS5=1.4dBm) was decreased to (BS1=0.97dBm, BS2=1.25dBm, BS3=1.23dBm, BS4=0.98dBm, BS5=0.7dBm). The network can improve data rate and provide a more seamless user experience with fewer interference.*

KEYWORDS Clustering, Machine Learning, Inter-cell Interference, 5G Network.

Date of Submission: 27-01-2025 Date of acceptance: 01-02-2025

I. INTRODUCTION

In a study by authors in [1], the fifth-generation mobile communication network (5G) technologies are expected to address problems such as low data rate, limited capacity, low latency, and poor quality of service experience in a network. On the other hand, introducing 5G networks has also brought up new difficulties, such as inter-cell interference, due to universal re-use frequency adopted for the deployment of 5G networks which can negatively impact user experience and network performance. User-centric clustering is a promising approach that has been employed for mitigating inter-cell interference in fifth-generation networks. By making use of the potential of artificial intelligence (AI), specifically machine learning (ML) algorithms, this approach can accomplish better performance compared to conventional techniques. The concept behind user-centric clustering is to group users based on their proximity to the base station and their interference levels [2]. This way, interference can be reduced by optimizing resource allocation and scheduling within each cluster. User-centric clustering is one of the approaches used for clustering, aside the network-centric clustering, user-centric clustering is one of the most suitable approaches in clustering as this approach is solely dependent on the preference and information obtained from the user. The user-centric clustering can be combined with other techniques, such as beamforming and power control, to further enhance network performance and user experience.

To resolve the problem encountered due to inter-cell interference, a user-centric clustering approach can be deployed, making using of the power of artificial intelligence, specifically machine learning techniques, and by grouping users based on their proximity and interference levels, this approach can optimize resource allocation and scheduling within each cluster, thus reducing inter-cell interference and improving overall network performance.

Inter-cell interference is a phenomenon that occurs in cellular networks when signals from different cells interfere with each other, resulting in degraded network performance and reduced user experience[3]. This interference arises due to the limited frequency spectrum available for wireless communication and the high density of cells in the 5G cellular networks. As a result, neighboring cells can transmit signals on the same frequency, leading to interference, which can cause poor signal quality, dropped calls, reduced data rates, and other issues. The mitigation of inter-cell interference is a critical area of research in 5G networks, and various techniques, including user-centric clustering, beamforming, and power control, are being explored to address this challenge. Traditional methods for interference mitigation often involve static configurations and heuristics, which may not adapt well to the dynamic nature of 5G networks. In contrast, user-centric approaches consider individual users' specific needs and behavior, providing a more personalized and effective solution.

Beamforming is the process of directing the transmitting and receiving beams toward the intending user equipment while reducing interference to neighboring user equipment within a location. Coordinated beamforming is a technique for base stations to jointly design or reassign their respective beamforming vectors to control inter-cell interference and information leakage [4]. This technique involves multiple base stations working together to manage interference and optimize the transmission of signals to and from multiple users.

Note that the best technique for clustering and optimizing RA in UE-centric based clustering for multi-cell interference reduction in fifth generation mobile networks using machine learning approaches depends on many variable, which include size and density of the system, level of interference, finally the availability of data. However, some commonly used techniques include k-means clustering, hierarchical clustering, and fuzzy clustering.

K-means clustering is a technique that divides users into k clusters based on their proximity and interference levels [5]. The algorithm iteratively assigns users to clusters and computes the centroid of each cluster, optimizing for the sum of squared distances between users and them respective centroids.

Hierarchical clustering is another technique that can be used for user-centric based clustering. It creates a tree-like structure by recursively merging smaller clusters into larger ones, based on their proximity and interference levels [6]. This approach can be useful when the number of clusters is not known in advance, and it can also provide insights into the hierarchical structure of the network.

Fuzzy clustering is a technique that assigns users to multiple clusters, with each user having a degree of membership in each cluster [6]. This approach can be useful when users have varying degrees of proximity and interference, and it can provide a more nuanced approach to clustering.

In terms of resource allocation and scheduling, machine learning algorithms such as reinforcement learning and deep learning can be used to optimize the allocation of resources within each cluster [7]. These algorithms can learn from past experiences and optimize resource allocation and scheduling based on network conditions and user requirements.

Despite considerable advancements in the field of 5G heterogeneous networks (HetNets), there remains a notable gap in the development of user-centric clustering schemes integrated with machine learning-based interference mitigation schemes for effective inter-cell interference mitigation. While existing research such as proposed by authors in [1], [8] - [13], have contributed significantly to understanding user-centric clustering schemes, resource allocation, and interference mitigation techniques in 5G and beyond networks, a notable research gap exists regarding the integration of machine learning (ML) techniques into user-centric methodologies for inter-cell interference mitigation.

Comprehensive methods that put the user experience first by dynamically modifying resource allocation and clustering algorithms in response to current network conditions and user demands are lacking. Additionally, while machine learning (ML) algorithms have shown promise in optimizing network performance, their application in conjunction with user-centric methodologies for interference mitigation in 5G HetNets remains relatively unexplored. Thus, there is a need for novel research that bridges this gap by proposing innovative ML-

www.jiengtech.com. All Rights Reserved2025.

driven clustering schemes and interference mitigation strategies tailored to individual user requirements, thereby enhancing the overall quality of service in 5G HetNets.

Therefore, in this work we differ from what a lot of authors did by devising a KM-means clustering machine learning approach for clustering and inter-cell interference mitigation in 5G networks.

II. MATERIALS AND METHODS

A. Materials

The programming language used to develop this mitigation scheme is the Python Programming Language. With an Integrated Development Environment like PyCharm which enhances productivity by providing features like code completion, debugging, and visualization tools. The PyCharm Community Edition (version 2023.3.3) was employed for coding, especially during debugging and development. Its integrated development environment (IDE) features helped manage the project's structure, maintain the codebase, and ensure proper configuration of dependencies. It uses built-in tools or libraries like Matplotlib and Seaborn to visualize data and model performance thereby providing debugging tools to identify and fix issues in the code efficiently. Also, the Integration of Jupyter Notebook (version 6.5.4) was used for Python coding and documenting the experimental process. It allowed interactive data analysis and was crucial for running models, visualizing results, and sharing findings. This Python Libraries Used includes;

Pandas: Pandas is a robust toolkit for data analysis and manipulation that offers data structures such as series and data frames. It is frequently used to organize, clean, and transform datasets, which facilitates the handling of structured data and the execution of intricate tasks like grouping, merging, and reshaping.
NumPy: A core Python package for numerical computation is called NumPy. In addition to supporting arrays and matrices, it offers a number of mathematical functions for effectively working with various data structures. Pandas and other data science libraries are built on top of NumPy, which is widely used in scientific computing.
Seaborn: Seaborn is a Matplotlib-based data visualization library. It offers a user-friendly interface for producing visually appealing and educational statistical visuals. Seaborn has features like color palettes, themes, and straightforward syntax for creating intricate plots, and it works well with Panda's data structures.
Matplotlib: Matplotlib is a flexible Python charting toolkit that offers extensive features for producing interactive, animated, and static displays. It is a fundamental Python data visualization toolkit since it provides a large variety of plot kinds and customization choices.
TensorFlow: Primarily used for machine learning but also for integration between preprocessing and modeling stages.
Keras: A high-level API built on TensorFlow, simplifying model building and training.
Scikit-learn (Sklearn): Provided tools for machine learning tasks like classification, clustering, and data splitting.
Joblib: Used to save and load machine learning models, preventing the need for retraining.
Warnings: This module was used to filter and suppress unnecessary warning messages during preprocessing.
The hardware used was a Toshiba Satellite A660 x64-based laptop, running on Windows 10 Pro (version 10.0.19045 Build 19045) with an Intel(R) Core (TM) i5 CPU M480 @ 2.67GHz. This laptop was used as it met the computational demands for data preprocessing, model training, and visualization tasks for programs in this research. It serves as a physical platform were all this model was imported to and rendered.

B. METHODS

In this section, the step-by-step procedure on how the machine learning based user-centric clustering scheme will be implemented. This describes the methodology, including data preprocessing, feature selection, and model training. the block diagram in Fig. 1 explains succinctly how these steps will flow.

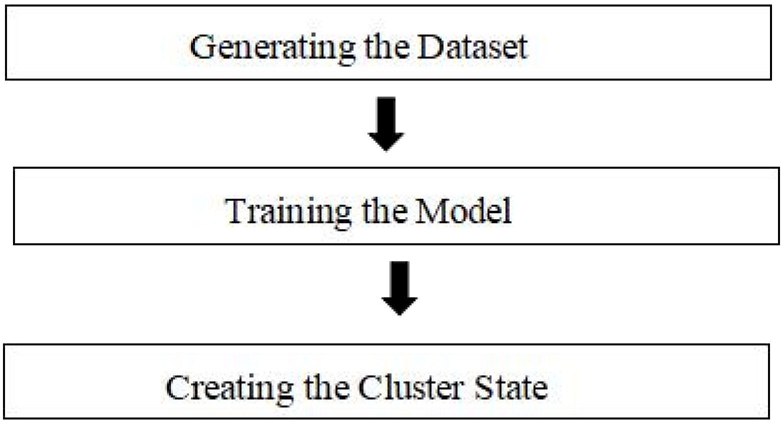

Fig. 1: Block Diagram Illustrating the step-by-step procedure on how the machine learning based user-centric clustering scheme is developed.

Data Generation

The primary dataset used in this project was generated from "wireless communication dataset for machine learning. https://www.salzburgresearch.at" with 3000 records, each representing a user in a 5G network. Key columns in the dataset include:

User Distribution:
Number of Users: 3,000 users.
Latitude Range: [9.020165, 31.926366] degrees.
Longitude Range: [18.379366, 41.539440] degrees.
Distribution Method: Uniform random distribution within the specified latitude and longitude ranges.

Home Base Station (1 to 5): This identifies which of the five base stations a user is connected to. It defines the user's starting point in terms of signal reception.

Base Station Locations:

Table 1: The Geographical Range Covered by Each Base Station

BASE STATION	LATITUDE RANGE	LONGITUDE RANGE
Base Station 1	9.020165 to 10.926139	18.379366 to 21.066517
Base Station 2	13.690127 to 15.782322	24.275958 to 26.046194
Base Station 3	19.040614 to 21.231621	28.938052 to 31.094901
Base Station 4	24.242576 to 26.360085	33.966279 to 36.030374
Base Station 5	28.987429 to 31.926366	38.849039 to 41.539440

Distance from Base Station (1 to 5): This represents the physical distance between the user and each base station. Signal strength diminishes with distance, so this feature is critical in determining interference levels. To determine the distance of the various user equipment we use the Haversine Formula.

Using the Haversine Formula:

$$d = 2 \text{ x } 6371 \text{ x } \arcsin\left(\sqrt{\sin^2\left(\frac{\emptyset_2 - \emptyset_1}{2}\right) + \cos(\emptyset_1)\cos(\emptyset_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \quad (1)$$

For each user, the distance to each base station was calculated using the formula above.

www.jiengtech.com. All Rights Reserved2025.

Where:

$\emptyset_1$, λ1 are the latitude and longitude of the user (in radians)
$\emptyset_2$, λ2 are the latitude and longitude of the base station (in radians)
6371 is the earth's radius in km

The barplot in Fig. 2 shows the average distance from users to each base station, providing a comparative view of the proximity of users to different base stations.

Interference from Base Station (1 to 5): This measures the interference a user experiences from each base station. Users near non-home base stations, especially at the cell edge, likely receive stronger interference.

The total interference from each base station in a cluster is calculated using the formula:

$$I = \sum_{j=1}^{N} \frac{P_j G_j}{d_j^{\alpha}} + \epsilon \qquad (2)$$

where:
I is the total interference at the user equipment receiver from j-th base station (in dBm),
dj is the distance from the j-th base station (in meters) to the receiver,
N is the number of interfering base stations,
α is the path loss exponent typically between 2 and 4, depending on the environment.
P_j is the power transmitted by the j-th base station.
G_j is the gain of the j-th base station's antenna,
ϵ is a random background noise component uniformly distributed between [-100, 10] dBm to simulate environmental variability.

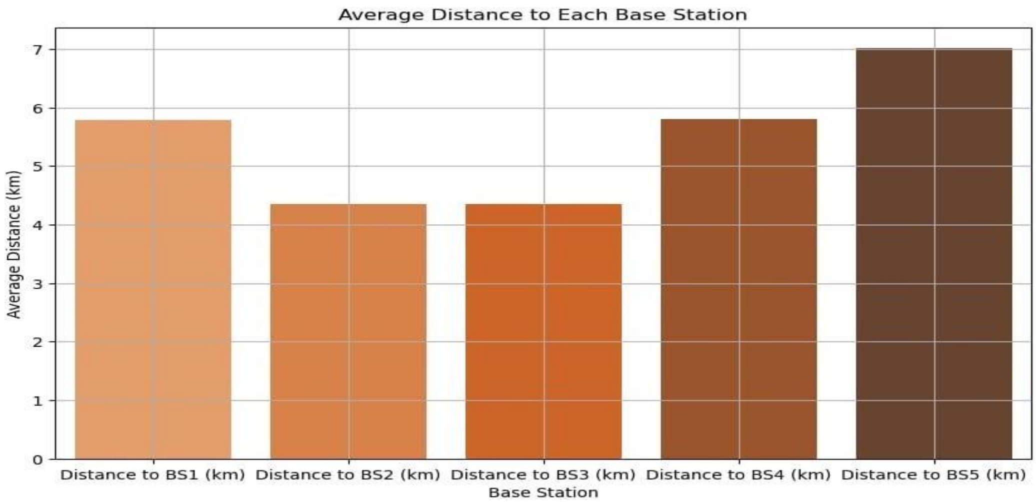

Fig.2: Diagram of the average distance to each base station

Feature Selection
For feature selection, SelectKBest from sklearn's feature selection module was employed. This technique selects the most relevant features that will improve the model performance. These features are based on statistical significance and will select the top ones with the most impact on the target variable. In this project, it was used to reduce the number of features, focusing on interference levels and distances from base stations. This enhanced the model accuracy and reduces overfitting.

Model Training Architecture
Gradient Boosting Classifier module is a machine learning algorithm designed to handle classification tasks by building an ensemble of weak learners, typically decision trees, to create a strong predictive model. The algorithm operates on the principle of gradient boosting, where new models are added sequentially to correct errors made by earlier ones. This iterative approach allows the model to optimize a loss function and

progressively improve its predictions over time. Each subsequent tree in the ensemble targets the residual errors, or the difference between actual values and previous model predictions.
The core of gradient boosting lies in minimizing a differentiable loss function, with the goal of reducing the errors from the prior model. During each iteration, the algorithm fits a new weak learner to the negative gradient of the loss function, which is computed based on the current model's predictions. This additive approach allows each new model to contribute to reducing the overall error. The learning rate plays a critical role here, as it scales the contribution of each weak learner to avoid overfitting. Compared to other classification algorithms, the GradientBoosting Classifier has distinct advantages and some drawbacks, especially when considering performance and dataset size.

When comparing it to Random Forest, which builds decision trees through bagging (bootstrap aggregation), the main difference is that Gradient Boosting builds trees sequentially, with each new tree correcting the mistakes of the previous ones. Random Forest, known for its robustness and ability to handle large datasets, can fall short on smaller datasets. With 3000 entries, Gradient Boosting may deliver better accuracy because it actively minimizes residual errors, while Random Forest's reliance on randomness requires larger datasets for comparable performance. However, Gradient Boosting demands careful tuning of hyperparameters like the learning rate, number of trees, and tree depth to avoid overfitting, while Random Forest usually performs well with default settings and is less prone to overfitting.

Support Vector Machines (SVMs), which classify data by finding a hyperplane that maximizes the margin between classes, are particularly effective with small to moderately sized datasets. They excel when the data is linearly separable or can be transformed into a separable space via kernel functions. Nevertheless, SVMs tend to struggle with larger datasets due to their computational cost and training time. In contrast, Gradient Boosting can handle a dataset of 3000 entries more efficiently as it doesn't rely on complex kernel functions and can be tuned for larger datasets. Moreover, while SVM requires feature scaling, Gradient Boosting can process raw data, making it more versatile. SVM, however, can still outperform Gradient Boosting in scenarios where the class boundaries are complex and a margin-based approach is crucial.
A single decision tree is often easy to interpret but tends to suffer from either high bias (if shallow) or high variance (if deep). Gradient Boosting addresses this by combining many weak trees into a strong model. By refining predictions iteratively, Gradient Boosting can mitigate both bias and variance, offering greater accuracy than a standalone decision tree, especially for datasets like the one with 3000 entries. The data in such a case may not be complex enough for a deep tree, but it can benefit from the iterative improvement that Gradient Boosting provides.

HistGradientBoosting, an extension of Gradient Boosting, enhances efficiency by discretizing continuous features into histograms, making it ideal for large datasets as it conserves memory and reduces computation time. However, with only 3000 entries, the advantages of HistGradientBoosting may not be as pronounced. The standard GradientBoostingClassifier is sufficiently efficient for this dataset size, and its flexibility in tuning parameters such as learning rate and tree depth allows it to achieve high accuracy. Although HistGradientBoosting scales better to larger datasets, the computational overhead of creating histograms offers little advantage for smaller datasets, making the standard GradientBoosting Classifier a more balanced choice.
For a dataset of 3000 entries, Gradient Boosting offers several key advantages. First, it is highly effective for medium-sized datasets, delivering high accuracy through its iterative learning process. By making small adjustments at each stage, Gradient Boosting reduces the bias present in earlier models, making it a strong candidate for tasks requiring high accuracy on datasets that are neither too small nor too large.
Second, Gradient Boosting performs well with imbalanced data, where one class may be underrepresented. By assigning higher weights to misclassified examples, the algorithm adjusts to focus on the underrepresented class in subsequent iterations. This gives it an edge over Random Forest, which can struggle with imbalanced classes without additional techniques like class weighting or resampling.

Third, Gradient Boosting is highly customizable. Through hyperparameters like the learning rate, number of trees, and tree depth, the model can be finely tuned to optimize performance and avoid overfitting. For the dataset in question, careful tuning can result in a model that strikes the right balance between bias, variance, and accuracy.
Fourth, Gradient Boosting is computationally efficient. Although it may not be as fast as Random Forest or HistGradientBoosting on larger datasets, it offers a good trade-off between computational cost and performance for medium-sized datasets like the one with 3000 entries. Its tree construction process can be parallelized, further improving scalability without excessive training times.

Fifth, while Gradient Boosting is not as inherently interpretable as a single decision tree, tools like feature importance rankings can provide valuable insights into the model's decision-making process. This is particularly important in applications where understanding which features contribute most to the predictions is a key concern. In comparison, SVM's reliance on complex mathematical transformations in kernel space offers less interpretability, making Gradient Boosting a more intuitive choice in certain situations.

Finally, Gradient Boosting naturally captures feature interactions through its tree-based structure, a capability that is crucial for datasets where such interactions are important for accurate predictions. While algorithms like SVM may require explicit feature engineering or the use of kernel functions to account for feature interactions, Gradient Boosting does so inherently, making it a more adaptable model for real-world data.

From the diagram below, the Gradient Boosting Classifier has the highest score making it the most suitable approach.

	Decision Tree	Random Forest	Gradient Boosting Classifier	HistGradientBoosting Classifier	SVC	Linear SVC
Score	0.993	0.993	0.997	0.205	0.091	0.792

Fig.3: Diagram showing the score of various classifiers

KM-Means Clustering

K-means clustering helps to group users based on similarities in their features. For example, users experiencing the same or similar level of interference from different base stations can be clustered together, however K-means clustering itself does not predict or mitigate interference. Rather, it serves as a powerful tool for identifying patterns and sources of interference.

In this research, KM-Means Clustering is coined from K-Means Clustering, the "M" attached to the K indicate that mitigation is involved after identifying patterns and sources of interference, and this mitigation is achieved through machine learning approach.

After the dataset was modified by the classification model, KM-Means clustering was applied to predict base station clustering schemes that would mitigate interference. This unsupervised algorithm groups users with similar interference profiles, suggesting optimal base station groupings. The combination of these models allowed for both classification and clustering, creating a strategy to reduce interference for 5G network users at the edge.

Note that the noise threshold is set to *10 dBm* during the mitigation phase and any interference above this level is considered significant.

For each user, create a list representing the *Cluster State*:

- *False* if ***Ij*** is below 10 dBm. (***Ij*** is the interference from base station BSj (in dBm),
- *HOME* for the base station that is designated as the user's primary connection.
- *True* if ***Ij*** is above 10 dBm.

Example: For a user with the home base station BS3, if the calculated interferences are [-85, -90, -70, -83, -88] dBm, the *Cluster State* would be [False, False, 'HOME', False, False].

Model Testing and Evaluation

The classification model was evaluated using the F1-score from sklearn's metrics module. The F1-score balances precision and recall, making it ideal for datasets with uneven class distributions. Precision measures the percentage of relevant results out of the total classified as relevant.

Recall measures the percentage of actual positive results correctly classified. The F1-score was chosen for its effectiveness in cases where false positives and false negatives matter. In this project, a high F1-score indicated the model's reliability in predicting users most affected by interference, making it a crucial metric for evaluation.

Description of Columns (Features) in the Dataset

- User ID: Integer values from 1 to 3000.
- Home Base Station: Randomly assign one of the five base stations to each user.
- Latitude and Longitude: Pre-generated coordinates within the specified ranges.
- Distance to BSj (m): The pre-calculated distances using the Haversine formula in (1).
- Interference from BSj (dBm): Simulated interference values using the provided formula in (2).
- Cluster State: The list of states for each user, based on the noise threshold (10dBm).

III RESULTS AND DISCUSSION

A. Pairplot of Interference from All Base Stations

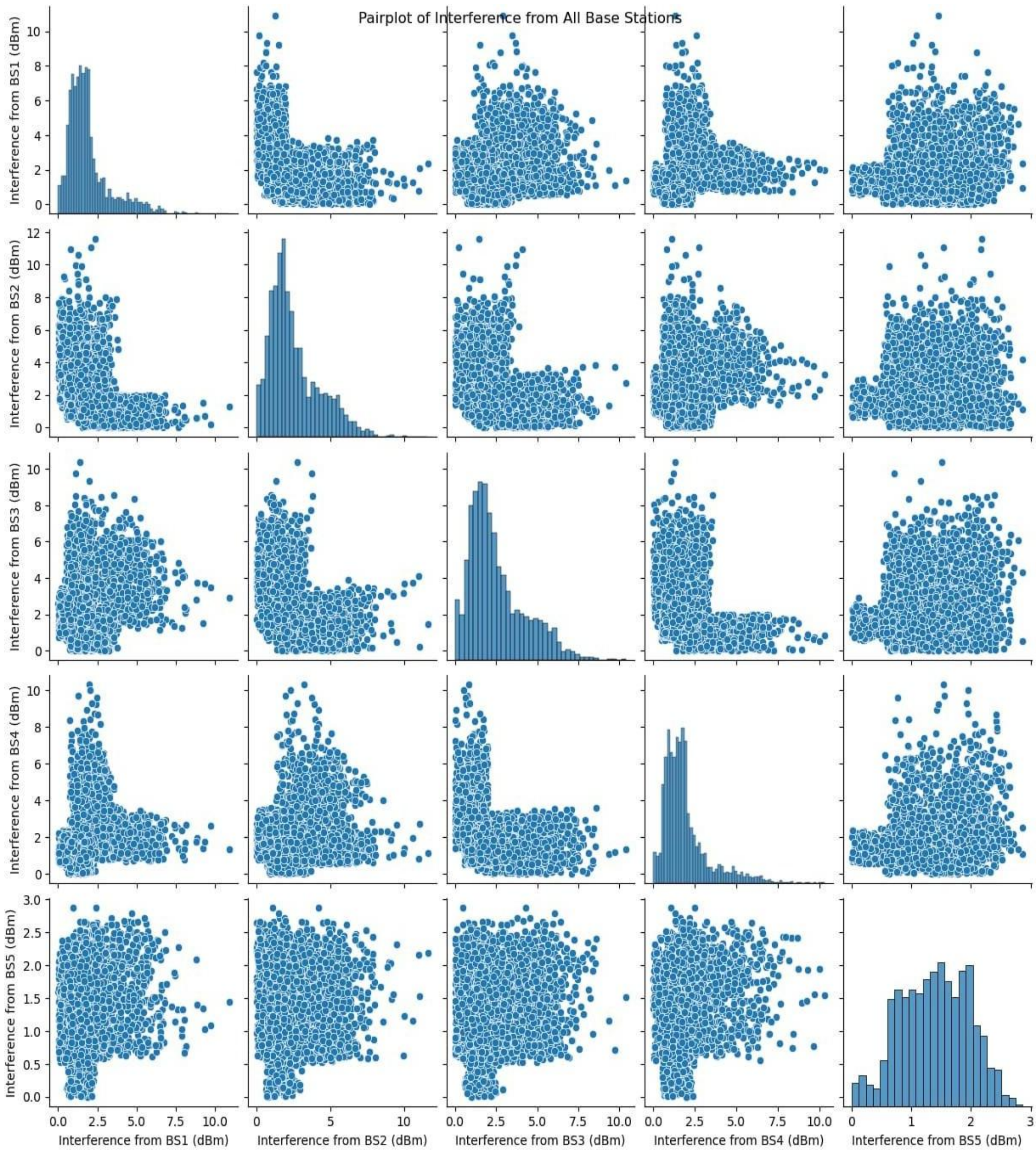

Fig. 4: Pairwise relationships between interference levels from all five base stations

The pairplot above represents the pairwise relationships and distributions of interference values from multiple base stations (BS1 to BS5) in dBm (decibels- milliwatts). This type of visualization is commonly used for exploring multi- dimensional data by showing scatterplots for pairwise comparisons and histograms for individual distributions, and from the pairplot we noticed the following.

i. **Diagonal Elements**

a. Each diagonal cell contains a histogram representing the distribution of interference values for a specific base station. For instance, the interference from BS1 shows a right-skewed distribution, while interference from BS5 exhibits a more centralized pattern.

ii. **Off-Diagonal Elements**

a. The scatterplots illustrate the relationship between interferences from different base stations. For example, the scatterplot between BS1 and BS2 shows a nonlinear relationship, with most points clustered in the lower interference range.

iii. **Symmetry**

The pairplot is symmetric along the diagonal, as the scatterplot of X vs. Y (e.g., BS1 vs. BS2) is the same as Y vs. X (BS2 vs. BS1).

Analysis of pairplots

- The relationship between interferences from different base stations depends on factors like distance, environmental conditions, and power levels.
- Nonlinear patterns (e.g., between BS2 and BS4) may indicate environmental factors like shadowing or reflection affecting signal strength.

B. Interference Received Before KM-Means Clustering

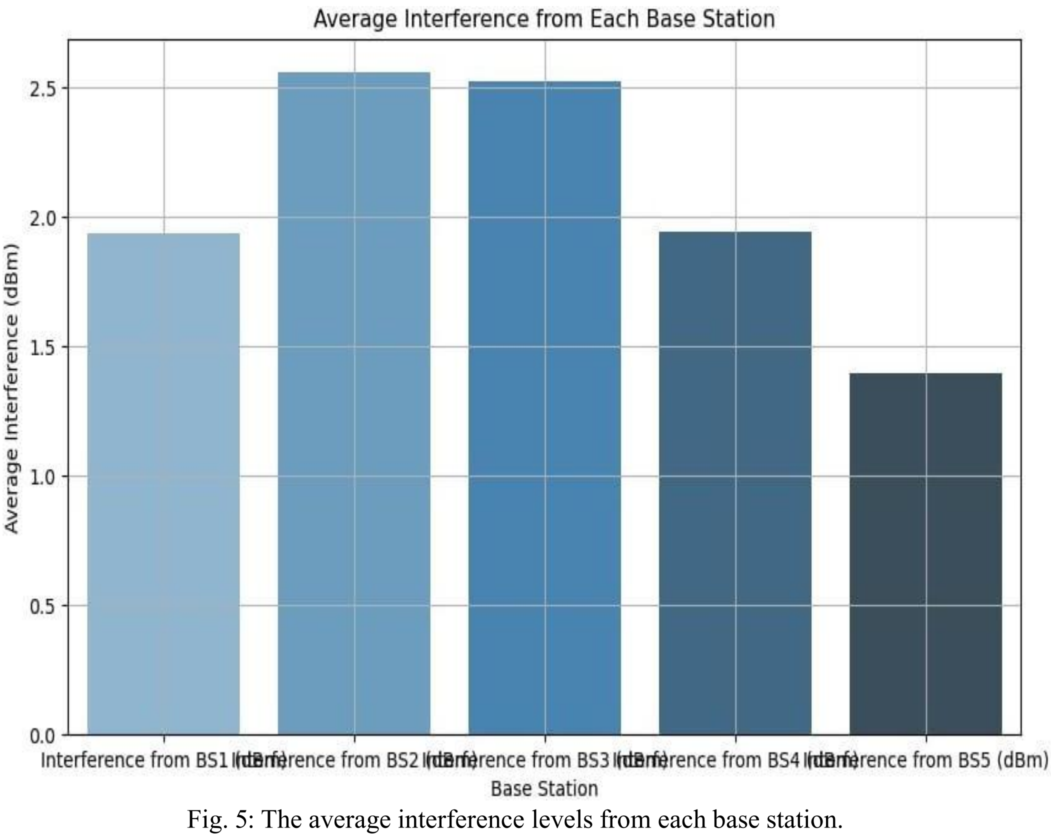

Fig. 5: The average interference levels from each base station.

Fig. 5. illustrate the average Interference measured in dBm (decibel- milliwatts) for five base stations labeled BS1 to BS5. The heights of the bars represent the interference contributions of each base station.

Observations:

- BS2 and BS3 show the highest average interference, both approximately 2.5 dBm.
- BS1 has slightly lower interference compared to BS2 and BS3, around 2.0 dBm.
- BS4 shows an intermediate interference level, approximately 1.9 dBm.
- BS5 exhibits the lowest average interference, around 1.4 dBm.

www.jiengtech.com. All Rights Reserved 2025.

C. Interference Received After KM-Means Clustering

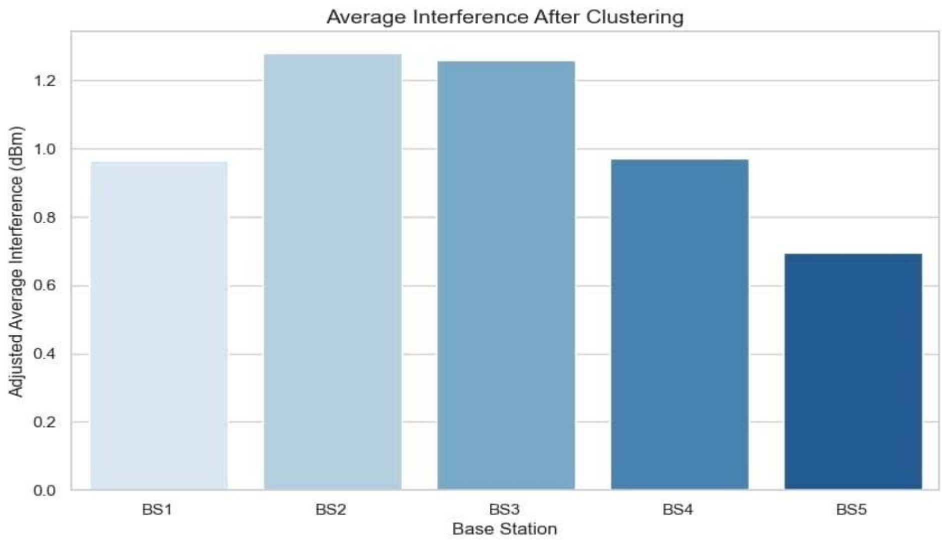

Fig. 6: The average interference levels from each base station after clustering

The Fig. 6 presented displays the **Adjusted Average Interference** in decibels (dBm) for five base stations (BS1, BS2, BS3, BS4, and BS5) after clustering. Let the interference values corresponding to each base station be denoted as I_{BS1}, I_{BS2}, I_{BS3}, I_{BS4}, and I_{BS5}.

From visual analysis:

- $I_{BS2} \approx 1.3$ dBm (maximum interference).
- $I_{BS3} \approx 1.25$ dBm
- $I_{BS1} \approx 1.0$ dBm.
- $I_{BS4} \approx 1.0$ dBm
- $I_{BS5} \approx 0.75$ dBm (minimum interference).

IV CONCLUSION

This Research work successfully developed a user-centric clustering scheme for inter-cell interference mitigation in 5G networks, utilizing machine learning techniques. By adapting KM-Means clustering within each cluster, the proposed scheme effectively mitigated inter-cell interference, improving spectral efficiency, data throughput, and overall network performance. The results demonstrate that ML-based user-centric clustering is a viable solution to address interference issues in ultra-dense 5G environments, thereby enhancing the user experience and ensuring higher quality of service (QoS) in modern telecommunications networks.

REFERENCES

[1]. Oguejiofor. O.S., and Okechukwu G. N. (2023) “ User-Centric based clustering scheme and resource allocation for inter-cell interference mitigation in 5G heterogeneous network ”. Journal of Inventive Engineering and Technology (JIET) 4(1), 1 - 10.

[2]. Dai, B. and Yu, W. (2014). “Sparce beamforming and user-centric clustering for downlink cloud radio access network.” IEEE Access 2, 1326 - 1339.

[3]. Anand S., and Mohammed, A. (2023). “A machine Learning solutions for video delivery to mitigate co-tier interference in 5G HetNets” Journal of Wireless Networks, 29 (3), 567 - 589

[4]. Dahrouj, H., and Yu, W. (2010). “coordinated beamforming for the multi-cell multi-antenna wireless system.” IEEE Transaction on Wireless system.” IEEE Transaction on wireless communications. 9 (5), 1748 - 1759.

[5]. Li, H., Chen, L. (2018). "Multi-dimensional resource allocation aware user clustering in user centric overlapped virtual cells". IEEE Transactions on Communication, 66 (11), 5483 -5495.
[6]. Yasser, H and Akroit, M. (2020). "Dynamic cell-free network architecture for non-orthogonal multiple access in ultra dense environment". IEEE Transactions on wireless communication, 19 (11), 3786 - 3802.
[7]. Djigal, H., and Zhang, Y. (2022) "Machine and Deep learning for resource allocation in multi-access edge computing: A survey." IEEE Communication Surveys and Tutorials, 24 (4), 2449 - 2494.
[8]. Humadi, W. (2023). "Simultaneous wireless information and power transfer in mmWave networks under user-centric base station clustering." IEEE Transactions on wireless communications, 22 (4), 1453 -1464.
[9]. Yuhan, L., and Li, Chun (2022). "User-centric based clustering and resource allocation for cell edge users in 5G ultr dense networks." IEEE transactions on Communications, 70 (3), 652 -661.
[10]. Jiaqi, z., and Chen L., (2019). "Conflict graph based machine learning approach for interference management in ultra dense networks." IEEE Communications Magazine , 57(12), 234-245.
[11]. Anirudh, S., and Saba, K. (2023). "UAV Integrated 5G Networks Deep Reinforcement Learning for power control and Interference Mitigation." Journal of Wireless Networks, 31 (4), 567 - 580.
[12]. Elsayed, M., and Melike, O. (2022) "machine Learning based inter-beam Inter-cell Interference mitigation in mmWave Network." IEEE Transaction on Wireless Communications, 21(8), 1231 - 1246.
[13]. Wang, H., Liu,Y (2020). "Decentralized Learning based Indoor Interference mitigation for 5G and beyond systems." IEEE wireless communications 27 (4), 44 - 52.

www.jiengtech.com. All Rights Reserved 2025.