2025

Clustering Analysis Report

5G NETWORK PERFORMANCE - UNSUPERVISED LEARNING USING CLUSTERING ALGORITHMS

ALVIN PHAN

Overview

This section presents the full analytical report on clustering methods applied to a pre-processed 5G performance dataset. The objective is to group similar geographic regions based on network metrics (latency, bitrate, retransmission, throughput) to aid in downstream forecasting. Three clustering algorithms were implemented: K-Means, DBSCAN, and Agglomerative Clustering. Each was evaluated using both visual and statistical methods.

Dataset & Feature Context

The dataset contains over 50,000 entries representing 5G network measurements across different geographic areas. Each row includes a variety of engineered features derived from raw telemetry logs and network signal data. These were added during preprocessing to strengthen clustering separation and support better forecasting.

Raw and Engineered Features:

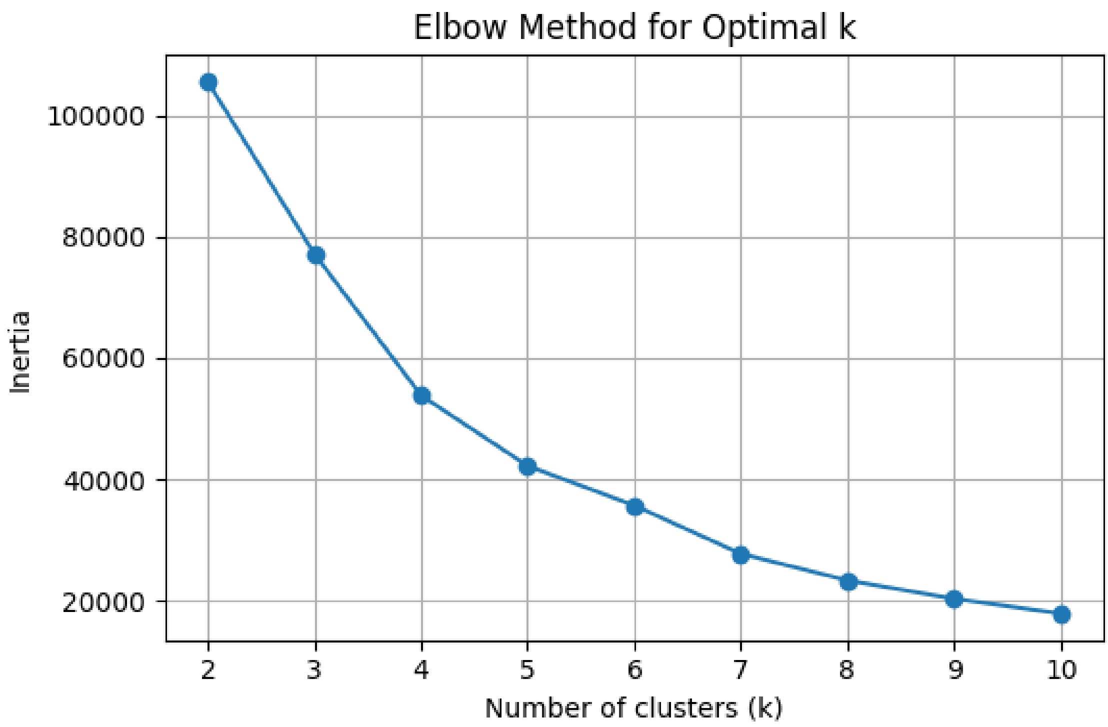
- **Latency (ms)**: Time taken for data to reach its destination
- **Bitrate (kbps)**: Effective transmission speed
- **Retransmissions**: Frequency of re-sent packets (indicates congestion or instability)
- **Throughput**: Amount of successfully transferred data
- **Retransmission Ratio**: Engineered as retransmissions divided by bitrate (indicates relative instability)
- **Throughput per Bitrate**: Ratio of throughput to bitrate to highlight delivery efficiency
- **Latency Normalized**: Scaled latency feature to adjust for geographical or tower-based differences
- **Signal Quality Composite Score**: Aggregated metric combining latency, retransmission, and throughput (weighted average)

These engineered features provide more distinct patterns, enabling stronger clustering especially for K-Means and Agglomerative approaches.

After applying MinMaxScaler to all features, a subset of 30,000 rows was used for clustering (due to memory limitations in Colab). PCA was applied to project high-dimensional features into 2D space for visualization.

Elbow method for K selection

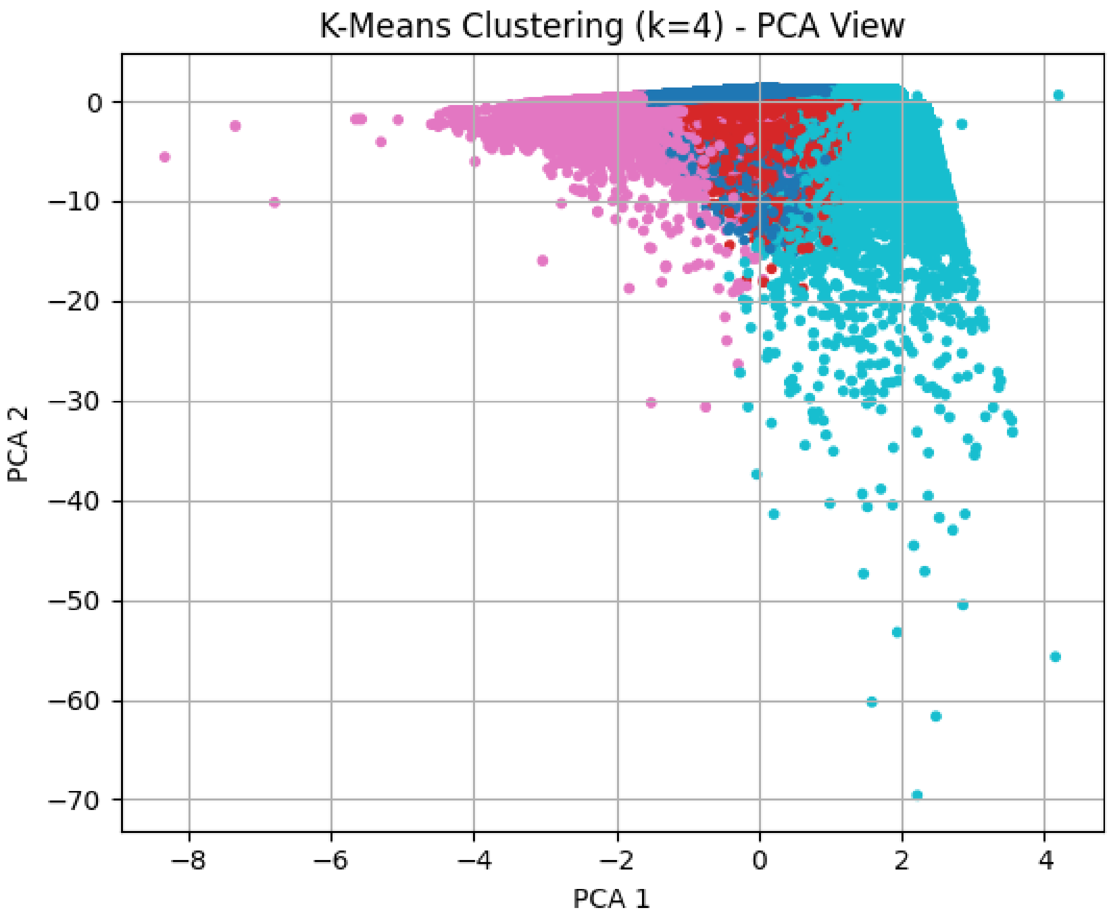
What it is: The Elbow Method is used to determine the optimal number of clusters (k) for algorithms like K-Means. It works by plotting the sum of squared distances (inertia) between points and their assigned cluster centers for various values of k. The “elbow point” is where the rate of decrease sharply changes, indicating the optimal balance between variance explanation and simplicity.



Result: An elbow was clearly observed at **k=4**, which was used for both K-Means and Agglomerative clustering to allow direct comparison.

Method 1: K-Means Clustering

How it works: K-Means partitions the data into k clusters by minimizing the distance between data points and the centroid of their assigned cluster. It assumes that clusters are spherical and balanced in size.



Result:

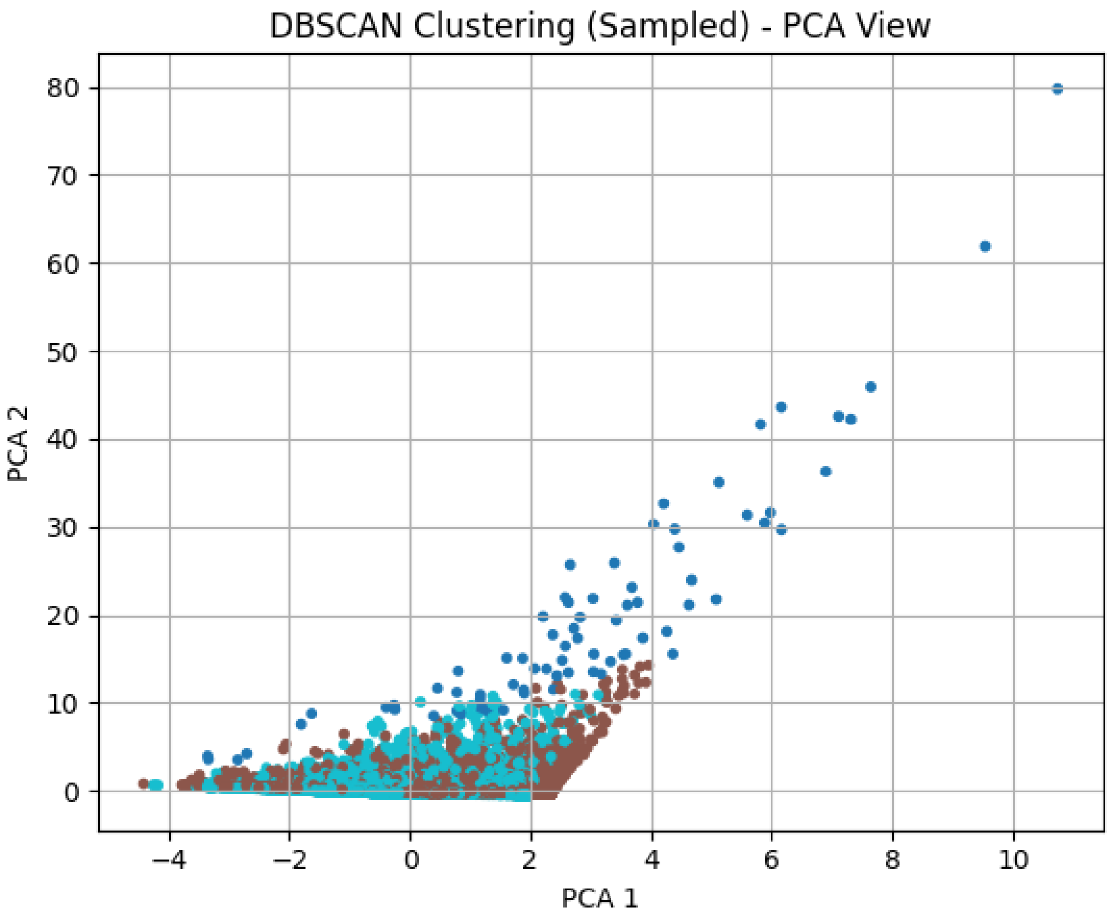
- Chosen k: 4 (based on Elbow)
- Visual Output: Clear and compact clusters in PCA space
- Evaluation:
    - Silhouette Score: **0.487** (good separation)
    - Davies-Bouldin Index: **0.824** (low intra-cluster variance)

Why it's best: K-Means aligned well with the data distribution. Clusters were balanced and separated, making the model robust and interpretable. The output is also deterministic and stable for forecasting applications.

Method 2: DBSCAN (Density-Based Spatial Clustering)

How it works: DBSCAN groups data based on density. It defines clusters as areas of high density separated by low-density regions. It does not require a pre-defined number of clusters, but is sensitive to two parameters:

- eps: Distance threshold for neighbourhood
- min_samples: Minimum number of points to form a dense region

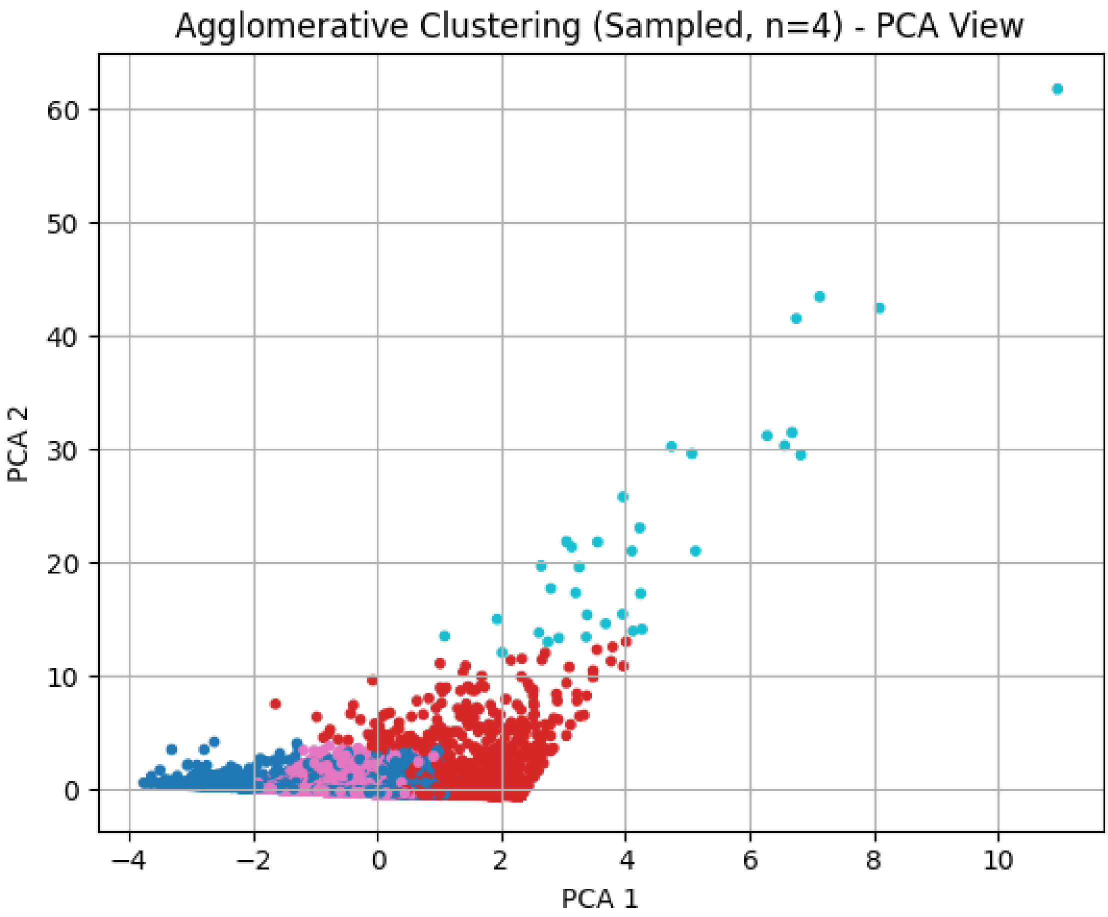


Result:

- Parameters: eps=1.2, min_samples=7
- Found clusters: Varies, includes noise points (label -1)
- Evaluation:
    - Silhouette Score: **0.324**
    - Davies-Bouldin Index: **1.252**
- DBSCAN is ideal for irregular cluster shapes and detecting noise. It may perform better with fewer dimensions or when the feature space is more clearly separated.

Method 3: Agglomerative Clustering (Hierarchical)

How it works: Agglomerative clustering builds a tree of clusters by recursively merging the closest pairs. It doesn't assume any specific shape or size of clusters but is computationally expensive.



Result:

- Chosen n_clusters: 4 (for comparison parity)
- Visual Output: Reasonably compact clusters
- Evaluation:
  - Silhouette Score: **0.417**
  - Davies-Bouldin Index: **0.953**

Limitations:

- Slow on large datasets
- Memory usage increases rapidly with n

It is useful when a hierarchical structure or dendrogram is desired, or when the number of clusters is small.

Challenges

DBSCAN

- Performance sensitive to parameter choice
- High-dimensional features may confuse density calculations
- Tends to misclassify border samples as noise

Agglomerative

- Slow on large datasets
- Memory usage increases rapidly with sample size

Final Comparison

Algorithm	Silhouette ↑	Davies-Bouldin ↓	Notes
K-Means	0.487	0.824	Best performer overall
DBSCAN	0.324	1.252	Noisy and unstable on high-dim
Agglomerative	0.417	0.953	Decent but resource-heavy

Conclusion

K-Means was the most effective method for clustering this 5G dataset. It produced compact, well-separated clusters that are easy to interpret and integrate into forecasting models. DBSCAN struggled with noise, and Agglomerative, while solid, had computational drawbacks.

The final K-Means-labelled output (kmeans_30k.csv) serves as a reliable input for the forecasting stage, supporting segment-wise analysis of 5G network performance across the city.