Addis Ababa University

College of Technology and Built Environment

School of Electrical and Computer Engineering

Telecommunication Network Engineering Graduate Program

Graph-based Approach for Efficient Tracking Area Design in Cellular Networks

By

Belayneh Taddesse WoldeSilassie

Advisor

Dr. -Ing. Dereje Hailemariam

A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Masters of Science in Telecommunication Network Engineering

June 2025

Addis Ababa, Ethiopia

Addis Ababa University

College of Technology and Built Environment

School of Electrical and Computer Engineering

Telecommunication Network Engineering Graduate Program

Graph-based Approach for Efficient Tracking Area Design in Cellular Networks

By

Belayneh Taddesse WoldeSilassie

Dr. -Ing. Dereje Hailemariam	____________
Advisor	Signature
____________________________	____________
Internal Examiner	Signature
____________________________	____________
External Examiner	Signature
____________________________	____________
Chairman	Signature

Declaration

I, the undersigned, declare that this thesis comprises my own work in compliance with internationally accepted practices. Except as referenced and acknowledged in the text, everything of the work contained within is my own original work and to the best of my knowledge no one submitted this work for any other degree or professional certification.

Belayneh Taddesse WoldeSilassie

____________________________ ___________

Signature

Advisor

____________________________ ___________

Dr. -Ing. Dereje Hailemariam Signature

June 2025

Addis Ababa, Ethiopia

Abstract

Modern cellular network technologies have significantly improved in terms of number and variety of connected devices, bandwidth, latency, and spectrum efficiency, especially compared to earlier generations. Despite these developments, cellular networks face significant challenges with respect to signaling overhead, most of which is associated with location management and user tracking. This signaling cost amounts to 1/3 of the total signaling cost at the Mobility Management Entity (MME) and consumes valuable communication resources, resulting in paging discards, handover failures, increased load on the MME, and increased power consumption at the User Equipment (UE).

To avoid these signaling overheads, traditionally, network operators use heuristic and distance-based approaches to partition their networks into location areas. However, these techniques often fall short in densely populated urban regions where there is high and dissimilar user mobility. On the other hand, connectivity-based clustering techniques such as graph-based clustering, if adopted properly, can offer an innovative solution to get an optimal design.

This thesis aims to implement graph-based techniques to determine the appropriate number of clusters that will help to significantly reduce the overall signaling overhead associated with paging and location update messages, and hence improve the overall performance of the network. To achieve this, cell location data, paging count, handover data, and key performance indicators (KPIs) are collected from Ethio telecom's Operation Support System (OSS). These datasets are analyzed using both distance- and graph-based clustering algorithms, namely, K-Means, Spectral Clustering, and Markov Clustering (MCL) techniques, to simulate and predict the mobility and density information of users.

To evaluate the proposed approach, a real-world LTE network deployed in the city of Addis Ababa was used as a source of data and as a baseline. The evaluation results show that, MCL reduces the Paging and Tracking Area Update (TAU) signaling overheads by 24.04% and 29.85%, respectively; while reducing the total signaling overhead by 27.87%, compared to the current ground truth configuration, improving network performance and efficiency. Additionally, this thesis advances the field by adapting MCL algorithm to optimal TA design and optimization, which may find use in next-generation and other wireless networks, which can also incorporate other networks' data to get a more accurate result.

Keywords – Cellular Network, LTE, 5G NR, Mobility Management, Signaling Overhead, Location Management, Tracking Area, Tracking Area Update, Paging, Handover, Mobility Analysis, MCL, Spectral Clustering, K-Means, Graph-based Clustering.

Acknowledgment

First and foremost, I would like to thank the Almighty God and His mother St. Mary, for everything that has happened and for everything that hasn't.

Next, I want to express my sincere gratitude to everyone who has mentored and supported me. Especially, my heartfelt gratitude goes to my advisor, Dr. -Ing. Dereje Hailemariam, whose advice, in-depth knowledge, tolerance, and understanding has been crucial in my studies generally and in this thesis specifically. My work has been greatly impacted by your guidance and unwavering support. My dear wife, Sr. Sheshayit WoldeGebriel, whose unwavering love, support, and sacrifices has always been the source of my motivation has my sincere gratitude as well. Having you by my side is a true blessing.

I want to thank my father for inspiring me to pursue my master's degree since your faith in my abilities has always been a driving force behind my academic achievements. Thank you for instilling the importance of education in my earlier ages and for being my constant source of inspiration later on.

Last but not least, I would like to acknowledge Addis Ababa University and Ethio telecom for sponsoring my studies. I am truly grateful for the excellent faculty, academic environment, and resources of AAU and for the generosity and financial support of Ethio telecom.

Finally, without the support and encouragement of each one of you, this accomplishment would not have been possible. Thank you!

Table of Content

List of Figures

List of Tables

List of Acronyms

2D	Two-Dimensional
3G	Third-Generation
3GPP	3rd Generation Partnership Project
4G	Fourth-Generation
5G	Fifth-Generation
5GC	5G Core
AMF	Access and Mobility Management Function
BS	Base Station
Cell ID	Cell Identification
CN	Core Network
CS	Circuit-Switched
CSV	Comma-Separated Values
eNodeB	Evolved NodeB
EPC	Evolved Packet Core
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
IoT	Internet of Things
IP	Internet Protocol
KPI	Key Performance Indicators
LA	Location Area
LAC	Location Area Code
LAI	Location Area Identity
LAU	Location Area Update
LTE	Long-Term Evolution
MCC	Mobile Country Code
MCL	Markov Clustering
MME	Mobility Management Entity
mMTC	Massive Machine-Type Communications

MNC	Mobile Network Code
MSC	Mobile Switching Center
NAS	Non-Access Stratum, protocol used for signaling between UE & MME
NP-hard	Nondeterministic Polynomial Time-hard
NR	New Radio
OSS	Operation Support System
PDCCH	Physical Downlink Control Channel
PGW	Packet Data Network Gateway
PRB	Physical Resource Block
PS	Packet-Switched
Q	Modularity
QoS	Quality of Service
RA	Registration Area
RAN	Radio Access Network
RO Traff Vol	Regulatory Oversight Traffic Volume
RRC	Radio Resource Control
S	Silhouette Score
S1-AP	S1 Application Protocol
SGW	Serving Gateway
TA	Tracking Area
TAC	Tracking Area Code
TAI	Tracking Area Identity
TAL	TA List
TAU	Tracking Area Update
UE	User Equipment
UL/DL PS Traf	Uplink/Downlink Packet Switched Traffic
UMTS	Universal Mobile Telecommunications System
URLLC	Ultra-Reliable Low-Latency Communications
VLR	Visitor Location Register

Chapter 1. Introduction

1.1 Background

In cellular communications, the concepts of Location Area (LA), Tracking Area (TA), and Registration Area (RA) in third-generation (3G), fourth-generation (4G), and fifth-generation (5G) networks, respectively, are fundamental to mobility management, enabling networks to track user equipment (UE) for paging and efficient resource allocation. This method is a logical grouping of nodes and it allows the network to locate a UE without sending constant signaling messages and to balance mobility tracking with signaling overhead [34,35, 36].

This location management is done by using the following two main processes:

- **Location Updating**: where a UE informs the network about the whereabouts of its location so as to maintain connectivity to the network. Location updating is done either periodically or upon movement from one node to another.
- **Paging**: where the network sends messages within a defined area to locate a UE for incoming calls or data.

These processes have evolved across generations of cellular network technologies so as to address the increasing demand for efficiency, accuracy and support for diverse applications [34, 35, 36].

1.1.1 Location Area

In Universal Mobile Telecommunications System (UMTS) networks, LA is a group of cells defined by the network operator to manage UE mobility in both circuit-switched (CS) and packet-switched (PS) domains of the network. The LA is identified by a Location Area Identity (LAI). LAI consists of Mobile Country Code (MCC), Mobile Network Code (MNC), and Location Area Code (LAC) [34].

Location Area Update (LAU) message is initiated when a UE moves from its current LA to a new LA to inform the network of its new location. This update is critical for later paging operations, as the network sends paging messages to all nodes within the UE's registered LA when an incoming call or data session is initiated.

The LA concept is designed to minimize signaling overhead in 3G networks. However, frequent LAUs in dense or high-mobility networks can increase signaling load [34].

1.1.2 Tracking Area

In Long-Term Evolution (LTE) networks, the TA replaces the LA to support the all-IP architecture and Evolved Packet Core (EPC). A TA is a set of nodes identified by Tracking Area Identity (TAI). TAI includes MCC, MNC, and Tracking Area Code (TAC). The TA is used by the Mobility Management Entity (MME) to track the location of idle UE's [35].

When a UE moves from its current TA to a new TA, it initiates a Tracking Area Update (TAU) procedure to register its new location with the MME. This enables the network efficiently page the UE for incoming data or control signaling.

TAs can be also grouped into TA lists (TAL) to reduce signaling further by allowing UEs to move within multiple TAs without the need to trigger a TAU. TAL improves efficiency further in high-mobility scenarios [35].

1.1.3 Registration Area

In 5G New Radio (NR) networks, the RA serves a similar role in mobility management. It is defined as a set of TAIs managed by the Access and Mobility Management Function (AMF). The RA is used to track UEs in the 5G Core (5GC) network, particularly in the Radio Resource Control (RRC) [36].

The RA concept is similar to the TA in 4G but is enhanced to support 5G's diverse use cases, such as ultra-reliable low-latency communications (URLLC) and massive machine-type communications (mMTC).

When a UE moves from its current RA to another, it performs a Registration Update to notify the AMF of its new location. The RA allows flexible configuration which enables operators to define larger or smaller RAs based on network topology, UE mobility patterns, and service requirements. The use of TAL in 5G further reduces signaling overhead by allowing UEs to move within multiple RAs without the need for frequent updates [36].

In general, LA, TA, and RA serve similar purposes. The LA in 3G supports dual-domain (CS/PS) systems, suitable for voice and early data services. The TA in 4G is optimized for an all-IP EPC that focuses on data-driven services with improved scalability. The RA in 5G builds on the TA concept which offers greater flexibility for 5G's service-based 5GC, including low-latency applications and Internet of Things (IoT). Each successive generation reduces signaling overhead that helps the network to adapt to the increasing demand for mobility and UE density. The transition from LA to TA to RA also reflects advancements in core network functions, from the 3G Mobile Switching Center (MSC)/Visitor Location Register (VLR) to the 4G MME and 5G AMF [34, 35, 36].

1.2 Problem Statement

Inefficient location area design in cellular networks leads to excessive signaling overhead. Studies show that signaling overhead caused by location management accounts to one-third of the overall signaling overhead in an LTE system [3, 5, 6, 14]. This increase in signaling overhead causes, besides the depletion of valuable communication resource, an increase in paging discards, i.e., a decrease in paging success rate, an increase in MME load, an increase in handover latency, and an increase in UE power consumption caused by frequent location updates. Various studies have been conducted to reduce this signaling overhead caused by paging and location update [3, 6, 7, 13, 14, 18]. However, current methods lack systematic integration of massive operational data, such as, handover and paging, and fail to adapt to varying-density network topologies and UE mobility patterns. Hence, the need for a robust, graph-based framework that optimizes TA configurations using real-world-inspired data.

1.3 Research Objectives

1.3.1 General Objective

The main objective of this research is to design an optimal tracking area based on subscriber density and mobility patterns derived from real-world paging and handover data and by using graph-based clustering algorithms so as to alleviate the signaling overhead in Ethio telecom's cellular network of the city of Addis Ababa.

1.3.2 Specific Objectives

The specific objectives of the research are:

- To collect and preprocess extensive real-world data (paging count, handover data, and key performance indicators (KPIs)) from Ethio telecom's Operation Support System (OSS);
- To develop a system model integrating Paging Count, Handover Count, KPIs and Cell Locations as input data for TA design;
- To implement distance-based K-Means, and graph-based Spectral Clustering and MCL algorithms in Python to cluster evolved Node Bs (eNodeBs) into optimal TAs;
- To demonstrate how these clustering techniques can effectively minimize signaling overhead and improve the accuracy of user tracking;
- To evaluate and compare the performance of these algorithms against baseline configurations in terms of cluster quality and signaling cost;
- To present intuitive visuals that help the assessment of the clustering results;
- To identify the most effective clustering method for tracking area design in cellular networks;
- To identify findings, discuss the applicability of the proposed solutions, highlighting their potential impact on current and future generation cellular networks, and suggest recommendations for future work;

1.4 Literature Review

The most important benefit of optimal TA planning is preventing unnecessary signaling radio resource utilization that can instead be allocated for the communication of subscribers [13]. According to various studies, signaling load associated with tracking UEs amounts up to *one-third* of the total signaling load on the MME [3, 5, 6, 14]. Network performance can be improved by decreasing this signaling overhead caused by TAU and paging messages.

TAU and paging messages are inversely proportional and are greatly affected by the size of TA. However, a trade-off between these two messages can be made by employing various machine learning algorithms [13]. Several studies have addressed TA optimization in LTE networks using clustering techniques.

In [3, 13, 14], distance-based clustering algorithms such as K-means and density-based algorithms such as Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) are used to decide on the optimal number of TAs. In [3], the author used K-means algorithm to partition a network of 650 eNodeBs into 74 TAs and improved the total signaling overhead by 49.51%. This result further improved into 59.54% with 237 TAs by employing density-based algorithm, namely HDBSCAN. However, this study does not perform customer profiling based on usage behavior to take traffic volume into consideration.

In [7], user mobility is analyzed by using a technique of two-dimensional (2D) Markov model. This technique is used to calculate TAU and for estimating the number and next location probability of users in a cell within a time slot. However, the simulations are implemented on greenfield networks, with gross assumptions and with minimal sample data as an input to the optimization algorithm.

Despite advances in various methods, there still persist several gaps in the research area. One biggest shortcoming from most researches is lack of extensive mobility and density data extracted from real-world networks. There is also a lack of comparative study. For example, hardly any works compare distance-based K-Means, Spectral Clustering, and MCL under identical LTE conditions; i.e., same dataset and metrics.

The Paging-TAU signaling trade-off also lacks systematic analysis across clustering methods, particularly in networks with varying mobility patterns.

MCL-based TA design, though not directly studied, showed promise in other networks. It has a potential for utilizing connectivity patterns effectively [31]. However, little research has been conducted to explore the performance of graph-based methods, such as Spectral Clustering and MCL, in large-scale cellular network deployments [19].

These limitations and research gaps highlight the importance of integrating spatial, topological, and operational data into clustering algorithms. This thesis addresses these gaps by implementing and comparing these algorithms against baselines, using real-world-inspired data to establish their efficacy in reducing signaling overhead.

1.5 Scope and Limitations

This research focuses on graph-based tracking area design for cellular networks using paging count and handover data and the subscriber density and mobility patterns derived from them. However, due to unavailability of operational data, the simulation is done on LTE network only. Comparison of various techniques for optimal TA design other than the mentioned location-based and graph-based techniques is beyond the scope of this work.

1.6 Contributions of the Study

This research will contribute to cellular network optimization by providing a data-driven TA design methodology based solely on paging count and handover data, potentially reducing signaling overhead by $25 - 30\%$. It will offer insights to the performance benefits and limitations of using various location and graph-based clustering approaches in cellular location area design and in optimizing the trade-off between signaling overhead, paging delay, handover success rate and user experience so as to benefit network operators and pave the way for extensions to future generation cellular systems.

1.7 Thesis Organization

This thesis is organized in six chapters. Chapter 1 outlines the background of the thesis. It also describes the problem statement, the research objectives, and the literature review. Chapter 2 details the foundations and backgrounds of location management and location area design. Chapter 3 provides the theoretical aspects behind graph-based clustering. In Chapter 4, we outline the methodology used to the desired objectives and the performance evaluation techniques employed and the parameters used in this study. Chapter 5 presents the results and discussions of the experiments conducted. And finally, Chapter 6 concludes the research with recommendations and directions for future work.

Chapter 2. Mobility Management in LTE

Global cellular data traffic, driven by the proliferation of smartphones, IoT devices, and emerging applications, such as augmented reality, is projected to reach 303 Exabyte per month by 2030 [1]. According to Ericsson's Mobility Report projection, significant portion of these traffic passes through LTE networks [1]. This surge in data demand and the ever increasing UE density and mobility places immense pressure on the performance of cellular networks. Hence, efficient mobility management is critical [8].

2.1 Architecture of LTE Networks

The architecture of LTE networks includes several key elements that include UE, eNodeBs, and EPC, as shown in Figure 2-1 below [17].

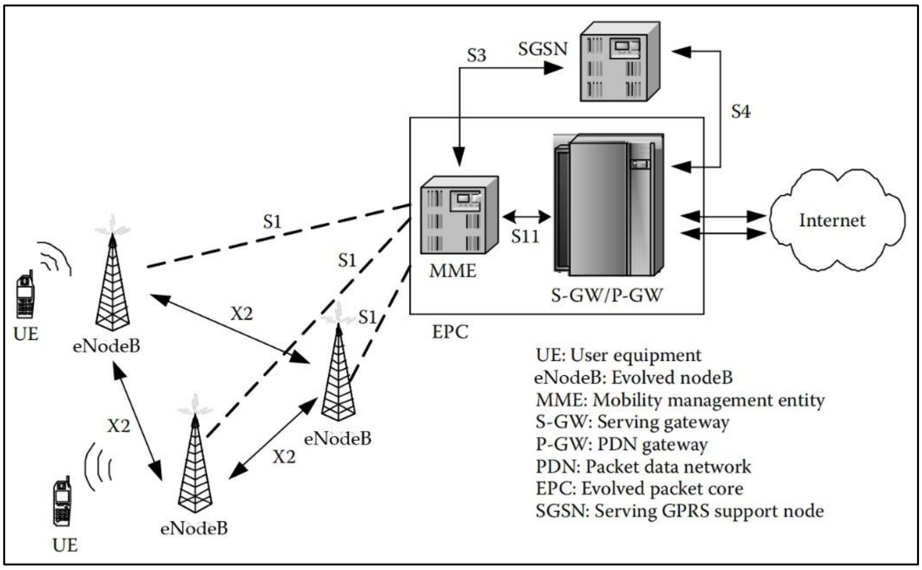

Figure 2 - 1: The LTE architecture [17].

UE represents end-user devices, such as smartphones and tablets, while eNodeB serves as the Base Station (BS) responsible for enabling radio communication with the UE. Radio Access Network (RAN) consists of multiple eNodeBs, which are deployed throughout a coverage area, and is connected to EPC via backhaul links [16].

The EPC serves as the backbone of the LTE network. It is composed of several key functional entities that collectively facilitate essential tasks, such as network control, management, and data routing. The EPC consists of three primary components, namely MME, Serving

Gateway (S-GW), and Packet Data Network Gateway (P-GW), which work together to facilitate signaling and data routing.

The MME, which is the main control element in the EPC, handles the mobility management by keeping track of UE locations and facilitating handovers [16]. Unlike its predecessor cellular systems, LTE adopts a TA-based approach to reduce signaling load [16, 35].

2.2 Mobility Management

Each cellular network has hundreds to thousands of eNodeBs deployed across a geographic region. And each eNodeB provides wireless coverage to a number of UEs. Monitoring the location of a UE and knowing where the subscriber is located at any given time is an essential part of the wireless network service which allows the wireless network to locate the subscriber when a connection request arrives at the switching center.

The MME manages UE mobility via the following RRC states:

- RRC_CONNECTED (active), in this state the UE is actively connected to the network and can send and receive data with dedicated radio resources allocated. The network also knows the UE's exact cell location.
- RRC_IDLE (idle), in this state the UE is not actively transmitting data but is registered with the network and is monitoring paging channels for incoming calls or data. The network tracks the UE at a TA level [8].

2.3 Tracking Area in LTE Networks

TA is a logical grouping of eNodeBs, designed to optimize signaling overhead that arises from TAU and Paging procedures [8]. The figure below shows N TAs, each having a number of eNodeBs, and each eNodeB having three cells. The TAs are indicated by different shades of colors. Every time a UE moves across the TAs, it notifies the MME in the LTE network of its current location by sending a TAU message (TAU Request message) [4].

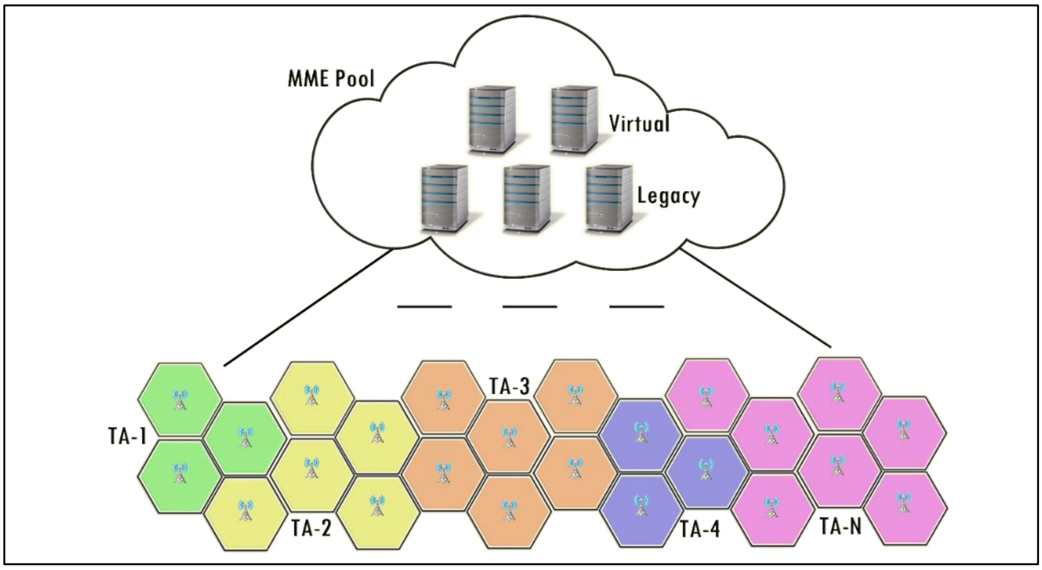

Figure 2 - 2: TA concept in LTE.

A TA is identified by a unique TAI. When a user is in an *active* state, the location of the UE is known at the cell level, as the network needs to quickly react to UE cell change. However, when a UE is in an *idle* state, the LTE network (the MME, to be accurate) knows its location only at the TA level, paging all cells within that TA to re-establish a connection [12]. TA design thus influences two key signaling processes, *Paging and TAU.*

- **Paging**: Initiated when the network needs to locate a UE within its TA, broadcasting messages across all nodes in that TA. The cost scales with TA size (number of eNodeBs) and paging rate.

  The paging process involves:

  - MME sends a Paging Request to eNodeBs via the S1-AP interface.
  - eNodeBs transmit RRC Paging messages on the Physical Downlink Control Channel (PDCCH).

  The Paging cost per TA is:

$$C_{Paging,i} = |TA_i| \cdot R_{Paging} \cdot S_{Paging} \qquad \text{(Eq. 2.1)}$$

  Where:

  - $|TA_i|$ is the number of nodes in TA i,
  - R_{Paging} is the paging rate (events/hour),
  - S_{Paging} is the signaling load per message.

- **TAU**: is triggered when a UE detects a new TAI outside its current TAL, requiring signaling to update the MME [8]. The cost depends on UE mobility and TA boundary crossings.

  The TAU triggers the following processes:

  - RRC Connection Request to the new eNodeB.
  - NAS TAU Request to the MME via S1-AP.
  - MME updates the UE's location and sends a TAU Accept.

  The TAU cost is:

$$C_{TAU} = \sum_{e \in E_{cut}} W_e \cdot S_{TAU} \qquad \text{(Eq. 2.2)}$$

Where:

- E_{cut} is the set of edges (handovers) between TAs,
- W_e is the handover count,
- S_{TAU} is the signaling load per message.

The design of TAs directly influences these signaling costs. Generally, larger TAs increase Paging overhead due to more cells being paged at a time, which in turn may result in slower call establishment time and inefficient resource usage. On the other hand, smaller, fragmented TAs raise TAU frequency due to excessive boundary crossings, which in turn results in handover delays, and frequently interrupted services [2, 8, 12, 24].

In addition to this, signaling overhead, if not managed properly, will be exasperated by high user mobility and increased number of cells in the network. Studies estimate that suboptimal TA design, *either excessively large TAs inflating Paging costs or fragmented TAs increasing TAU frequency*, can elevate signaling overhead by $20\ to\ 30\ \%$, straining the MME and degrading Quality of Service (QoS) [18].

On the other hand, a reduced signaling overhead has multiple benefits. It provides the network with significant amount of additional communication resources, reaped from the signaling overhead. It also helps to improve the performance of the network by decreasing paging discards and loading of the MME. It even decreases the power consumption at the UE side by decreasing the frequency of paging and TAU messages.

2.4 Signaling Overhead in LTE Networks

As discussed above, signaling overhead in LTE networks arises primarily from the two processes, *Paging and TAU* [12]. These signaling overhead poses significant challenge in cellular networks, particularly in dense urban areas or high-mobility scenarios such as large open markets and highways.

According to [3, 5, 6, 14], a substantial amount of the signaling overhead stems from user location management related messages, i.e., paging and TAU messages as illustrated in Figure 2-3 below.

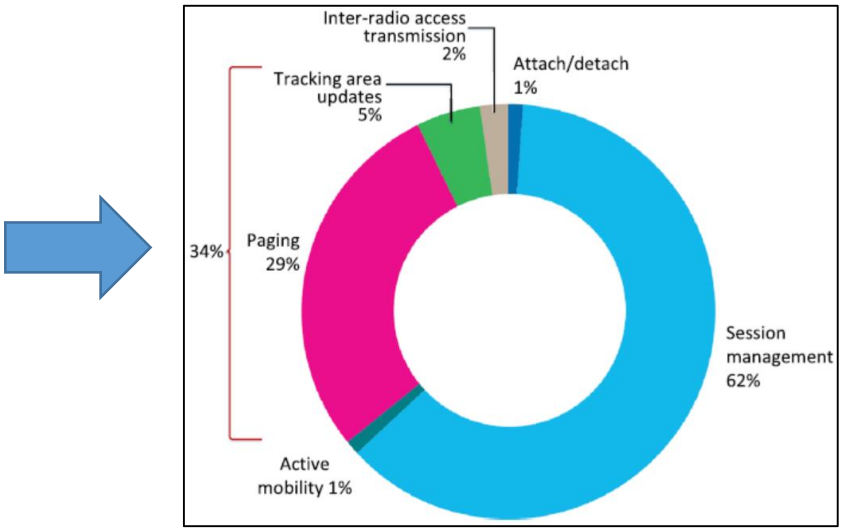

Figure 2 - 3: Distribution of MME signaling events [5].

Figure 2-3 shows paging and TAU messages comprise 29% and 5%, respectively, of the total signaling events in the MME. These two are the main components of the location management related signaling at the MME.

2.5 Baseline Implementations for TA Design

Baseline TA designs provide benchmarks for evaluating clustering algorithms:

- **Single TA Configuration**: The simplest baseline where all eNodeBs form one TA, eliminating TAU but maximizing Paging cost.

- **No TA Configuration**: The opposite of single TA configuration, where each eNodeB is a TA by itself. It decreases the paging cost to the minimum; however, the TAU cost will be the maximum.
- **Manual TA Planning**: TAs are defined by heuristics algorithm or operator knowledge, often suboptimal due to static assumptions and TAs grow following the network's expansion [15]. Real-world implementations show $10 - 20\%$ higher signaling overhead than optimized designs [24].

These baselines underscore the importance of adopting intelligent, data-driven methods that exploit both network connectivity and user mobility patterns to reduce overall signaling costs. One such approach is graph-based clustering wherein eNodeBs are represented as nodes and their interconnections as edges to achieve efficient TA design.

2.6 Need for Optimization

Traditional TA planning often depends on manual configurations or fixed geographic boundaries without the consideration of dynamic network characteristics such as handover behavior and paging loads [15]. However, graph-based approaches, where eNodeBs are represented as nodes and their connectivity relationships as edges, offer an alternative that better aligns TA boundaries with actual network topology and UE mobility patterns [21].

This thesis examines various clustering algorithms, *including both distance- and graph-based techniques,* to optimize TA design. It further evaluates and compares these methods against each other and against baseline clusters, to identify the most optimal approach.

Chapter 3: Graph-based Clustering

Graph-based clustering is a network analysis technique designed to partition a graph's nodes into cohesive groups or meaningful clusters based on the relationship or connectivity between its elements. In this context, a graph $G = (V, E)$ consists of a set of vertices V, that represent nodes in a cellular network, and a set of edges E, that represent handover relationships. Edges are often weighted to capture the strength of the interaction between nodes [21].

In the context of cellular networks, graph clustering optimizes TA design by grouping cells into areas that minimize signaling overhead caused by paging and location update messages. This chapter explores the theory, algorithms, and applications to cellular network of graph clustering, focusing on Spectral Clustering and MCL, and distance-based K-Means, which are the methods employed in this research to process *Cell Locations, Paging Counts,* and *Handover Counts* data.

3.1 Fundamentals of Graphs and Clustering

Clustering in graphs differs from traditional data clustering by emphasizing topological structure over Euclidean proximity. For instance, two eNodeBs with high handover frequency should ideally belong to the same TA to reduce TAU costs, even if they are geographically distant [21].

A graph can be represented as either an *adjacency matrix* or an *edge list*. Adjacency matrix is a square matrix where each row and column represents a node in the graph. The values in the matrix indicate the presence or strength of connections between nodes. It can be represented as $A \in R^{n \times n}$, where A_{ij} is the weight or handover count of edge (i, j), and $A_{ij} = 0$ if no handover exists between nodes i and j [21].

Similarly an *Edge List* is a list of tuples (i, j, A_{ij}) for all edges. In cellular networks, V represents eNodeBs, and A_{ij} denotes handover frequency, while Cell Locations provide spatial attributes [21].

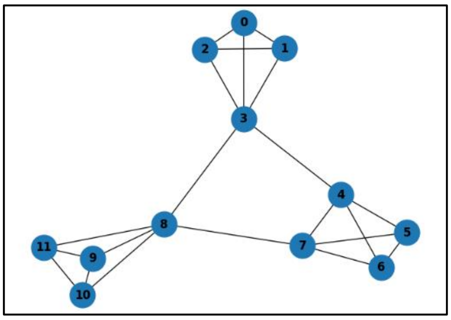

Figure 3 - 1: Unweighted sample graph, G.

The sum of edge weights between clusters is called the Cut Size. A clustering is said to be good if it effectively minimizes cut size, $Cut(C_1, C_2) = \sum_{i \in C_1 . j \in C_2} W_{ij}$, which in turn yields a reduced TAU. In other words, a good clustering should maximize intra-cluster edges while minimizing inter-cluster ones.

With respect to metrics, common objectives of graph clustering include increasing the modularity and silhouette score.

Silhouette Score (S) measures cohesion, i.e., how similar a node is to its own cluster versus neighboring clusters.

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \qquad \text{(Eq. 3.1)}$$

Where:

- $a(i)$ is intra-cluster distance,
- $b(i)$ is inter-cluster (nearest-cluster) distance [32].

Modularity (Q) measures the strength of division by assessing the graph partitioning.

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \qquad \text{(Eq. 3.2)}$$

Where:

- A is the adjacency matrix of the graph,
- k is degree of nodes,
- m total edges,
- (c_i, c_j) clusters, and
- $\delta(c_i, c_j) = 1$ if nodes i and j are in the same cluster, 0 otherwise [27].

However, these metrics should always be used in conjunction with each other to get a comprehensive view of clustering performance.

Generally, TAs should minimize Paging and TAU costs. These objectives can be achieved by using graph clustering which enables TAs to be aligned with handover patterns (edges or connectivity) and helps the TA size (node count) to be controlled [24].

3.2 Graph Clustering Techniques

This section details the algorithms used in this research, their mathematical foundations, and their application to TA optimization.

3.2.1 Distance-based K-Means Clustering

Distance-based K-Means clustering is a popular, centroid-based, unsupervised machine learning algorithm that partitions n data points into k distinct clusters based on Euclidean distance, making it suitable for geographical grouping of eNodeBs [33]. It works by iteratively assigning elements to the nearest centroid and updating the centroids based on the mean value of the assigned elements.

Distance-based K-Means clustering is efficient and widely used for its simplicity and effectiveness. In the context of TA design, K-Means can minimize Paging overhead by forming compact, spatially cohesive clusters.

The algorithm's objective function is:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2 \qquad \text{(Eq. 3.3)}$$

Where:

- C_i is the i^{th} cluster,
- x is a data point, such as eNodeB coordinates (x, y), and
- μ_i is the centroid of cluster C_i [33].

The algorithm iterates:

1. **Initialization**: Choose k initial centroids randomly from the data points. These centroids represent the centers of the clusters.

2. **Assignment**: Each data point is assigned to the nearest centroid based on a distance metric, usually Euclidean distance; $c(x) = argmin_i|x - \mu_i|^2$. This creates k clusters.
3. **Update**: For each cluster, compute the new centroid by taking the mean of all data points assigned to that cluster. $\mu_i = \frac{1}{|c_i|}\sum_{x \in C_i} x$.
4. **Iteration**: Repeat the assignment and update steps until the centroids no longer change significantly, i.e., until convergence is achieved, or until a predefined number of iterations is reached, i.e., until J stabilizes [20].

Algorithm: Distance-based K-Means Clustering

```
Input: Dataset X (n points, each with d features), number of clusters k,
       max_iterations, threshold ε
Output: Cluster labels for n points, k centroids
1. Initialize centroids:
   Choose k points as initial centroids μ_1, μ_2, ..., μ_k
2. Initialize labels:
   labels = array of size n, initially empty
3. For iteration = 1 to max_iterations:
   Assign points to clusters
   For each point x_i in X:
   Compute distance d_ij = |x_i − μ_j|^2 for all centroids μ_j
     labels[i] = j where d_ij is minimized
   Update centroids
   For each cluster j:
   C_j = {x_i | labels[i] = j}
   If C_j is not empty:
     μ_j = mean(C_j) (Average of points in C_j)
   Else:
     μ_j =random point or unchanged (Handle empty clusters)
   Check convergence
   If max(|μ_j − μ_j_old|) < ε for all j:
   Break
4. Return labels, centroids
```

Distance-based K-Means is computationally efficient. It has a complexity of $O(nkd)$ per iteration, where n is the number of eNodeBs, k is the desired number of clusters, and d is the dimensions of the input data (2 for coordinates). Its simplicity, scalability, speed, and intuitive nature for spatial data makes it a practical baseline [20]. However, its sensitivity to initialization, need for pre-specified k, ignorance to handover/edge/connectivity data, *potentially increasing TAU costs by splitting high-mobility regions in networks with irregular mobility patterns,* and tendency to work best only in isotropic clusters, *which may not always be the case*, misaligns with cellular network's topological needs [20].

3.2.2 Spectral Clustering

Spectral Clustering is a graph-based technique used for clustering data points based on their connectivity. It leverages the spectral properties, *eigenvalues and eigenvectors,* of the graph's Laplacian matrix to embed nodes into a lower-dimensional space, where traditional clustering algorithms like distance-based K-Means can be applied. It is particularly effective for identifying clusters in non-convex shapes and complex structures in high-dimensional spaces [25].

Spectral Clustering begins by representing the data as a graph, $G = (V, E)$, where each data point is a node, and edges represent similarities between points [25]. An adjacency matrix, A, *often called the affinity/similarity matrix,* is constructed, where $A_{ij} = 1$ if there is an edge between nodes i and j, and $A_{ij} = 0$ otherwise.

For simplicity, we assume an unweighted graph here, but the method extends to weighted graphs by setting A_{ij} to the edge weight [30].

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The adjacency matrix A for this graph G is:

- $A_{12} = A_{21} = 1$ indicates an edge between nodes 1 and 2.
- $A_{34} = A_{43} = 1$ indicates an edge between nodes 3 and 4.
- All other entries are 0, reflecting no additional edges.

Next, the degree matrix, D, is computed. The degree matrix is a diagonal matrix where each diagonal entry D_{ii} is the degree of node i, calculated as the number of edges incident to node i. Mathematically, $D_{ii} = \sum_j A_{ij}$.

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Each node has a degree of 1 (since each is connected to exactly one other node):

- Node 1: degree = 1 (connected to node 2)
- Node 2: degree = 1 (connected to node 1)
- Node 3: degree = 1 (connected to node 4)
- Node 4: degree = 1 (connected to node 3)

From the adjacency and the degree matrixes we derive the Laplacian Matrix, $L = D - A$, which represents the structure of the graph. The diagonal entries of L represent node degrees, and the off-diagonal entries (negative) indicate connections between nodes.

$$L = D - A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Next, compute the eigenvalues and eigenvectors of the Laplacian Matrix, L, (eigenvalue decomposition). Since, the Laplacian is symmetric and positive semi-definite, its eigenvalues are real and non-negative ($\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$), and the smallest eigenvalue is 0 for a connected graph.

Then select the k eigenvectors corresponding to the k smallest eigenvalues, where k is the desired number of clusters. Here, the smallest eigenvalues relate to the graph's connectivity, and their eigenvectors reveal nodes that are loosely connected across clusters but tightly connected within them.

For the above example, since the graph has two disconnected components, the Laplacian is block-diagonal, with each block corresponding to a pair of nodes:

$$L = \begin{bmatrix} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \end{bmatrix}$$

For each 2×2 block $\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$, the characteristic equation is:

$$det\left(\lambda I - \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\right) = det\begin{bmatrix} \lambda - 1 & 1 \\ 1 & \lambda - 1 \end{bmatrix} = (\lambda - 1)^2 - 1 = \lambda(\lambda - 2)$$

Eigenvalues are $\lambda = 0$ and $\lambda = 2$. Thus, for the full 4×4 L, the eigenvalues are $0, 0, 2, 2$, each with multiplicity 2 due to the two components.

For $\lambda = 0$, the eigenvectors span the space of the connected components:

- $u_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, constant on nodes 1 and 2
- $u_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$, constant on nodes 3 and 4

Since we want $k = 2$ clusters, we select the eigenvectors corresponding to the two smallest eigenvalues (both 0) and form a matrix $U \in R^{n \times k}$ with these eigenvectors as columns, where n is the number of eNodeBs and k the number of clusters.

Each node i is represented by a k-dimensional vector *–the i^{th} row of the eigenvector matrix.* This embedding transforms the graph into a space where clusters are more separable, as nodes in the same cluster tend to have similar eigenvector values.

The eigenvector matrix, U, with the two eigenvectors as columns is,

$$U = [u_1 \quad u_2] = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

A standard clustering algorithm, such as K-Means, is then applied to these new points (U) to form clusters [30]. Treat each row of U as a node in R^k. Apply K-Means clustering algorithm to these n points to partition them into k clusters. The resulting clusters correspond to groups of nodes in the original graph.

In the example above,

- Node 1: (1,0)
- Node 2: (1,0)
- Node 3: (0,1)
- Node 4: (0,1)

Applying k-means with $k = 2$ to these points:

- (1,0) and (1,0) are identical, forming one cluster, Cluster 1: {nodes 1 and 2}.
- (0,1) and (0,1) are identical, forming another cluster, Cluster 2: {nodes 3 and 4}.

This matches the graph's structure, where {1,2} and {3,4} are the two disconnected components.

Thus, Spectral Clustering minimizes inter-TA handovers, reducing TAU cost, by using handover count between $eNodeB_i$ and eNodeB_j as A_{ij}.

Algorithm: Spectral Clustering

```
Input: Adjacency matrix A(n × n), number of clusters k
Output: Cluster labels for n nodes
1. Construct similarity matrix:
   Set S = A; use edge weights, e.g., handover counts
   Ensure S is symmetric: S_ij = S_ji
2. Compute degree matrix:
   D=diagonal matrix, D_ii = Σ_j A_ij
3. Form Laplacian:
   L = D − A
4. Compute eigenvectors:
   Find k smallest non-zero eigenvalues of L and their eigenvectors u_1, u_2, ..., u_k
   Form U = [u_1, u_2, ..., u_k]  (n × k matrix)
5. Cluster in eigenvector space:
   Apply K-Means to rows of U to get k clusters
   Assign each node i to cluster based on K-Means labels
6. Return cluster labels
```

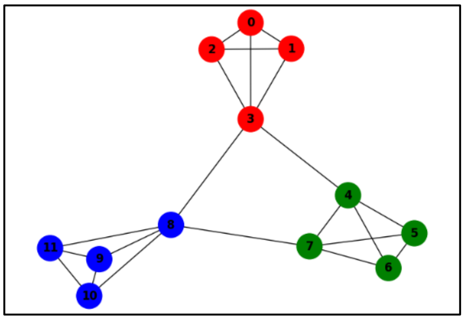

Figure 3 - 2: Communities of a Spectral Clustering.

Spectral Clustering's flexibility in capturing complex connectivity structures that are not well-suited to traditional clustering methods, its noise resistance, and robustness to non-spherical clusters makes it suitable for graph clustering. However, it has also some notable drawbacks. For instance, it requires specifying the number of clusters k, struggles with sparse graphs, and is computationally intensive due to eigenvalue decomposition with a complexity of $O(n^3)$ [29].

3.2.3 Markov Clustering

MCL, is an unsupervised graph-based clustering algorithm that identifies clusters by simulating random walks on a graph. It uses Handover Counts to form TAs with strong internal connectivity, reducing TAU. This is achieved through an iterative process of two operations –*expansion and inflation* [31]. The inflation parameter controls cluster granularity by amplifying flow within clusters and weakening it between them, *strengthening strong flows and weakening weak ones*, in other words suppressing noise, making it adaptable to varying network densities [28, 31].

For example, let's take the following graph with 6 nodes, designed to have two distinct clusters connected by a weak link.

- Nodes 1, 2, 3 form a triangle (all pairs connected).
- Nodes 4, 5, 6 form another triangle.
- An edge between node 3 and node 4 connects the two triangles.

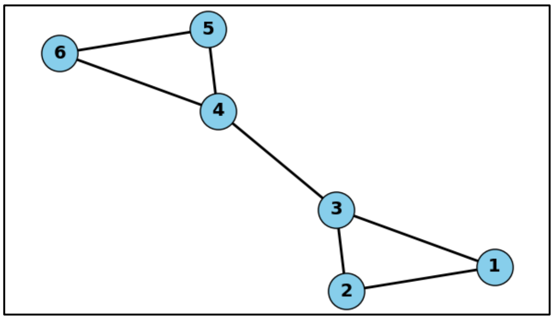

Figure 3 - 3: Graph G with 6 nodes.

The MCL algorithm starts by representing the network with an undirected graph, G, and constructing the adjacency matrix A, where $A_{ij} = 1$ if there is an edge between nodes i and j, and $A_{ij} = 0$ otherwise. For weighted graphs, A_{ij} represents edge weights.

The adjacency matrix of graph G is,

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

- $A_{12} = A_{13} = A_{23} = 1$ (triangle 1-2-3)
- $A_{45} = A_{46} = A_{56} = 1$ (triangle 4-5-6)
- $A_{34} = A_{43} = 1$ (bridge between clusters).

It is also common practice to add a self-loop to every node. This ensures that a random walk has a non-zero probability of remaining at its current node. It also helps to avoid issues with certain graph structures, such as bipartite graphs.

The next step is to compute the degree matrix D, a diagonal matrix where D_{ii} is the degree of node i, i.e., the number of edges incident to i, $D_{ii} = \sum_j A_{ij}$. Using this, a transition matrix, $M = D^{-1}A$, where $M_{ij} = A_{ij}/D_{ii}$ is constructed. M represents the probability that a random walk moves from node i to node j. And since each row sums to 1, M is row-stochastic.

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

- Node 1: degree 2 (connected to 2, 3)
- Node 2: degree 2 (connected to 1, 3)
- Node 3: degree 3 (connected to 1, 2, 4)
- Node 4: degree 3 (connected to 3, 5, 6)
- Node 5: degree 2 (connected to 4, 6)
- Node 6: degree 2 (connected to 4, 5).

The transition matrix of graph G is,

$$M = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{bmatrix}$$

Each row of the transition matrix sums to 1, reflecting transition probabilities.

The expansion operation involves applying a power of the adjacency/transition matrix, i.e., the matrix representation of the graph, to spread the probabilities across the graph. Computing $M = M \cdot M$, matrix multiplication, simulates a random walk taking two steps, spreading the flow across the graph and reinforcing connections within clusters.

$$M^2 = \begin{bmatrix} 5/12 & 1/6 & 1/4 & 1/6 & 0 & 0 \\ 1/6 & 5/12 & 1/4 & 1/6 & 0 & 0 \\ 1/6 & 1/6 & 4/9 & 0 & 1/9 & 1/9 \\ 1/9 & 1/9 & 0 & 4/9 & 1/6 & 1/6 \\ 0 & 0 & 1/6 & 1/4 & 5/12 & 1/6 \\ 0 & 0 & 1/6 & 1/4 & 1/6 & 5/12 \end{bmatrix}$$

Calculation for M_{11}^2: $M_{11}^2 = \mathrm{M}_{12}\mathrm{M}_{21} + \mathrm{M}_{13}\mathrm{M}_{31} = \left(\frac{1}{2}\right)\left(\frac{1}{2}\right) + \left(\frac{1}{2}\right)\left(\frac{1}{3}\right) = \frac{1}{4} + \frac{1}{6} = \frac{5}{12}$.

The inflation phase controls cluster granularity, with higher values yielding more clusters. It modifies the probabilities by raising them to a power, i.e., greater than one. And then normalizing each row to sum to 1, i.e., $M_{ij} = M_{ij}^r / \sum_k M_{kj}^r$; $r > 1$, element-wise power, enhancing strong flows. Inflation amplifies the influence of larger probabilities and diminishes smaller ones, effectively forming distinct clusters by enhancing intra-cluster connections and weakening inter-cluster ones.

For example, if we inflate the above graph with $r = 2$, for row 1: $\frac{5}{12}, \frac{1}{6}, \frac{1}{4}, \frac{1}{6}, 0, 0$;

- Square each: $\left(\frac{5}{12}\right)^2 = \frac{25}{144}, \left(\frac{1}{6}\right)^2 = \frac{1}{36}, \left(\frac{1}{4}\right)^2 = \frac{1}{16}, \left(\frac{1}{6}\right)^2 = \frac{1}{36}, (0, 0)$
- Sum of squares: $\frac{25}{144} + \frac{1}{36} + \frac{1}{16} + \frac{1}{36} = \frac{7}{24}$
- Normalize: $\left[\frac{25/144}{7/24}, \frac{1/36}{7/24}, \frac{1/16}{7/24}, \frac{1/36}{7/24}, 0, 0\right] = \left[\frac{25}{42}, \frac{2}{21}, \frac{3}{14}, \frac{2}{21}, 0, 0\right]$.

After inflating all rows, the matrix becomes as shown below.

$$M_{inflated} = \begin{bmatrix} 25/42 & 2/21 & 3/14 & 2/21 & 0 & 0 \\ 2/21 & 25/42 & 3/14 & 2/21 & 0 & 0 \\ 1/10 & 1/10 & 32/45 & 0 & 2/45 & 2/45 \\ 2/45 & 2/45 & 0 & 32/45 & 1/10 & 1/10 \\ 0 & 0 & 2/21 & 3/14 & 25/42 & 2/21 \\ 0 & 0 & 2/21 & 3/14 & 2/21 & 25/42 \end{bmatrix}$$

From the matrix we can see that probabilities within clusters, e.g., columns 1-3 for rows 1-3, are amplified, while those between clusters, e.g., columns 4-6 for rows 1-2, diminished.

The algorithm iterates through expansion ($M_{inflated}^2$) and inflation steps. After several iterations, the probabilities stabilize resulting in a matrix that reveals the clusters present in the graph. The matrix converges to a form where:

- Rows 1, 2, 3 have non-zero entries only in columns 1, 2, 3.
- Rows 4, 5, 6 have non-zero entries only in columns 4, 5, 6.

Finally, clusters are extracted from the converged matrix's connected components. Nodes whose rows have non-zero entries in the same columns belong to the same cluster [28].

- Nodes 1, 2, 3 have strong connections among themselves.
- Nodes 4, 5, 6 have strong connections among themselves.
- The bridge (3-4) is weakened.

Thus, the clusters are:

- Cluster 1: {1, 2, 3}
- Cluster 2: {4, 5, 6}

Algorithm: MCL Clustering

```
Input: Adjacency matrix A(n × n), expansion parameter e, inflation parameter r,
       convergence threshold ε
Output: List of clusters (sets of node indices)
1. Initialize:
   M = A
   Add self-loops to M if needed: M_ii = ε for all i
   Normalize M to make it row-stochastic:
    For each row i:
     sum = Σ_j M_ij
     M_ij = M_ij/sum for all j
2. While not converged:
   M = M_old
   Expansion: M = M × M (matrix multiplication)
   Inflation:
    For each row i:
     For each j:
      M_ij = (M_ij)^r
     sum = Σ_j M_ij
     M_ij = M_ij/sum for all j
   Prune: Set M_ij = 0 if M_ij < small threshold (optional)
   Check convergence: If |M − M_old| < ε, break
3. Extract clusters:
   Construct graph G from non-zero entries in M
   Find connected components in G
   Each component is a cluster
4. Return clusters
```

MCL's adaptive cluster size and ability to automatically detect the number of clusters based on the structure of the graph (no k requirement) makes it suitable for clustering graphs with variable topologies. However, its sensitivity to r and scalability issues that arise from its complexity, i.e., $O(n^3)$ per iteration, limit its implementation [19].

Generally, distance-based K-Means clusters nodes based on spatial proximity, Spectral Clustering minimizes edge cuts between clusters based on connectivity, balancing both costs; and MCL adapts clusters to edge connections based solely on connectivity between the nodes, lowering TAU. However, both Spectral Clustering and MCL can be computationally expensive, especially for large networks. The clustering accuracy might also be affected by noise and outliers.

Table 3 - 1: Comparison of clustering methods.

METHOD	INPUT	OBJECTIVE	STRENGTH	WEAKNESS
K-MEANS	Locations	Spatial cohesion	Fast, simple	Ignores topology and connectivity
SPECTRAL	Location, Paging and Handover Counts	Minimize edge-cuts	Captures connectivity	High complexity
MCL	Paging and Handover Counts	Flow-based cohesion	Adaptive clusters	Scalability issues, High complexity

3.2.4 Sparse Matrix Representation

Since urban networks are dense, their graph manipulation is computationally expensive. Luckily, most nodes in big networks aren't directly connected, so the matrix is full of zeros and can be represented by sparse matrix. For example, for a 1,000 node graph, we have a $1{,}000 \times 1{,}000$ matrix with most of its entries equal to zero. Hence, instead of storing a million entries, a sparse matrix only keeps track of the non-zero values to save a huge amount of memory. This helps operations like eigenvalue decomposition become much faster [29].

Similarly, MCL algorithm works with a transition matrix based on the graph's adjacency matrix. This matrix represents connections –*like handover frequencies between eNodeBs*. MCL simulates random walks by repeatedly multiplying and tweaking this transition matrix. Hence, sparse matrix representations are essential for the efficient computation of large graphs in MCL algorithm, particularly when simulating random walks over massive, sparsely connected networks [31].

Generally, sparse matrix is more memory efficient, has better calculation speed, and is scalable to let Spectral Clustering and MCL algorithms tackle large, real-world datasets.

3.3 Graph Clustering for TA Design

In an LTE context, a graph is a mathematical structure that consists of nodes –*representing cells in the network*– and edges –*representing connections between cells*. The weight of each edge can be used to represent the amount of handover between two cells. The weight of an edge can be based on the probability or frequency of a device moving from one cell to another.

Once a graph has been constructed, TA optimization can be formulated as a graph partitioning problem. By utilizing graph theory, we can analyze the relationships between TAs, identify areas of high mobility and potential congestion, and make intelligent optimization decisions. The objective is to partition the graph into TAs in such a way that minimizes the total weight of inter-TA edges. This is equivalent to identifying the minimum cut of the graph.

Overall, graph-based clustering offers a powerful framework for optimizing LTE TA design, with distance-based K-Means offering spatial efficiency, while Spectral Clustering emphasizing connectivity, and MCL adapting to the underlying network structure. By incorporating KPI data such as *Cell Locations*, *Handover Counts*, and *Paging Loads*, these approaches effectively address the inherent trade-off between *Paging* and *TAU* signaling messages.

Chapter 4: Methodology

This research is conducted in four major stages. In the first stage, data on cell locations, handover count, paging count, and related KPIs are collected. In the second stage, these datasets are used to model the network in the form of a graph, and to generate the user density and mobility patterns. In the third stage, both distance- and graph-based clustering algorithms are implemented using Python to generate TAs. Finally, the clustering results are visualized using scatter plots so as to facilitate qualitative analysis. Quantitative evaluation is also performed using relevant clustering metrics and signaling cost analysis, enabling comparisons among the proposed algorithms and against baseline configurations.

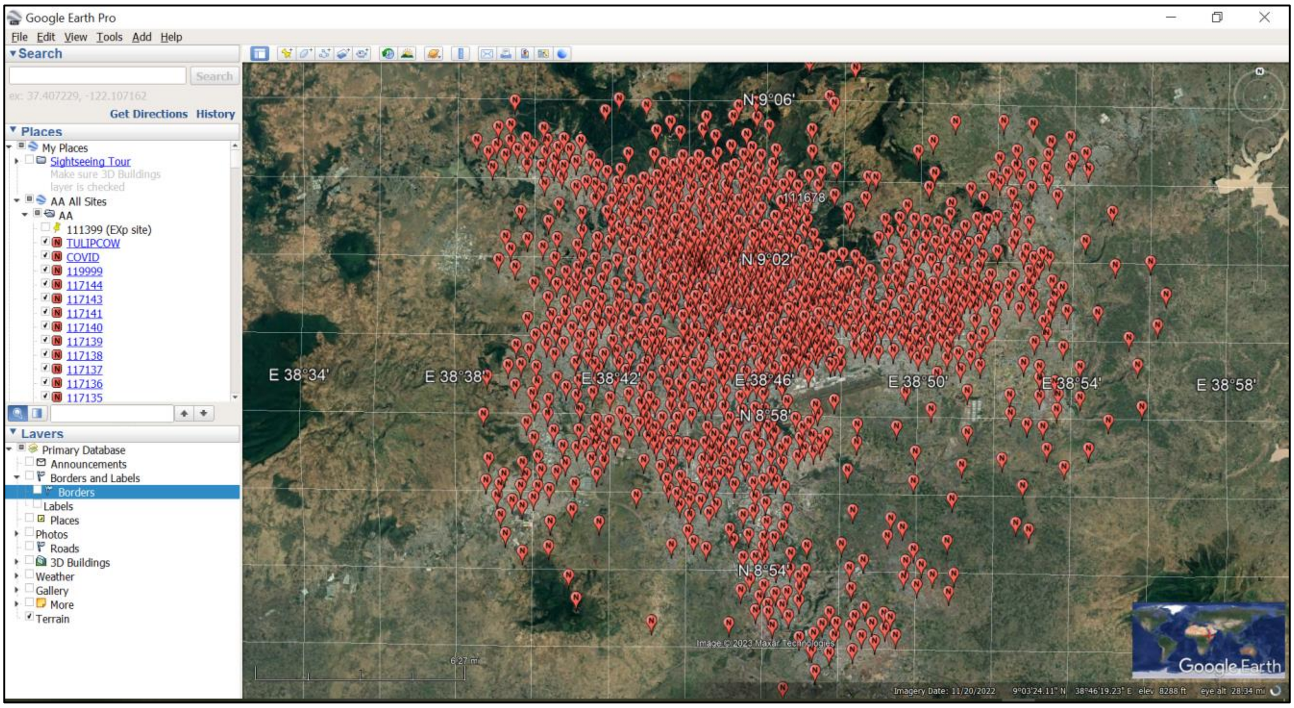

Figure 4 - 1: LTE network topology of the city of Addis Ababa.

This chapter presents the research design, data collection and preparation procedures, algorithmic implementation, performance evaluation methods, and the tools used, to ensure consistency with the research objectives outlined in Chapter 1.

4.1 Research Design

The research employs a quantitative, experimental research design, using data from a real-world LTE network to evaluate the performance of the various clustering algorithms.

The research workflow consists of the following steps:

- **Data Collection**: Gathering real word data from Ethio telecom's Radio Access Network-Operation Support System (RAN-OSS). The dataset includes eNodeB locations, handover relationships, paging frequencies, KPIs, and neighbor relationships.
- **Preprocessing:** Converting large volumes of raw KPI data into smaller bits and suitable formats for clustering, such as adjacency matrices and coordinate arrays.
- **Clustering**: Applying K-Means, Spectral Clustering, and MCL to generate TAs.
- **Evaluation**: Assessing the resulting TA configurations through visual inspection, internal clustering metrics, and signaling cost analysis.
- **Comparison**: Benchmarking the clustering results against baseline configurations to determine the optimal method.

This research is designed in such a way that it allows a controlled testing of our initial hypothesis, *i.e., the efficacy of graph-based clustering in reducing location management related signaling costs in handover-intensive cellular networks*.

4.2 Data Collection and Preparation

4.2.1 Data Collection

We start by collecting and inspecting the necessary datasets to understand our LTE network better. This involves gathering eNodeB ID, eNodeB location, paging count, handover count, neighbor relationship, and KPI data that reflect network traffic and resource utilization.

Handover and paging count data of the LTE network of the city of Addis Ababa is collected from Ethio telecom's RAN-OSS. From the dataset the following information is extracted:

- **Cell Locations**: Coordinates (x, y) for $n = 1{,}158$ eNodeBs which represent a large-sized LTE deployment. These data provide spatial input for distance-based K-Means and for Spectral Clustering algorithms ensuring compact TAs.

Table 4 - 1: Cell Location format and sample data.

S.N.	eNodeB ID	Longitude	Latitude
1	111001	38.72548	9.03251
2	111002	38.77861	8.895278
…	…	…	…
1,158	119999	38.72325	9.007635

- **Handover Count**: Pairwise handover frequencies between eNodeBs is taken by adding the handover count of each eNodeB throughout a one-month duration. The directed count is converted to undirected count so as to form symmetry. The count is converted into a weighted square matrix called the adjacency matrix $A \in R^{n \times n}$. A is crucial for Spectral Clustering and MCL algorithms.

  Those eNodeBs with a total handover count of zero are considered as an outlier, and can effectively be assigned to any TA without causing any considerable effect to the overall performance of the design.

  We have a total of 10,633,406 handover entries for a one-month data in the format shown in the table below.

Table 4 - 2: Handover count format and sample data.

S.N.	Date	eNodeB Name	Local eNodeB ID	Target eNodeB ID	Target Cell Name	L.HHO. NCell. ExecSuccIn	L.HHO. NCell. ExecSuccOut
1		113471_WL_UL_ BSCRNC04.HW. AKAKIKALITY WOREDA8. SAAZ.AA	113471	111567	111567_H7_DG_WL_ GUL_BSCRNC4.HW. KLTMINAGRI. SAAZ.AA_L3	3,699	3,889
2							
…	…	…	…	…	…	…	…
10,633,406							

In addition to the handover count we have neighbor relationship in the format shown below.

Table 4 - 3: Neighbor relationship format and sample data.

Src eNodeB ID	Src Cell ID	Src PLMN	Target Cell Info	Target Cell Type	HO Prohibited
1384	1	63601	MCC=636, MNC=01, eNodeB_ID=112082, Cell_ID=1	Intra-freq	FALSE

The neighbor relationship helps us take a sneak-peek as to how the similarity between adjacent eNodeBs is and to assure the data integrity. We have a total of 1,144,612 entries of neighbor relationships data for a duration of one-month.

- **Paging Count**: Paging data is found by adding all the paging count in the cells of an eNodeB. Since the paging count data collected for this research is from a live network with an already existing TA configuration, we need to reverse engineer to get the "organic " paging data for each eNodeB, i.e., the per-eNodeB paging frequencies P_i.

From the data, each eNodeB's paging count is a result of all cells in the current TA the eNodeB is assigned into. Hence, we need some sort of fraction to extract its "native" paging count. For example, KPIs related to resource usage tell a lot about the load on an eNodeB. RRC Connected User data can be used to generate this ration.

The ratio is generated for each eNodeB based on the TA it belongs to:

$$RRC\_Ratio_{eNodeB\_i} = RRC_{eNodeB\_i} / \sum_{i=1}^{n} RRC_i \qquad \text{(Eq. 4.1)}$$

for n eNodeBs in the TA in which the eNodeB currently resides.

$$eNodeB_{Native\_Paging\_i} = P\_Ave_{Current\_TA} * RRC\_Ratio_{eNodeB\_i} \qquad \text{(Eq. 4.2)}$$

The above equations enable us to calculate the paging cost of each eNodeB with utmost approximation, since native paging data is almost impossible to acquire from a real network, unless it is greenfield, synthetically generated, or deliberately configured with one eNodeB per TA, for a reasonable duration, which clearly is not a realistic solution.

Table 4 - 4: Paging Count format and sample data.

S.N.	Date	eNodeB ID	TAC	L.Paging.S1.Rx
1		111222	11252	279,472
2		111073	11255	387,450
…		…	…	…
210,603		113032	11203	424,921

As can be seen in Table 4-4 above, our network has a Paging count data of 210,603 entries with features eNodeB ID, TAC, and Paging Count. It influences TA size and is integrated into cost evaluation.

The paging distribution in each eNodeB is shown in the figure below.

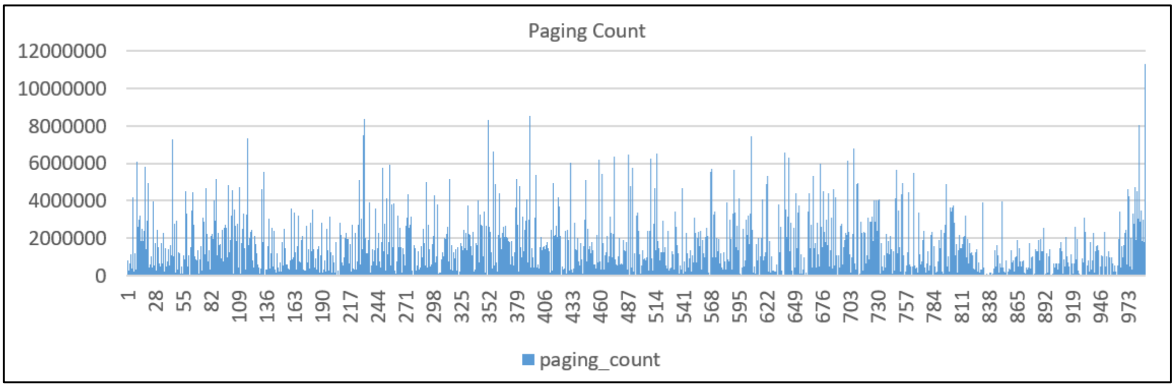

Figure 4 - 2: Paging Count of all eNodeBs in the network.

- Resource KPIs: Various resource query KPIs will be used to identify the paging share of each eNodeB in a TA. We have 229,816 entries with features such as paging utilization, downlink PRB utilization, downlink PS traffic, RRC connected user, etc.

Table 4 - 5: Resource KPI format and sample data.

Date	eNodeB ID	Paging Util.	DL PRB Util.	UL PRB Util.	DL PS Traff.	UL PS Traff.	RRC Conn User	RRC Conn User	RO.RRC Conn Success Rate	LTE HO Success Rate
		2.814	52.10	13.83	119.8	10.3	25.9	25.9	99.93	100
		2.814	58.48	19.04	138.3	12.7	38.4	38.4	99.93	99.92
		…	…	…	…	…	…	…	…	…
		2.814	54.99	11.71	107.3	6.50	16.2	16.2	99.90	100

- RRC Connected User: RRC connected user is a metric that indicates the total number of UEs that have successfully completed the RRC connection establishment procedure and are registered with the eNodeB. This means the UE is in a state where it can readily send and receive data or control information.

  RRC connected user's values are highly correlated with resource KPIs such as paging utilization, downlink/uplink PRB utilization, and downlink/uplink PS traffic.

In [23]: `df_resource_for_paging`

Out[23]:

	eNodeB_ID	P_Util	DL_PRB_Util	UL_PRB_Util	DL_PS_Traf	UL_PS_Traf	RO_Traf_Vol	RRC_Conn_User	RRC_Conn_Conges
0	111001	6.0845	38.4389	17.2956	95.0747	5.6741	100.7488	31.6643	0
1	111001	6.0847	19.9484	10.57	47.7055	2.6503	50.3558	11.7293	0
2	111001	6.0845	75.6007	29.9039	334.9830	19.7318	354.7148	90.6488	0
3	111001	6.2867	47.3013	18.4751	111.3560	6.7202	118.0762	35.6609	0
4	111001	6.2866	19.802	11.469	50.7730	3.7040	54.4769	12.4695	0
...	...	...	...	...	...	...	...	...	...
229811	118888	2.8067	58.5297	40.1212	67.1436	8.3817	75.5253	26.5673	0
229812	118888	2.8067	1.6701	3.4121	1.9822	0.6693	2.6515	1.4974	0
229813	118888	2.4489	30.0326	13.0325	44.2374	6.2292	50.4666	18.6878	0
229814	118888	2.4489	62.9388	44.7407	67.4873	10.4627	77.9500	28.3110	0
229815	118888	2.4489	1.5326	3.2872	1.7794	0.8620	2.6415	1.5804	0

229816 rows × 9 columns

Figure 4 - 3: RRC Connected User.

These data are stored in three CSV files, namely *enodebs.csv* which includes the *enodeb_id, longitude* and *latitude*, *handovers.csv* that holds the *source_enodeb_id, target_enodeb_id*, and the total *handover_count* fields, and *paging.csv* that has the *enodeb_id* and total *paging_count* data. These data will later be used to identify the UE traces, i.e., mobility pattern and subscriber density.

4.2.2 Data Preprocessing

Preprocessing ensures data compatibility with clustering algorithms. First, the data is normalized. x and y coordinates are normalized to $[0, 1]$ using min-max normalization or standard scaler to balance their scale.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad \text{(Eq. 4.3)}$$

Handover and Paging Counts are similarly normalized to $[0, 1]$ to mitigate scale disparities.

Paging count is used to compute the density of users in an eNodeB. Those with a paging count value of 0 are considered as an out-of-service eNodeB and will be automatically discarded from the list of eNodeBs.

For handover, we will start by dividing the file into 11 smaller chunks and by removing (dropping) all with a value of NULL. We will also verify the data integrity by checking only for non-negative counts.

Next, handover count between $source\_enodeb\_id$ and $target\_enodeb\_id$ pairs will be used to compute the weight of edges. Since handover data is directed, we create an undirected graph by adding counts for similar pairs. For example, if $1 \rightarrow 830: 49{,}600$ and $830 \rightarrow 1: 9$, then we have an edge weight of $49{,}609$ between $eNodeB$ 1 and $eNodeB$ 830, i.e., $1 \leftrightarrow 830: 49{,}609$.

Finally, we make sure that no isolated nodes are in the network, i.e., $\sum_j W_{ij} > 0 \; \forall \, i$, since eNodeB IDs that are not available in the handover count data are assumed isolated, i.e., has no edge weight between it and any other eNodeB.

Once we have our data ready, we use Python's NetworkX library to generate an undirected graph,

$$G = (V, E) \qquad \text{(Eq. 4.4)}$$

Where:

- $V = \{ eNodeBs\}$ and
- $E = \{(i, j): W_{ij} > 0\}$, that represents the eNodeBs and their neighbor relationships, respectively.

4.3 Clustering Algorithms

In this thesis, we will implement three algorithms to cluster eNodeBs into TAs, each leveraging different aspects of the preprocessed input data.

4.3.1 Distance-based K-Means Clustering

Since, distance-based K-Means partitions eNodeBs based on spatial proximity by minimizing the within-cluster sum of squares, a feature matrix, $X \in R^{n\times 2}$, generated from the normalized coordinates of eNodeBs, will be fed to the algorithm to generate the clusters.

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2 \qquad \text{(Eq. 4.5)}$$

Where:

- $x = (x, y)$ is the coordinate of the eNodeBs,

- μ_i is the centroid of cluster i, and
- k is the number of clusters [33].

The parameters are $k = 95$, computed from the best silhouette score, and a random seed value of 42 for reproducibility.

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=95, random_state=42)
labels = kmeans.fit_predict(X)
```

Figure 4 - 4: K-Means process using sklearn.cluster.KMeans.

The output provides the cluster labels $L \in \{0,1, \dots, k-1\}^n$. Distance-based K-Means forms compact TAs, reducing paging cost by limiting TA size. However, it ignores handover data, since it focuses solely on spatial cohesion.

4.3.2 Spectral Clustering

Spectral Clustering partitions the graph based on connectivity, minimizing the normalized cut,

$$NCut(C_1, C_2) = \frac{Cut(C_1,C_2)}{vol(C_1)} + \frac{Cut(C_1,C_2)}{vol(C_2)} \qquad \text{(Eq. 4.6)}$$

Where:

- $Cut(C_1, C_2) = \sum_{i \in C_1, j \in C_2} W_{ij}$ and
- $vol(C_i) = \sum_{j \in C_i} D_{jj}$ [25].

The Laplacian $L = D - W$; where a diagonal, degree matrix $D_{ii} = \sum_j W_{ij}$ which is the sum of edge weights for node i is used to compute eigenvectors.

The algorithm takes location data and adjacency matrix, $A \in R^{n \times n}$, from Handover Count, and Paging Count as an input. The parameters are $k = 95$, $affinity =$ '$precomputed$', and a random seed value of 42.

```
from sklearn.cluster import SpectralClustering
spectral = SpectralClustering(n_clusters=5, affinity='precomputed',
random_state=42)
labels = spectral.fit_predict(W)
```

Figure 4 - 5: Spectral clustering process using sklearn.cluster.SpectralClustering.

The output provides the cluster labels $L \in \{0,1, \dots, k-1\}^n$. Spectral Clustering minimizes inter-TA handovers, reducing TAU cost by aligning TAs with connectivity patterns.

Varying k values can also be used for sensitivity analysis to test robustness. Even though Spectral Clustering is effective for graphs with clear community structure, it also struggles with those having uneven cluster sizes or densities.

4.3.3 Markov Clustering (MCL)

MCL simulates random walks, iterating expansion and inflation to identify clusters [31]. The algorithm takes location data and adjacency matrix, A, from Handover Count and Paging Count as an input. And then iterate expansion ($M = M \cdot M$) and inflation ($M_{ij} = M_{ij}^r / \sum_k M_{kj}^r,\ r > 1$) until convergence. Convergence yields a matrix where rows indicate cluster membership.

The Inflation parameter, r, will have a range of tunable values, $r \geq 1.1$. Varying the values of the inflation parameter, r, can also be used for sensitivity analysis.

```
import markov_clustering as mc
import numpy as np
matrix = np.array(W)
result = mc.run_mcl(matrix, inflation=2.0, iterations=100)
clusters = mc.get_clusters(result)
labels = np.zeros(len(W), dtype=int)
for i, cluster in enumerate(clusters):
    for node in cluster:
        labels[node] = i
```

Figure 4 - 6: MCL process using Markov Clustering.

The output provides cluster labels $L \in \{0,1, \dots, k-1\}^n$, where m is the number of clusters. MCL adapts cluster size to handover patterns, balancing Paging and TAU costs without requiring a pre-defined k value.

4.4 Baseline Configurations

Including baseline evaluations helps us to measure how much clustering actually improves signaling cost over approaches that build up some kind of naivety over a period of time. To benchmark the three algorithms used in this thesis, the current ground truth clustering configuration is taken as a baseline. In addition, two naïve approaches *–Single and No TA clustering–* are taken as a comparison.

In a single TA configuration all eNodeBs are incorporated in a single cluster.

$$C_{TAU} = 0,\ C_{Paging} = n \cdot P_{avg} \qquad \text{(Eq. 4.7)}$$

In a No TA configuration each eNodeB is assigned to a unique TA, resulting in the maximum possible C_{TAU} and the minimum possible C_{Paging} values.

The currently deployed TA configuration of Addis Ababa LTE network has 17 TAs, as can be seen in the figure below.

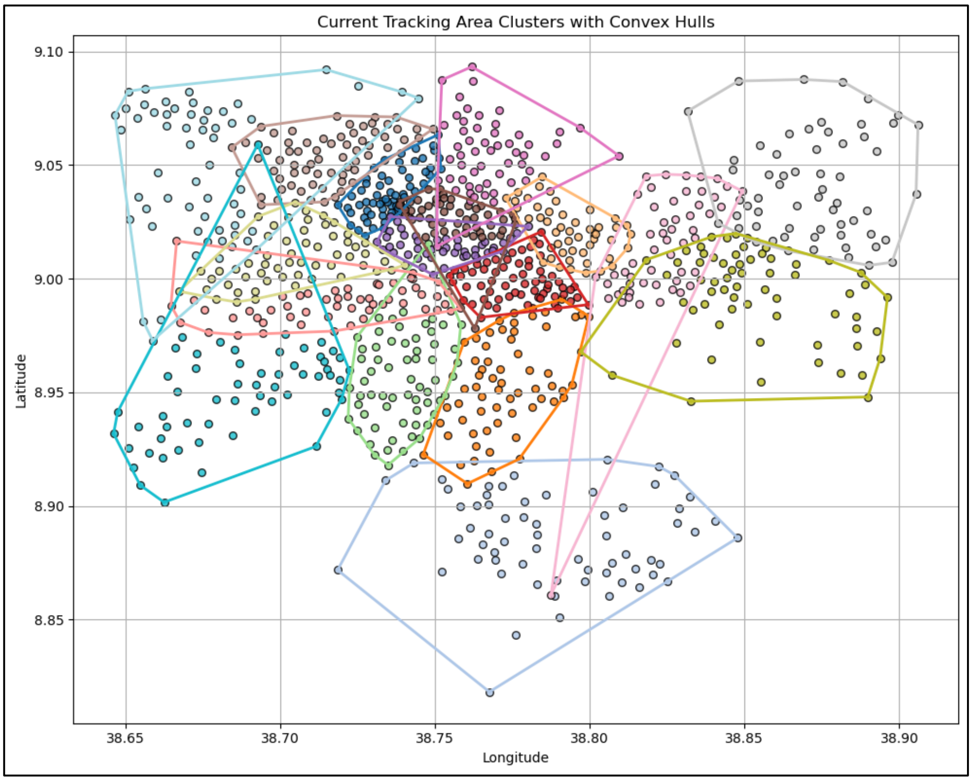

Figure 4 - 7: Ground Truth TA configuration.

4.5 Performance Evaluation

Performance Evaluation quantifies how good the clusters are using metrics. Since the algorithms used in this thesis are unsupervised, i.e., no initial ground-truth labels, all external metric and internal graph-based metrics are used. While the external metric captures signaling overhead, internal graph-based metrics gauge the quality of clusters. Additionally, comparative analysis against each other and against baselines helps to ensure the practical significance of the proposed methods.

4.5.1 Signaling Cost Metrics

A. Paging Cost

Proportional to the number of eNodeBs per TA;

$$C_{Paging} = \sum_{i=1}^{k} |TA_i| \cdot P_{avg,i} \qquad \text{(Eq. 4.8)}$$

Where:

- $|TA_i|$ is the number of eNodeBs in TA_i and
- $P_{avg,i} = \frac{1}{|TA_i|} \sum_{j \in TA_i} P_j$ is the average Paging Count in TA_i.

B. TAU Cost

Proportional to handover frequency across TA boundaries;

$$C_{TAU} = \sum_{e \in E_{cut}} W_{ij} \qquad \text{(Eq. 4.9)}$$

Where:

- $E_{cut} = \{(\mathrm{i}, \mathrm{j}) : \mathrm{L_i} \neq \mathrm{L_j}\}$ is the set of edges between TAs,
- W_{ij} is the sum of handover counts across TA boundaries.

C. Total Signaling Cost

Sum of Paging and TAU costs;

$$C_{Total} = \sum_{i \in N} \sum_{j \in N} (\alpha . C_{TAU_{ij}} (1 - S_{ij}) + \beta . C_{Paging_i} S_{ij}) \qquad \text{(Eq. 4.10)}$$

Where:

- α and β are adjustable weights to reflect network priorities, and
- the membership matrix, $S_{ij} = \begin{cases} 1, & if\ t_i = t_j \\ 0, & otherwise \end{cases}$, where t_i is TA of $cell_i$.

The common practice is to use the values $\alpha = 1$ and $\beta = 0.1$ in academic simulations, since handover events are 10 times more likely to occur on the network than paging [18].

4.5.2 Cluster Quality Metrics

Cluster quality metrics help to assess the structural integrity of clusters, such as intra-cluster density and inter-cluster sparsity.

A. Silhouette Score (S)

Silhouette Score is an internal evaluation metrics which assesses cluster quality based on the data itself. It measures how similar a node is to its own cluster (cohesion) compared to other clusters (separation).

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \qquad \text{(Eq. 4.11)}$$

Where:

- $a(i)$ is the average distance or dissimilarity between node i and all other nodes in the same cluster, and
- $b(i)$ is the smallest average distance between node i and all nodes in any other cluster, i.e., the nearest other cluster [32].

Both $a(i)$ and $b(i)$ could be based on edge weights, shortest paths, or other distance metrics. In our case we use the similarity matrix A as a metric.

The overall silhouette score is typically the average $s(i)$ across all nodes, and it ranges from $-1\ to\ 1$. An $s(i)$ value close to 1 indicates that the node is well-clustered, far from other clusters, and close to nodes in its own cluster. On the other hand, negative $s(i)$ values imply the node is closer to nodes in another cluster than its own, suggesting poor clustering. However, $s(i)$ value around 0 suggests the node is on or near the boundary between clusters, indicating ambiguous or overlapping clusters.

B. Modularity (Q)

Modularity is a graph-based evaluation metrics that measures the strength of division of a network into modules or communities. It compares the fraction of edges within clusters to the expected fraction if edges were distributed at random. In other words, Modularity quantifies how well-connected nodes are within clusters compared to what would be expected in a random graph with the same degree distribution.

Modularity,

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j) \qquad \text{(Eq. 4.12)}$$

Where:

- A is the adjacency matrix of graph,
- k is degree of nodes,
- m total edges,
- (c_i, c_j) clusters, and
- $\delta(c_i, c_j)$ is 1 if nodes i and j are in the same cluster, 0 otherwise [27].

To compute modularity in Python, we typically need to provide the graph structure and the partition of nodes into clusters generated by the clustering algorithm. By calculating the modularity score, we can quantitatively assess how well the algorithm has clustered the graph into distinct communities based on the density of edges within clusters versus between clusters.

Modularity typically ranges from -0.5 to 1. Positive values (closer to 1) indicate a structure with dense connections within clusters and sparse connections between clusters. Near 0 values suggest clustering is no better than random partitioning. Negative values imply nodes are less connected within clusters than expected, indicating poor or unnatural clustering.

4.5.3 Domain Knowledge

The appropriate number of clusters often depends on the underlying nature of the data. However, domain expertise can be critical in choosing a sensible value. Hence, we will use expert opinion as a crucial part of the performance evaluation metrics, using knowledge of the network topology and user experience as an input. Since our network has a known varying mobility and density structure, we expect a cluster size distribution that reflects this very nature of the network. The best clustering algorithm will make sure the TAs are neither too small (leading to high TAU) nor too large (leading to high Paging cost).

4.6 Implementation Details

4.6.1 Tools and Environment

- Programming Language: Python 3.9.13 as a development environment.
- Additional Data Analysis: Microsoft Excel 2016.
- Libraries:
  - numpy: for numerical computing to perform mathematical operations on arrays
  - pandas: to manipulate and analyze large datasets
  - scikit-learn: to build predictive models., clustering, and evaluations, e.g., k-means, spectral clustering, various metrics
  - networkx: for graph construction and clustering
  - markov_clustering: for MCL implementation
  - matplotlib: to visualize data and create graphs
- Hardware: Windows 10 64-bit PC, processor type is Intel i7-2600, 3.40 GHz CPU, 8 GB of memory.

4.6.2 Workflow

The overall process includes data loading, preprocessing, clustering, evaluation, and visualization as shown in the figure below.

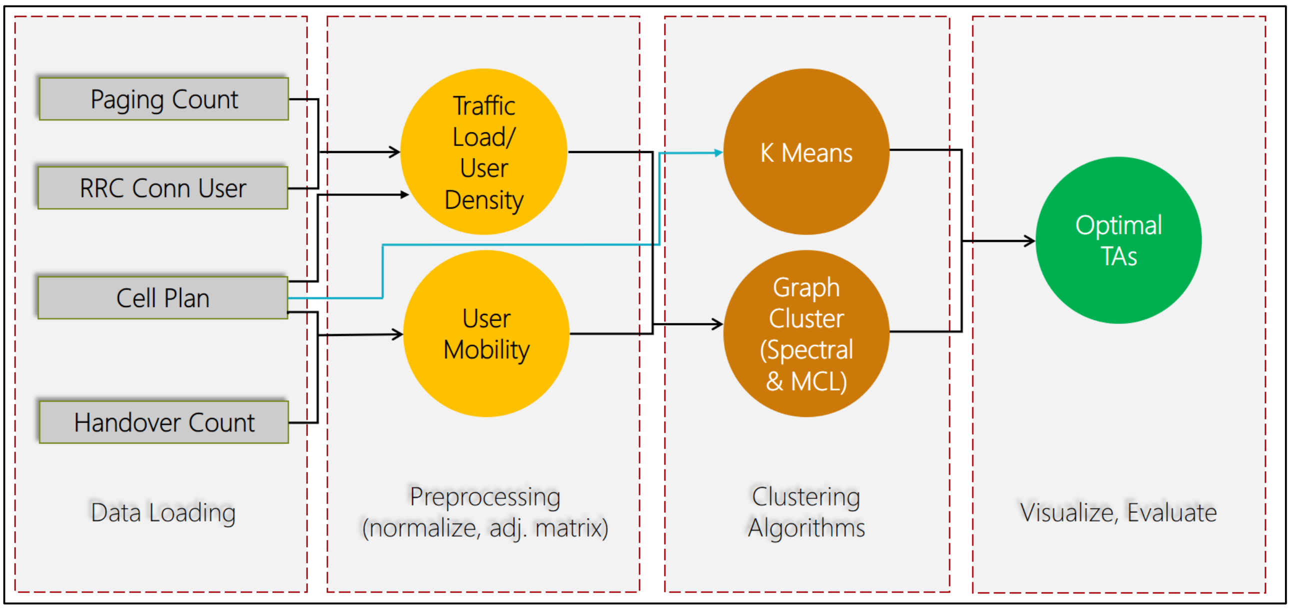

Figure 4 - 8: Workflow of the overall process.

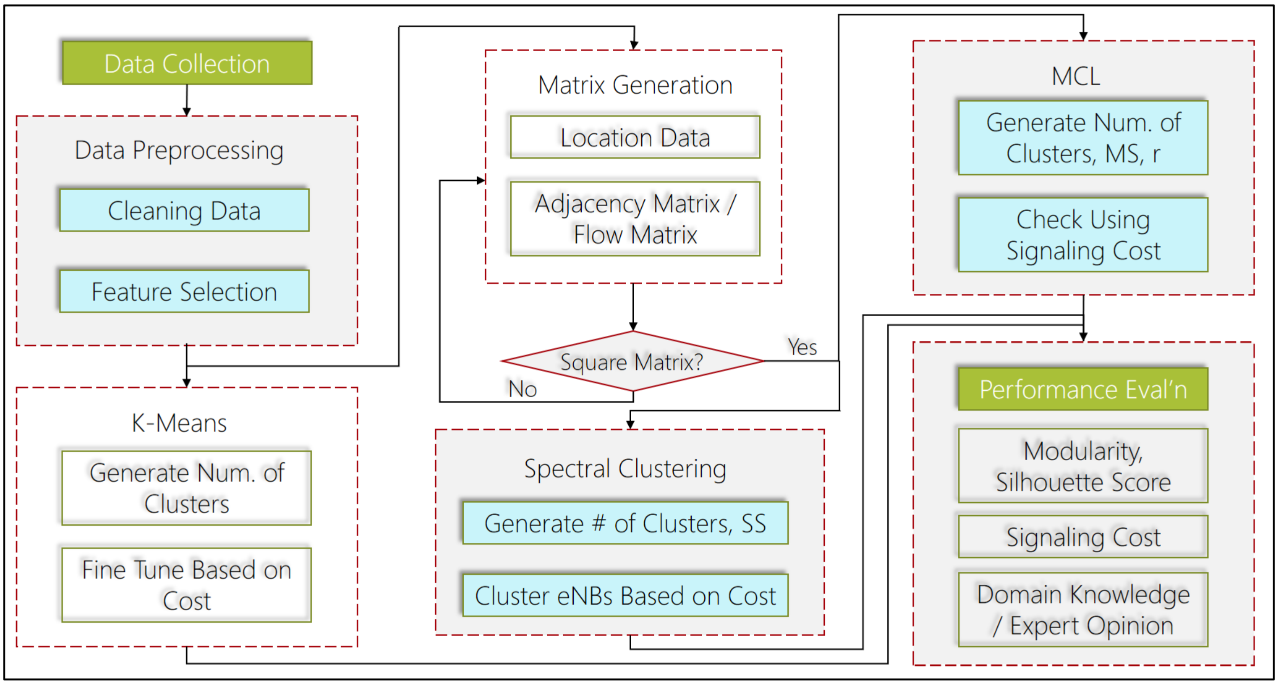

Figure 4 - 9: Flowchart for the overall process.

This methodology provides a thorough framework to optimize LTE TA design using both distance-based and graph-based clustering techniques. By integrating eNodeB Locations, Handover Counts, and Paging Counts, and evaluating distance-based K-Means and graph-based Spectral Clustering and MCL against baselines, it addresses the research objectives with a blend of theoretical consistency and practical implementation.

Chapter 5: Results and Discussion

This chapter presents the results of applying graph-based clustering algorithms to optimize TA design, using cell location, paging, RRC connected user, and handover counts data to analyze user density and mobility patterns. The primary objective is to minimize the total signaling overhead, expressed as $C_{Total} = C_{TAU} + C_{Paging}$, where C_{TAU} denotes the TAU cost and C_{Paging} denotes the Paging cost. Performance is evaluated against each other and against baseline clusters. Results are analyzed through quantitative metrics *–signaling costs, silhouette score, modularity–*, followed by qualitative assessments of cluster quality and geographical coherence derived from visualizations and expert opinion and a discussion of their implications for cellular network optimization. This chapter addresses the research questions *–how these algorithms perform comparatively, and which method yields the optimal TA design*.

5.1 Experimental Setup

The dataset comprises cell locations for 1,158 eNodeBs with (x, y) coordinates of Addis Ababa LTE network, handover count that is used for generating the adjacency matrix, A, with weights W_{ij} reflecting handover frequencies, and paging counts in the form of a vector.

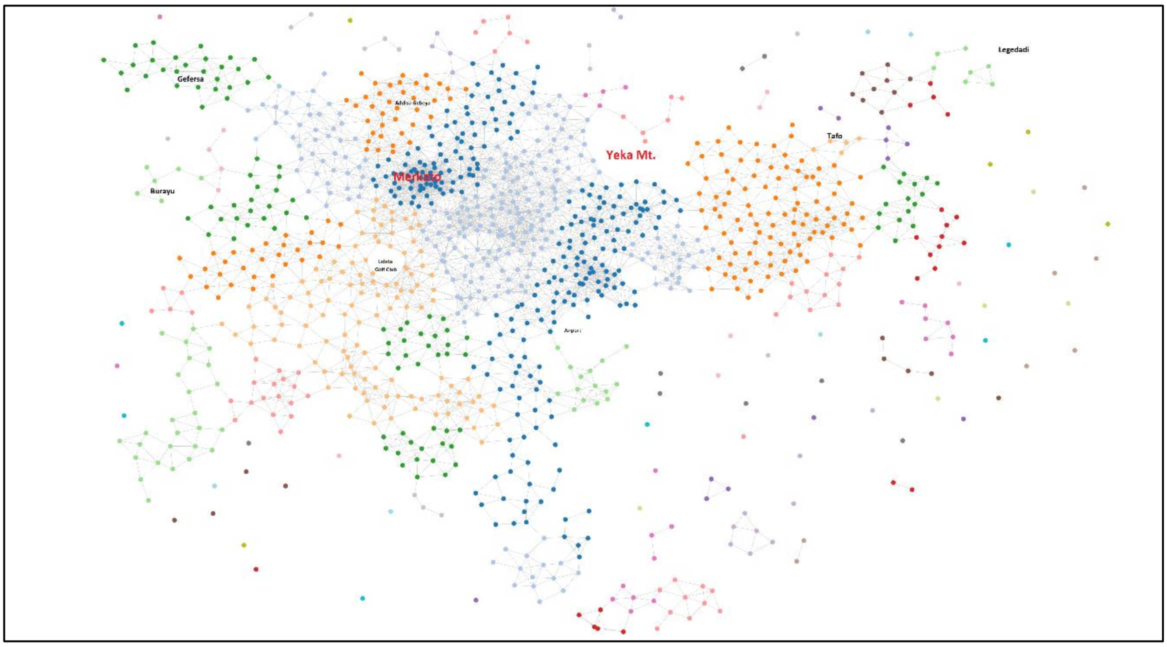

Figure 5 - 1: Parts of Addis Ababa's LTE network topology.

Our input data have the following features: -

- Location: contains enodeb_id, longitude, and latitude.
- Paging Count: contains enodeb_id and paging_count.
- RRC Conn. User: contains enodeb_id, TAC, RRC Conn User
- Handover Count: contains source_enodeb_id, target_enodeb_id, and handover_count.

Algorithms were implemented in Python. Each algorithm uses the following input values, features, and parameters:

- K-Means is implemented using cell locations as an input and a parameter k.
- Spectral Clustering is implemented using cell locations and A, derived from the paging, RRC connected user, and handover counts, as an input and a parameter k.
- MCL is implemented using A, derived from the paging, RRC connected user, and handover counts, as an input and a parameter r, tuned to optimize cluster granularity.
- Baselines include the current ground truth TA configuration of Addis' LTE network ($k = 17$), single TA ($k = 1$), and no TA ($k = 1{,}158$).

Metrics include $C_{Paging}, C_{TAU}, C_{Total}$, S, and Q computed as per Chapter 4.

5.2 Results

5.2.1 Distance-based K-Means Clustering

Distance-based K-Means clustering algorithm was applied to the dataset of cell location features *–longitude and latitude–* to partition eNodeBs into a predefined number of clusters, k. The number of clusters, k, was set to 95 based on silhouette analysis as shown below.

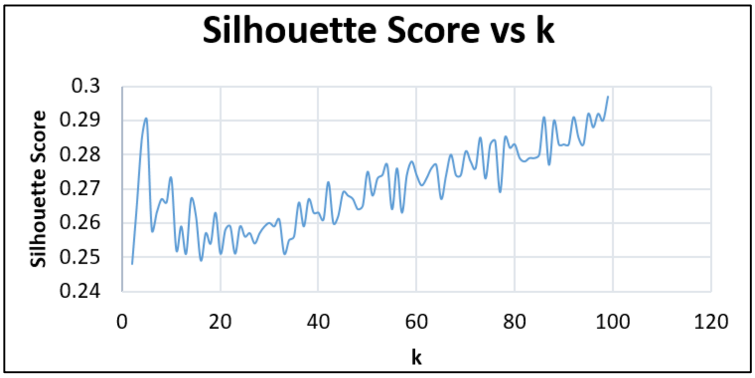

Figure 5 - 2: Evaluating optimal k for distance-based K-Means and Spectral Clustering.

A. Quantitative Performance

The quantitative performance of distance-based K-Means clustering approach was evaluated using internal validation metrics and the domain-specific signaling cost. The results are summarized in Table 5-1 below.

Table 5 - 1: Quantitative performance of distance-based K-Means clustering.

Metric	Value
Silhouette Score	0.307
Modularity	0.666
Signaling Cost (C_{Paging})	2.10E+09
Signaling Cost (C_{TAU})	3.54E+09
Signaling Cost (C_{Total})	3.75E+09

B. Qualitative Assessment

The geographical distribution and coherence of the clusters formed by distance-based K-Means algorithm are presented in Figure 5-3 below. This visualization depicts the assignment of nodes to different TAs based on K-Means' distance-based clustering output.

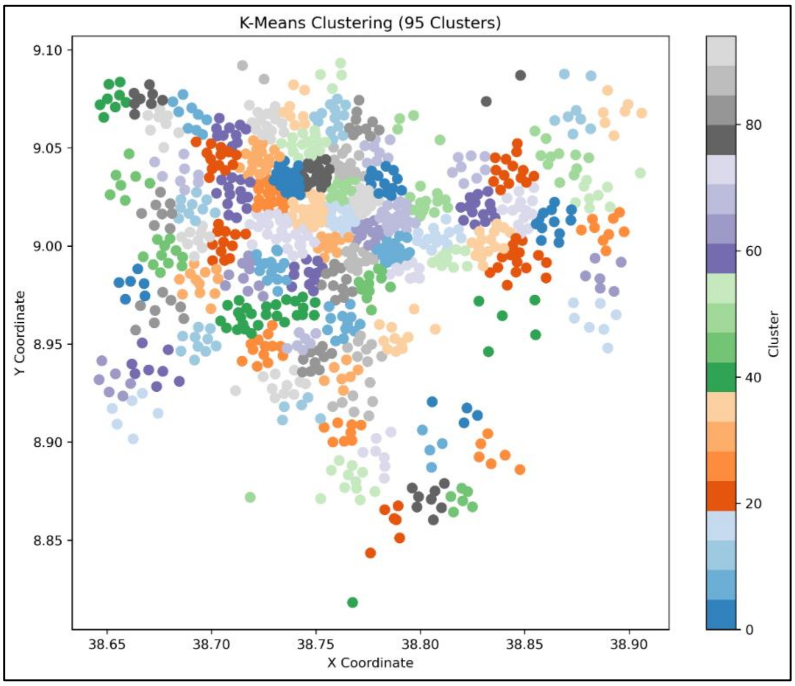

Figure 5 - 3: Cluster visualization for K-Means showing geographical coherence of TAs.

Observations from Figure 5-3 indicate equal sized clusters for both dense and sparse parts of the network.

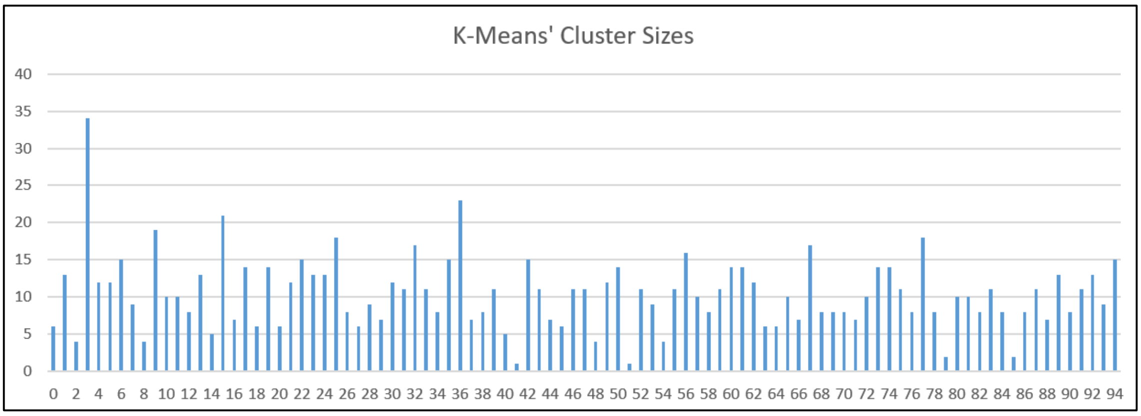

Figure 5 - 4: Distribution of eNodeBs per cluster for k-means algorithm.

Figure 5-4 above illustrates the distribution of eNodeBs across the generated clusters, providing insight into the relatively uniform distribution of eNodeBs per cluster.

5.2.2 Spectral Clustering

Spectral Clustering was employed to group eNodeBs based on the similarity matrix derived from cell location, paging, RRC connected user, and handover counts, i.e., user density and mobility patterns. The number of clusters was set to 95 based on the silhouette analysis shown in Figure 5-2 previously.

A. Quantitative Performance

The quantitative metrics for the TAs designed using Spectral Clustering are presented in Table 5-2 below.

Table 5 - 2: Quantitative performance of spectral clustering.

Metric	Value
Silhouette Score	0.201
Modularity	0.750
Signaling Cost (C_{Paging})	2.16E+09
Signaling Cost (C_{TAU})	2.59E+09
Signaling Cost (C_{Total})	2.81E+09

B. Qualitative Assessment

Figure 5-5 below illustrates the geographical arrangement of clusters formed by Spectral Clustering, with each cluster denoted by a unique color.

From Figure 5-5 below, the clusters formed by Spectral Clustering exhibit uneven sizes, aligned with handover patterns. This is because spectral clustering takes into consideration both cell location and users' interaction to generate clusters.

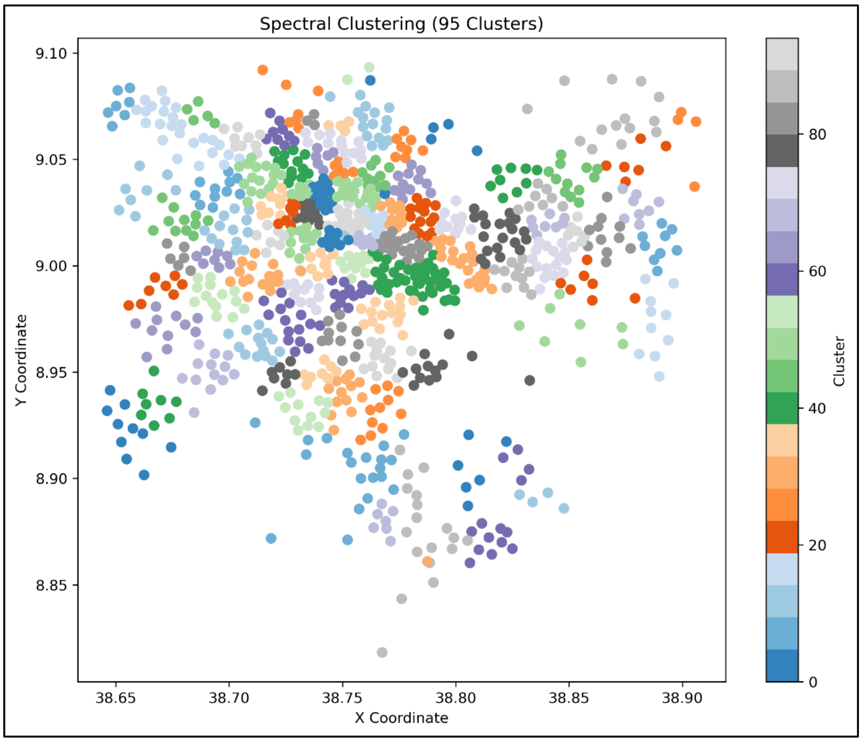

Figure 5 - 5: Cluster visualization for spectral clustering.

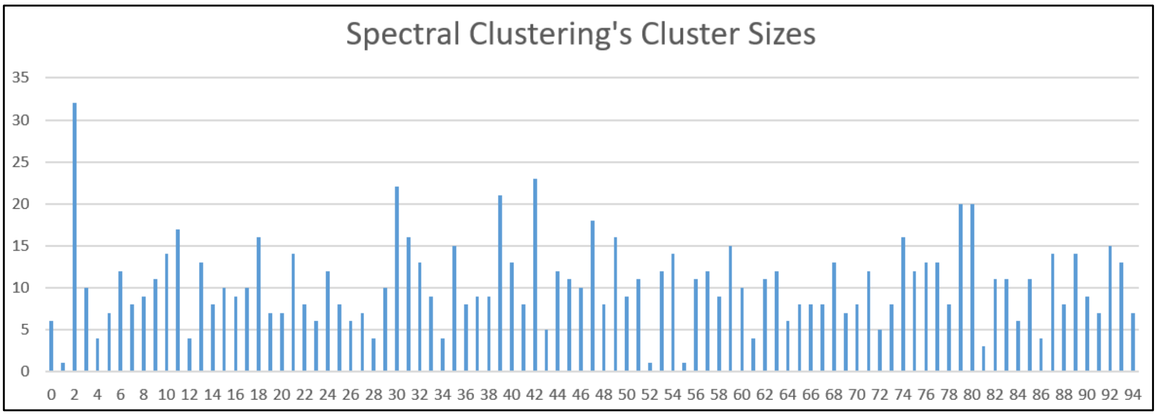

Figure 5 - 6: Distribution of eNodeBs per cluster for spectral clustering algorithm.

The distribution of eNodeBs within each TA generated by Spectral Clustering is shown in Figure 5-6. The figure indicates a relatively similar number of eNodeB distribution per TA.

As we can see from **Tables 5-1** and **5-2**, spectral clustering outperformed distance-based k-means in modularity and cost efficiency, indicating better handling of network topology and mobility patterns.

5.2.3 Markov Clustering (MCL)

MCL was applied to the user mobility graph, using flow simulation to detect densely connected regions. The number of clusters is determined by the algorithm based on the graph structure and the inflation parameter. The inflation parameter, r, is used for the range of $1.1 \leq r \leq 2.5$, tuned to balance cluster granularity.

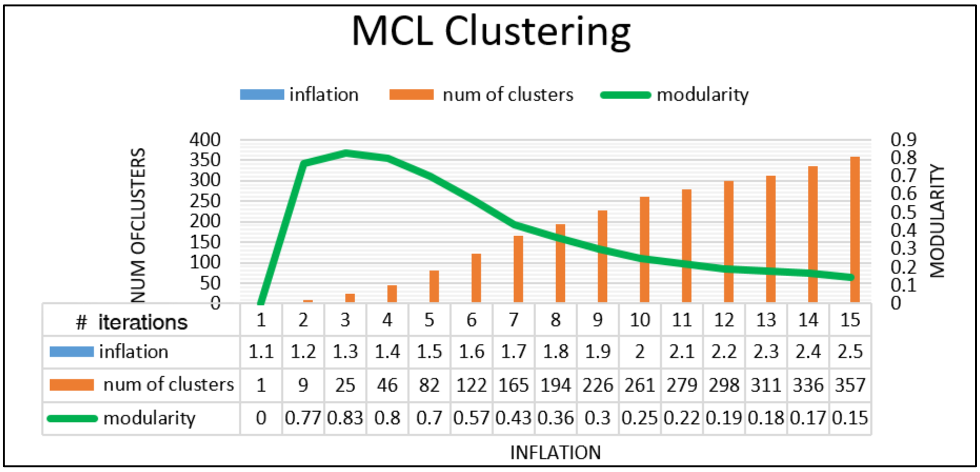

Figure 5 - 7: Inflation versus modularity versus number of clusters.

A. Quantitative Performance

The performance of MCL in forming TAs, measured by the selected quantitative metrics, is detailed in Table 5-3 below. The number of clusters resulting from MCL was 25.

Table 5 - 3: Quantitative performance of Markov Clustering.

Metric	Value
Silhouette Score	0.235
Modularity	0.830
Signaling Cost ($\boldsymbol{C_{Paging}}$)	7.43E+09
Signaling Cost ($\boldsymbol{C_{TAU}}$)	1.33E+09
Signaling Cost ($\boldsymbol{C_{Total}}$)	2.08E+09

B. Qualitative Assessment

Figure 5-8 displays the geographical representation of TAs obtained through MCL.

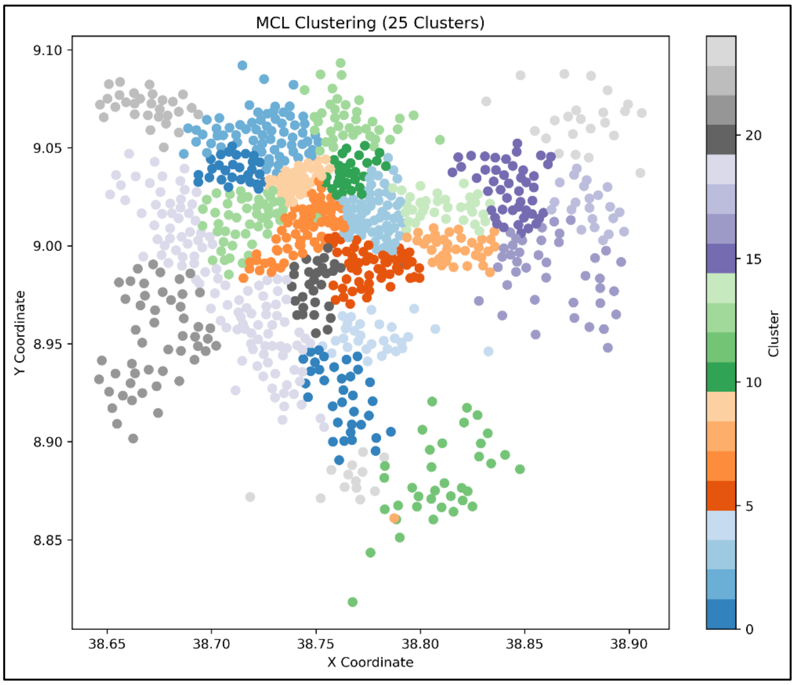

Figure 5 - 8: Cluster visualization for MCL.

MCL produced the most geographically coherent clusters with minimal overlap. Visual inspection of Figure 5-8 above reveals that the TAs formed by MCL are characterized by varying shapes, tendency to follow similarities between eNodeBs, and identification of natural barriers.

The distribution of eNodeBs among the TAs identified by MCL is presented in Figure 5-9.

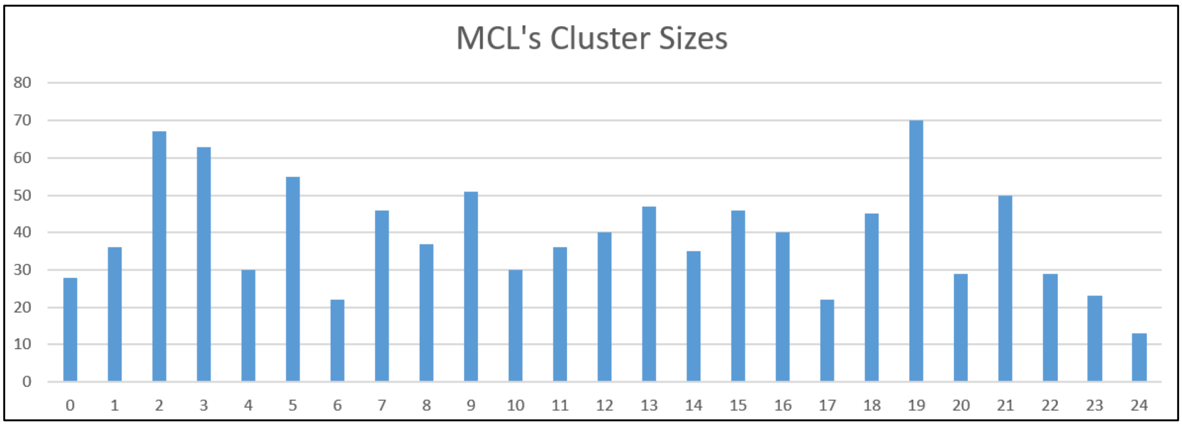

Figure 5 - 9: Distribution of eNodeBs per cluster for Markov clustering algorithm.

Figure 5-9 shows a high range of eNodeB count per TA, i.e., the presence of many small TAs or a few large ones.

From **Tables 5-1, 5-2,** and **5-3**, we can see that MCL demonstrated superior performance across all evaluation metrics, particularly in modularity and signaling cost, indicating its suitability for TA optimization.

5.2.4 Results Summary

This section presents the summary of the clustering results. Quantitative performance was assessed using Silhouette Score, Modularity, and Signaling Cost.

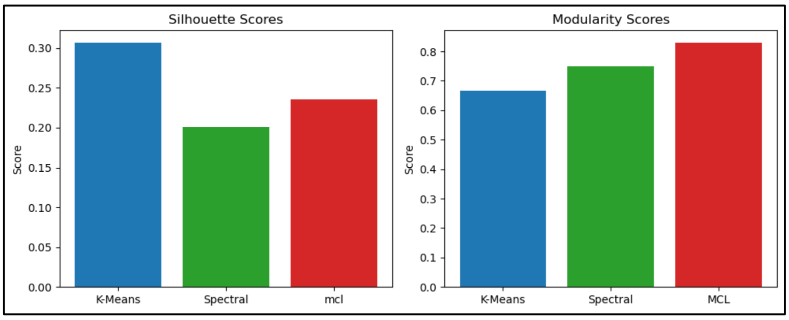

Figure 5 - 10: Box plot of metrics across methods.

As we can see from Figure 5-10, MCL outperformed both distance-based K-Means and Spectral Clustering with the most crucial metric for graph clustering, i.e., modularity, while maintaining a reasonable silhouette score.

The following table summarizes the performance of each clustering algorithm, as well as the baseline configurations, across these evaluation metrics.

Table 5 - 4: Cluster quality metrics across algorithms.

Method	Silhouette Score	Modularity
K-Means	0.307	0.666
Spectral Clustering	0.201	0.750
MCL	0.235	0.830
Single TA	NA	0.000
No TA	NA	0.013
Current Configuration	0.203	0.768

Qualitative evaluation focused on geographical coherence, size and structure of the clusters, and expert opinion on the user density and mobility patterns.

5.3 Discussion

This section presents an interpretation of the results obtained from applying distance-based K-Means as well as graph-based Spectral Clustering and MCL algorithms to cellular network data for optimizing TA design. It evaluates the performance of each algorithm, compares their relative effectiveness, and discusses their implications for TA design in cellular networks.

The Silhouette Score measures cluster cohesion and separation, indicating how well each eNodeB fits within the cluster it is assigned into versus other clusters. Modularity evaluates how well the clusters correspond to the network's inherent community structure.

5.3.1 Summary of Algorithms' Results

A. Distance-based K-Means Clustering

Distance-based K-Means algorithm, applied with $k = 95$, produced a Silhouette Score of 0.307, a Modularity of 0.666, and a Signaling Cost of $3.75E + 09$. Silhouette Score values shown here show a moderate cluster separation that suggests that distance-based K-Means performs well in capturing spatial proximity but is less effective in capturing structural relationships within the network. The relatively lower Modularity (0.666) reflects moderate community structure in the graph that indicats that distance-based K-Means disregards graph connectivity and handover patterns. The TAU Signaling Cost, a critical metric for TA design, suggests acceptable, but not minimal, overhead due to frequent TA updates.

Qualitatively, the illustration in Figure 5-3 shows clusters with equally weighed cluster sizes, which indicate sensitivity to Euclidean distance metrics and limitations to capture mobility-driven relationships. The geographical coherence of distance-based K-Means clusters appeared less consistent in high-mobility areas, likely due to its reliance on centroid-based partitioning, which does not take handover patterns into account.

Overall, these findings suggest that distance-based K-Means is effective for networks with relatively uniform UE density but may struggle to accurately model complex, mobility-driven cellular networks.

B. Spectral Clustering

Spectral Clustering, applied with a k value of 95, achieved a Silhouette Score of 0.201, a Modularity of 0.75, and a Signaling Cost of $2.81\text{E} + 09$. The higher Modularity Score relative to distance-based K-Means suggests a trade-off between spatial proximity and graph structure, which indicats improved clustering based on user density and mobility. The relatively lower Signaling Cost suggests reduced overhead for TA management, consistent with the algorithm's ability to generate more cohesive clusters.

As shown in Figure 5-5, the clusters are geographically compact, that demonstrates better alignment with cell adjacency and handover relationships. This improved geographical coherence likely results from Spectral Clustering's ability to leverage graph-based similarity metrics.

C. Markov Clustering (MCL)

MCL, applied with an inflation parameter of 1.3, achieved a Silhouette Score of 0.235, a Modularity of 0.83, and a TAU Signaling Cost of $1.33E + 09$ and a total Signaling Cost of $2.08E + 09$. The highest Modularity score indicates MCL's strong ability to detect mobility-based communities, a result of its flow-based approach that emphasizes handover-driven connections. The low TAU Signaling Cost indicates an efficient TA design, with reduced signaling overhead compared to distance-based K-Means and Spectral Clusterings.

Qualitatively, Figure 5-8 illustrated highly coherent geographical clusters, particularly in regions with frequent handover near the center of the network. The clusters demonstrated strong alignment with mobility patterns, which reflects MCL's ability to model transition probabilities within the network graph. However, the algorithm's sensitivity to the inflation parameter may influence cluster granularity, potentially resulting in smaller clusters in densely populated areas. These findings suggest that MCL excels at capturing mobility-driven structures but requires careful parameter tuning to achieve the desired clustering results.

5.3.2 Comparative Analysis

This section presents a comparative overview of the performance of distance-based K-Means and graph-based Spectral Clustering and MCL algorithms by considering both quantitative evaluation metrics and qualitative aspects.

A. Comparison of Quantitative Metrics

Table 5-5 summarizes the key quantitative performance metrics for the three clustering algorithms and the baseline configurations. Distance-based K-Means and graph-based Spectral Clustering were applied with $k = 95$ clusters, whereas MCL resulted in 25 clusters.

Table 5 - 5: Summary of quantitative performance of the various clustering methods.

Method \ Metric	S	Q	C_{Paging}	C_{TAU}	C_{Total}	# of TAs
K-Means	0.307	0.666	2.10E+09	3.54E+09	3.75E+09	95
SC	0.201	0.750	2.16E+09	2.59E+09	2.81E+09	95
MCL	0.235	0.830	7.43E+09	1.33E+09	2.08E+09	25
Single TA	NA	0.000	1.65E+11	0.00E+00	1.65E+10	1
No TA	NA	0.013	1.66E+08	1.10E+10	1.11E+10	1,158
Current Configuration	0.203	0.768	9.78E+09	1.90E+09	2.88E+09	17

Distance-based K-Means achieves a moderate C_{Total} $(3.75E + 09)$ by forming compact TAs, reducing C_{Paging} $(2.10E + 09)$ effectively due to spatial clustering. However, it exhibits a relatively high C_{TAU} $(3.54\text{E} + 09)$, primarily due to suboptimal boundary placement which disregards handover patterns. This results in the splitting of highly connected eNodeBs across TAs. Its silhouette score (0.307) indicates reasonable spatial cohesion, but the lower modularity (0.666) indicates weaker alignment with the network's topology, which is consistent with its limitation in graph-based contexts.

In contrast, Spectral Clustering's C_{Total} $(2.81\text{E} + 09)$ outperforms distance-based K-Means', driven by a lower C_{TAU} $(2.59\text{E} + 09)$ due to minimizing edge cuts. Its C_{Paging} $(2.16\text{E} + 09)$ is comparable to that of distance-based K-Means' which is an indication of a balanced TA size. The high silhouette (0.201) and modularity (0.75) scores further confirm strong cluster quality, which leverages the graph Laplacian's ability to capture handover connectivity. However, its predefined k value may not adapt to varying network densities.

As we can see from Table 5-5, MCL's adaptive clustering nature suits the handover-based graph's natural partitions, yielding fewer, denser clusters than both distance-based K-Means and Spectral Clustering approaches. Its superior modularity score (0.83) highlights topological robustness. MCL yields the lowest C_{TAU} (1.33E + 09) *–and hence* C_{Total} *(*$2.08E + 09$*)–* due to its flow-based approach aligning TAs with handover patterns. Its relatively higher C_{Paging} (7.43E + 09) reflects availability of larger TAs. However, we can still adjust the inflation parameter and get a lower value of C_{Paging} at the expense of C_{TAU} and C_{Total}.

The currently deployed configuration's exaggerated C_{Total} (2.88E + 09) confirms the need for data-driven methods.

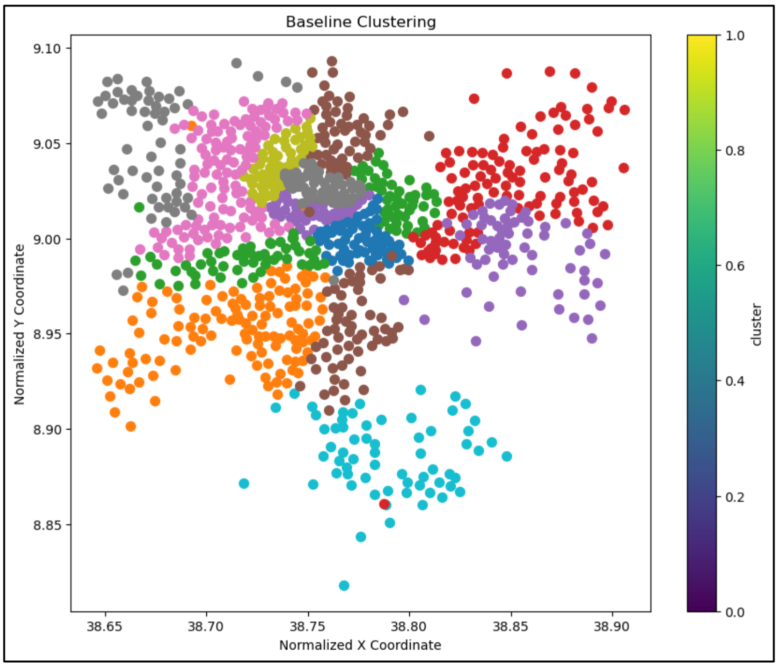

Figure 5 - 11: Scatter plot for ground truth clustering.

In addition to this, both single TA configuration and No TA configuration are also assessed to underscore the Paging-TAU trade-off. Single TA has the highest Paging cost (1.65E + 11) *–zero* C_{TAU} *but excessive* C_{Paging}*–* due to paging all 1,158 eNodeBs at once for every connection request. No TA has the highest TAU cost (1.10E + 10) *–excessive* C_{TAU} *but minimal* C_{Paging}*–* due to the need for TAU whenever a user moves from one eNodeB to another.

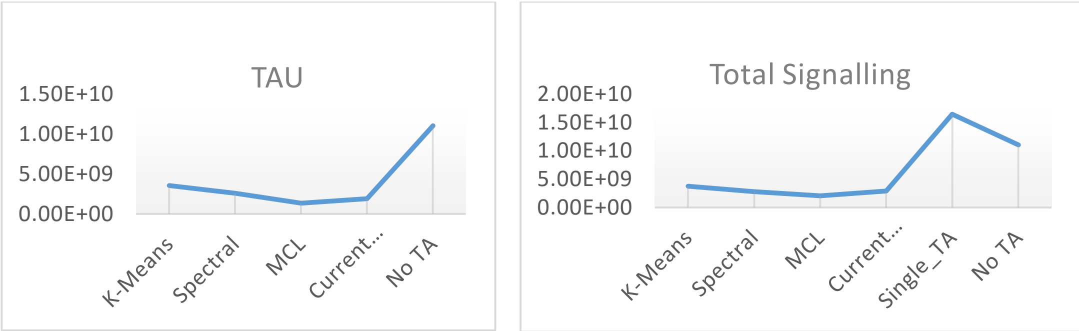

Figure 5 - 12: Bar chart of signaling costs C_TAU and C_Total across methods.

Generally, compared to the other approaches, MCL emerges as the optimal method both with respect to TAU and total signaling costs. MCL consistently outperformed distance-based K-Means, Spectral Clustering, and the current ground truth configurations in critical metrics, particularly in modularity and handover signaling messages and its ability to detect natural clusters without predefined numbers, which is ideal for handover networks.

MCL's success in leveraging Handover Counts to reduce C_{TAU} while managing C_{Paging} underscores the efficacy of graph-based approach in LTE TA design. Hence, while distance-based approach's simplicity makes it more practical for rapid prototyping and greenfield designs, graph-based approach's complexity offers higher precision for complex networks.

B. Comparison of Qualitative Aspects

Qualitatively, the geographical coherence and cluster characteristics were observed from a side-by-side comparison of the cluster visualizations as shown in Figure 5-13 below.

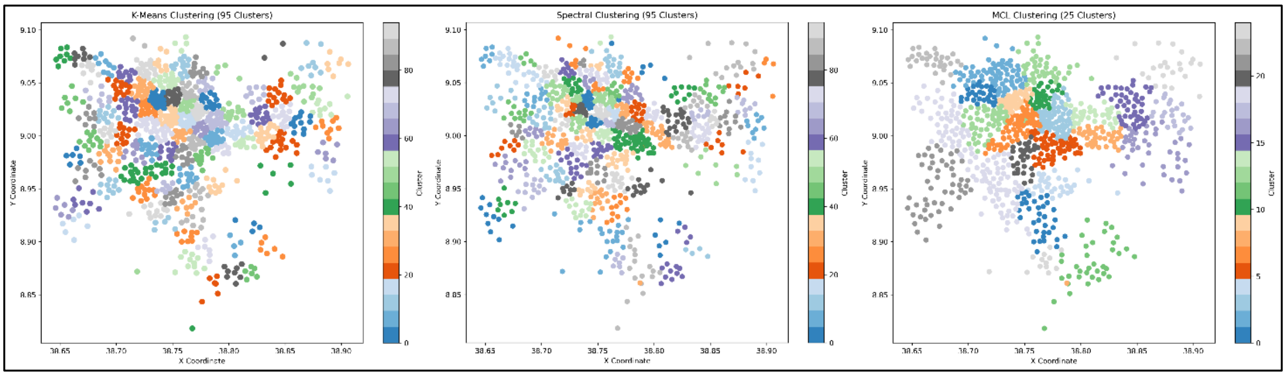

Figure 5 - 13: Conceptual side-by-side comparison of cluster visualizations.

As we can see above, the clusters generally appeared of equal size in K-Means Clustering (Figure 5-11, a). While the clusters formed by Spectral Clustering (Figure 5-11, b) tended to be more irregular in shape, able to capture non-convex structures, and exhibited good separation. However, MCL (Figure 5-11, c) produced clusters that varied significantly in size and density. And it appears that MCL aligns well with inherent network structures.

The distribution of eNodeBs per cluster showed more uniform size in both K-Means and Spectral Clustering compared to MCL which resulted in a wider range of cluster sizes. In Figures 5-4, 5-6, and 5-9, each bar represents a cluster, and its height shows the number of eNodeBs in that cluster.

In K-Means, bars are fairly uniform in height, since K-Means tends to balance cluster sizes. In Spectral Clustering, greater variation in cluster sizes is observed, with some significantly larger than others, which shows the presence of dominant network communities. MCL exhibits even more uneven cluster sizes, which shows the natural diversity in community structures. This variation is reflected in the range of cluster populations in each method.

K Means and Spectral Clustering has a range of 33 and 31, respectively, that indicates a relatively uniform cluster distributions. In contrast, MCL has a much larger range of 57, since it captures the most natural shapes of network communities.

5.3.3 Trade-off Analysis

The Paging-TAU trade-off is clearly observed across the clustering algorithms. Distance-based K-Means tends to prioritize paging cost reduction, often at the expense of higher TAU cost. On the other hand, Spectral Clustering achieves a more balanced trade-off, managing both TAU and paging costs effectively. In contrast, MCL favors TAU reduction, accepting slightly higher Paging costs. This aligns well with LTE's and 5G's operational needs. Urban networks with high mobility benefit from MCL's TAU focus, while sparse networks suit K-Means' Paging efficiency.

5.3.4 Implications for LTE TA Design

The results have significant implications for TA design in cellular networks. As have been seen in the results, graph-based approaches, particularly MCL, aligns well with dense, handover-heavy, and varying density urban networks. It can effectively reduce the TAU signaling cost (C_{TAU}) by 87.93% when compared to a No TA configuration; and by 29.85% when compared to the ground truth configuration.

Spectral Clustering is a viable alternative, balancing costs with lower computational overhead than MCL and incorporating the location data into consideration. Distance-based K-Means is suitable for uniformly distributed networks, offering simplicity and speed.

Table 5 - 6: Signaling cost improvements.

Method \ Metric	$C_{P.S.}$	$C_{P.NoTA}$	$C_{P.Cur}$	$C_{TAU.NoTA}$	$C_{TAU.Cur}$	$C_{Tot.S.}$	$C_{Tot.NoTA}$	$C_{Tot.Cur}$
K-Means	98.72	-1165.09	78.49	67.94	-86.36	77.22	66.08	-30.32
SC	98.69	-1197.98	77.93	76.55	-36.29	82.96	74.63	2.54
MCL	95.49	-4368.15	24.04	87.93	29.85	87.39	81.23	27.87

Figure 5-14 below shows convex hulls for the various clustering methods. Convex hull *–the smallest convex polygon that contains a given set of eNodeBs in a TA–* helps to visualize the nature and quality of the separation in the clusters.

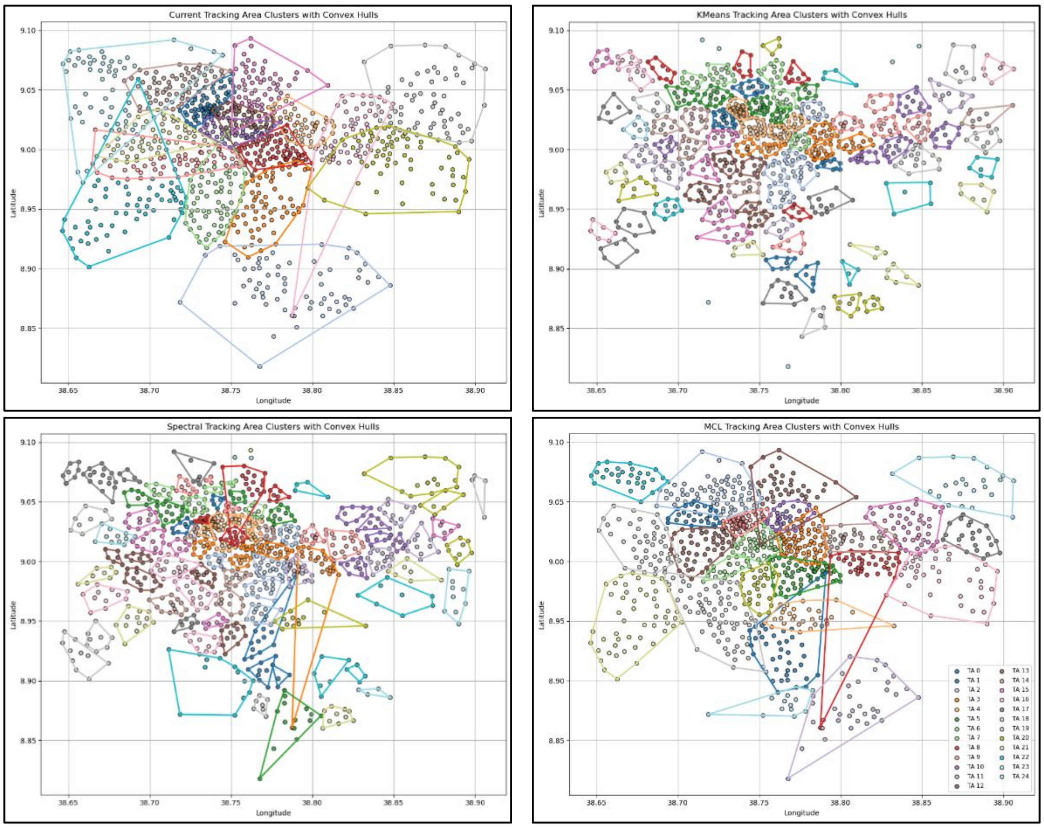

Figure 5 - 14: Convex Hull for various clustering configurations.

For example, in the ground truth (fig. a) we see a randomly shaped and overlapping TA configuration which might be a result of manual assignment of eNodeBs to TAs as the network expands progressively, while (fig. b) shows a neatly separated equal clusters which are a result of distance-based K-Means clustering, reflecting its mere dependence on geographic affinity.

In Spectral Clustering (fig. c) we see most areas to be uniformly partitioned *–reflecting its dependency on geographic coordinates*. However, some clusters are also amorphous *–reflecting its graph-based nature*. On the other hand, in MCL (fig. d) we see varying sizes of TAs, a result of natural – *handover/connectivity-based–* clustering. However, some outlier eNodeB's seen in both Spectral Clustering and MCL results can be adjusted using expert knowledge about the topology and connectivity of the network.

These performance improvements can be manifested in various forms. For example, seamless handover that leads to improved handover latency and call drop rate, lower connection speed that is a result of decreased paging and improved user locating, less UE battery consumption, etc.

5.4 Conclusion

This chapter has presented and analyzed the results by emphasizing the strengths and limitations of distance-based K-Means as well as graph-based Spectral Clustering and MCL algorithms for TA design. The findings underscore the potential of graph-based approaches to improve cellular network efficiency while identifying opportunities for further refinement. The next chapter concludes the thesis and summarizes its key contributions.

Chapter 6: Conclusion and Future Work

This chapter concludes the study on graph-based approaches for designing optimal TA in cellular networks. It summarizes the key findings from applying distance-based K-Means as well as graph-based Spectral Clustering and MCL algorithms to a real-world dataset comprising 1,158 eNodeBs. The outcomes of the study, their implications for improving mobility management, and potential directions for future research are also discussed.

Efficient TA design is essential for streamlining cellular network operations and ensuring seamless user experience. This objective has been addressed through clustering eNodeBs into TAs, and evaluating performance improvements by analyzing cluster quality metrics and signaling costs.

The graph-based clustering techniques presented here enable intelligent design of TAs that leverages network topology and inter-eNodeB relationships to form cohesive clusters. The advantages of this approach, including the methods used for computational efficiency and real-world feasibility, position it as a promising alternative for TA planning in modern cellular networks.

Hence, continued innovation and research in this area will further enhance the efficiency and scalability of its application in cellular networks, so as to accommodate the ever-increasing demand for reliable, high-quality services.

6.1 Summary of Key Findings

The research successfully demonstrated the effectiveness of graph-based clustering for TA design optimization with each algorithm offering distinct advantages.

Distance-based K-Means achieved a total signaling cost of $C_{Total} = 3.75E + 09$, by using the features of cell location to form spatially compact TAs. It reduced Paging cost ($C_{Paging} = 2.10E + 09$) significantly but with a corresponding higher TAU cost ($C_{TAU} = 3.54E + 09$).

Its relatively lower modularity (0.666) indicates moderate cluster quality, which is suitable for scenarios prioritizing spatial efficiency.

Spectral Clustering outperformed distance-based K-Means with $C_{Total} = 2.81E + 09$, balancing C_{Paging} $(2.16E + 09)$ and C_{TAU} $(2.59E + 09)$. It significantly reduces inter-TA handovers. Its strong cluster quality metric ($Q = 0.75$) reflects its topological alignment, making it a robust choice for connectivity-driven network environments.

MCL emerged as the optimal method with $C_{Total} = 2.08\mathrm{E} + 09$, excelling in TAU reduction ($C_{TAU} = 1.33E + 09$) *–a result of its flow-based clustering nature–*, and a slightly higher C_{Paging} $(7.43E + 09)$. Its superior quality metric ($Q = 0.83$) further underscores MCL's adaptability to the underlying network structure.

Compared to baseline configurations, all algorithms significantly reduced signaling overhead. A No TA configuration resulted in $C_{Total} = 1.11\mathrm{E} + 10$, a value that is clearly impractical for real-world network operations. The current configuration also exhibited a relatively high C_{Total} $(2.88E + 09)$ compared to both Spectral Clustering and MCL, highlighting the necessity of data-driven TA design methodologies.

Quantitative performance metrics and visualization analysis confirmed these findings. Spectral Clustering reduced C_{Total} by 2.54% relative to the current ground truth configuration, while MCL reduced C_{TAU} and C_{Total} by 29.85% and 27.87%, respectively, compared to the current ground truth configuration.

6.2 Achievement of Research Objectives

The research has successfully met its objectives in the following areas:

- **Data Integration**: The study effectively modeled location, RRC, paging, and handover data in a graph-based framework. While K-Means relies on spatial data, Spectral Clustering and MCL incorporated topological features, RRC connected user counts, paging frequencies, and handover relationships.

- **Algorithm Implementation**: Both Spectral Clustering and MCL were successfully implemented, producing feasible and optimized TA configurations suitable for real-world application.
- **Performance Evaluation**: The algorithms were evaluated against baselines and against each other using C_{Total}, S, and Q.
- **Optimal Method Identification**: MCL emerged as the most effective approach by achieving the lowest C_{TAU} and the highest quality metrics. Spectral Clustering demonstrated strong performance as an alternative, while distance-based K-Means remains suitable for simpler scenarios.

6.3 Implications for Cellular Networks

Optimal TA design is essential for efficient mobility management. By modeling the network as a graph, graph-based clustering techniques can effectively optimize the size and shape of TAs. This approach makes use of factors such as network topology, UE density, and user mobility patterns, which allows the resulting TA configurations to better align with real-world traffic behavior and connectivity structures.

6.3.1 Practical Implications

The results offer several important implications for network operators:

- **Signaling Overhead Reduction**: By achieving a 25-30% reduction in both C_{TAU} and C_{Total} can substantially improve network efficiency by lowering MME processing load and improving QoS. This is especially critical in urban environments where handover activity is frequent.
- **Algorithm Selection**: The findings suggest MCL as the most suitable method for dense, topologically complex networks, such as Ethio telecom's Addis Ababa LTE network; Spectral Clustering provides balanced performance with relatively lower computational cost. In contrast, distance-based K-Means remains appropriate for isotropic, low user mobility areas.

- **Data-Driven Design**: Integrating operational data –*location, paging, RRC connected user, and handover*– into TA planning clearly outperforms traditional manual or heuristic techniques.

6.3.2 Theoretical Contributions

This study advances the application of graph-based clustering in cellular networks by:

- Demonstrating the complementary strengths of spatial and topological approaches.
- Scalability testing of Spectral Clustering and MCL on real and urban network and using traffic data derived from a real-world network.
- Demonstrating a unified framework for evaluating TA design, that bridges graph theory and cellular network optimization.
- Utilizing RRC Connected User as a basis to predict the paging density of eNodeB's in a live network.
- Highlighting MCL's potential in adaptive clustering, not explored in cellular network contexts compared to K-Means and Spectral methods.

6.4 Future Work

The results demonstrate the effectiveness of graph-based approach in improving TA design. However, with the ever-growing number of connected devices to the cellular network, there is still room for further research.

Future research may focus on employing hybrid approaches, for example implementing one clustering algorithms for initial TA design supplemented by another for refinement, such as combining K-Means' simplicity with MCL's graph-based precision. Additionally, designing TAs in a multi-operator environment and their implications on battery life of UEs can be explored.

Adapting the framework presented here to 5G network RAs and future generation cellular technologies, would also extend the research's relevance to next-generation networks. Since the number of devices connected to cellular networks is increasing tremendously, the cost of signaling will be very high if not managed properly. TA is here to stay.

6.5 Final Remarks

This study has shown that graph-based clustering greatly improves TA design in LTE networks. MCL is the best way to reduce signaling overhead, and Spectral Clustering is a close second, making it a viable option. These algorithms work better than the baseline *–the ground truth configuration–* and the distance-based K-Means algorithm by combining Cell Location, Paging Count, RRC Connected User, and Handover Count data. This gives us a data-driven way to solve the Paging-TAU trade-off. The results help us understand both theory and practice better, showing how useful graph theory is in cellular communications.

Future work can extend this research by incorporating emerging technologies, exploring hybrid clustering frameworks, and enhancing the evaluation methodology with additional performance indicators. Overall, this thesis provides a solid foundation for improving mobility management by offering network operators practical tools and insights while establishing a starting point for further innovation in cellular network design.

References

[1] Ericsson, "Ericsson Mobility Report 2024," *Ericsson AB*, 2024.

[2] "Tracking Area," [online] Available on: https://www.sharetechnote.com/html/Handbook_LTE_Tracking_Area.html [Accessed on 2022]

[3] Z. Tesfaye, "Design Optimization of Tracking Area in LTE Networks Using Clustering Techniques, The Case of Addis Ababa, Ethiopia," MSc Thesis, Dept. Elect. Comp. Eng., Addis Ababa Univ., Addis Ababa, 2021.

[4] S. Tu, M. Waqas, Q. Lin, S. U. Rehman, M. Hanif, C. Xiao, M. M. Butt, C. Chang, "Tracking Area List Allocation Scheme Based on Overlapping Community Algorithm," *Computer Networks*, 2020.

[5] A. Alsaeedy, E. Chong, "A Review of Mobility Management Entity in LTE Networks: Power Consumption and Signaling Overhead," *Int J Network Mgmt.*, 2019.

[6] H. A. Hashim, M. A. Abido, "Location Management in LTE Networks using Multi-Objective Particle Swarm Optimization," *Computer Networks*, 2019.

[7] M. D. Elradi, L. O. Widaa, "Design Optimization of Tracking Area List in LTE Using 2D Markov Model," *Advances in Wireless Communications and Networks*, Vol. 5, No. 2, pp. 52-56, 2019.

[8] 3GPP, "General Packet Radio Service (GPRS) Enhancements for E-UTRAN Access," TS 23.401, 2018.

[9] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and E-UTRAN; Overall Description," TS 36.300, 2018.

[10] Zhang X., "LTE Optimization Engineering Handbook," *China Mobile Group Design Institute Co., Ltd., Beijing, China,* John Wiley & Sons Singapore Pte. Ltd, 2018.

[11] K. R. Kumar, S. Saranya, "Design Optimization of Tracking Area List for Reducing Total Signaling Overhead in LTE Systems," *Global Journal of Engineering Science and Researches*, 2017.

[12] Dahlman, E., Parkvall, S., Sköld, J., "4G: LTE/LTE-Advanced for Mobile Broadband," *Academic Press*, 2016.

[13] M. S. Algebary, I. M. Ellabib, A. B. Elwhishi, "Particle Swarm Optimization Based Approach for Location Area Planning in Cellular Networks," *IJISAE*, 2014.

[14] M. Nawaz, "Exploiting Tracking Area List Concept in LTE Networks," MSc Thesis, Linköping Univ., Sweden, 2013.

[15] 3GPP, "Self-configuring and Self-Optimizing Network Use Cases and Solutions," TR 36.902, 2011.

[16] Holma, H., Toskala, A., "LTE for UMTS: Evolution to LTE-Advanced," *Wiley*, 2011.

[17] L. Song, J. Shen, "Evolved Cellular Network Planning and Optimization for UMTS and LTE," *CRC Press Taylor & Francis Group*, 2011.

[18] S. M. Razavi, "Tracking Area Planning in Cellular Networks – Optimization and Performance Evaluation," PhD Dissertation, Dept. Sci. Tech., Linkoping Univ., Norrkoping, Sweden, 2011.

[19] Satuluri, V., Parthasarathy, S., "Scalable Graph Clustering Using Stochastic Flows: Applications to Community Discovery," *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 737-746, 2009.

[20] Jain, A. K., "Data Clustering: 50 Years Beyond K-Means," *Pattern Recognition Letters*, 31(8), 651-666, 2010.

[21] Newman, M. E. J., "Networks: An Introduction," *Oxford University Press*, 2010.

[22] S. M. Razavi, D. Yuan, F. Gunnarsson, J. Moe, "Exploiting Tracking Area List for Improving Signaling Overhead in LTE," 2010.

[23] 3GPP, "3G Release 8," 2008, TR 36-series, [online] Available on: http://www.3gpp.org/ftp/specs/archive/36 series/ [Accessed on 2022]

[24] S. M. Razavi, D. Yuan, "Performance Improvement of LTE Tracking Area Design- A Re-Optimization Approach," Linköping Inst. Tech., Sweden, 2008.

[25] Von Luxburg, U., "A Tutorial on Spectral Clustering," *Statistics and Computing*, 17(4), 395-416, 2007.

[26] Bishop C. M., "Pattern Recognition and Machine Learning," *Springer Science+Business Media, LLC*, 2006.

[27] Newman, M. E. J., “Modularity and Community Structure in Networks,” *Proceedings of the National Academy of Sciences*, 103(23), 8577-8582, 2006.

[28] Enright, A. J., Van Dongen, S., Ouzounis, C. A., “An Efficient Algorithm for Large-scale Detection of Protein Families,” *Nucleic Acids Research*, 30(7), 1575-1584, Oxford University Press, 2002.

[29] Ng, A. Y., Jordan, M. I., Weiss, Y., “On Spectral Clustering: Analysis and an Algorithm,” *Advances in Neural Information Processing Systems*, 14, 849-856, 2002.

[30] Shi, J., Malik, J., “Normalized Cuts and Image Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 888-905, 2000.

[31] Van Dongen, S., “Graph Clustering by Flow Simulation,” PhD Thesis, University of Utrecht, 2000.

[32] Rousseeuw, P. J., “Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis,” *Journal of Computational and Applied Mathematics*, 20, 53-65, 1987.

[33] MacQueen, J., “Some Methods for Classification and Analysis of Multivariate Observations,” *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281-297, University of California, Los Angeles, 1967.

[34] 3GPP, “Universal Mobile Telecommunications System (UMTS); Mobility Management Procedures,” 3GPP TS 23.012, v17.0.0, Sep. 2022.

[35] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); Mobility Management Entity (MME) Procedures,” 3GPP TS 23.401, v17.5.0, Jun. 2022.

[36] 3GPP, “5G; System architecture for the 5G System (5GS),” 3GPP TS 23.501, v18.0.0, Dec. 2023.