Applications of Graph Theory in Maps and Navigation

Session: 2022 – 2026

Submitted by:

Harmain Iftikhar	2022-CS-07
Ehsan Ullah	2022-CS-36
Zohaib Arshad	2022-CS-51
Muhammad Kamran	2022-CS-53

Supervised by:

Dr. Ayesha Altaf

Department of Computer Science

University of Engineering and Technology,

Lahore Pakistan

Table of Contents

Acknowledgement

I would like to express my sincere gratitude to all those who contributed to the successful completion of this project on " Applications of Graph Theory in Maps and Navigation" in the domain of Discrete Mathematics. This project has been an exciting and challenging journey, and it would not have been possible without the support and encouragement of many individuals and resources.

First and foremost, I would like to extend my heartfelt appreciation to our project supervisor, Dr. Ayesha Altaf, whose guidance and mentorship were invaluable throughout this project. Your expertise, patience, and constant support played a pivotal role in shaping this project and helping me overcome various obstacles.

I would also like to thank my fellow team members for their collaboration and dedication. This project would not have been as comprehensive and well-executed without their hard work and commitment.

Abstract

This report delves into the multifaceted application of graph theory in the realm of maps and navigation, exploring how discrete mathematics contributes to solving complex spatial problems. The purpose of this study is to elucidate the fundamental concepts of graph theory and demonstrate their practical implementation in representing, analyzing, and navigating maps. Employing a comprehensive approach, the report investigates various algorithms and techniques, shedding light on their efficacy in solving real-world challenges.

The methods employed encompass an in-depth examination of fundamental graph theory principles, including graph representation of maps, connectivity, shortest paths, Eulerian and Hamiltonian paths, graph coloring, and their integration into Geographical Information Systems (GIS) and social network analysis. Additionally, case studies provide tangible examples of how graph theory has been successfully applied in diverse scenarios.

Key findings highlight the critical role of graph theory in optimizing route planning, identifying efficient paths, and minimizing interference in map representation. The report underscores the practical relevance of algorithms like Dijkstra's, Traffic Prediction Algorithm, and Floyd-War shall in the navigation landscape. Furthermore, the exploration of GIS integration and social network analysis elucidates how graph theory transcends traditional mapping boundaries.

Project Details

1. Project Title

After a casual discussion, we titled this project with a combination of different words called "Application of Graph Theory in Maps and Navigation"**.** This project navigates the practical applications of Graph theory which is core topic of Discrete Math.

2. Project Statement

The project aims to explore the practical applications of Graph Theory in the field of maps and navigation. Graph Theory provides a powerful framework for modeling and solving problems related to routes, connectivity, and distances in various spatial contexts. The project will delve into the fundamental concepts of Graph Theory and illustrate how they can be effectively applied to address real-world challenges in maps and navigation system.

Description

Graph Representation of Maps:

One of the primary applications of Graph Theory in maps is the representation of geographical features and connectivity. Roads, intersections, cities, and landmarks can be modeled as vertices, while the connections between them (roads or paths) are represented as edges. This graph-based representation allows for efficient analysis and computation of routes, distances, and connectivity.

Shortest Path Algorithms:

Graph Theory facilitates the implementation of various shortest path algorithms, such as Dijkstra's algorithm and the Traffic Prediction Algorithm. These algorithms play a crucial role in determining the most efficient routes between locations on a map, aiding in navigation systems to find optimal paths for vehicles or pedestrians.

Network Analysis for Transportation Systems:

Transportation networks, including road systems and public transit, can be modeled as graphs to optimize scheduling, minimize travel times, and improve overall efficiency. Graph algorithms help in analyzing the flow of traffic, identifying critical nodes or intersections, and optimizing transportation systems.

Geographical Information Systems (GIS):

Graph Theory is integral to the design and functionality of Geographic Information Systems. GIS systems utilize graphs to represent spatial relationships, enabling the analysis of geographical data, route planning, and location-based decision-making.

Map Coloring and Graph Coloring:

Graph coloring techniques can be applied to maps to solve problems related to areas with adjacent borders (countries, states, etc.). Map coloring problems involve assigning colors to regions in such a way that no two adjacent regions share the same color, which has applications in political districting and resource allocation.

Tour Planning and Traveling Salesman Problem:

The Traveling Salesman Problem (TSP) is a classic optimization problem that falls under the domain of Graph Theory. In the context of maps and navigation, TSP can be applied to plan efficient routes for delivery services or tourist itineraries, ensuring that each location is visited exactly once.

Graph Database Applications:

Graph databases are increasingly used in location-based services and mapping applications. These databases leverage Graph Theory to represent and query spatial relationships, making them well-suited for applications that require efficient traversal and retrieval of geographical data.

Graph Description

A graph is a data structure that is defined by two components:

- A node or a vertex. is a fundamental unit in a graph. It represents a distinct entity or point in the graph. Node can represent various entities, such as cities, intersections, computers, or any other object of interest, depending on the context of the graph. Nodes may have associated attributes or properties, depending on the specific application. For example, a node representing a city may have attributes like population, name, and geographical coordinates.

- An edge E or ordered pair is a connection between two nodes u,v that is identified by unique pair (u, v). The pair (u, v) is ordered because (u, v) is not same as (v, u) in case of directed graph. The edge may have a weight or is set to one in case of unweighted graph. Edges indicate how nodes are related or interact with each other. For example, in a road network graph, edges represent the roads connecting different cities or intersections.

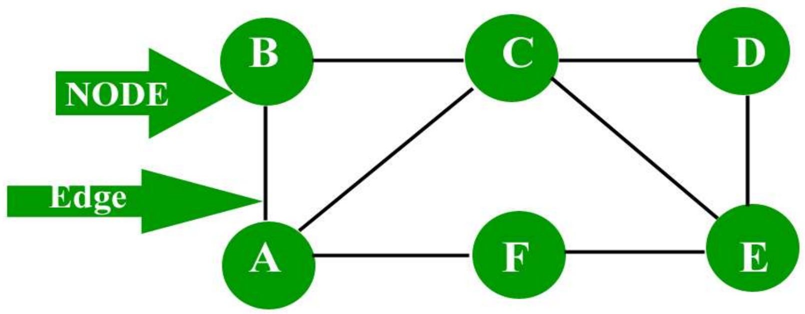

Figure: 1

Methodology:

- **Types of Graphs**

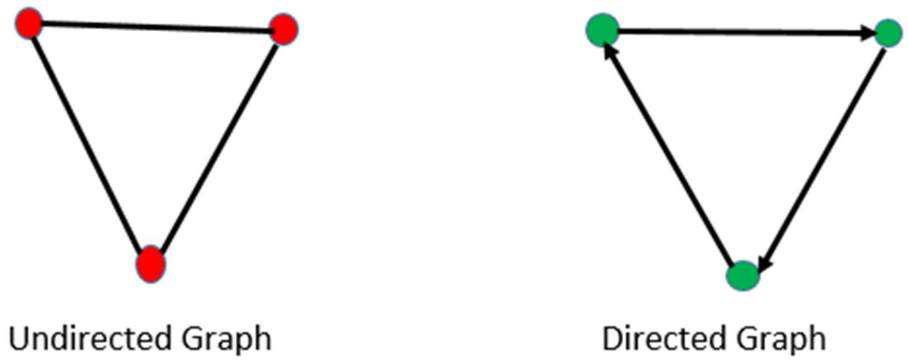

Figure: 2

- **Implementation**

The graph is implemented using nodes or vertices connected the help of edges or arcs. The graph may have a weight or cost which denotes the distance and cost of specific item according to the nature of graph.

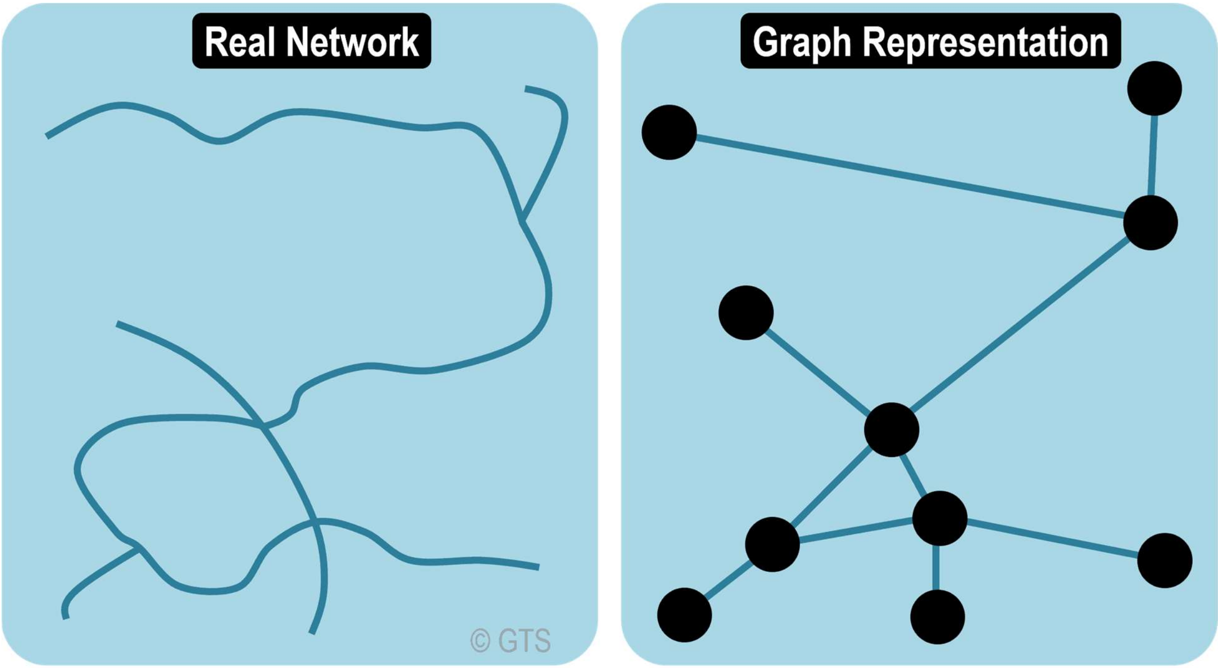

Figure: 3

- ## Maps

Google Maps operates on the principles of graph theory, where locations are represented as nodes, and roads or connections between them are depicted as edges. The map is essentially a graph, with each intersection or point of interest as a node and the roads as edges linking them. Nodes hold geographic coordinates, forming a spatial network. The routing algorithm, likely a variation of Dijkstra's algorithm, calculates the shortest path between two nodes, considering factors like distance, traffic conditions, and one-way streets. Real-time data, collected from user devices and various sources, updates the graph dynamically, ensuring accurate navigation.

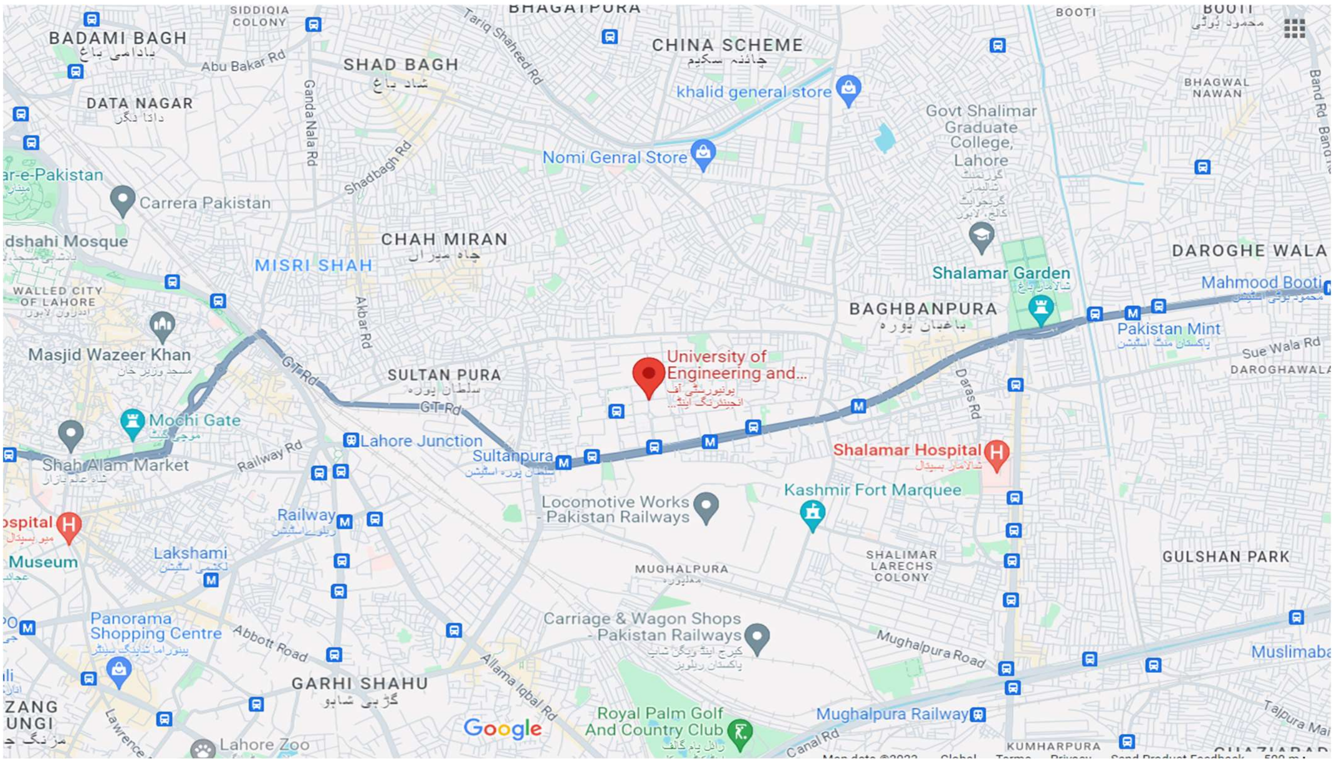

Figure: 4

- **Traffic Prediction**

  The traffic prediction algorithm shows the number of vehicles on the road that demonstrates either the traffic is moving quickly or jam or too slow. This helps emergency vehicles like ambulance and police to choose the road where traffic is not jam.

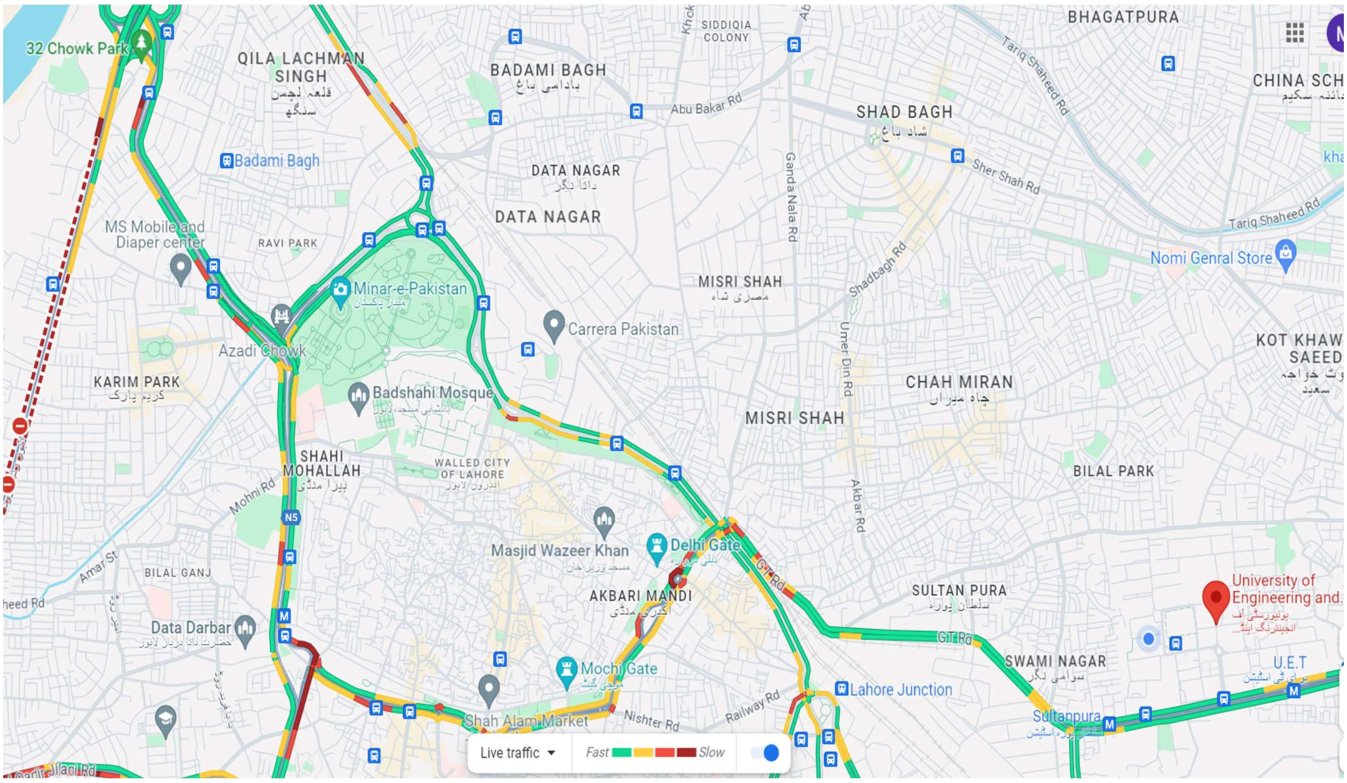

Figure: 5

The Red Path shows traffic is too slow or jam, we can extract different predictions at this regard like there may be a road accident or road maintenance work at this location etc. The light Red path shows the traffic is slow but moving on the road. The Yellow path shows the traffic is moving on average speed like 70 to 90 km per hour. The Green Path shows the road is clear and the vehicle can move on a speed of 100 km per hour or more.

- **Distance Measurement**
  This tool provide the facility to measure the distance from one place to another place. For example I want to measure the distance from UET entrance gate to Railway Station, we can drag the arrow from UET gate to Railway Station the measured distance will show on the drag line.

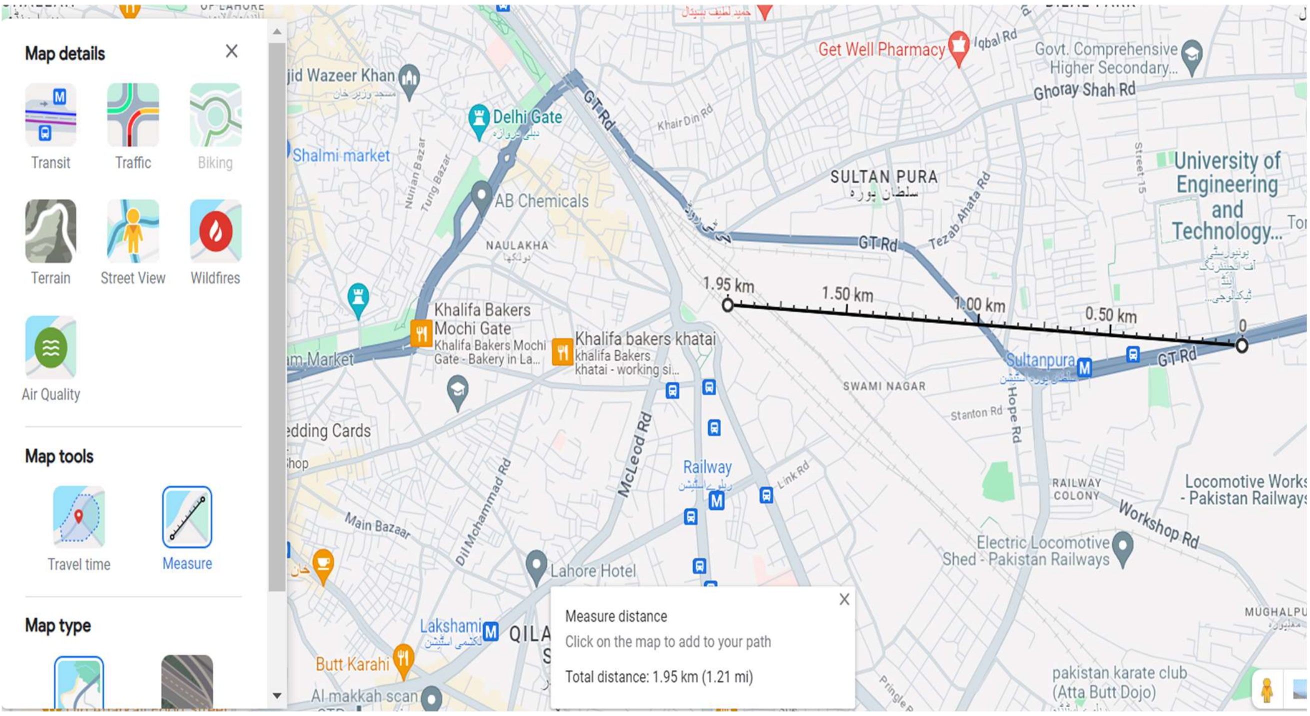

Figure: 6

- **From Source to Destination**

Suppose we want to travel from UET to Allama Iqbal International Airport, Google map provides three different roads path for travelling, every road distance and time is different. We always choose the distance which is nearest and required minimum time.

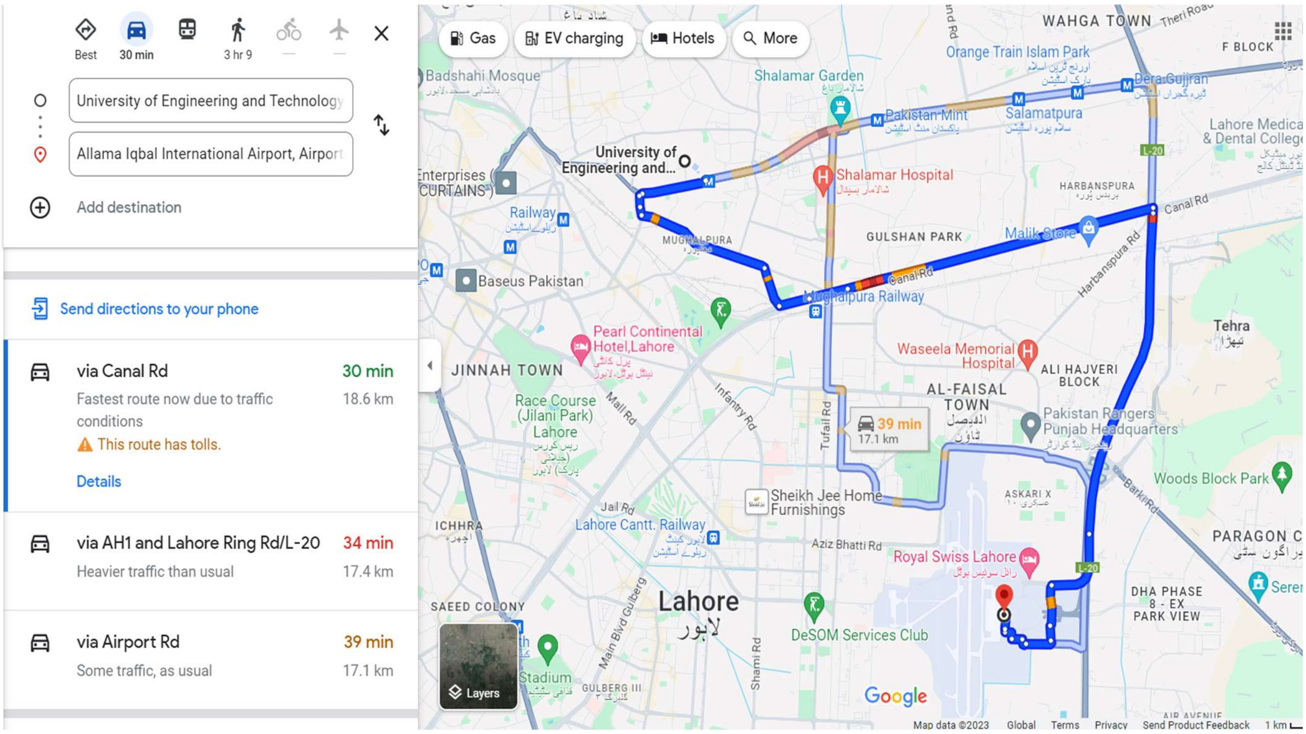

Figure: 7

- **Other Options**

Google Maps also provide the facility of Wildfire, Street view and Air Quality with respect to location. The Wildfire shows the location where the trees and and plants are burning or fired. Air Quality option provides the facility to check the quality of air in the specific area.

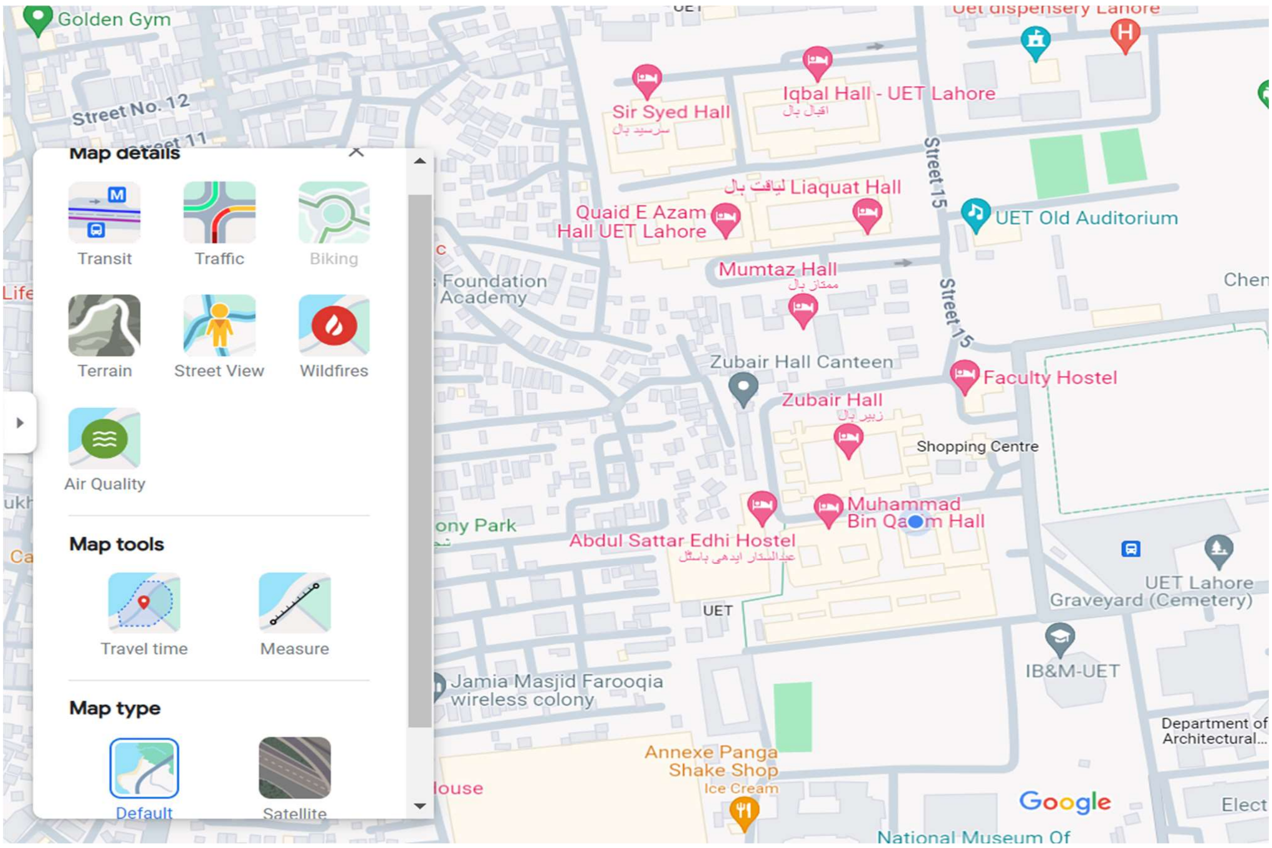

Figure: 8

Algorithm:

For finding the shortest path between two points we used Dijkstra Algorithm. Many other algorithms are also used in the process of implementation of Google Maps and Navigation. Traffic prediction and Map Matching Algorithm are also used in finding paths in google map for better user experience.

a) Dijkstra Algorithm

Dijkstra's algorithm is a graph search algorithm used to find the shortest path between two nodes in a weighted graph. It starts from an initial node and explores neighboring nodes, updating the shortest path and distance at each step. The algorithm uses a priority queue to efficiently select the node with the smallest tentative distance. Dijkstra's algorithm guarantees the discovery of the shortest path when all edge weights are non-negative.

Pesudocode

```
import heapq
def dijkstra(graph, start):
    # Initialize distances from start to all nodes as infinity
    distances = {node: float('infinity') for node in graph}
    distances[start] = 0
    # Create a priority queue and add the start node
    priority_queue = [(0, start)]
    while priority_queue:
        # Get the node with the smallest distance
        current_distance, current_node = heapq.heappop(priority_queue)
        # Check if the current distance is already greater than recorded distance
        if current_distance > distances[current_node]:
            continue
        # Update distances to neighbors
        for neighbor, weight in graph[current_node].items():
            distance = current_distance + weight
            if distance < distances[neighbor]:
```

```
                distances[neighbor] = distance
                heapq.heappush(priority_queue, (distance, neighbor))
    return distances

# Example usage:
graph = {
    'A': {'B': 1, 'C': 4},
    'B': {'A': 1, 'C': 2, 'D': 5},
    'C': {'A': 4, 'B': 2, 'D': 1},
    'D': {'B': 5, 'C': 1}
}
start_node = 'A'
shortest_paths = dijkstra(graph, start_node)
# Print the shortest paths from the start node to all other nodes
for node, distance in shortest_paths.items():
    print(f"Shortest path from {start_node} to {node}: {distance}")
```

b) Traffic Prediction Algorithms:

Traffic prediction algorithms employ machine learning and statistical models to anticipate traffic conditions. These algorithms analyze historical traffic data, considering factors such as time of day, day of the week, and special events. They incorporate real-time information from various sources, including GPS devices, mobile apps, and traffic cameras. By identifying patterns and trends, the algorithms predict future traffic congestion levels, enabling navigation systems to recommend optimal routes. Machine learning models may continuously adapt and improve their predictions based on the incoming data, offering accurate and up-to-date information for commuters.

Advance Approach: <u>Modelling of the Sybil Attack in Social Networks Using Graphs</u>

Social Networks such as Facebook, YouTube, and BitTorrent have become vulnerable to Sybil attacks. Many existing studies reported that Sybil users affect the correct functioning of any SN by contributing malicious content. Hence, illegitimately increasing influence and power with legitimate users. Malicious activities from the Sybil users are posing serious threats to the SN users, who trust the service and depend on it for online interactions just like normal users. There exist two categories of solutions that are available to defend Sybils attacks in a SN: sybil detection/identification, and sybil resistance/admittance control. Sybil detection schemes leverage the SN structure to identify whether a given user is sybil or non-sybil. We present an example of the graph that can be used to model the sybils in a SN.

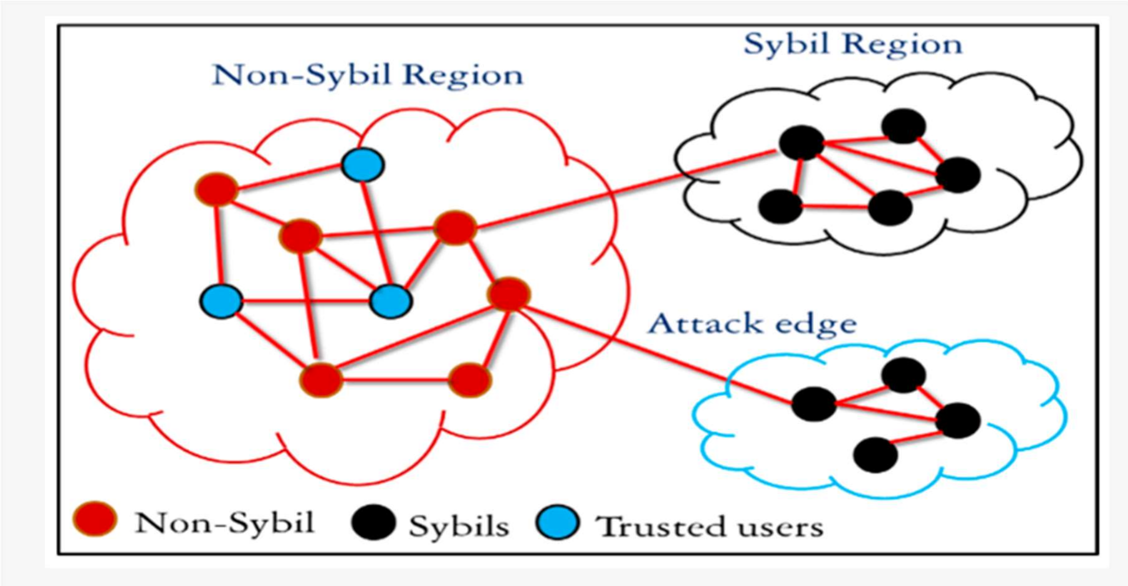

Figure: 9

Future Directions

1) Dynamic Graph-Based Routing:

Future developments in Google Maps may focus on dynamic routing algorithms that continuously adapt to real-time traffic conditions. By employing advanced graph theory techniques, the system could dynamically adjust route recommendations based on evolving traffic patterns, incidents, and user preferences.

2) **Integration with Smart City Infrastructure:**
Google Maps could further integrate with smart city infrastructure, utilizing graph theory to optimize navigation through interconnected smart traffic lights, intelligent parking systems, and other urban technologies. This integration would enhance overall traffic flow and reduce congestion in urban environments.

3) **Predictive Graph Analytics:**
Advancements in predictive analytics, leveraging graph theory, may enable Google Maps to anticipate future traffic conditions. Machine learning models could analyze historical data, user behavior, and external factors to predict traffic congestion, offering users more accurate and proactive route suggestions.

Acronym and Abbreviation

- DM (Discrete Mathematics)
- Algo (Algorithm)
- SN (Social Network)
- Weight (Distance, cost)
- UI (User Interface)
- Instructor (Teacher)
- Tools (Applications)
- Arc (Edge)

References

- **https://www.mdpi.com/2411-5134/5/1/10**
- **https://www.google.com/maps/@31.5461046,74.3261492,12z?entry=ttu**
- **https://www.geeksforgeeks.org/mathematics-graph-theory-basics-set-1/**