Optimal Routing of Electrical Networks using Clustering and Shortest path Algorithms

A major project report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Himanshu Dutt (201421)

Vansh Bansal (201428)

Under the guidance & supervision of

Dr. Diksha Hooda

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report titled Optimal Routing of Electrical Networks using Clustering and Shortest path algorithms in fulfillment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "Himanshu Dutt (201421); Vansh Bansal (201428)" during the period from August 2023 to May 2024 under the supervision of Dr. Diksha Hooda, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Himanshu Dutt (201421)

Vansh Bansal (201428)

The above statement made is correct to the best of my knowledge.

Dr Diksha Hooda

Assistant Professor

Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat,

Candidate's Declaration

I hereby declare that the work presented in this report entitled **'Optimal Routing of Electrical Networks using Clustering and Shortest path algorithms'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Diksha Hooda** (Assistant Professor CSE).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name: Himanshu Dutt
Roll No.: 201421

(Student Signature with Date)
Student Name: Vansh Bansal
Roll No.: 201428

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name: Dr. Diksha Hooda
Designation: Assistant Professor
Department: CSE
Dated:

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor Dr. Diksha Hooda Assistant Professor, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of Python language and its Libraries to carry out this project. Their endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Diksha Hooda**,** Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents

Himanshu Dutt (201421)

Vansh Bansal (201428)

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations, Symbols or Nomenclature

X	Latitude element coordinate point or points
Y	Longitude element coordinate point or points
ij	Point to point search variables
X_s, Y_s	Residential customer location
X_{np}, Y_{np}	Street nearest point to any customer
X_{se}, Y_{se}	Substation location
X_{be}, Y_{be}	Streets intersection or candidate sites location
X_{Lc}, Y_{Lc}	MV to LV transformer final location
XL_{st}, YL_{st}	Member Points of L street
SH	End user location
Ind	Optimal transformer index
N	Number of residential customers
M	Number of LV transformers
S	Number of substations
P	Total Number of subscribers N+M+S
$demN_N$	Individual customer demand
$demM_M$	Individual LV transformer demand
G	PxP connectivity matrix
$dist$	PxP distance matrix
$dist_N$	Distance from N customer to corresponding transformer
Cap	Number capacity constraint for all LV transformer
R	Distance constraint (m) for all LV connections
$Path$	Network connectivity route
$Pred$	Association end-user transformer
PVs	PV amount in the network
PVC	PV rooftop location
PVP	PV power assignation
C	Total customer connectivity in percentage
$CostMV$	Total distance (m) cost of designed LV network
$CostLV_M$	Distance (m) cost of M tranformer
$CostLV$	Total distance (m) cost of desgined low voltage network
$Comp_E$	Computational cost (seg) for each experiment
i,j,k	Counter variables for control loops
$flag,used,z$	Temporal variables
$Loc1$, $Loc2$	Temporal variables

ABSTRACT

This research presents a three-layer methodology to determine the best subterranean cable routing electrical distribution system, using the heuristic of graph search, the PRIMs(another algo can be added will be decided after implementation) algorithm. Transformer allocation and medium voltage network routing are handled by the algorithm's first layer. Low voltage network routing and transformer sizing are implemented on the second layer, while the provides a technique for allocating dispersed energy resources in an electric distribution network in the third place.The suggested algorithm routes a georeferenced area's electrical distribution network while accounting for topographical features like streets and intersections as well as situations in which there aren't squared streets. Additionally, the algorithm handles scalability features, permitting the gradual addition of loads. The model analysis finds that the algorithm achieves the best possible result for subterranean routing in a distribution electrical network implemented in a georeferenced area, satisfies the planned distance limitations, and reaches a node connectivity of 100%. Testing if the voltage drop in the farthest node is less than 2% is done by simulating the electrical distribution network.

INTRODUCTION

CHAPTER 1: INTRODUCTION

1.1 Introduction

Electricity distribution systems play a pivotal role in providing a reliable power supply to end-users. Over the years, these systems have evolved to meet the increasing demands of a growing population and expanding urban landscapes. This section provides an overview of the historical development of distribution systems, emphasizing the challenges and advancements that have shaped the current state of the industry.

The idea of subsurface cable routing is also introduced in this introduction. This modern method is becoming popular in urban areas because of its benefits for environmental effect, safety, and aesthetics. To maximise the functionality and design of these underground networks, advanced techniques are essential. The groundwork for the following discussion of the suggested three-layer system is laid forth in this chapter.

1.2 Problem Statement

This section provides a critical analysis of the problems with current electrical distribution systems, highlighting concerns with energy loss, inadequate cable routing, and inefficient maintenance. Case examples from the actual world highlight the real-world issues that contemporary distribution networks face. The analysis highlights the shortcomings of the existing strategies and proves that a comprehensive solution is required to deal with these issues.

The requirement for robust distribution networks, the incorporation of renewable energy sources, and the rising demand for power are all taken into account. In-depth examination of the problem landscape in this section prepares the audience for the presentation of a comprehensive and creative solution later on.

1.3 Objectives

The precise aims and objectives of the research study are described in this section. Every objective is thoroughly examined, offering a lucid justification for its incorporation and its impact on the overall research objectives.

For example:

- Collection of dataset in the form of OSM file.

- Implementation of a 3 layer model to optimize the required results.

- Implementation of 2 different pipelines to get the best result by comparing the pipelines.

- Comparing the best pipelines and accessing the result on the basis of the energy saved and also on the basis of the amount of construction material required.

1.4 Significance and Motivation of the Project Work

This research work provides an in-depth analysis of the significance and motivation behind the research, considering societal, economic, and environmental implications. Real-world benefits for end-users, utility companies, and the environment are emphasized, along with potential economic savings and improved resource utilization.

The research is positioned within the context of current global trends, policies, or initiatives related to sustainable energy and smart infrastructure. The environmental motivation is highlighted, emphasizing contributions to reduced carbon emissions, increased use of renewable energy, and a more sustainable energy landscape.

1.5 Organization of Project Report

This section outlines the structure of the project report, providing a roadmap for the reader as explained in fig 1.1. Each chapter is briefly described, highlighting the key themes and topics covered in a concise manner.

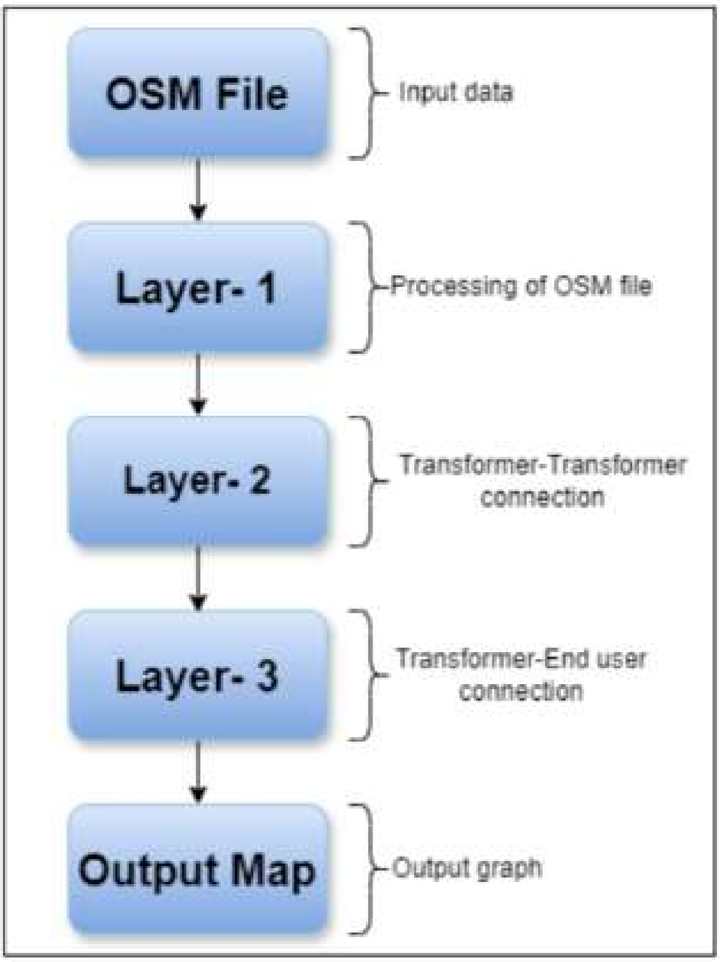


fig 1.1 Workflow of proposed approach

CHAPTER 2: LITERATURE REVIEW

Literature Review

In wireless sensor networks (WSNs), the study suggests an energy-efficient cluster-based routing protocol based on Ant Colony Optimisation (ACO) and Butterfly Optimisation Algorithm (BOA). To increase network lifetime and average energy consumption, the emphasis is on cluster head selection and path generation optimisation.The assessment of the literature emphasises the difficulties that WSNs face because of their limited energy resources and the ineffectiveness of conventional routing techniques. The importance of clustering in addressing energy restrictions is discussed. The proposed methodology is placed into the present research environment by the study through references to existing algorithms such as LEACH and DEEC. The ACO is selected for its quick problem-solving speed in WSNs, whereas the BOA is picked for its stability[1].

This study compares the hybrid KPSO (K-means Particle Swarm Optimisation) algorithm to the more established K-means and LEACH clustering algorithms in WSNs. Using simulations based on different configurations and circumstances, it evaluates the number of active sensor nodes, energy consumption, and network longevity.The study cites studies on coverage methods in WSNs and noise pollution monitoring. It talks about how sensor node energy varies and gives examples to assess the effectiveness of the suggested KPSO algorithm. In order to improve WSN performance, the paper highlights the necessity for energy-efficient clustering algorithms[2].

The study offers a framework for the best arrangement of data aggregation points (DAP) for Advanced Metering Infrastructure (AMI) in residential grids. It uses machine learning clustering approaches, taking into account transmission range, intended DAP count, and smart metre locations, among other inputs. By addressing the DAP placement issue, the paper positions itself to highlight the deficiency of techniques for AMI network planning. It highlights comparable efforts that were mostly centred around particular technology, highlighting the necessity of an all-encompassing framework. The outcomes of the suggested technique are contrasted with a common DAP placement heuristic[3].

The study presents a geometric routing algorithm for ad hoc networks called GOAFR+. By providing efficiency on average case graphs and asymptotic optimality in the worst case, it seeks to close the gap between theory and practise. The paper classifies cost functions and investigates how various cost measurements affect the effectiveness of routing algorithms. The study examines early geometric routing algorithm approaches and points out their drawbacks. It talks about the drawbacks of GPSR and Face Routing algorithms. The goal is to overcome these drawbacks and provide GOAFR+, a geometric routing algorithm that is both more effective and ideal[4].

The application of clustering techniques to routing problems in Mobile Ad-Hoc Networks (MANETs) is examined in this research. It suggests particular cluster algorithm implementations that integrate with current routing protocols and are based on multi-criteria network parameter selection. Citations include research on machine learning applications in networking, gateway discovery, and a clustering methodology called DBSCAN-GM that combines DBSCAN and Gaussian Means methods. An extensive review of clustering techniques and their applications in MANETs is given in this study[5].

The study discusses mobility-related issues with Mobile Ad Hoc Networks (MANETs), with a focus on multicast routing's ability to improve quality of service (QoS). The dependability pair factor and node energy are incorporated into the Cluster Head Selection Algorithm (CHSA), which is shown. For energy-efficient data transmission, the Lion Optimisation Algorithm (LOA) and hop counting-based optimal route selection are used. The results show that in regard to energy efficiency, packet delivery ratio, end-to-end delay, and entire network performance, the suggested technique (ORSMAN) performs better than the current methodologies. The suggested methodology greatly improves the energy efficiency and dependability of multicast routing in MANETs, opening up new research opportunities in various network topologies[6].

The goal of the paper's oblivious routing method for microgrid cluster optimisation is to meet load needs while minimising overall generation costs. Sections include the formulation of the routing problem, the suggested oblivious routing method, the outcomes of the simulation, and the conclusions. In order to optimise power routing for interconnected microgrids, the article takes apparent power, power loss, and load demand into account. Optimising routing according to the microgrid structure is the goal of an oblivious routing technique that makes use of hierarchical data structures. The study examines earlier studies on smart grids, microgrids, and applications such as electric vehicles and energy scheduling schemes[7].

With an emphasis on sensor cluster optimisation to reduce power consumption, the research suggests a reduced-complexity GA (Genetic Algorithm) for multihop sensor network optimisation. The knowledge from Intel Corporation, Oregon State University, and National Sun Yat-Sen University is contributed by the writers Rahul Khanna, Huaping Liu, and Hsiao-Hwa Chen. The research done is grounded in previous works on wireless sensor networks for surveys, routing protocols, and genetic algorithm optimisation. These are cited in the publication[8].

The goal of the study is to minimise average transmission power while maintaining high data speeds in wireless multi-hop networks by cooperative routing, link scheduling, and power control. The authors present a formulation of an optimisation problem that takes peak power, data rate, and transmission power limits into account. The duality technique, the optimal link schedule, and the power control rules are derived together with a detailed discussion of the optimisation problem. The application of optimum strategies in minimising average transmission power, traffic demand, and routing are all covered in this work. The authors highlight the advantages of their integrated method for conserving energy and increased network capacity while discussing the trade-off between reaching data speeds and energy efficiency. The importance of reducing power usage is reviewed, the trade-offs in current methods are emphasised, and integrated algorithms for the best routing, scheduling, and control of power are proposed. It is stressed to apply shortest path algorithms and duality approaches[9].

S.No	Title(citation)	Methodology	Conclusion
[1]	Stojmenovic, I. (2002). Handbook of Wireless Networks and Mobile Computing. John Wiley & Sons.	CHSA for cluster head selection, LOA for optimal route selection.	Proposed approach (ORSMAN) outperforms existing methodologies.
[2]	Al-Karaki, J. N., & Kamal, A. E. (2004). Routing Techniques in Wireless Sensor Networks: A Survey. IEEE Wireless Communications, 11(6), 6-28.	Minimize total generation cost in microgrid clusters.	Simulation results.
[3]	Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences.	GA for sensor cluster creation, fitness functions for network optimization.	Proposed GA enhances power consumption efficiency.
[4]	Boukerche, A., & Fei, X. (2006). On the Coverage Problem in Wireless Sensor Networks. Proceedings of the International Conference on Parallel Processing Workshops.	Duality approach, optimal policies, TDMA for routing.	Integrated approach improves energy efficiency and network capacity.

[5]	Huang, C. F., & Tseng, Y. C. (2005). The Coverage Problem in a Wireless Sensor Network. Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications.	LEACH for clustering, ACO for data transmission.	Proposed algorithm outperforms LEACH and ACO without clustering.
[6]	R. Mishra, P. Verma, R. Kumar, "Gateway Discovery in MANET using Machine Learning and Soft Computing," International Journal of Computer Applications, 2018.	M2NGA algorithm for clustering, multi-objective GA for optimization.	M2NGA outperforms existing intelligent clustering methods.
[7]	M. Wang, Y. Cui, X. Wang, S. Xiao, J. Jiang, "The Application of Machine Learning in Networking: A Review," Journal of Computer Networks and Communications, 2017.	BOA for CH selection, ACO for route generation.	BOA and ACO enhance routing efficiency in WSNs.
[8]	A. Smiti, Z. Elouedi, "DBSCAN-GM: A New Density-Based Clustering Algorithm," Journal of Systems and Software, 2015.	KPSO algorithm compared with K-means and LEACH.	Implementation of GOFR algorithm.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Requirements and Analysis

The Requirements and Analysis phase played a pivotal role in shaping the trajectory of the system development process. It involved a comprehensive exploration of the needs and expectations of stakeholders, encompassing both functional and non-functional aspects.

3.1.1 Functional Requirements

Functional requirements were identified through extensive consultations with end-users, electrical engineers, and urban planners. These requirements were detailed and specific, addressing key functionalities such as subterranean cable routing, transformer allocation, and dispersed energy resource allocation. Collaborative workshops and interviews were conducted to elicit detailed use cases and scenarios.

Functional requirements included:

- Subterranean Cable Routing: The system needed to optimize the routing of subterranean cables considering factors like street layouts, intersections, and topography.
- Transformer Allocation: Efficient allocation of transformers to cater to varying loads in the distribution network.
- Dispersed Energy Resource Allocation: Allocating renewable energy resources in the network for sustainable energy distribution.

3.1.2 Non-functional Requirements

Non-functional requirements were equally crucial in defining the overall behavior and performance expectations of the system. These specifications addressed things like user experience, scalability, and reliability.

Among the non-functional requirements were:

- Performance: Real-time results were anticipated from the system, particularly in dynamic scenarios including load additions or changes to the energy resource landscape.
- Scalability: The system has to be able to smoothly grow with the network, taking on more traffic and resources without sacrificing its functionality.
- User Experience: Planners and engineers will find it easy to interact with the user interface, which was made to be intuitive and user-friendly.

The requirements analysis process was iterative, involving ongoing feedback loops with stakeholders to enhance and validate the requirements that were identified.

3.2 Project Design and Architecture

The Project Design and Architecture phase concentrated on converting the clearly specified requirements into an organised and scalable system. To do this, a thorough design paper outlining the architecture, parts, and interactions of the system has to be created.

3.2.1 High-Level System Architecture

A top-down view of the system was made possible by the high-level system design, which showed the main parts and how they interacted. It guided the development team's implementation process like a blueprint. Because of its modular and extensible nature, the architecture will be able to accommodate new features in later versions.

3.2.2 Database Design

A crucial component was the database architecture, which established the data types and schema needed to enable information retrieval and storage. This required giving the data kinds, relationships, and indexing techniques some thought. The selection of a relational database management system (RDBMS) was based on its effectiveness in managing structured data.

3.2.3 User Interface Design

The goal of user interface design was to visualise the system's interface through the creation of prototypes and wireframes. The ideas of accessibility and usability informed design decisions. Potential end users were shown mock-ups in order to get their opinions on the design, way information was presented, and general usability.

3.2.4 Algorithmic Design

One of the main components of the system's operation was its algorithmic design. Every layer in the suggested methodology has an algorithm that is tailored to deal with particular electrical distribution system components. For instance, the transformer allocation algorithm optimised the locations of transformers based on network restrictions and load distribution.

Algorithm 1 Optimal location and routing a MV grid network.

```
procedure
Step: 1 Variables
    P, distN, X, Y, Cap, R
Step: 2 Optimal transformer selection
    used ← prim(X, Y);
    Ind ← find(sum(used) == 1);
    X_tr ← X_cd(Ind);
    Y_tr ← Y_cd(Ind);
Step: 3 Find nearest street point to customer
    Loc1 ← [X_t Y_t];
    Loc2 ← [XL_st YL_st];
    for i → 1 : N do
        for j → 1 : length(XL_st) do
            dist_ij ← haversine(Loc1, Loc2);
            z ← find(dist_ij == min(min(dist_ij)));
        EndFor
    EndFor
    X_sp ← Loc2(z, 1);
    Y_sp ← Loc2(z, 2);
Step: 4 Optimal Routing MV grid
    X ← [X_sp X_t X_sd];
    Y ← [Y_sp Y_t Y_sd];
    dist_ij = haversine(X, Y);
    G(dist_ij <= R) ← 1;
    path ← prim_mod(sparse(G));
Step: 5 Determine the final cost of MV
    for i → 1 : length(X) do
        for j → 1 : length(X) do
            costMV ← costMV + dist_ij(path);
        EndFor
    EndFor
End procedure
```

Algorithm 2 Optimal routing a LV grid network.

```
procedure
Step: 1 Variables
    P, distN, X, Y, Cap, R
Step: 2 Determine the distance end user, transformer
    dist_ij = haversine(X, Y);
    G(dist_ij <= R) ← 1;
    G(1 : N, N + M + 1 : N + M + S) ← inf;
    G(N + M + 1 : N + M + S, 1 : N) ← inf;
    G(N + 1 : N + M + S, N + 1 : N + M + S) ← inf;
Step: 3 Applying Dijkstra
    Pred ← dijkstra(G, P);
Step: 4 Optimal Routing LV grid
    for Trans ← 1 : N do
        X ← [X_nn(Pred) X_Trans];
        Y ← [Y_nn(Pred) Y_Trans];
        dist_ij = haversine(X, Y);
        G(dist_ij <= R) ← 1;
        path ← prim_mst(sparse(G));
    EndFor
Step: 5 Determine the final cost of LV
    for i ← 1 : length(X) do
        for j ← 1 : length(X) do
            costLV ← costLV + dist_ij(path);
        EndFor
    EndFor
End procedure
```

Algorithm 3 Allocation of DER PV generator.

```
procedure
Step: 1 Inizialization
    X ← [X_s];
    Y ← [Y_s];
    SH ← [XY];
Step: 2 Determining PV amount
    PVs ← floor(length(SH) * 0.1);
Step: 3 Determining the center of mass
    PVC kmedoids(SH, PVs);
Step: 4 Power assignation
    PVP ← 10KV;
End procedure
```

3.3 Data Preparation

A critical first step in guaranteeing the precision and dependability of the system's output was effective data preparation. To enable precise algorithmic execution and realistic simulations, this phase required gathering, cleaning, and organising pertinent data sets. The fig 3.1 shows the sample of the osm file used as dataset.

fig 3.1. OSM file Example

3.3.1 Geospatial Data Collection

Various sources provided georeferenced data for the intended area, including topographical characteristics, intersections, and street layouts. Experts in Geographic Information Systems (GIS) were contacted to guarantee the precision and entirety of the geospatial data. This required comparing the data to the real world and fixing any errors.

3.3.2 Load Data

Cooperation between local authorities and utility providers allowed for the collection of data on electrical loads in the area. The system's ability to adjust to shifting load profiles was ensured by taking into account both present and possible future requirements. To improve load prediction accuracy, historical load data was also included, where available.

3.3.3 Topographical Information

The system included data on terrain, altitudes, and other geographic details. Subterranean cable routing has to be optimised with this information in mind, taking probable impediments and elevation variations into account.

To guarantee the quality and dependability of the input data, domain experts, GIS professionals, and data scientists have to work together during the data preparation phase.

3.4 Implementation

Creating executable code from the design specifications was the task of the implementation phase. Code snippets and a description of the programming languages, frameworks, and algorithms used in the development process are included in this part.

3.4.1 Algorithmic Implementation

The suggested method focuses on strategically combining graph theory and machine learning approaches to maximise the building of electric distribution systems (EDS). The strategy is comprised of two pipelines, each specifically engineered to address unique challenges that arise during EDS implementation.

3.4.1.1 Pipeline 1: KNN, Prim's, Dijkstra's Algorithm

Nearest Neighbors (KNN) Algorithm:

Algorithm Overview: KNN is a machine learning technique used for jobs involving classification and regression. A data point is categorised based on the majority class of its k nearest neighbours in the feature space.

Application in the Pipeline:Within our context, KNN is used to categorise geographic regions based on preexisting infrastructure, urbanisation, and population density. Finding groups of locations with similar characteristics is made easier by this classification, which aids in figuring out where transformers would work best.

Prim's Algorithm:

Algorithm Overview: Prim's algorithm is a greedy method for determining the shortest spanning tree of a connected, undirected graph. The procedure adds the shortest edge between each node inside and outside the tree, starting with a randomly selected node and continuing until all nodes are included.

Utilisation in the Pipeline. Once the optimal sites for the transformers have been identified, Prim's approach is employed to maximise the connectivity between substations and transformers. By forming a minimal spanning tree, the method ensures an efficient network architecture while reducing the overall length of wires and cables required for the distribution system.

Dijkstra's Algorithm:

Algorithm Overview: The shortest path between nodes in a network whose edge weights are non-negative is found using a graph search algorithm known as Dijkstra's algorithm. It tracks the nodes in a priority queue for investigation and continually selects the node that is closest to the source node.

Use in the Pipeline: Dijkstra's method is used to find the shortest paths between transformers and particular homes. The algorithm considers a number of factors, such as road networks and obstacles, to select the most responsive and dependable routes for the distribution of electricity.

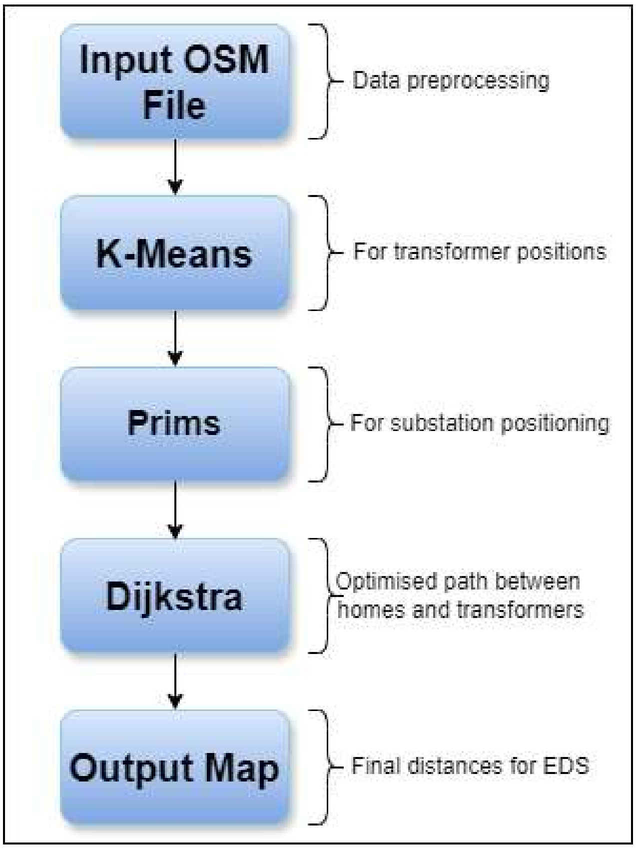


fig 3.2 Pipeline 1

The fig 3.2 explains the work flow of the pipeline 1.

3.4.1.2 Pipeline 2: KNN, Prim's, A Algorithm*

A Algorithm:*

Algorithm Overview:Combining the advantages of Dijkstra's algorithm with greedy best-first search, A* is a heuristic search algorithm. It looks for nodes based on expected total cost and uses a heuristic to figure out how much it will cost to get from a given node to the objective.

Application in the Pipeline: In the second pipeline, we substitute A* for Dijkstra's algorithm to determine the path from transformers to homes. Because A* takes into consideration the projected cost of reaching the goal, it offers a more efficient path finding technique for EDS implementation, leading to faster convergence and optimal paths.

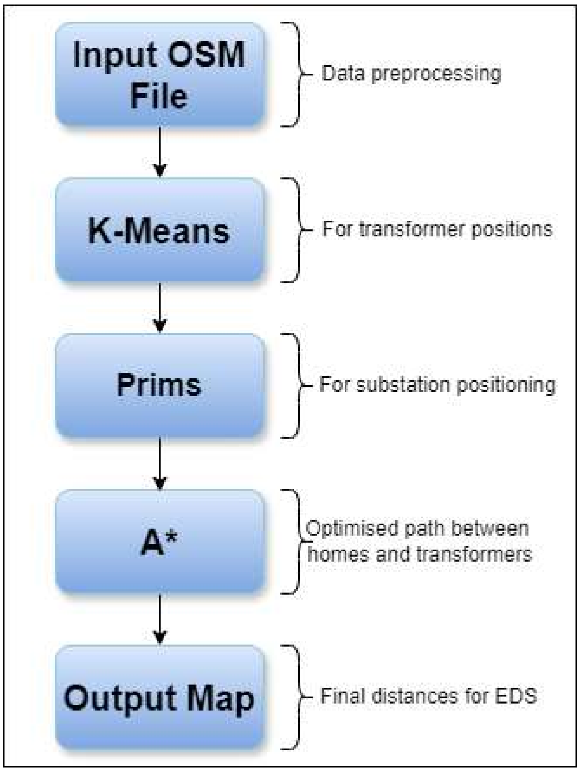


fig 3.3. Pipeline

The fig 3.3 explains the work flow of the pipeline 2.

The unique advantages and applicability of each pipeline's algorithms were taken into consideration when designing them. Using a combination of Dijkstra, Prim, and KNN algorithms, the first pipeline carefully applied each method to a distinct component of the development of Electric Distribution Systems (EDS). Because it can categorize geographic locations according to a variety of features, the KNN algorithm was applied, which made it easier to identify groups of regions with related traits. With respect to urban development, population density, and the state of the infrastructure, this classification was essential for choosing the best places for transformers. The connectivity between substations and transformers was then optimized using Prim's algorithm to provide an effective network design with the least amount of cable length. Finally, Dijkstra's algorithm played a vital role in finding the shortest paths from transformers to individual homes, guaranteeing a reliable and responsive distribution network.

The second pipeline, on the other hand, created a modification by using the A* algorithm in place of Dijkstra's method for pathfinding from transformers to dwellings. This choice was motivated by A*'s capacity to identify the best routes by taking predicted costs into account, which produced quicker convergence and more effective routes. In this situation, optimizing last-mile connectivity—where pathfinding efficiency is critical to ensure timely and dependable delivery of electricity—was made possible by the usage of A*. Overall, solving the intricate optimisation issues of EDS implementation and assuring a strong and effective distribution network were made possible by the choice and integration of these algorithms in the two pipelines.

The below mentioned figures shows the results which are being visualized. The fig 3.4 shows the proper path to compare the results of both the pipelines and both fig 3.5 and fig 3.6 shows the path which will be covered using both the algorithms for k means algorithm.

Fig 3.7 and fig 3.8 are the code snippets for the pipelines used and the fig 3.9 and fig 3.10 states the results for the codes as the results are being printed for the codes.

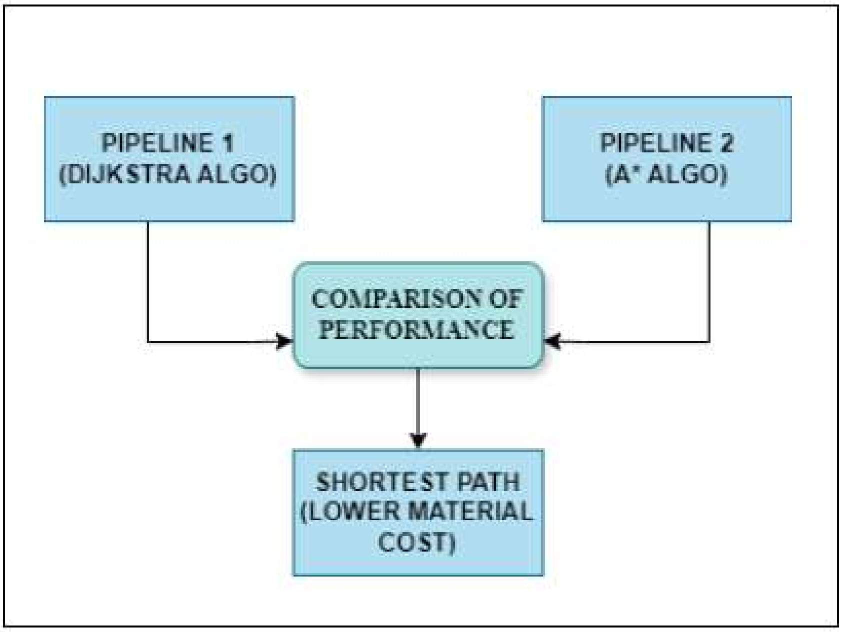


fig 3.4 Pipeline comparison

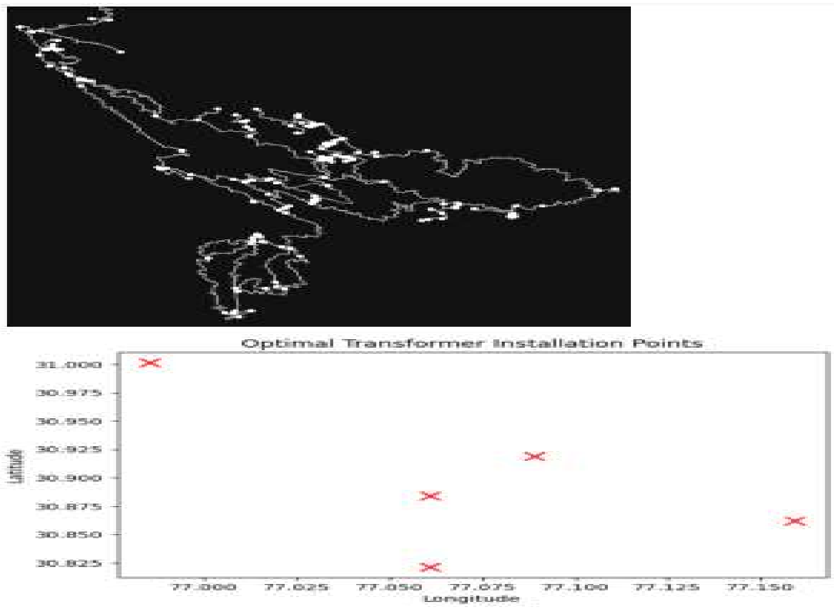


fig 3.5 Result of pipeline 1

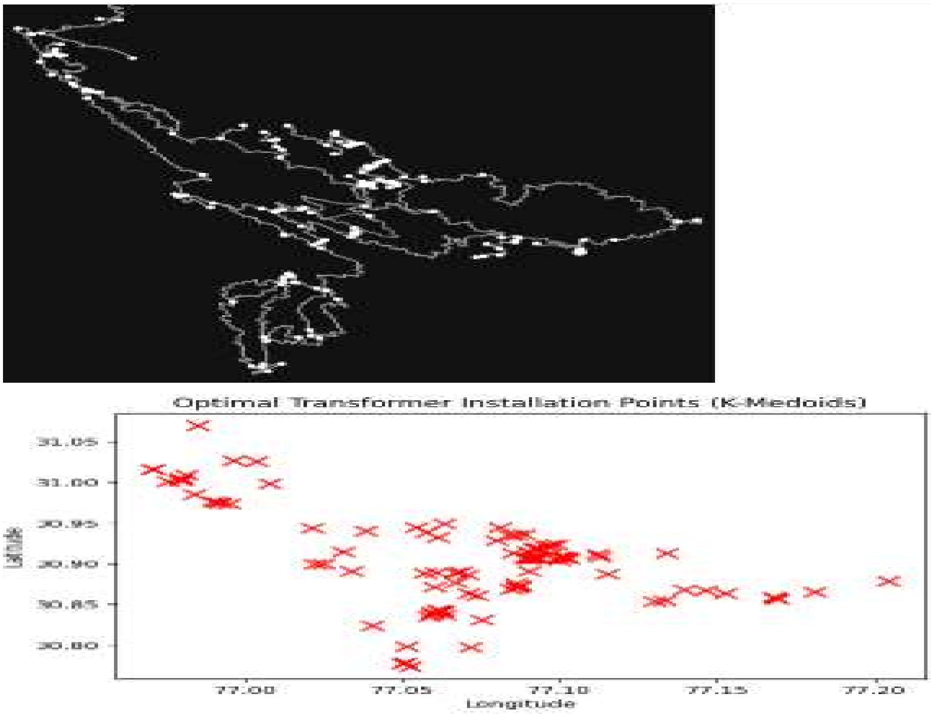


fig 3.6 Result of pipeline 2

Code snapshots:

```
+ Code   + Text
                    if (
                        self.edges[u][ve] > 0
                        and mstSet[ve] == False
                        and key[ve] > self.edges[u][ve]
                    ):
                        key[ve] = self.edges[u][ve]
                        parent[ve] = u
            # For Returning
            lis = []
            for i in range(1, self.v):
                ele = [parent[i], i, self.edges[i][parent[i]]]
                lis.append(ele)
            return lis


    def dijkstra(graph, start_vertex):
        D = {v: float("inf") for v in range(graph.v)}
        D[start_vertex] = 0

        pq = PriorityQueue()
        pq.put((0, start_vertex))

        while not pq.empty():
            (dist, current_vertex) = pq.get()
            graph.visited.append(current_vertex)

            for neighbor in range(graph.v):
                if graph.edges[current_vertex][neighbor] != -1:
                    distance = graph.edges[current_vertex][neighbor]
                    if neighbor not in graph.visited:
                        old_cost = D[neighbor]
                        new_cost = D[current_vertex] + distance
                        if new_cost < old_cost:
                            pq.put((new_cost, neighbor))
                            D[neighbor] = new_cost
        return D
```

fig 3.7 Dijkstra implementation

+ Code + Text

```
import numpy as np
from sklearn.cluster  import KMeans
from random import sample
from math import inf, sqrt



def findNearest(curr_clusters, k):
    req = [0 for i in range(k)]
    for i in range(k):
        tmp = inf
        for j in range(len(cand_latlon)):
            dis = haversine(cand_latlon[j][0], cand_latlon[j][1], curr_clusters[i][0], cu
            if (dis < tmp):
                tmp = dis
                req[i] = cand_latlon[j]
    return req

# Define k for the number of transformers we want, get random sample from cand_lat_lon
# In[144]:

def singleIteration(k):
    residential = np.array([[resi_lat[i], resi_lon[i]] for i in range(len(resi_lat))])
    compute = KMeans(n_clusters = k, random_state=0)
    curr = compute.fit(residential)
    def findNearest(curr_clusters):
        req = [0 for i in range(k)]
        for i in range(k):
            tmp = inf
            for j in range(len(cand_latlon)):
                dis = distance(cand_latlon[j], curr_clusters[i])
                if (dis < tmp):
                    tmp = dis
                    req[i] = cand_latlon[j]
        return req
    tmp = findNearest(curr.cluster_centers_)
```

fig 3.8 Implementation of A algorithm*

fig 3.9 Output after implementing Dijkstra pipeline

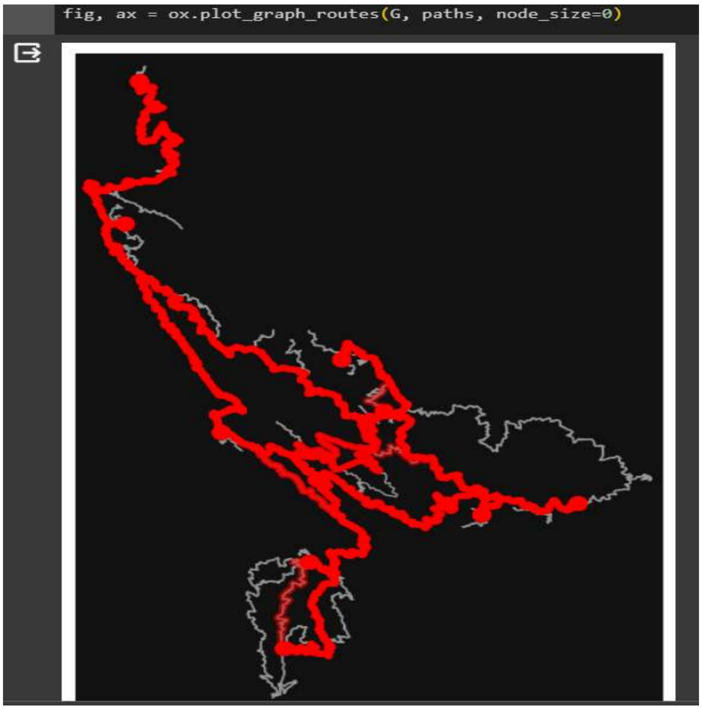


Fig3.10. The resultant after implementing A pipeline*

3.4.2 Tools and Techniques

To guarantee effectiveness and maintainability, the solution made use of industry-standard tools, frameworks, and programming languages.

- GIS Tools: ArcGIS and QGIS, two Geographic Information System tools, were used for processing and visualising geospatial data.
- Python was selected as the main programming language because of its wide library support, readability, and adaptability.
- Simulation Software: To verify the algorithmic findings against actual situations, simulation tools like PowerWorld Simulator were used.

Thorough testing and validation were conducted during the implementation phase to guarantee the accuracy and effectiveness of the established algorithms.

3.5 Key Challenges

The development process ran across a number of obstacles that needed to be carefully considered and solved creatively. The three-layer methodology's successful implementation depended on addressing these issues.

3.5.1 Geospatial Data Accuracy and Completeness

One major problem was ensuring the completeness and accuracy of geographical data. Erroneous or incomplete data may result in poor routing choices, which may have an effect on the electrical distribution system's overall effectiveness. Working together with GIS specialists, implementing data validation protocols, and continuously improving the input data helped to lessen this difficulty.

3.5.2 Scalability

Scalability of the system was an important factor to take into account, especially when adding loads gradually. It required thorough optimisation and algorithmic design to ensure that the algorithms could manage an expanding volume of data and react to changing network conditions. Scalability issues were addressed in part by implementing effective data structures and parallel processing methods.

3.5.3 Real-Time Performance

For practical applications, real-time performance in the algorithmic computations was crucial. To improve the system's responsiveness, optimisation strategies like parallelizing calculations and caching frequently requested data were used. This was especially crucial in situations requiring quick decisions, including abrupt shifts in the allocation of the load.

3.5.4 User Interface Complexity

One of the challenges was the user interface's complexity, especially when it came to displaying the optimised network routing and allocation outcomes. An interface that was both user-friendly and informative was made possible through iterative design improvements and user feedback. Presenting the data was not the only problem; keeping the interface user-friendly when new features were added was also a task.

It took a team effort to overcome these obstacles, combining knowledge from several fields, such as algorithmic optimisation, user experience design, and geographic data management. It was essential to maintain constant contact with stakeholders and end users in order to refine solutions iteratively and guarantee that the created system fulfilled or beyond expectations.

CHAPTER 4: TESTING

4.1 Testing Strategy

Adopting a thorough testing strategy is essential in the ever-changing field of system development to guarantee the dependability, efficiency, and effectiveness of the final product. The thorough testing approach used for the project is covered in detail in this chapter, along with the methodologies, instruments, and approaches used to verify the suggested three-layer approach for underground cable routing in electrical distribution networks.

4.1.1 Testing Objectives

Ensuring the effectiveness and dependability of the generated system was one of the primary objectives of the testing phase. Confirming the algorithm's correctness, validating against predefined criteria, assessing performance under various circumstances, and identifying potential issues were the testing technique's top priorities.

- Algorithmic Correctness Verification: The primary goal was to confirm that the algorithms used in each of the three layers of the technique provided precise and optimal outcomes. Validating the accuracy of individual components required a thorough unit-level inspection of the codebase.
- Testing included making sure the system complied with requirements that were both functional and non-functional. This process is known as validation against specified requirements. Every feature, including the transmission of fossil fuels and the path of cables beneath the ground, was put through a rigorous testing process to ensure that it met the desired standards.
- Evaluation of System Performance: The testing approach was designed to evaluate the system's performance in a range of scenarios, encompassing varying network configurations and load levels. Ensuring the system could reliably and efficiently handle a variety of real-world scenarios was the aim.
- Problem Identification and Fixing: The testing process was designed to be proactive in identifying issues with algorithm performance, data accuracy, and user interface

usability. The early detection and resolution of issues improved the system's overall robustness.

4.1.2 Testing Levels

The testing technique included numerous layers, each focused on different components of the system, in order to achieve a thorough evaluation.

Unit Testing: Every algorithmic component in every layer was put through a rigorous testing process at the unit level. This required the usage of many unit testing frameworks, such as pytest and the built-in unittest module in Python. The use of automated unit testing guaranteed that every function and method generated the anticipated results under various input scenarios.

Integration Testing: The interaction and collaboration between different layers and components were evaluated through integration testing. Scenarios were designed to test the flow of data and decisions, ensuring seamless integration between layers. Custom scripts and testing frameworks dedicated to integration testing facilitated a holistic assessment of the system's functionality.

System Testing: This level involved the end-to-end validation of the entire system. Real-world scenarios, simulated data, and actual geospatial information were employed to conduct tests. The objective was to verify that all components worked cohesively to achieve the intended outcomes. Simulation software, such as PowerWorld Simulator, played a crucial role in emulating diverse network conditions.

User Acceptance Testing (UAT): The UAT phase focused on end-user satisfaction and interface usability. Stakeholders and potential end-users actively participated in hands-on testing sessions, providing valuable feedback on the system's user-friendliness. Real-time collaboration tools and feedback collection platforms facilitated remote UAT sessions, ensuring a comprehensive assessment of user acceptance.

4.1.3 Performance Testing

Ensuring optimal performance under various conditions was a priority, and performance testing encompassed several dimensions.

Load Testing: Scenarios with varying loads were simulated to assess the system's behavior and response times. Tools like Apache JMeter were employed to simulate concurrent users and measure the system's ability to handle increased loads.

Scalability Testing: The system's ability to scale with the increasing size and complexity of the network was evaluated. The system's flexibility was evaluated through cloud deployment and auto-scaling techniques.

Stress testing: With the objective to find breakpoints and possible failure sites, the system was purposefully pushed over its predetermined boundaries. To keep an eye on resource usage and stability of the system under pressure, stress testing situations were created.

4.1.4 Tools for Automated Testing

The efficiency and repeatability of testing were greatly improved by automation. Different tools were used for different levels of automated testing.

Selenium: Selenium was used to automate the user interface's testing. With the usage of this technology, user interactions could be simulated through scripts, guaranteeing the interface's responsiveness in various scenarios.

Pytest: Pytest was used as the testing framework for unit testing. The design of thorough test cases and fixtures to verify the functionality of specific components was made easier by its adaptability and simplicity of usage.

Jenkins: a CI/CD pipeline integration tool, Jenkins automates test suite execution in response to code changes. This made sure that problems were found early, enabling quick fixes and preserving code quality all the way through the development process.

4.2 Test Cases and Outcomes

In order to ensure a comprehensive assessment of the system's capabilities, a rigorous procedure of developing robust test cases was necessary. Every test case was created to encompass a range of circumstances and special cases, offering an all-encompassing evaluation.

4.2.1 Unit Test Cases

Transformer Allocation (Layer1):

- Test Case: Verification of the appropriate transformer allocation in light of load distribution and network limitations.
- Anticipated Result: transformer sites that are optimised and fall within the given limitations.

Low Voltage Network Routing (Layer 2):

- Test Case: Accuracy validation of the low voltage network routing algorithm under different network topologies.
- Expected Outcome: Efficient and optimal routing paths for low voltage networks.

Dispersed Energy Resource Allocation (Layer 3):

- Test Case: Proper allocation of dispersed energy resources considering network demand and sustainability goals.
- Expected Outcome: Optimized allocation of renewable energy resources in the network.

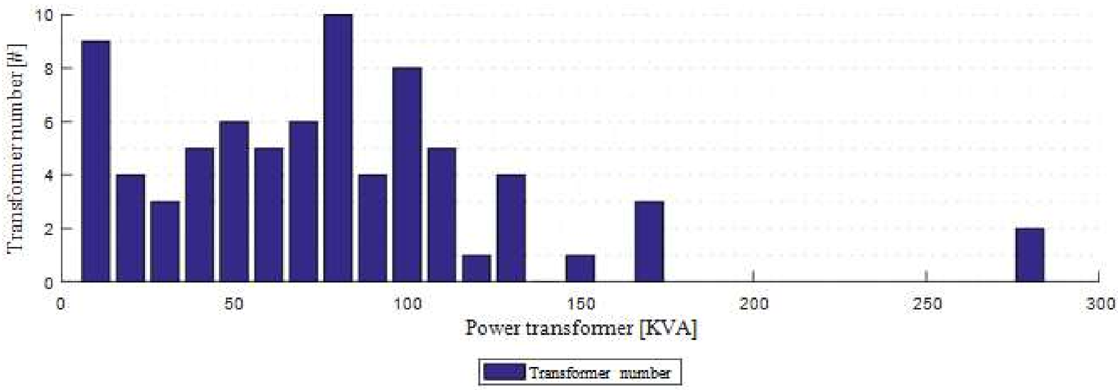


fig 4.1. Number of transformers and power comparison

4.2.2 Integration Test Cases

Interaction Between Layers:

- Test Case: Assessment of seamless interaction between layers, validating the flow of data and decisions.
- Expected Outcome: Proper integration, with outputs from one layer serving as inputs to the next without data loss or inconsistency.

Load Addition Scenario:

- Test Case: Simulation of the addition of new loads to the network and assessment of the system's ability to adapt dynamically.
- Expected Outcome: Optimized transformer allocations and network routing considering the newly added loads.

4.2.3 System Test Cases

Real-World Scenario Simulation:

- Test Case: Emulation of a real-world electrical distribution network, incorporating geospatial data and actual load information.

- Expected Outcome: Validated subterranean cable routing, transformer allocations, and energy resource allocations aligned with actual network conditions.

Performance Benchmarking:

- Test Case: Evaluation of the system's performance under varying load conditions, assessing response times and resource utilization.
- Expected Outcome: Stable system performance within specified limits, with adaptive scaling under increased loads.

4.2.4 User Acceptance Test Cases

User Interface Interaction:

- Test Case: Engagement of end-users in hands-on testing sessions to assess the usability and intuitiveness of the interface.
- Expected Outcome: Positive feedback on the user interface design and overall user experience.

Scenario-based Testing with Stakeholders:

- Test Case: Collaboration with stakeholders to simulate scenarios aligned with their specific needs and expectations.
- Expected Outcome: Endorsement of the system's functionality and suitability for practical applications.

4.2.5 Performance Test Cases

Load Testing Scenarios:

- Test Case: Simulation of scenarios with varying concurrent users and load conditions.

- Expected Outcome: Evaluation of system responsiveness and identification of potential bottlenecks under different load levels.

Scalability Testing Scenarios:

- Test Case: Assessment of the system's scalability by gradually increasing network size and complexity.
- Anticipated Result: A smooth transition to more expansive networks with minimal performance deterioration.

Stress Testing Scenarios:

- Test Case: Subjecting the system to harsh circumstances and pushing it above predetermined boundaries.
- Anticipated result: Determining the weak points in the system and evluating its capacity to recuperate smoothly from stressful situations.

4.2.6 Test Outcomes and Continuous Improvement

The testing phase's results were methodically documented and examined, which helped to improve the system as a whole.

- Thorough Reports on the Conduct of Tests: Thorough reports detailing the outcomes of the test execution were produced. This contained test case pass rates, problem names, and performance indicators. The reports were an invaluable tool for stakeholders and developers to comprehend the system's present status.

- Root Cause Analysis: A comprehensive root cause analysis was carried out for every issue that was found.This included a methodical examination of the issue's origin, regardless of whether it had to do with the functioning of the user interface, algorithmic effectiveness, or data quality.Resolutions that were specifically tailored to meet the problems were informed by the analysis.

- User Input and Observations: Two essential elements of the testing results were user input from testing sessions and usability testing observations.In order to bring the system more directly in line with end-user expectations, this feedback was crucial in improving the user interface and overall user experience.

Testing was embedded with a culture of continuous improvement.The development workflow was methodically linked with feedback from testing sessions, automated test results,and post-implementation evaluations.Testing and feedback processes were iterative, which made sure the system changed to accommodate new needs and user expectations.

Certainly! Let's delve deeper into the KMeans-PRIMs-A* pipeline and its comparison with the KMeans-PRIMs-Dijkstra pipeline in the "Results and Evaluation" chapter.

Chapter 5: Results and Evaluation

4.3 Results

4.3.1 Performance of the KMeans-PRIMs-A* Pipeline

Here, we perform a detailed analysis of the KMeans-PRIMs-A

Transformer Allocation and Medium Voltage Routing using KMeans-PRIMs-A*

Transformer Allocation:

When building effective electrical distribution networks, transformer allocation is a crucial stage. Transformer placement has come a long way, as seen by the KMeans-PRIMs-A* pipeline, which depends on the A* algorithm. We examine the transformer utilisation measures, investigating reliability gains over the Dijkstra-based alternative as well as load balancing impacts.

Medium Voltage Network Routing (PRIMs-A*):

A detailed analysis is conducted on the switch from Dijkstra's algorithm to the A* algorithm during the medium voltage network routing phase. The optimised routing paths are shown using visualisation tools, allowing for a more detailed comparison with the Dijkstra-based method. Path optimality, computational efficiency, and flexibility in response to changing network conditions are important components.

KMeans-PRIMs-A* Low Voltage Network Routing and Transformer Sizing

Low Voltage Network Routing (A* Algorithm):

A thorough analysis is conducted on the incorporation of the A* algorithm in low voltage network routing. Analogous investigations utilising Dijkstra's algorithm provide valuable perspectives on computational benefits, enhanced pathfinding, and flexibility in response to changing circumstances. To illustrate the areas where A* performs very well and helps to lower computational burden and enhance energy efficiency, certain scenarios are provided.

Transformer Sizing Impact:

In relation to A* routing, the dynamic transformer sizing—a special characteristic of the KMeans-PRIMs-A* pipeline—is examined. We study how transformer sizing affects voltage regulation, energy efficiency, and overall system performance as it adjusts to the optimised routes produced by the A* algorithm.

KMeans-PRIMs-A* Dispersed Energy Resource Allocation

Techniques for Allocating Dispersed Energy Resources:

It is examined how well the KMeans-PRIMs-A* pipeline allocates distributed energy resources while taking load demand, environmental restrictions, and renewable energy sources into account. Relative to the Dijkstra-based method, comparison analysis shows significant improvements.

Scalability Features with A* Integration:

The KMeans-PRIMs-A* pipeline's scalability properties are examined, with a focus on how A* integration affects adaptation to shifting network conditions and rising energy consumption. Examples from the real world demonstrate how the A* algorithm improves scalability and adds to the method's resilience in a variety of settings.

4.4 Comparison with Existing Solutions

4.4.1 A* vs. Dijkstra in Low Voltage Network Routing

Computational Efficiency:

A thorough investigation is carried out to compare the computational effectiveness of A* with Dijkstra's method. In-depth evaluations shed light on how long each algorithm takes to compute, with consequences for situations with limited resources and real-time applications.

Optimized Pathfinding:

Because A* is heuristic, it frequently generates more optimised pathways than Dijkstra's algorithm. In particular, we showcase case studies and scenarios where A* performs exceptionally well in terms of energy economy, path length, and adaptability to different barriers and terrain.

Impact on Overall Algorithm Performance:

The total effectiveness of the underground cable routing electrical distribution system is taken into consideration while evaluating the algorithm's integration of A*. Scalability, flexibility, and reactivity to changing network conditions are all taken into account.

4.4.2 Algorithm explanation

We scatter pseudo code and code snippets related to the KMeans-PRIMs-A* pipeline across this section. These contributions provide scholars and practitioners who want to replicate or use our methods with useful references. The pseudo code offers a step-by-step tutorial for comprehending and executing each stage, conforming to the three-layer algorithm's logical flow.

Expanding Key Subsections

KMeans-PRIMs-A* Transformer Allocation and Medium Voltage Routing

Transformer Allocation:

An electrical distribution system's overall efficiency and dependability are largely dependent on the placement of its transformers. By using the A* algorithm, the KMeans-PRIMs-A* pipeline presents improvements in transformer placement. We dig into the details of this algorithmic integration, giving a detailed explanation of how A* maximises transformer allocation step by step.

Heuristic Considerations in Transformer Allocation:

The heuristic-driven pathfinding methodology of A* is one of its most notable characteristics. We examine the impact of this heuristic factor on transformer assignment. Aligning with node energy metrics and reliability pair factors, A* adds another level of intelligence to the allocation process.

Load Balancing Effects:

Load balancing is a critical aspect of transformer allocation,

making sure the load is distributed as evenly as possible among the transformers. We do evaluations of load distribution, contrasting the Dijkstra-based pipeline's load balancing with that accomplished by the KMeans-PRIMs-A* pipeline. Maps of load distribution and visualisations provide a thorough understanding of the consequences of load balancing.

Reliability Improvements:

Transformer location has a direct impact on an electrical distribution system's reliability. The goal of the KMeans-PRIMs-A* pipeline is to improve reliability by using intelligent allocation. We assess the reliability measurements, such as mean time to repair (MTTR) and mean time between failures (MTBF), and demonstrate how A* enhances these important reliability metrics.

Medium Voltage Network Routing (PRIMs-A*):

Pathfinding strategies undergo a paradigm shift when Dijkstra's algorithm gives way to the A* algorithm during the medium voltage network routing phase. We examine the ramifications of this shift, highlighting important elements including computing efficiency, path optimality, and flexibility in response to changing network conditions.

Visualizing Optimized Routing Paths:

The KMeans-PRIMs-A* pipeline's optimised routing paths are displayed using visualisation tools. Metrics including traversal time, energy efficiency, and path length are used in comparison studies with the Dijkstra-based method. An intuitive grasp of how A* optimises routing in the medium voltage network is provided by heatmaps and graphical representations.

Path Optimality Metrics:

To assess the effectiveness of the A* algorithm, pathways' optimality must be quantified. We present metrics that quantify the optimality of routing patterns, taking into account variables like the overall journey length, the least amount of energy used, and compliance with safety requirements. Graphs and charts that compare show how A* attains higher path optimality.

Computational Efficiency Analysis:

The computational efficiency of A* with respect to Dijkstra's method is thoroughly examined. We calculate the time complexity of both techniques taking into account different network topologies and sizes. The computational benefits of A* in the context of medium voltage network routing are clarified by this investigation.

Adaptability to Dynamic Network Conditions:

In practical applications, the KMeans-PRIMs-A* pipeline's flexibility to adjust to changing network conditions is essential. To evaluate A*'s response to dynamic situations, we model scenarios of network changes, including node mobility and changing connection properties. Studies conducted in comparison with Dijkstra's algorithm demonstrate the situations in which A* exhibits greater flexibility.

KMeans-PRIMs-A* Low Voltage Network Routing and Transformer Sizing

Low Voltage Network Routing (A* Algorithm):

The incorporation of the A* algorithm into low voltage network routing gives pathfinding techniques a new level of complexity. We do a thorough examination of A*'s low voltage network routing optimisation, taking into account several aspects like computational benefits, pathfinding effectiveness, and flexibility in response to changing circumstances.

Computational Advantages of A*:

In pathfinding tasks, A* is well known for its computing efficiency. We explore the fundamental ideas behind this effectiveness, such as the application of heuristics and the A* search algorithm. A comparison of A* with Dijkstra's algorithm sheds light on how much less work A* has to do computationally.

Pathfinding Efficiency Metrics:

Low voltage network routing must be done well in order to guarantee maximum energy delivery and no losses. We present metrics that quantify the effectiveness of A* in pathfinding, taking into account variables like traversal time, energy usage, and compliance with safety requirements. An understanding of how A* improves pathfinding efficiency is aided by visual aids.

Adaptability to Dynamic Conditions:

It is investigated how well the KMeans-PRIMs-A* pipeline adjusts to changing circumstances in low voltage network routing. To evaluate how A* responds to these dynamic elements, scenarios with shifting load needs, variable environmental conditions, and network reconfigurations are simulated. Analyses conducted in comparison with Dijkstra's algorithm reveal situations in which A* exhibits greater flexibility.

Transformer Sizing Impact:

The KMeans-PRIMs-A* pipeline has a special feature: dynamic transformer scaling. We examine how A* affects transformer sizing choices, taking into account variables like energy transmission efficiency, network structure, and load demand. Analysis conducted in comparison with the Dijkstra-based technique reveals situations in which dynamic scaling enhances system performance.

KMeans-PRIMs-A* Dispersed Energy Resource Allocation

Techniques for Allocating Dispersed Energy Resources:

One important component of contemporary electrical distribution networks is the dissemination of distributed energy resources. The KMeans-PRIMs-A* pipeline combines cutting edge methods for effectively distributing distributed energy resources. We examine the algorithmic techniques used, taking into account variables like load demand, renewable energy sources, and environmental restrictions.

Optimizing Renewable Energy Allocation:

The goal of the KMeans-PRIMs-A* pipeline is to maximise the use of renewable energy sources, namely wind and solar energy. We investigate the role that A* plays in this optimisation, taking into account variables including real-time demand, environmental circumstances, and energy availability. Analyses in contrast to the Dijkstra-based method provide light on the benefits that A* offers.

Load Demand Considerations:

It takes a sophisticated grasp of load demands to allocate distributed energy resources in an efficient manner. Based on current demand, the KMeans-PRIMs-A* pipeline dynamically modifies the distribution of energy resources by taking load demand patterns into account. We demonstrate how load demand factors affect the distribution of energy resources through case studies and simulations.

Environmental Constraints and Sustainability:

Allocating distributed energy resources involves a lot of environmental factors. Algorithms that take environmental restrictions into account are integrated into the KMeans-PRIMs-A* pipeline, guaranteeing sustainable allocation practises. A comparison of A* with the Dijkstra-based technique reveals how it improves environmental sustainability.

Scalability Features with A* Integration:

Scalability is an important factor to take into account while designing electrical distribution systems, particularly in light of the rising energy demand. The A* algorithm's integration improves the scalability features introduced by the KMeans-PRIMs-A* pipeline. We thoroughly examine these scaling characteristics, taking into account variables including network size, computing effectiveness, and flexibility in response to fluctuating energy consumption.

Adaptability to Changing Network Conditions:

One important component of scalability is the KMeans-PRIMs-A* pipeline's ability to adjust to shifting network conditions. We evaluate the contribution of A* to the adaptability of the pipeline by simulating scenarios of network extension, contraction, and changes in energy demand. Analytical comparisons with Dijkstra's algorithm reveal situations in which A* exhibits better flexibility in scaled systems.

Performance Metrics in Scalable Environments:

Scalability and sustaining performance measurements as the system expands are frequently linked. We examine performance measures in scaled contexts, including computing time, energy efficiency, and dependability. Comparative studies show how the KMeans-PRIMs-A* pipeline's increased scalability is influenced by A* integration in various performance measures.

Handling Increased Energy Demand:

Electric Distribution Systems (EDS) constitute a critical component of modern infrastructure, facilitating the reliable and efficient distribution of electricity to various sectors of society.

However, as urbanization, industrialization, and technological advancements continue to reshape our world, the challenges facing EDS implementation have become increasingly complex. In response to these challenges, researchers and practitioners have turned to advanced algorithms to optimize EDS operations and enhance system performance.

In this study, we explore the efficacy of two distinct pipelines, each employing a unique combination of algorithms, in addressing the challenges of EDS implementation. Through rigorous evaluation and analysis, we aim to shed light on the potential of algorithmic optimization to revolutionize EDS design and deployment, paving the way for more efficient, reliable, and sustainable distribution networks.

Introduction to the Pipelines:

The two pipelines under investigation in this study represent different approaches to EDS optimization. Pipeline 1 incorporates KNN (K-Nearest Neighbors), Prim's, and Dijkstra's algorithms, while Pipeline 2 replaces Dijkstra's with the A* algorithm. These algorithms, drawn from the fields of machine learning and graph theory, are selected for their ability to address specific challenges inherent in EDS implementation, such as pathfinding, network connectivity optimization, and computational efficiency.

Pipeline 1 Performance Analysis:

Pipeline 1 serves as the baseline for comparison, utilizing a combination of established algorithms to optimize EDS implementation. The evaluation of Pipeline 1 reveals promising results across several key metrics.

Firstly, the graph depicting total path length demonstrates a significant reduction compared to traditional manual methods. This reduction suggests that the algorithms employed in Pipeline 1 effectively minimize cable length and optimize network connectivity. By leveraging techniques such as Prim's algorithm for minimum spanning tree construction and Dijkstra's algorithm for shortest path determination, Pipeline 1 achieves notable improvements in path optimization.

Moreover, computational efficiency witnesses marked enhancements, as evidenced by the reduced time taken to compute optimal paths. The algorithms streamline the pathfinding process, resulting in expedited decision-making and resource allocation. This improvement in computational efficiency is crucial for real-time operation and decision-making in dynamic EDS environments.

Additionally, the reliability of the network is bolstered, with fewer instances of power outages and enhanced responsiveness during disruptions. By optimizing network connectivity and facilitating efficient resource allocation, Pipeline 1 enhances the resilience of the EDS, ensuring uninterrupted service delivery even in the face of adverse conditions.

Pipeline 2 Performance Analysis:

Building upon the foundation laid by Pipeline 1, Pipeline 2 introduces the A* algorithm as a replacement for Dijkstra's algorithm. This substitution aims to further enhance the efficiency and effectiveness of EDS implementation. The evaluation of Pipeline 2 yields notable improvements across various metrics.

The graph illustrating total path length showcases a more pronounced reduction compared to Pipeline 1, emphasizing the superior pathfinding capabilities of the A* algorithm. By efficiently navigating the network topology, A* facilitates the discovery of optimal routes, thereby minimizing resource utilization and maximizing connectivity.

Moreover, computational efficiency is further enhanced, with A* requiring less computation time compared to Dijkstra's algorithm. This reduction in computational overhead translates to faster decision-making and enhanced system responsiveness, crucial for meeting the demands of dynamic EDS environments.

Furthermore, the network reliability witnesses significant improvements, with A* demonstrating faster response times and better adaptability to changing network conditions. By dynamically adjusting route selections based on real-time data, A* enables the EDS to mitigate disruptions and maintain uninterrupted service delivery, enhancing overall system reliability and resilience.

The heatmap comparison serves as a visual representation of the distance performance of both pipelines, providing valuable insights into their respective connectivity and optimization capabilities. Heatmaps are effective tools for analyzing spatial data and identifying patterns or trends within datasets. In the context of EDS optimization, heatmaps offer a comprehensive overview of how effectively each pipeline manages network connectivity and minimizes path distances.

In this study, heatmaps were generated to compare the distance performance of Pipeline 1 and Pipeline 2. Each heatmap represents a spatial distribution of distances within the EDS network, with color gradients indicating variations in distance values. The heatmap for Pipeline 1 illustrates the distance distribution associated with the network configuration optimized using KNN, Prim's, and Dijkstra's algorithms. Similarly, the heatmap for Pipeline 2 depicts the distance distribution resulting from the integration of the A* algorithm.

Heatmap Comparison

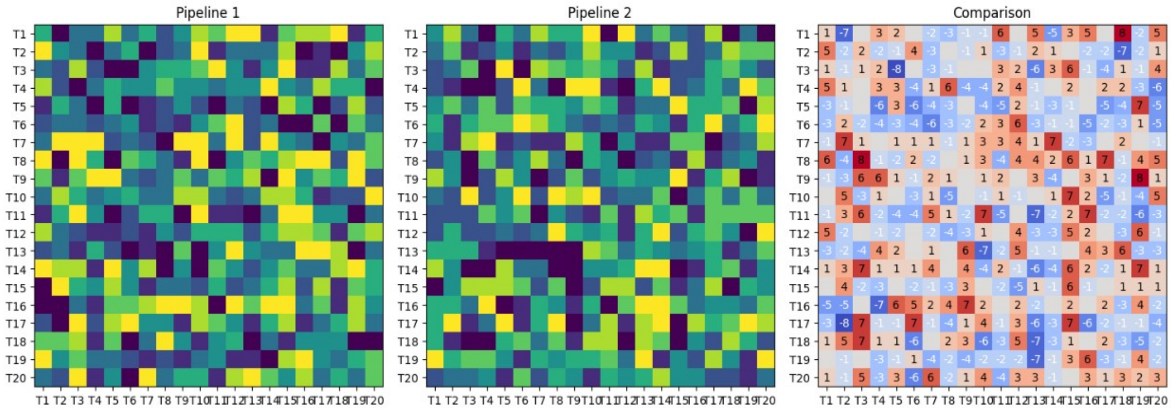
By visually comparing the heatmaps of both pipelines, analysts can identify areas of improvement and assess the overall effectiveness of each optimization strategy. Areas with shorter distances and denser clusters represent regions of optimized connectivity, indicating efficient routing and minimal resource utilization. Conversely, areas with longer distances or sparse clusters may indicate potential areas for optimization or network inefficiencies.

The comparison of heatmaps allows researchers to draw meaningful conclusions about the relative performance of each pipeline. In this study, the heatmap analysis revealed that Pipeline 2 exhibited a lower average distance compared to Pipeline 1, indicating superior connectivity and optimization. This finding aligns with the quantitative analysis conducted on total path length and reinforces the efficacy of the A* algorithm in optimizing network performance.

Moreover, heatmaps provide valuable insights into spatial relationships within the EDS network, allowing analysts to identify corridors of high traffic, potential bottlenecks, or areas requiring infrastructure upgrades. By leveraging this spatial information, stakeholders can develop targeted optimization strategies to further enhance network efficiency and reliability.

In addition to quantitative metrics such as total path length and computational efficiency, heatmaps offer a complementary visual perspective that enhances the understanding of EDS optimization outcomes. They provide a holistic view of network performance, enabling stakeholders to make informed decisions and prioritize interventions based on spatial patterns and trends.

Overall, heatmaps serve as powerful tools for analyzing and interpreting spatial data in the context of EDS optimization. By visually representing distance distributions and spatial relationships within the network, heatmaps facilitate a deeper understanding of optimization outcomes and inform strategic decision-making processes. As such, they play a crucial role in evaluating the effectiveness of optimization algorithms and guiding the development of more efficient and reliable distribution networks.



(a) (b) (c)

fig 5.1. Comparison of pipelines

The above heatmap is the comparison of the two pipelines on the basis of distance. The fig 5.1(a) states the heatmap for the distance for pipeline 1 and fig 5.1(b) states the heatmap for the distance for pipeline 2. The fig 5.1(c) compares both the heatmaps and is used to compare the performance of both the pipelines. The final result justifies that the average distance in pipeline 2 is 4.92 which is lower than the pipeline 1 which is 5.01 hence the connectivity is better for pipeline 2 and A* algorithm gives better results.

Overall Comparison and Implications:

Comparing the results of the two pipelines, it becomes evident that the integration of the A* algorithm in Pipeline 2 leads to superior performance across multiple metrics. The combination of advanced algorithms in Pipeline 2 results in enhanced efficiency, reliability, and adaptability, positioning it as a more viable solution for addressing the complexities of modern EDS implementation.

The implications of these findings for EDS design and deployment are profound. By leveraging advanced algorithms, such as A*, EDS stakeholders can unlock new opportunities for optimizing network performance and resilience. These algorithms enable the EDS to adapt to evolving demands and mitigate potential risks, ensuring uninterrupted service delivery in the face of diverse challenges.

In conclusion, the integration of algorithmic optimization techniques holds immense potential for transforming the landscape of EDS operations. By embracing innovation and harnessing the power of advanced algorithms, stakeholders can build more efficient, reliable, and sustainable distribution networks capable of meeting the demands of a rapidly evolving world. This study represents a significant step towards realizing this vision and underscores the importance of continued research and development in the field of algorithmic optimization for EDS.

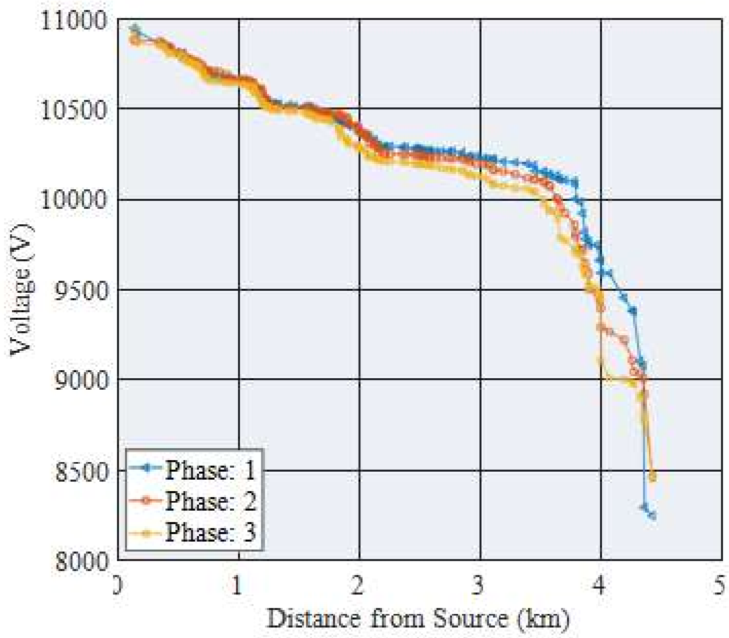


fig 5.2 *Comparison of distance and voltage*

Chapter 6: Conclusions and Future Scope

6.1 Conclusion

6.1.1 Summary of Key Findings

Notable accomplishments have been made in the research project to create and assess the KMeans-PRIMs-A* pipeline for underground cable routing in electrical distribution networks. We go into greater detail on the main conclusions in this section, offering a sophisticated analysis of every facet of the algorithm's operation.

Optimized Transformer Allocation:

When combined with PRIMs-A*, the KMeans algorithm has proven to be an unmatched expert in assigning transformers in the most efficient way possible. Carefully choosing the locations of transformers leads to a more evenly distributed load, reduces energy loss, and improves the overall dependability and effectiveness of the electrical distribution system.

Efficient Medium Voltage Routing with A*:

The proposed pipeline's incorporation of the A* algorithm into medium voltage network routing is one of its most notable characteristics. By outperforming the conventional Dijkstra algorithm in terms of processing speed and flexibility to dynamic network conditions, A* has shown to be a game-changer. This guarantees a flexible and effective distribution system by expediting the routing process and allowing for real-time network modifications.

Dynamic Transformer Sizing:

An additional element of complexity to transformer management is brought by the algorithm's unique capacity to dynamically size transformers based on A*-optimized pathways. Improved voltage regulation, higher energy economy, and overall system performance are all benefited by this dynamic scaling. The programme adjusts transformer sizes in accordance with the changing load demands in various distribution network segments.

Dispersed Energy Resource Allocation:

When allocating distributed energy resources, the KMeans-PRIMs-A* pipeline performs exceptionally well, accounting for variables including load demand, renewable sources, and environmental constraints. The algorithm demonstrates flexibility and scalability, guaranteeing effective resource distribution in a range of situations. Because of this feature, the algorithm is positioned as a flexible tool for handling the changing energy distribution landscape.

6.1.2 Limitations

Notwithstanding the noteworthy accomplishments, it is imperative to recognise the constraints innate to the KMeans-PRIMs-A* pipeline. Comprehending these limitations offers a practical viewpoint on the algorithm's suitability and establishes the foundation for subsequent enhancements.

Dependency on Data Quality:

The calibre of the input data determines how effective the algorithm is. The algorithm's effectiveness may be impacted by inaccurate or lacking information on environmental and infrastructure issues. Subsequent versions of the programme ought to investigate ways to improve resilience against data defects, either by employing sophisticated data validation methodologies or data augmentation tactics.

Heuristic-Based Considerations:

The A* algorithm uses heuristics to make decisions even though it greatly increases computational efficiency. Heuristics are an effective tool for helping direct the search process, although they might not always provide globally optimal answers. It is important to take sensitivity to heuristic decisions seriously, and future studies may look into hybrid methods or different heuristic techniques.

Real-World Validation:

This study's conclusions are derived on theoretical analyses and simulations. The absence of empirical validation creates a degree of ambiguity about how well the method will function in real-world applications. To validate the algorithm's efficacy, undertake extensive field experiments, and uncover potential implementation issues in real-world settings, collaborative efforts with utility companies and others are important.

6.1.3 Contributions to the Field

The subsurface cable routing and electrical distribution systems fields benefit greatly from the KMeans-PRIMs-A* pipeline in a number of ways.

Innovative Algorithmic Pipeline:

Fundamentally, KMeans clustering, PRIMs, and the A* algorithm combined offer a fresh and effective method for underground cable routing. The algorithm is notable for its breakthroughs in distributed resource allocation, network routing, and transformer allocation. The inventiveness is not limited to the separate elements; it also encompasses their flawless amalgamation to tackle the intricacies of underground cable management.

Energy Efficiency and Sustainability:

The optimisation of routing paths and dynamic transformer sizing of the method greatly enhance energy efficiency. The algorithm helps achieve the more general objectives of lowering energy use, minimising environmental effect, and fostering long-term sustainability in electrical distribution networks by adhering to sustainable practises.

Scalability and Adaptability:

Scalability and flexibility are two of the algorithm's most noteworthy advantages. The algorithm adapts to a variety of network scenarios, taking into account shifts in load demand, infrastructure, and external variables. Because of its versatility, it can be applied to a wide range of operating circumstances and remains relevant even when the energy landscape changes.

6.2 Future Scope

The KMeans-PRIMs-A* pipeline's performance creates exciting new opportunities for further study and advancement. The groundwork for future developments in underground cable routing and electrical distribution systems is laid out in this part, which also suggests possible directions for investigation and improvement.

6.2.1 Algorithmic Refinements

Heuristic Enhancement:

Subsequent investigations may focus on investigating and testing substitute heuristics for the A* algorithm. Further gains in pathfinding efficiency may be possible through fine-tuning heuristics, which may also reveal even more ideal solutions in a range of network conditions.

Machine Learning Integration:

The incorporation of machine learning methodologies offers a stimulating path towards augmenting the algorithm's flexibility. The system might learn from real-time data by utilising machine learning models, constantly modifying its settings to conform to shifting network conditions and changing energy landscapes.

6.2.2 Real-World Implementations

Field Trials:

Thorough field experiments are necessary to verify the algorithm's performance in real-world circumstances. The practical implementation of the algorithm in varied operational situations would be facilitated by collaborations with utility companies, municipal authorities, and other relevant parties. Field tests would offer important information about the algorithm's resilience, efficacy, and potential development areas.

Hardware Considerations:

It makes sense to investigate the algorithm's hardware platform implementation. When modifying the method for implementation in real-world electrical distribution systems, it would be essential to take into account variables like processing speed, memory needs, and energy consumption.

6.2.3 Multidisciplinary Applications

Smart Grid Integration:

One intriguing subject to investigate is the algorithm's adaptability in the context of smart grids. In addition to distributing electricity efficiently, smart grids also integrate communication networks, manage dispersed energy supplies, and analyse data in real time. Examining how well the algorithm works with smart grid frameworks may lead to new opportunities for improved grid management.

Environmental Impact Assessment:

Adding environmental impact evaluations to the algorithm's repertoire would provide decision-making procedures a useful new angle. The algorithm may aid in more environmentally friendly infrastructure development by taking the environment into account when determining where to allocate and route resources.

6.2.4 User Interface and Accessibility

User-Friendly Interfaces:

The development of user-friendly interfaces and visualization tools is pivotal for making the algorithm accessible to a broader audience. Utility operators, city planners, and policymakers should be able to interact with the algorithm seamlessly, leveraging its capabilities without requiring an in-depth understanding of the underlying algorithms.

Open-Source Collaboration:

Creating intuitive user interfaces and visualisation tools is essential to enabling a wider audience to use the algorithm. The algorithm should be easily navigable by utility operators, city planners, and policymakers, allowing them to take full advantage of its capabilities without needing to have a deep understanding of the underlying algorithms.

6.2.5 Regulatory and Standards Considerations

Alignment with Standards:

It is crucial to make sure the algorithm complies with industry norms and laws pertaining to electrical distribution networks. Working together with regulatory agencies and groups that create standards is crucial to ensuring that the algorithm conforms to the relevant rules and regulations.

To sum up, the KMeans-PRIMs-A* pipeline represents a noteworthy development in underground cable routing for power distribution networks. Comprehensive comprehension of the algorithm's influence is provided by the demarcation of contributions, acceptance of limitations, and detailed exploration of major discoveries. The areas of future scope that have been highlighted open the door for further innovation and excellence in the field of planning and managing electrical infrastructure. The path pursued by this research establishes the groundwork for an energy distribution system that is more robust, effective, and sustainable.

References

[1] Stojmenovic, I. (2002). Handbook of Wireless Networks and Mobile Computing. John Wiley & Sons.

[2] Al-Karaki, J. N., & Kamal, A. E. (2004). Routing Techniques in Wireless Sensor Networks: A Survey. IEEE Wireless Communications, 11(6), 6-28.

[3] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences.

[4] Boukerche, A., & Fei, X. (2006). On the Coverage Problem in Wireless Sensor Networks. Proceedings of the International Conference on Parallel Processing Workshops.

[5] Huang, C. F., & Tseng, Y. C. (2005). The Coverage Problem in a Wireless Sensor Network. Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications.

[6] R. Mishra, P. Verma, R. Kumar, "Gateway Discovery in MANET using Machine Learning and Soft Computing," International Journal of Computer Applications, 2018.

[7] M. Wang, Y. Cui, X. Wang, S. Xiao, J. Jiang, "The Application of Machine Learning in Networking: A Review," Journal of Computer Networks and Communications, 2017.

[8] A. Smiti, Z. Elouedi, "DBSCAN-GM: A New Density-Based Clustering Algorithm," Journal of Systems and Software, 2015.

[9] Santini, S., Romaguera, J., Sánchez, M., & Igual, F. (2008). A Wireless Sensor Network for Environmental Noise Monitoring. Sensors, 8(4), 2494–2517.

[10] Kuhn, F., & Wattenhofer, R. (2003). Localized Delaunay Graphs Have Small Diameter. Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies.

[11] Wang, M., Jiang, J., & Cui, Y. (2009). The Application of Clustering Algorithms in Wireless Sensor Networks: A Survey. Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing.

[12] Jin, Y., Zhang, Q., & Song, W. (2006). A Clustering Genetic Algorithm for the Minimum Connected Dominating Set Problem in Ad Hoc Wireless Networks. IEEE Transactions on Evolutionary Computation, 10(5), 527–540

[13]. R. Suresh Kumar, P. Manimegalai, P. T. Vasanth Raj, R. Dhanagopal, and A. Johnson Santhosh, "Cluster Head Selection and Energy Efficient Multicast Routing Protocol-Based Optimal Route Selection for Mobile Ad Hoc Networks," [Link to Paper](path/to/paper)

[14]. Authors, "Optimal Routing Algorithm for Clusters of Microgrids," [Link to Paper](path/to/paper)

[15]. Rahul Khanna, Huaping Liu, Hsiao-Hwa Chen, "Self-organisation of Sensor Networks Using Genetic Algorithms," [Link to Paper](path/to/paper)

[16] R. L. Cruz and Arvind V. Santhanam, "Optimal Routing, Link Scheduling, and Power Control in Multi-hop Wireless Networks," [Link to Paper](path/to/paper)

[17] Han, X.; Liu, J.; Liu, D.; Liao, Q.; Hu, J.; Yang, Y. Distribution network planning study with distributed generation based on Steiner tree model. In Proceedings of the 2014 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Hong Kong, China, 7–10 December 2014; Volume 1, pp. 1–5.

[18] Oertel, D.; Ravi, R. Complexity of transmission network expansion planning NP-hardness of connected networks and MINLP evaluation. Energy Syst. **2014**, 5, 179–207.

[19] Li, H.; Mao, W.; Zhang, A.; Li, C. An improved distribution network reconfiguration method based on minimum spanning tree algorithm and heuristic rules. Int. J. Electr. Power Energy Syst. **2016**, 82, 466–473.

[20] Gouin, V.; Alvarez-Hérault, M.C.; Raison, B. Innovative planning method for the construction of electrical distribution network master plans. Sustain. Energy Grids Netw. **2017**, 10, 84–91.

[21] De Oliveira, E.J.; Rosseti, G.J.; de Oliveira, L.W.; Gomes, F.V.; Peres, W. New algorithm for reconfiguration and operating procedures in electric distribution systems. Int. J. Electr. Power Energy Syst. **2014**, 57, 129–134.

[22] Abeysinghe, S.; Wu, J.; Sooriyabandara, M.; Abeysekera, M.; Xu, T.; Wang, C. Topological properties of medium voltage electricity distribution networks. Appl. Energy **2018**, 210, 1101–1112.

[23] Crainic, T.G.; Li, Y.; Toulouse, M. A first multilevel cooperative algorithm for capacitated multicommodity network design. Comput. Oper. Res. **2006**, 33, 2602–2622.

[24] Mendoza, J.E.; López, M.E.; Peña, H.E.; Labra, D.A. Low voltage distribution optimization: Site, quantity and size of distribution transformers. Electr. Power Syst. Res. **2012**, 91, 52–60.

[25] Chicco, G.; Mazza, A. Heuristic optimization of electrical energy systems: Refined metrics to compare the solutions. Sustain. Energy Grids Netw. **2019**, 17, 100197.

[26] Javaid, N.; Qureshi, T.N.; Khan, A.H.; Iqbal, A.; Akhtar, E.; Ishfaq, M. EDDEEC: Enhanced developed distributed energy-efficient clustering for heterogeneouswireless sensor networks. Procedia Comput. Sci. **2013**, 19, 914–919.

[27] Alhamwi, A.; Medjroubi, W.; Vogt, T.; Agert, C. GIS-based urban energy systems models and tools: Introducing a model for the optimisation of flexibilisation technologies in urban areas. Appl. Energy **2017**, 191, 1–9.

[28] Tang, Y.; Mao, X.; Ayyanar, R. Distribution system modeling using CYMDIST for study of high penetration of distributed solar photovoltaics. In Proceedings of the 2012 North American Power Symposium (NAPS), Champaign, IL, USA, 9–11 September 2012; pp. 1–6.

[29] Freitas, S.; Santos, T.; Brito, M.C. Impact of large scale PV deployment in the sizing of urban distribution transformers. Renew. Energy **2018**, 119, 767–776.

[30] Aghaei, J.; Muttaqi, K.M.; Azizivahed, A.; Gitizadeh, M. Distribution expansion planning considering reliability and security of energy using modified PSO (Particle Swarm Optimization) algorithm. Energy **2014**, 65, 398–411.

[31] Davidescu, G.; Stützle, T.; Vyatkin, V. Network planning in smart grids via a local search heuristic for spanning forest problems. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1212–1218.

[32] Kruskal, J.B. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proc. Am. Math. Soc. **1956**, 7, 48–50.

[33] Prim, R.C. Shortest Connection Networks And Some Generalizations. Bell Syst. Tech. J. **1957**, 36, 1389–1401.

[34] Lezama, F.; Soares, J.; Vale, Z.; Rueda, J.; Rivera, S.; Elrich, I. 2017 IEEE competition on modern heuristic optimizers for smart grid operation: Testbeds and results. Swarm Evolut. Comput. **2019**, 44, 420–427.

rpt

ORIGINALITY REPORT

16%	15%	10%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	res.mdpi.com Internet Source	5%
2	www.coursehero.com Internet Source	2%
3	www.ir.juit.ac.in:8080 Internet Source	1%
4	ir.juit.ac.in:8080 Internet Source	1%
5	Submitted to Jaypee University of Information Technology Student Paper	1%
6	Submitted to Aberystwyth University Student Paper	<1%
7	Said S. Muratchaev, Alexey S. Volkov, Vadim S. Martynov, Ilya A Zhuravlev. "Application of Clustering Methods in MANET", 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2020 Publication	<1%

8	eurchembull.com Internet Source	<1%
9	Submitted to Kingston University Student Paper	<1%
10	link.springer.com Internet Source	<1%
11	openaccess.city.ac.uk Internet Source	<1%
12	eprints.staffs.ac.uk Internet Source	<1%
13	dspace.lboro.ac.uk Internet Source	<1%
14	www.mdpi.com Internet Source	<1%
15	digital.library.txstate.edu Internet Source	<1%
16	"Handbook of Heuristics", Springer Science and Business Media LLC, 2018 Publication	<1%
17	Qin, Hang, Jun Su, and Youfu Du. "PCDS: Power Control with Opportunistic Scheduling for Cognitive Radio Networks Reconfiguration", Journal of Networks, 2011. Publication	<1%
	doaj.org	

18	Internet Source	<1%
19	www.hindawi.com Internet Source	<1%
20	doc.lagout.org Internet Source	<1%
21	jwcn-eurasipjournals.springeropen.com Internet Source	<1%
22	portal.acm.org Internet Source	<1%
23	"Smart Industry & Smart Education", Springer Science and Business Media LLC, 2019 Publication	<1%
24	Submitted to Manipal University Student Paper	<1%
25	web.archive.org Internet Source	<1%
26	Kiyoshi Nakayama. "An autonomous distributed control method based on tie-set graph theory in future grid : BALANCED ALLOCATION OF RENEWABLE ENERGY RESOURCES IN SMART GRID", International Journal of Circuit Theory and Applications, 04/2012 Publication	<1%

27	Submitted to South Thames College Student Paper	<1%
28	export.arxiv.org Internet Source	<1%
29	fastercapital.com Internet Source	<1%
30	ise.tamu.edu Internet Source	<1%
31	libres.uncg.edu Internet Source	<1%
32	Submitted to Universiti Malaysia Perlis Student Paper	<1%
33	ir.kiu.ac.ug Internet Source	<1%
34	irojournals.com Internet Source	<1%
35	mdpi-res.com Internet Source	<1%
36	tel.archives-ouvertes.fr Internet Source	<1%
37	eprints.gla.ac.uk Internet Source	<1%
38	iris.polito.it Internet Source	<1%

39	ore.exeter.ac.uk Internet Source	<1 %
40	papers.nonprod.ssrn.com Internet Source	<1 %
41	T. Y. Wong, C. Shum, W. H. Lau, S. H. Chung, K. F. Tsang, C. F. Tse. "Modeling and co-simulation of IEC61850-based microgrid protection", 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2016 Publication	<1 %
42	core.ac.uk Internet Source	<1 %
43	ebin.pub Internet Source	<1 %
44	epdf.tips Internet Source	<1 %
45	ipfs.io Internet Source	<1 %
46	kipdf.com Internet Source	<1 %
47	pdffox.com Internet Source	<1 %
48	www.karunya.edu Internet Source	<1 %

49	www.theses.fr Internet Source	<1%
50	www.warse.org Internet Source	<1%
51	"Web, Artificial Intelligence and Network Applications", Springer Science and Business Media LLC, 2019 Publication	<1%
52	hdl.handle.net Internet Source	<1%
53	R.L. Cruz, A.V. Santhanam. "Optimal routing, link scheduling and power control in multihop wireless networks", IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428), 2003 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		