Predicting Congested Roads: A Simple Traffic Model

Minerva University

CS166: Simulation and Modeling

Prof. Tambasco

December 3rd, 2023

Procedural Design Enhancements in Code

Improved 'Car' Initialization

- Addressed edge cases by handling instances where the start and destination positions were randomly assigned to be the same.
- If such a scenario occurred, a random neighbor of the starting node was chosen as the destination.

Fallback Path Mechanism

- Introduced a fallback path method to provide alternative paths in cases where edges or paths did not exist between randomly assigned locations.

Encapsulation of Car Movement

- Encapsulated the movement of cars within the 'Car' class, streamlining the process of moving cars from one edge to another.

Enhancements in 'Traffic' Class

- Expanded the 'Traffic' class to handle the initialization of cars, track their movements, and assess congestion.
- Introduced a dictionary to keep track of traffic at each edge during the simulation.
- Implemented methods to add and remove cars from edges as they transition from one node to another.

Modified Congestion Metric

- Changed the congestion metric from the number of cars to the ratio of car length occupied by the total number of cars.
- Assumed an average car length of 4.8 meters (Aadarsh, 2023).
- Adjusted the congestion threshold to 75% of occupied road from 5 cars on a road.

Metric for Shortest Path

- Changed the shortest path metric from travel time to the length of the path for simplicity, achieving the same purpose.

Visualization Improvements

- Enhanced the output and visualization of the congested map by replicating the style of the osmnx map.
- Highlighted congested roads over the original map, providing a clearer representation.
- Simplified edge coloring by assigning the same color to all congested edges for improved visual clarity.

Model Explanation And Improvements

The simulation provided attempted to simulate a simple discrete-time step traffic model on a network graph where cars moved on a set path between nodes until the number of cars on each road exceeded the congestion threshold. The improved simulation extended this simple model by adding methods to track the car's movement and traffic updates. The initial model provided was suitable only for small simulations. The improved model can be used for experimentation to predict the roads most likely to get congested.

The new and improved model consists of two classes, 'Traffic' and 'Car.' The 'Traffic' class deals with the operations related to traffic management such as checking for congestion, moving cars from one edge to another, and tracking traffic values for each edge. The 'Car' class handles the operations associated with each car such as managing and updating the car's characteristics and keeping track of the car's journey.

Car Class

The Car class represents individual cars in the simulation, initialized with random starting locations, random destinations, and the shortest path between the starting location and destination using Dijkstra's algorithm.

Starting Locations, Destinations, and Paths

If the destination coincides with the starting point, a random neighboring node of the starting location is chosen as the destination. The path for each car is calculated using the NetworkX library's shortest_path function, considering edge lengths as weights. If no direct path is found, a fallback random path is selected.

The logic behind the fallback random path in the simulation is implemented within the `Car` class's `find_fallback_path` method. When the standard shortest path computation using the NetworkX library fails to find a direct route from the car's starting location to its destination, the fallback mechanism is activated to prevent simulation stagnation. Firstly, an attempt is made to select a random neighbor of the original destination node. If there exists an edge between this randomly chosen neighbor and the initial destination, a simple path is formed by connecting these two nodes. In cases where there is no direct edge between the random neighbor and the original destination, the fallback process resets. It then randomly selects an entirely new start node and connects it to one of its neighbors, resulting in a random path. This fallback random path strategy ensures the simulation's continuity, allowing cars to navigate the road network dynamically, even when the standard shortest path algorithm is unable to determine a precise route to the destination.

Traffic Class

The broader traffic simulation is managed by the Traffic class, which takes into account parameters such as the number of cars, simulation duration, and congestion threshold. During each time step of the simulation, the current traffic conditions on each edge are tracked, and the next stops for all cars are determined. The simulation checks for congestion on edges based on a predefined congestion threshold. Cars attempt to move to the next edge in their path unless it is congested. Traffic information is updated for both the current and next edges. The simulation repeats these steps for the specified duration.

Congestion Tracking

In the simulation, the congestion threshold is computed by evaluating the ratio of the number of cars on an edge to the edge's length. This process involves the dynamic tracking of traffic conditions on each edge throughout the simulation. The 'traffic_edges' dictionary is initialized at the simulation's outset, mapping edges in the road network to their corresponding traffic information. The traffic values are continually updated during each time step as cars move across edges, facilitated by the 'add_car' and 'remove_car' methods.

The congestion value for each edge is then calculated by considering the number of cars on the edge, the average length of a car (e.g., 4.8 meters), and the length of the edge itself. If this computed congestion value surpasses a predefined congestion threshold (i.e. 0.75), the edge is identified as congested.

The list of congested edges is dynamically updated throughout the simulation, offering insights into which sections of the road network are currently experiencing congestion. This process, embedded within the 'track_congestion' method, iterates over all edges, calculates congestion values, and flags edges exceeding the congestion threshold, providing a mechanism for effectively monitoring and responding to traffic congestion within the simulated road network.

Multiple Simulations And Visualizations

The simulation offers an optional visualization function to plot the road network with edge colors representing traffic conditions. Additionally, it allows running multiple simulations through the run_multiple_simulations function, which returns congested edges at the end of each simulation. The analyze_congestion function aggregates congestion results from multiple simulations, providing a ratio of congestion for each edge.

The `plot_traffic_on_graph` function in the simulation facilitates a comprehensive visual representation of the road network, leveraging color-coded edges to convey traffic conditions. The colormap 'Reds' is employed to map varying traffic values, ranging from light to dark red, symbolizing different levels of traffic intensity. Darker shades indicate higher traffic values, suggestive of congestion, while lighter hues represent lower traffic intensity. The width of each edge is dynamically adjusted based on its congestion status, with congested edges portrayed prominently through wider lines. This enables an immediate visual identification of areas experiencing traffic congestion. The underlying road network structure is overlaid with this color-coded representation, allowing for the identification of spatial patterns and congestion hotspots. Through this visual representation, the simulation empowers users to discern and analyze traffic dynamics, emphasizing congestion patterns across the simulated road network.

Effective Aspects of Real Roads and Traffic

The simulation effectively captures certain fundamental aspects of real roads and traffic, providing valuable insights into basic traffic dynamics.

Traffic Flow and Congestion Dynamics

The model successfully emulates the flow of traffic within a road network and the emergence of congestion based on the number of cars on an edge. This enables the observation of how traffic conditions evolve over time, showcasing congestion buildup and dispersion.

Pathfinding and Route Optimization

The simulation incorporates pathfinding algorithms, particularly the use of the NetworkX library's `shortest_path` function, to determine optimal routes for cars. This reflects the real-world behavior of drivers seeking the most efficient paths to reach their destinations.

Edge-Based Travel Time

By simplifying travel time calculations to edge lengths, the model captures a basic representation of the relationship between edge characteristics and the time it takes for cars to traverse them. This abstraction provides a foundational understanding of the impact of road length on travel duration.

While these aspects provide a solid foundation for understanding traffic dynamics, it's important to note that the model's effectiveness is limited by its simplifications and the absence of certain real-world complexities. The model's strength lies in its ability to simulate general traffic patterns and congestion dynamics within the constraints of its assumptions.

Abstractions And Limitations

The simulation, while providing a foundational representation of traffic dynamics, relies on several key assumptions, abstractions, and limitations that shape its scope and realism. Firstly, the model assumes the accuracy of the road network representation obtained from the OSMNx

API, which may not fully capture all the intricacies of real-world road systems. This assumption implies a simplified view of the road infrastructure.

A notable simplification lies in the calculation of travel time, which is reduced to edge lengths only. This abstraction overlooks more complex factors influencing travel duration, such as varying speed limits, traffic signal delays, and road geometries. Consequently, the model may not accurately reflect the diverse conditions influencing travel time in real-world scenarios.

Furthermore, the simulation does not account for real-time changes in road conditions, a crucial aspect of dynamic traffic systems. Factors like accidents, road maintenance, or sudden changes in demand are not considered, limiting the model's responsiveness to evolving situations.

The model also oversimplifies driver behavior, neglecting the diverse and dynamic decision-making processes exhibited by real-world drivers. The absence of detailed road features such as intersections and traffic signals further reduces the model's fidelity, as these elements significantly influence traffic patterns.

Additionally, the congestion determination solely relies on a predefined threshold, disregarding nuanced factors like bottlenecks or road capacity limitations. The absence of a more sophisticated congestion detection mechanism limits the model's ability to emulate realistic scenarios where localized traffic issues contribute to congestion.

Lastly, the model does not resolve congestion. Once the congestion threshold is met, the road is perpetually blocked and no more cars can access that road. This eventually leads to a back up of cars onto preceding roads, resulting in a sprawling network of congested roads. While this does capture the inherent nature of congestion, the model fails to account for the fact that congestion rarely results in complete blockage. Rather, cars slow down and move at a slower pace eventually reaching their destination or they reroute to alternate paths or destinations. A more realistic model would effectively work past this abstraction to model the real-world traffic.

Experiments And Analysis

The model was tested on two distinct locations: Adalbertstraße 58 in Berlin, Germany, and Bahadurabad in Karachi, Pakistan. In both cases, the simulation was successfully implemented. For each simulation, ten thousand cars were introduced and moved over the time frame of 3600 time steps. The congested edges were identified and plotted on the original map, with increasing traffic values at the end of the simulation indicated with increasing shades of red.

For empirical purposes, the simulation was run for hundred times for each location. All the congested edges for each trial of the simulation were collected and then aggregated to determine how many times each edge got congested out of the total hundred simulations. The results were plotted on a scatter plot with the edge betweenness values.

Congestion at a Glance

Adalbertstraße 58, Berlin, Germany

The map of Adalbertstraße 58 in Berlin, Germany has a total of two hundred and twenty-six locations (226) represented by nodes and five hundred and thirty-eight (538) roads represented by edges (Figure 1). The graph is a directed graph with the possibility of multiple edges connected to the same nodes. It is also not fully connected i.e. some nodes are completely inaccessible by others.

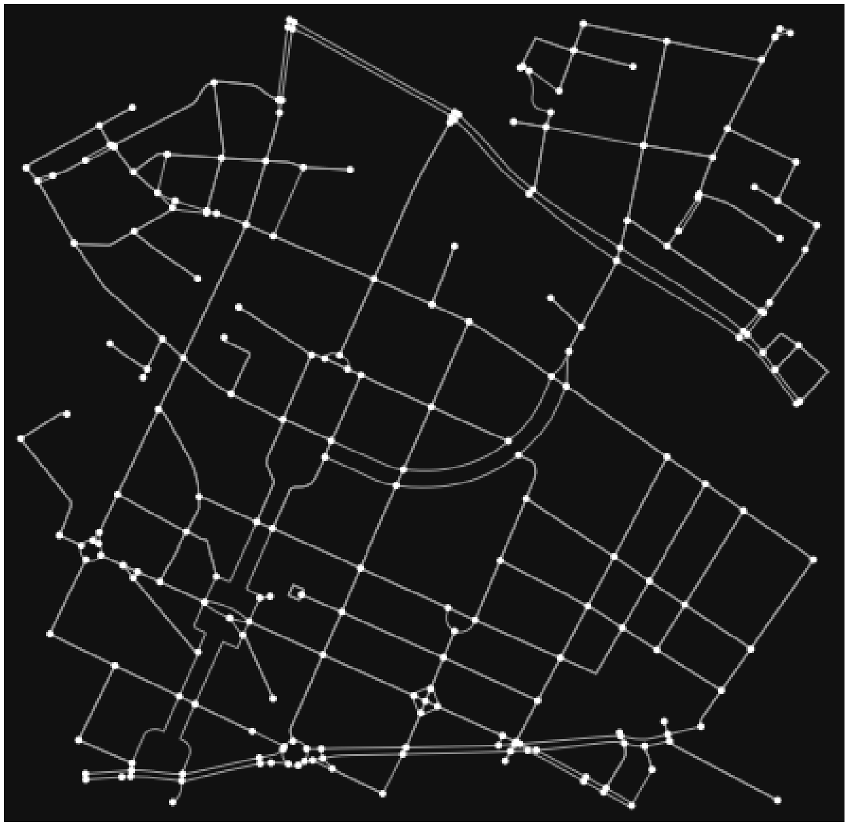

Figure 1: Map of Adalbertstraße 58 in Berlin, Germany represented as a network. The dots represent the nodes or locations and the lines connecting them are edges or roads.

Introducing and moving ten thousand cars to Adalbertstraße 58 for 3600-time steps results in 243 congested edges, meaning 45.17% of all the roads in the area have cars covering more than 75% of their length (Figure 2).

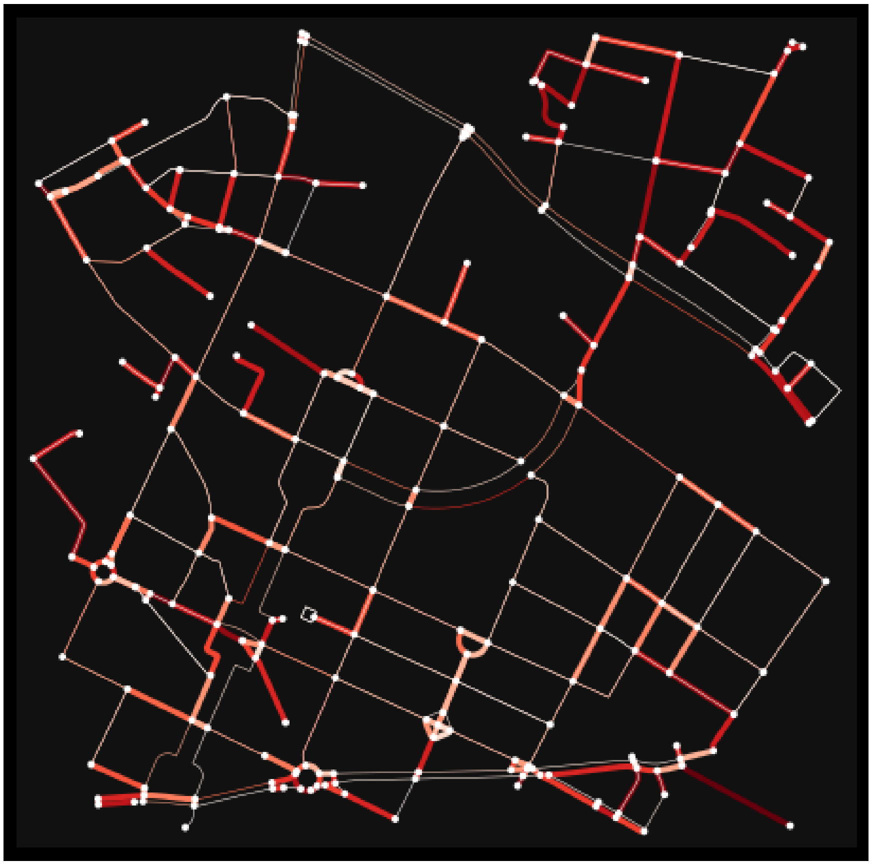

Figure 2: Map of Adalbertstraße 58 in Berlin, Germany represented as a network after 3600 time steps with 10,000 cars. Lighter shades of red represent lower traffic values while darker shades of red represent higher traffic values.

Bahadurabad, Karachi, Pakistan

The map of Bahadurabad in Karachi, Pakistan has a total six hundred and eight-six (686) locations represented by nodes and thousand six hundred and eighty-four (1,684) roads represented by edges (Figure 3). The graph is a directed graph with the possibility of multiple edges connected to the same nodes. It is also not fully connected i.e. some nodes are completely inaccessible by others.

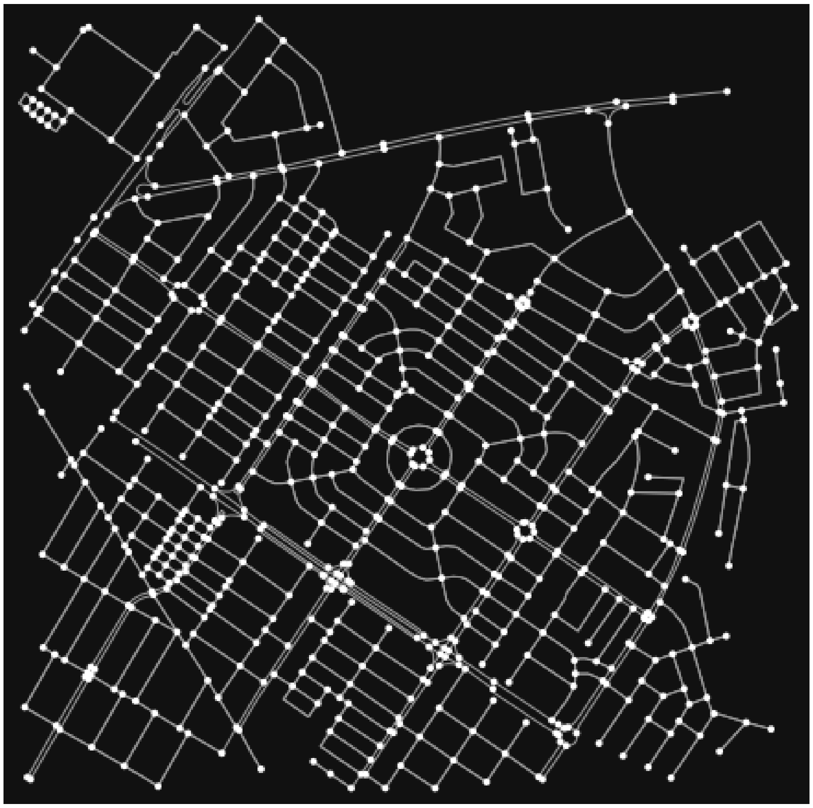

Figure 3: Map of Bahadurabad in Karachi, Pakistan represented as a network. The dots represent the nodes or locations and the lines connecting them are edges or roads.

Introducing and moving ten thousand cars to Bahadurabad for 3600 time steps results in 507 congested edges, meaning 30.11% of all the roads in the area have cars covering more than 75% of their length (Figure 4).

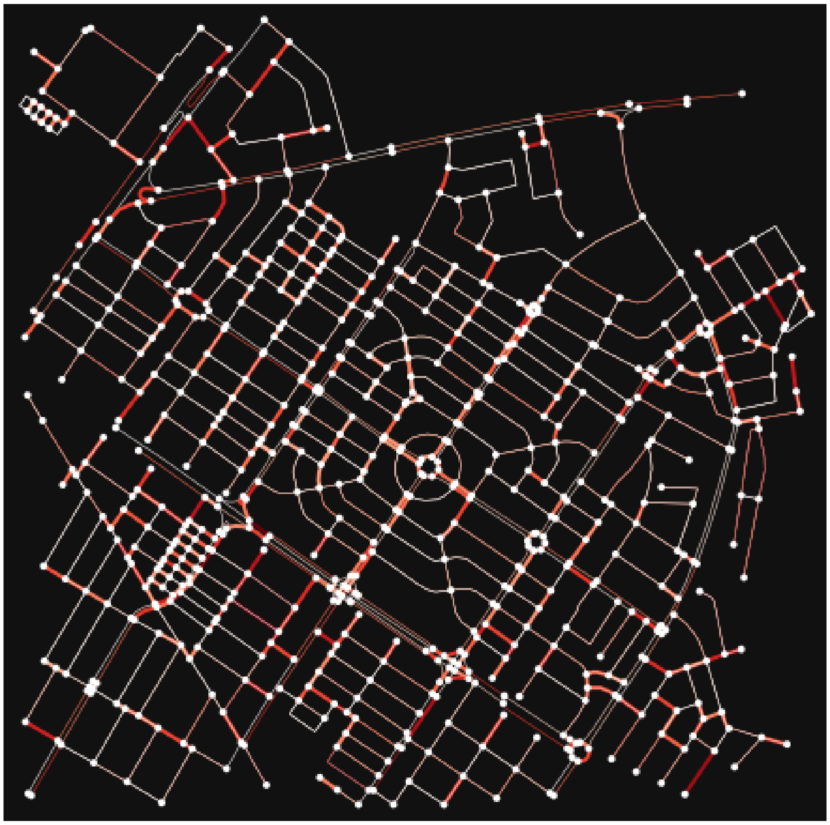

Figure 4: Map of Bahadurabad in Karachi, Pakistan represented as a network with the movement of 10,000 cars simulated over 3600 time steps. The increasing shades of red represent the traffic values on the roads.

Comparison

The empirical analyses offer a comparative perspective between two urban locations, Adalbertstraße 58 in Berlin, Germany, and Bahadurabad in Karachi, Pakistan, focusing on their road network characteristics and simulated traffic scenarios. Adalbertstraße 58 exhibits a more modest road network with 226 nodes and 538 edges, whereas Bahadurabad presents a larger and more intricate infrastructure, featuring 686 nodes and 1,684 edges. In terms of traffic congestion, Adalbertstraße 58 shows a higher percentage (45.17%) of congested edges, indicating that a significant proportion of roads have cars covering over 75% of their length. In contrast, Bahadurabad records a congestion percentage of 30.11%, suggesting a relatively lower level of traffic intensity on its roads.

Both locations share similarities in having road networks that are not fully connected, indicating areas that are inaccessible by certain routes. The simulations for both locations involve the introduction of 10,000 cars and a runtime of 3600 time steps, ensuring a consistent basis for comparison. The distinctions in congestion percentages underscore the impact of size, complexity, and geographical context on traffic dynamics, reflecting the diverse urban planning and transportation characteristics of Berlin and Karachi.

Overall, these comparative findings emphasize the importance of considering local factors when studying traffic patterns in different urban environments.

Empirical Analysis and Discussion

The empirical analysis included running the same simulation one hundred times for each of the two chosen cities. The data for congested roads was recorded at the end of each simulation and aggregated to see which roads got congested the most. The aggregated congestion results did not lead to any significant revelations about the roads' tendency to get congested. The maximum number of times any road got congested over the same of a hundred trials ranged between one and two. Hence, no conclusive remarks can be made about which roads get congested the most.

Limited Trials

There are not enough trials to determine the frequency of congestion. The simulation was run for a total of a hundred times for each location. Given the vast number of edges and nodes, that is simply not enough to see the roads get congested often. Hence, simulating the models over five hundred times might increase the chances of more robust results.

Randomness in the Model

The model initializes cars at random starting points and assigns random destinations to each car. Given that there are only ten thousand cars in the model, the random assignments of locations mean that not all roads get explored. In reality, vehicles do not randomly move from one place to another. Rather, there are clusters of nodes that are popular as destinations. Therefore, it is safe to say that the randomness in the model results in an inability to truly capture the roads that are likely to get congested.

Congestion Metric

The congestion metric of choice calculates the ratio of the road's length occupied by cars. By this definition, the roads that do tend to get congested are those of relatively shorter lengths. Given that cars are initialized on roads randomly, congestion on roads becomes a matter of the number of cars entering the road and the length of the road instead of the inherent tendency of the road to get congested due to centrality, popularity, or capacity, etc.

Theoretical Analysis and Discussion

To corroborate the empirical results, edge betweenness centrality was chosen as the theoretical network metric to predict which roads tend to get congested. Edge betweenness centrality is a network metric used to quantify the importance of edges within a network. It is a measure of the extent to which an edge lies on the shortest paths between pairs of nodes in the network. In other words, it measures the number of shortest paths that pass through the road in the traffic model. Because it assesses the influence that a particular road has over the flow of traffic between locations, edge betweenness centrality makes for an intuitive network metric to use for the prediction of road congestion.

Edge betweenness centrality is computed by identifying all the shortest paths between all possible pairs of nodes in the network and determining how many of these paths traverse a particular edge. A high edge betweenness centrality means that an edge lies on a large number of shortest paths. Mathematically, it looks like this:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Where:

- σ_{st} is the total number of shortest paths from node s to t.
- $\sigma_{st}(v)$ is the number of those paths that include edge v.

To visualize the edge betweenness centrality of the edges in the two graphs, the metric as computed for all the edges in each of the two locations using the networkx library in Python, and the edges with values of greater 0.1 were highlighted on the map (Figures 5 & 6).

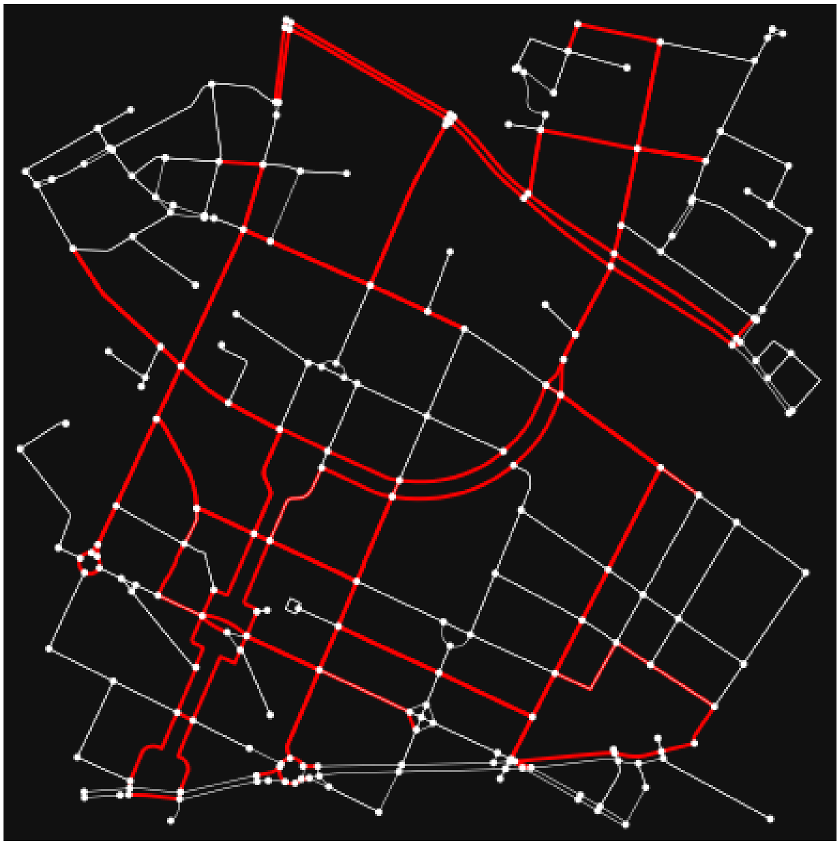

Figure 5: Map of Adalbertstraße 58 in Berlin, Germany with edges boasting edge betweenness centrality value higher than 0.1 highlighted. This map visualizes the main avenues of the area which are crucial to the connectivity of the network.

Figure 7: Map of Bahadurabad in Karachi, Pakistan with edges boasting edge betweenness centrality value higher than 0.1 highlighted. This map visualizes the main avenues of the area which are crucial to the connectivity of the network.

Since the higher the edge betwenness centrality, the higher the importance of the edge to the connectivity of the network, it seems fair to assume that roads with higher edge betwenness centrality would get congested the most. To investigate this relationship, empirical results were plotted against the edge betweenness centrality values (Figures 7 & 8).

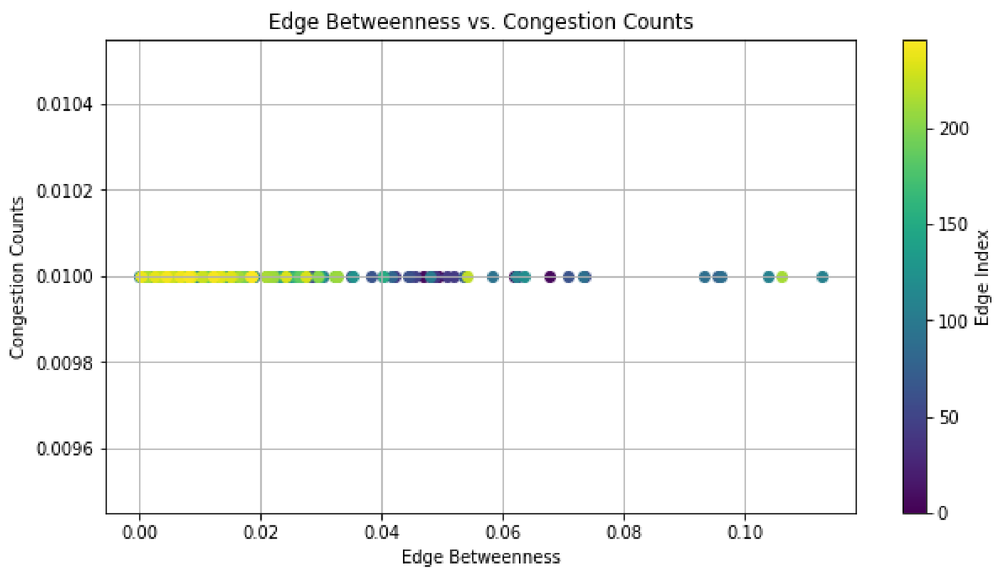

Figure 8: Edge Betweenness Centrality values and congestion counts for the roads of Adalbertstraße 58 in Berlin, Germany. Congestion counts are all 0.01, not elucidating anything useful about alignment with the theoretical results. The edge betweenness centrality values range from 0 to 0.12.

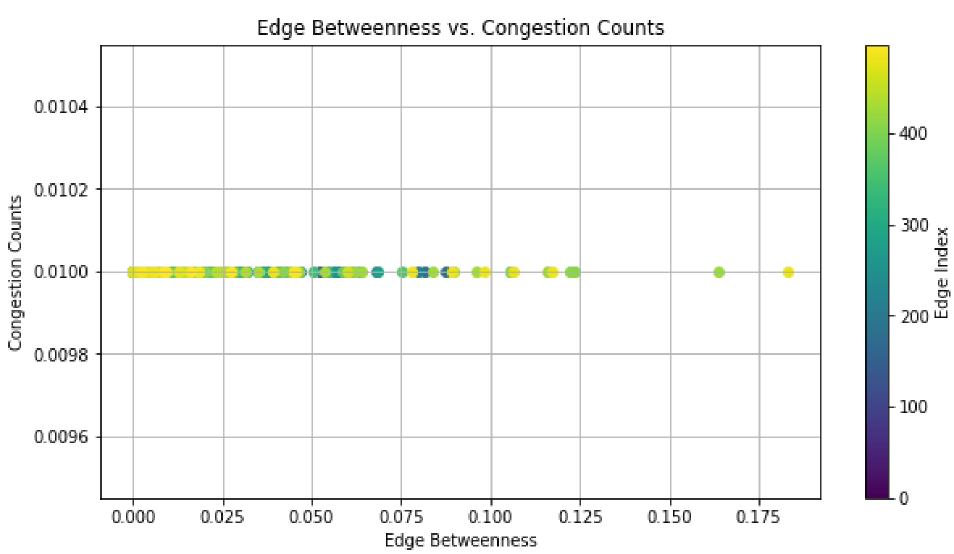

Figure 9: Edge Betweenness Centrality values and congestion counts for the roads of Bahadurabad in Karachi, Pakistan. Congestion counts are all 0.01, not elucidating anything useful about alignment with the theoretical results. The edge betweenness centrality values range from 0 to 0.180.

For both locations, it was clear that the edge between centrality values and the congestion counts did not corroborate each other. The congestion counts were all uniform throughout the roads that got congested over the hundred trials but the edge betweenness values for them varied from 0.00 to 0.12 for Berlin roads and 0.00 to 0.18 for Karachi roads.

The most obvious reason for the discrepancy between the theoretical and empirical results is that the theoretical metric assumes congestion is caused by the centrality of the road while the empirical metric assumes that congestion is caused by the capacity of the road. Both metrics are reasonable in their own right but do not align with each other. Hence, they cannot be compared.

To answer the question of which roads get congested, one can choose to look at either of the two metrics. Empirically, it looks like the shorter roads get congested more often because more cars stay on them for longer distances. Theoretically, it looks like roads that connect the most shortest paths get congested more often because more cars traverse them.

For more conclusive results, it is suggested to improve the modeling of the system to capture complex relationships between cars and the dynamic nature of traffic. It is also suggested to use a better congestion metric such as car per unit road capacity that is more aligned with the intuitive metrics of congestion. Lastly, for any empirical analysis to work, simulation must be run for many times.

AI Use

In this assignment report, AI was utilized primarily in two key aspects: data analysis and simulation. The report leverages AI techniques through the NetworkX library, a powerful tool for analyzing and modeling complex networks, to extract and analyze information from real-world road networks. The `load_berlin_road_network` function utilizes the OpenStreetMap API through the OSMNx library, incorporating AI-driven algorithms to generate a road network graph for Berlin. This data forms the basis for subsequent simulations.

The simulation, a core component of the assignment, employs AI techniques to model traffic dynamics. The `Car` class utilizes the NetworkX library's shortest path algorithm to calculate optimal routes, and the `Traffic` class simulates the movement of cars through the road network. The simulation involves elements of randomness, akin to real-world traffic, showcasing the application of stochastic AI components.

Furthermore, the report emphasizes the iterative nature of the simulation, suggesting the potential for multiple runs to capture diverse scenarios. The `run_multiple_simulations` function showcases the utilization of AI in generating statistical insights by running the simulation numerous times. Additionally, the `analyze_congestion` function aggregates simulation results, employing statistical analysis to identify trends across multiple runs.

In summary, AI techniques are integral to the assignment, contributing to the extraction and analysis of real-world road network data, as well as the simulation of traffic dynamics. The use of NetworkX and stochastic elements in the simulation exemplifies the practical application of AI in understanding and modeling complex systems such as urban traffic networks.