LBA PROJECT

Modeling Traffic Flow

Minerva University

CS166 - Professor Tambasco

Table of Contents

Explanation of Model **3**

Details of Simulation 3

Advantages of Model 4

Simplifications and Limitations of Model 4

Determining Road Congestion & Tips for Improvement 5

Improving Congestion Metrics - Employing Nagel and Schreckenberg (1992) 5

Improving Traffic Visualization **8**

Original Visualization 8

Improved Visualization 9

Analysis and Prediction of Traffic Congestion **10**

Determining Congestion-Prone Roads: 10

Network Metrics to Predict Congestion: 11

Comparing Empirical and Theoretical Results 13

Buenos Aires Traffic Simulation Results 15

Conclusion **18**

AI USAGE STATEMENT **19**

References **19**

Explanation of Model

Details of Simulation

Traffic dynamics are complex systems that involve interlevel interactions with local and global occurrences. Macroscopic properties of traffic dynamics such as the overall movement of cars, and the presence of traffic jams are defined by and influence microscopic properties of the movement of individual cars. The model provided by ChatGPT represents a traffic network and cars moving from a given initial positional node to some randomly determined final destination. The purpose of simulating such a model is to analyze and explain the behavior of the system given its network structure. By analyzing the network, we can better inform how to optimize for traffic flow, and make predictions of traffic.

Two class objects and initialization

In the code provided by ChatGPT that I have modified, there are two types of class objects that can be defined: a traffic flow and a car. Within a traffic flow, a predetermined number of cars are defined with random start and destination locations at the location value nodes. Both objects are initialized with a set of parameters that become attributes of the class object.

The car object requires parameters of a starting node, destination node, unique ID, as well as the city graph being used for the place nodes. The two nodes are used, along with the city graph, to determine the shortest possible path for the car to travel from its starting node to its destination. The unique ID is defined in the order of being defined within the Traffic_Flow class to allow users to easily recognize the movement of a given car between time steps– tracing where car 1 has moved to is much easier than car 2023840.

The Traffic_Flow Class also requires several parameter values: the defined city graph (osnmx file), the number of cars in the system, the total number of steps, and the threshold for a traffic jam. Instances of this class are initialized with the function initialize() to define a list of car objects that will be iteratively moved and a visualization of the system.

Updating Rules in Traffic Flow

The car itself has no update rules or functions to simulate the change in status of the car and a transitory boolean condition. Within the traffic flow class, there are several functions that simulate the change in the traffic flow at a one-time step. The movement of all cars at one time step is simulated with the function move_all_one_step, which moves all the cars that have not yet reached their destination by one edge (from current node to next node in path). This function contains a sub-class method, move_car_one_step() which moves a single car at one time step.

Measuring Congestion and Simulating Traffic Jams

In addition to moving each car from one step to another, the class Traffic_Flow also checks for current local traffic using the function curr_traffic. This function evaluates the congestion of the edge (road) that a given car is traversing through to determine whether there is a traffic jam. If there is a traffic jam, with the simulation currently defined, only the first car in line will be moved while the rest will stay at their current position.

Termination Condition for Car Movement

The termination condition for a given car to stop moving is if its predetermined path has been satisfied, meaning that the car has reached its destination. Once a car in the traffic flow has reached its destination, it will be removed from the simulation by removing it from the list of moving cars defined in the list attribute cars.

Advantages of Model

While no model can or should capture the full reality of the phenomenon it is modeling, this model does have a few significant strengths in accurately capturing traffic flow. One advantage of this model is its strategy in defining a travel route for the car to travel. Much like Google Maps, or other mapping software, the model defines the most optimal route by calculating the shortest possible route. Another advantage of this model is how it employs simultaneous updating. Just as we would expect multiple cars to move simultaneously in real traffic, the simulation moves all cars at one time step. Finally, this model determines congestion and reactions to it locally. Just as a car in the real world determines its next position based on what the driver sees directly ahead (how much congestion on the road), the model utilizes information about the congestion on the edge of movement (the edge used to get from current to next node) to decide the cars next position. Despite these advantages, there are also some clear limitations that must be discussed.

Simplifications and Limitations of the Model

Let’s discuss a few shortcomings that reduce the reliability of its real-life applications. One assumption that incorrectly simplifies this model is the assumption that given no congestion, all edges (roads) can and will be traversed in one step. This does not align with the real-world limitations of velocity. If a car is moving at some steady speed, it will travel through a short distance in less time than a long distance. In the current model, both edges of long and short distances are being traversed in a single time step. Defining these time steps to be of some unit, and determining the position of the car along the edge depending on its velocity would increase the reliability of this model, despite complicating it.

Another simplification being made to this model is the nature of traffic jams–this is an issue related to the earlier assumption. In real life, a traffic jam can often slow the velocity of a vehicle down significantly. Such occurrences do not imply inherently that there will be no movement, however. In the simulation, the traffic jam completely stops the flow, since we assume that only five cars (or whatever threshold we define) can traverse through a road, we assume that any other car will not move at all, and will begin the next time step at the previous current location.

Finally, notice that in this model, we assume the congestion rate to be the same for all roads. Realistically, however, there are some roads with 5 lanes, some with barely 1. This will affect the rate by which such edges will experience congestion. These limitations will be addressed in a further implementation of the code in the next section.

Determining Road Congestion & Tips for Improvement

Currently, the model determines road congestion before moving the car from its current location to its next location. The current metric for road congestion is to see how many cars are on the same edge as the one that is currently moving. With the current threshold at 5 cars, the code checks whether the path from the current to the next node is planned for less than 5 cars and if so, the car can freely move to the next node. If there are 5 or more cars, however, only the first four cars to be counted on the edge can move, while the rest must stay put. Clearly, this model is flawed, since it is arbitrary that 5 cars will indicate a jammed system. In the following paragraphs, I will present an alternative solution to measuring road congestion and responding to it.

Improving Congestion Metrics - Employing Nagel and Schreckenberg (1992)

An improved method for determining congestion is informed by Nagel and Schreckenberg's mode on traffic flows (1992). Their model is particularly informative for modeling the behavior of traffic jams, as the behavior of a car (determined by speed) will depend on the microscopic behavior of the car directly in front. The parameters of the model that will be utilized in the code improvement are as follows:

L: The length of the road (distance)

N: The number of cars on the road

$\rho = \frac{N}{L}$: The density of cars (average cars per unit distance)

v_{max}: maximum speed (distance per timestep)

Given these parameters, each car will have some current speed v_t–this value is determined by average speeds. At each time step, there will be a few updates made to the car: the speed of the car and the position of the car.

Speed:

Increasing: v_{t+1} depends on whether $v_t < v_{max}$. If this expression is true, $v_{t+1} = v_t + 1$.

Stable: If the expression $v_t = v_{max}$ is true, $v_{t+1} = v_t$.

Decreasing: If the distance between a car, and the car in front of it on the same road is less than the current velocity $v_t > d$, then $v_{t+1} = d - 1$. A small modification has been made to calculate next velocity is one minus the distance to the next car to work as a buffer (the original model assigned $v_{t+1} = d$)

Position:

Each car will move a distance v_{t+1} along the road from its current to the next node. If it overpasses the current edge, it will continue on to the next edge.

Now that we have our newly updated model rules, we can finally explain our new method for determining congestion: average traffic flow. Nagel and Schreckenberg describe in their paper a metric for determining the flow and congestion of their simple traffic model with the following equation:

$$\frac{\sum_{cars} car\ speed}{number\ of\ cars} \times \frac{number\ of\ cars}{length\ of\ road} = av\ speed\ \times car\ density$$

Note, however, that Nagel and Schreckenberg's model differed from the model that we employ. While Nagel and Schreckenberg use their model on one road, with a constant number of cars on the road, our model employs an entire road network with differing numbers of vehicles along the road and different road lengths. This difference does not hinder the applicability of this method of determining congestion levels, however. Since the primary objective of this paper is to determine which roads experience the highest traffic, we can use this equation to compare the congestion levels of the different roads.

Calculating Congestion

Utilizing the traffic flow equation presented by Nagel and Schreckenberg, an inverse relationship was determined as the congestion level. Following the logic that a low traffic flow indicated higher congestion—since as congestion increases, the average speed of cars will decrease at a more significant ratio than the density will increase—while a high traffic flow indicated low congestion. Traffic flow calculated to zero is indicative of a lack of cars on the road. With this reasoning, congestion was calculated with the following equation:

$$edge\ congestion\ = 1 - \frac{edge\ traffic\ flow}{theoretical\ max\ flow}$$

Where,

$$edge\ traffic\ flow\ = current\ density\ \times average\ speed$$

$$\textit{theoretical max flow} = \textit{max possible density} \times \textit{road speed limit}$$

The theoretical maximum flow was determined with the assumption that for any road, there would need to be safely 5 meters between one car and the next, for safe driving. In real-life scenarios, this would probably be even greater when cars move at high speeds, but for the sake of our model, it will work fine. This equation is informative in explaining how much of a road's potential flow is not being used, thus indicating higher congestion.

Improving Traffic Visualization

Original Visualization

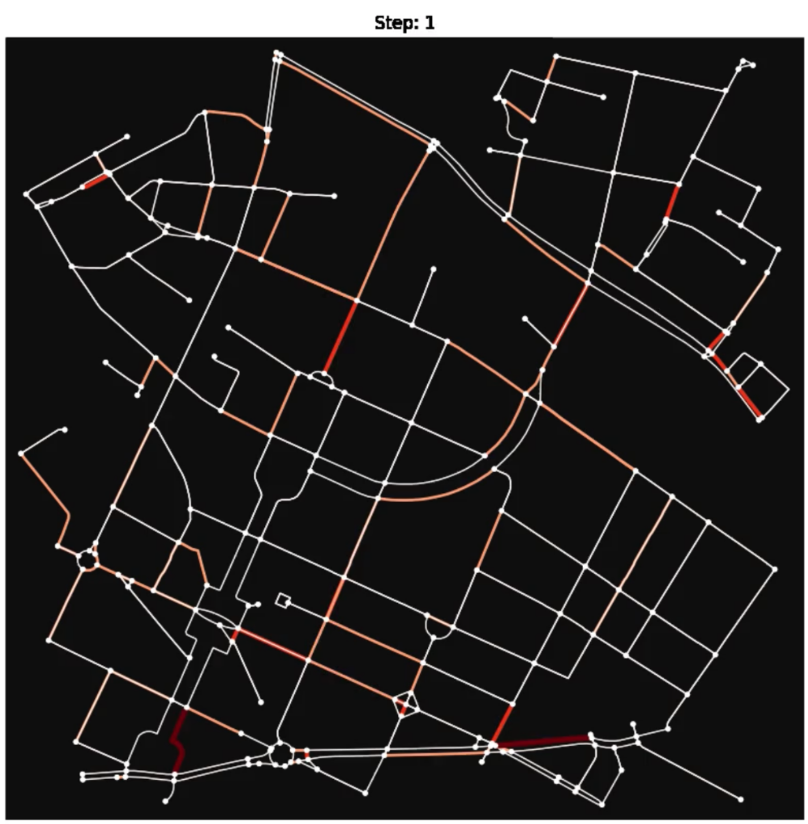

Figure 1. A visualization of the traffic model provided by ChatGPT. Redder and thicker lines indicate more cars traversing through the edge, signifying congestion levels. This visualization was taken from a simulation run with 100 cars.

In the original simulation visualization—of which a still image can be seen in Figure 1—the coloring and thickness of the road networks are determined as a ratio of the maximally congested road. This congestion level, as mentioned above, is determined by the arbitrary threshold of five cars on the road. As a ratio of the maximally congested roads, all edges are colored with different degrees of “redness”, plotted by the color mapping function in Matplotlib.

There are a few immediate issues with this visualization. Firstly, there is no legend or labeling, so it is not immediately understandable past the fact that some simulation is being run over a certain number of steps. We will change this in an improved visualization. Another observation with this visualization is that the changes in the coloring of the edges are not transparent to what is actually occurring in the simulation. In the simulation, it is the cars that are moving through the edges, but any onlooker to this visualization is not made aware of such a movement. While the lack of cars does help to visualize the state of the edges, in our visualization, we will include modeling of cars as well to be transparent of the microscopic processes causing congestion.

Improved Visualization

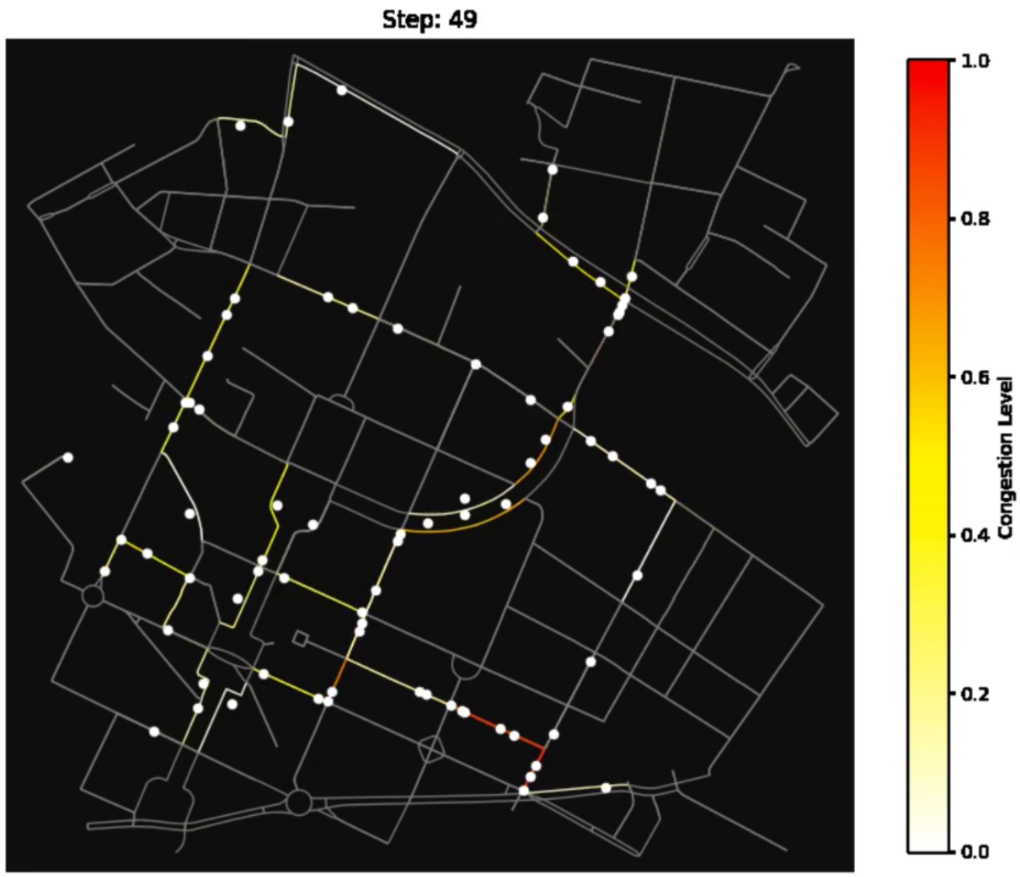

Figure 2. A figure showing the traffic network adapted to show the movement of the cars and the congestion level determined by Nagel and Schreckeneberg. As the legend shows, red indicates higher congestion, yellow indicates intermediate congestion, and white shows low congestion. The grey roads indicate no cars passing through. Congestion at each edge was determined at each time step as $1 - \frac{edge\ traffic\ flow}{max\ edge\ flow}$.

Figure 2 displays the changes that have been made to the visualization to better understand the congestion. Firstly, there are now white nodes showing the cars moving along the traffic network! This is helpful for understanding why a particular edge may be more congested than others. Seeing how the cars interact along the road, as they slow down, speed up, and how many occupy an edge is extremely helpful for conceptualizing traffic flow and congestion.

Additionally, in this model, the coloring of the roads now depends on the traffic flow, from red-yellow-white moving from low flow (high congestion) to high flow (low congestion) respectively.

Analysis and Prediction of Traffic Congestion

Determining Congestion-Prone Roads:

With the congestion equation displayed above, the most congested roads were calculated over multiple simulation trials, over the same conditions of 200-time steps and 100 cars initialized to the road. These initial parameters are chosen through trial and error with values that will allow all cars to reach their endpoint but still cause congestion. In addition to calculating the congestion at each time step, the congestion was aggregated over all time steps for each trial. From each trial, the average congestion of the road per time step was then computed and the maximal value was saved. A heatmap of the average congestion of the different roads can be seen in Figure 5.a. Additionally, the 10 most consistently congested roads are calculated and plotted in Figure 3. From looking at these empirical results, we conclude that edges marked by nodes (2148732, 26960762), (29276222, 29218323) are the most congested, as they were the most highly congested on average from 100 trials. We will be comparing such values to the network metric of betweenness centrality of edges in the following section.

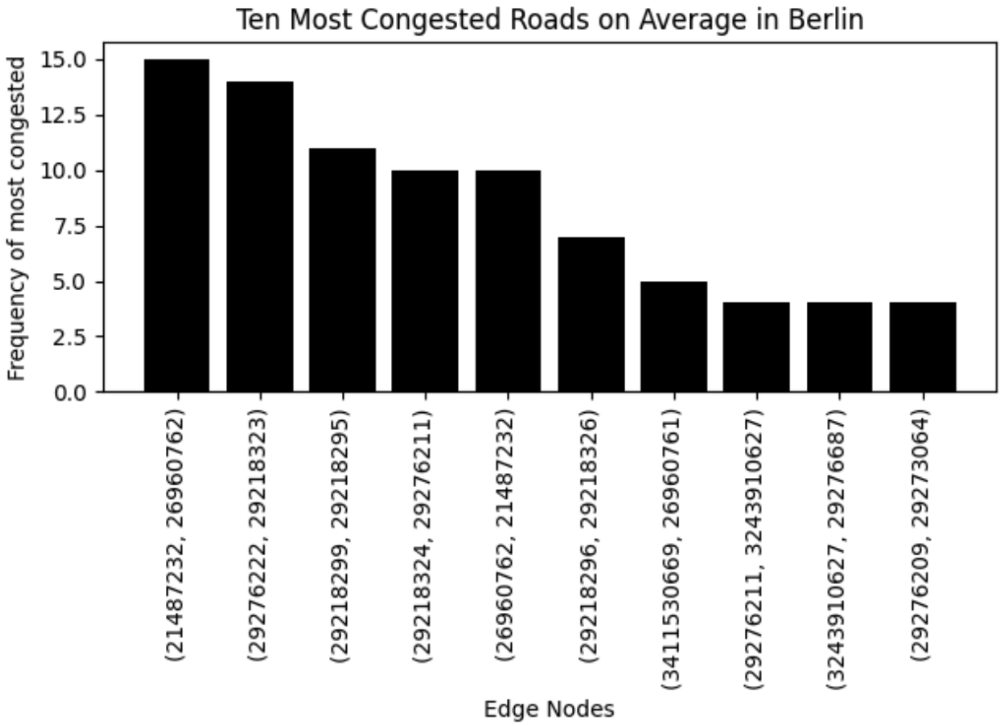

Figure 3. Histogram of 10 maximally average congested roads over 100 trials. From these results, we see that edges (2148732, 26960762) and (29276222, 29218323) were the most congested roads, being the most congested in 15/100 trials and 13/100 trials respectively.

<u>Network Metrics to Predict Congestion:</u>

While there are a number of metrics that can inform the "popularity" of a node or its connecting edges, the *betweenness centrality of edges* is the most fitting for our case. This is a probability of how likely the shortest path between two random nodes will pass through a specific edge. This betweenness centrality is incredibly important to road networks, as it may help us to identify the crucial roads that connect more disconnected areas, identifying edges that work much like highways!

Betweenness centrality of an edge e is calculated by taking the sum of the fraction of all node pairs' shortest paths passing through e

$$c_B(e) = \sum_{s,\, t \in V} \frac{\sigma(s, t \mid e)}{\sigma(s,t)}$$

Where V is the set of nodes, $\sigma(s, t)$ is the number of shortest paths between s and t, and $\sigma(s, t \mid e)$ is the number of these shortest paths passing through e.

Using NetworkX's inbuilt function, we determined the edge betweenness values for all edges, plotted them as a heatmap, and found the "top ten" most central edges. Figure 4 shows the betweenness centrality of these different edges. From this metric, we observe that there is no edge that is particularly more central than the others when comparing from the ten most central nodes.

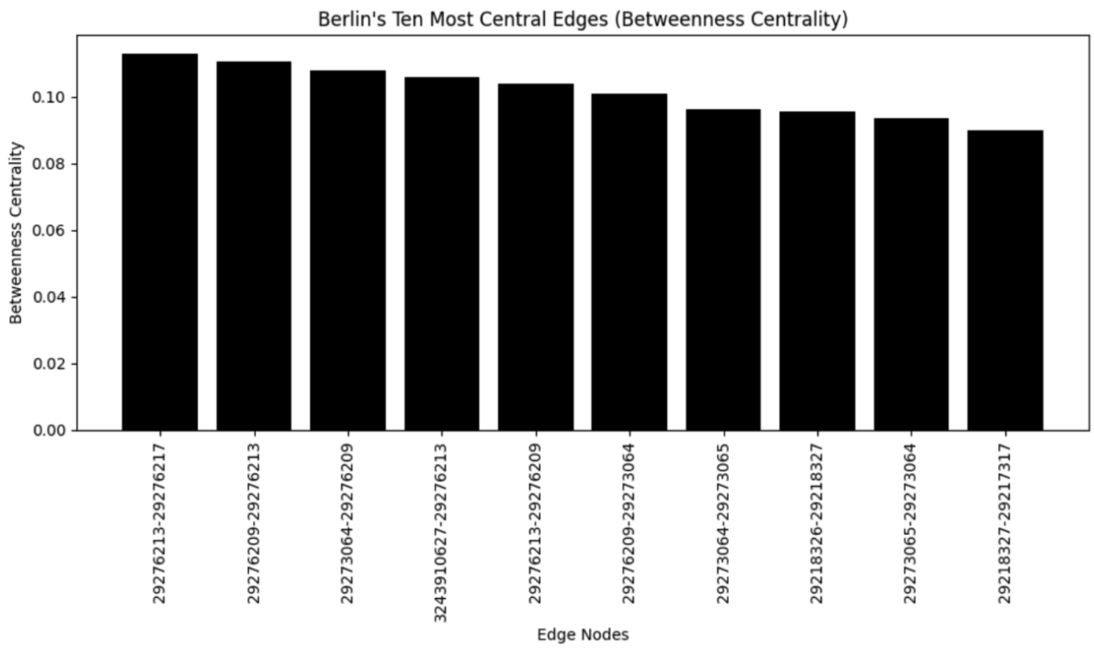

Figure 4. A bar chart of Berlin's ten edges (roads) with the highest betweenness centrality values. The top three edges with the highest betweenness centrality are (29276213, 29276217), (29276209, 29276213), (29273064, 29276209). All ten edges have similar betweenness centralities ranging from 0.11-0.12. This indicates that there is at maximum, a 0.12 probability of any shortest path between two random nodes including this edge in its path.

Comparing Empirical and Theoretical Results

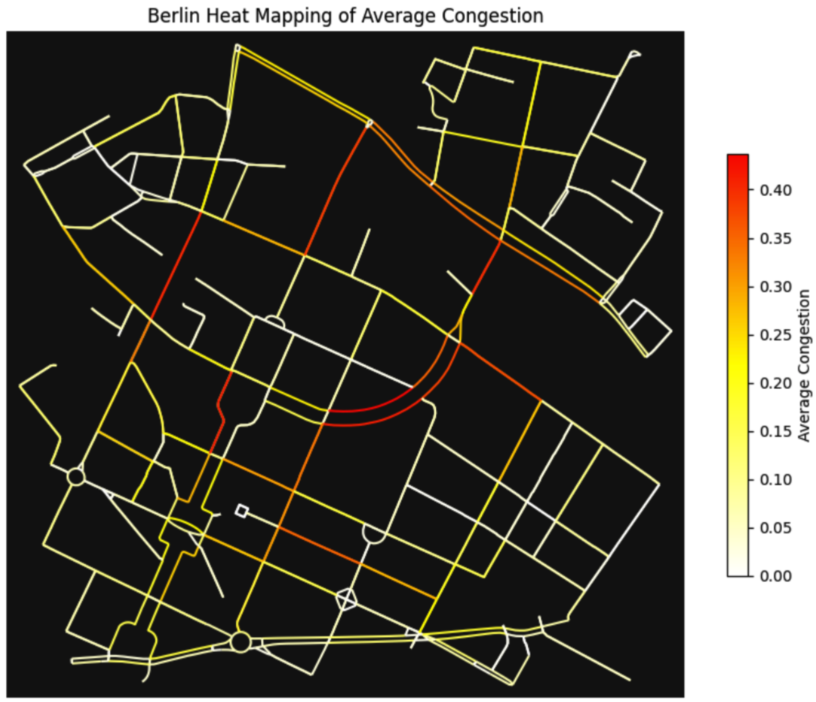

Figure 5. a. Heat mapping overlay of average congestion levels over 100 trials of Berlin's road network near Adalbertstrabe 58b, Berlin Germany.

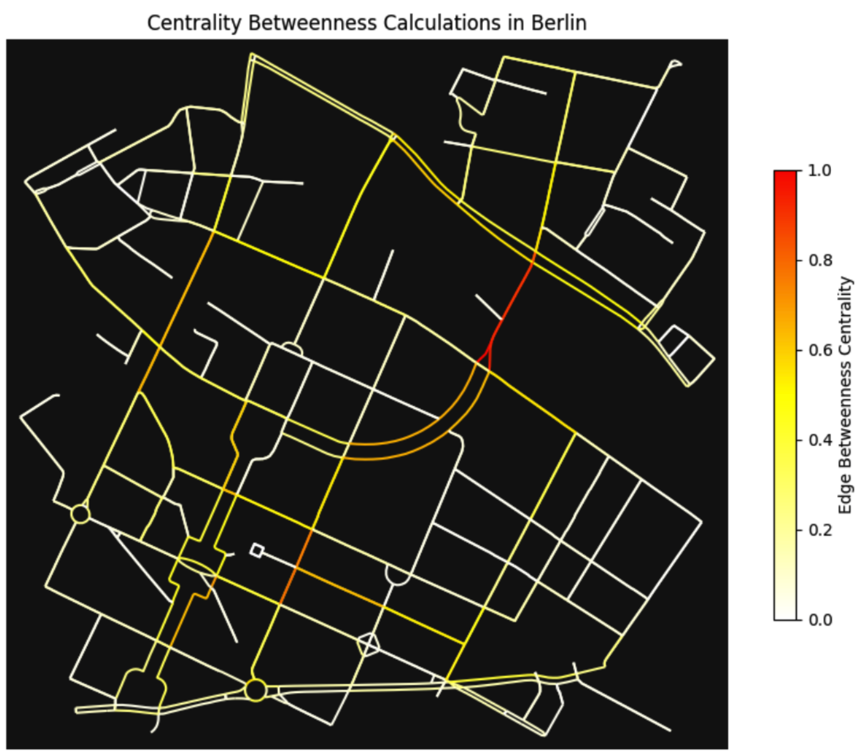

Figure 5. b. Heat mapping overlay of betweenness centrality of the edges of a road network near Adalbertstrabe 58b, Berlin Germany.

From identifying the ten most congested/central roads, we can observe figures 3 and 4 to see that there is only one common edge: (29276209, 29273064). The reasoning for this disparity in results may come down to the way that congestion was calculated, as this metric is not only accounting for how likely that a car will pass through it but also its capacity for cars on the road. Although these metrics do not perfectly align, notice Figure 5.a. and 5.b., and how there is a large overlap in 5.a. between roads with higher congestion (in red) and those in 5.b. with middle and high betweenness centrality (in orange in red). Such an overlap suggests that while betweenness centrality is not a perfect determinant of how congested the road might be–because there are many other factors influencing congestion such as the road length and speed limits), it can help to inform where higher capacities are required in road building.

Buenos Aires Traffic Simulation Results

Let's continue to validate the metric of betweenness centrality in another location. I have chosen to conduct this in an area in Buenos Aires, notorious for congestion when traveling to and from it: Palermo! The same simulation as before has been performed with 200 cars and 200 time-steps for 100 trials.

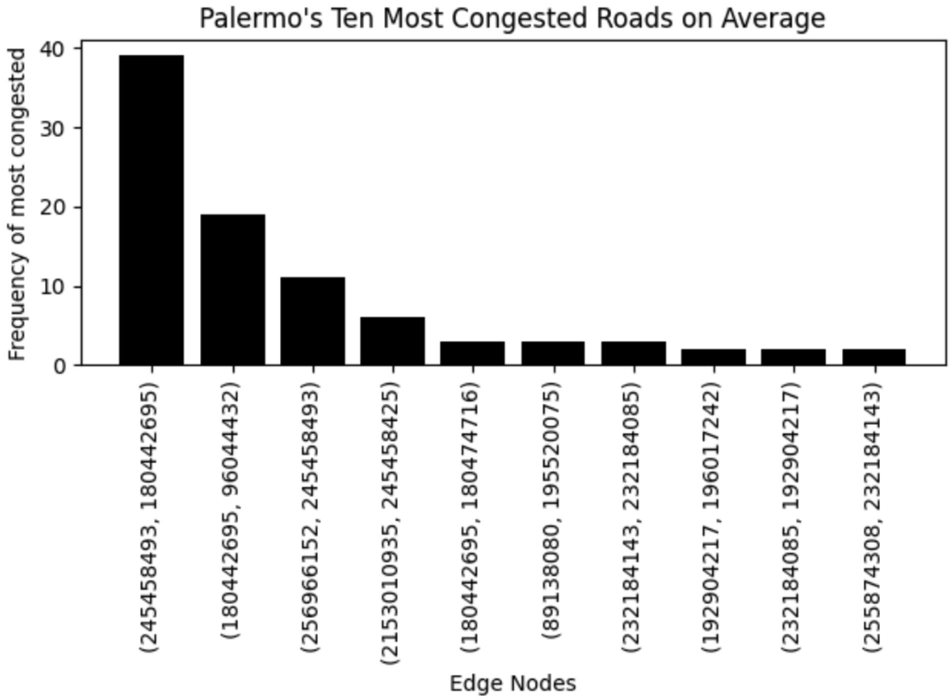

Figure 6. Bar chart showing the ten most congested cars on average from 100 trials. There is a steep difference between the most congested road and the other nine roads. Notice that this top congested edge (245458493, 180442695), is the most congested road ~40 out of 100 trials. The next most congested edge (180442695, 96044432) is the most congested ~20 out of 100 trials.

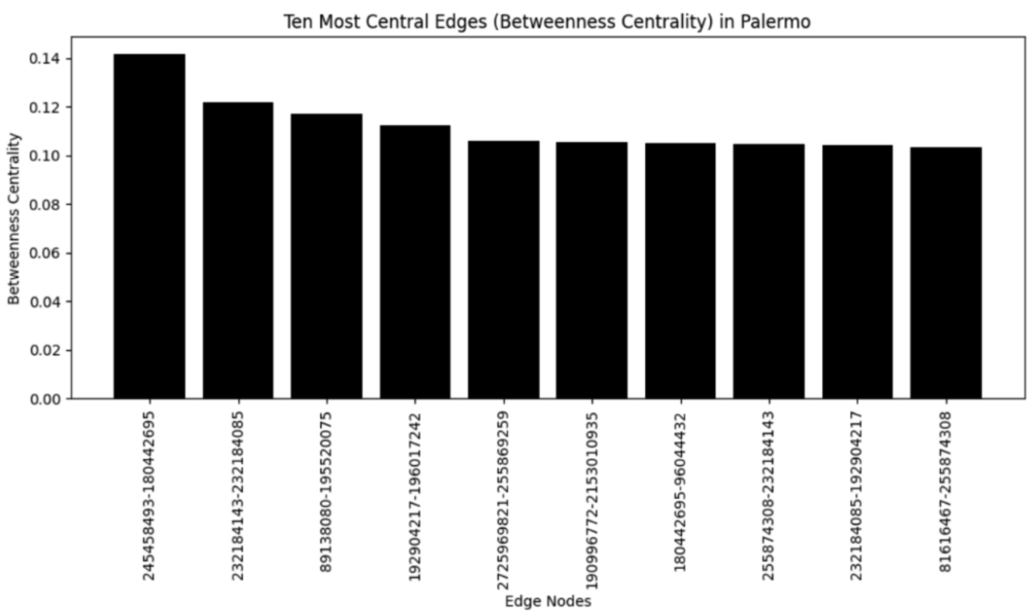

Figure 7. Bar chart of the ten most central edges in Palermo, Buenos Aires. The central edge is (245458493, 180442695), with a centrality value of 0.14, followed by (232184143, 232184085) with a centrality value of 0.12.

The results found from the traffic simulation in Palermo are clearly more consistent in results with betweenness centrality and congestion levels. Notice in Figure 6 that the most congested road by a high margin, (245458493, 180442695), is also the most central edge in Figure 7. Additionally, the heatmaps in Figures 8.a. and 8.b. are well aligned, with congested roads aligning with those of greater betweenness centrality. In such a case, we can more confidently conclude that betweenness centrality is a good metric for predicting where congestion may occur.

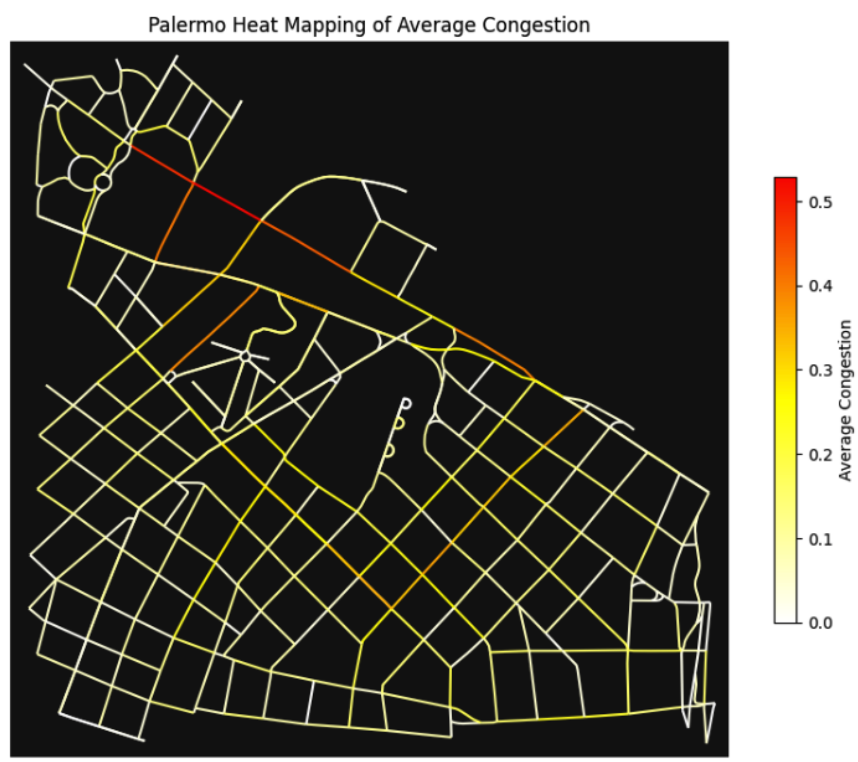

Figure 8. a. Heat mapping overlay of average congestion levels over 100 trials with 200 cars and 200 time steps in Palermo, near Junín 1930, C1113 CABA, Buenos Aires.

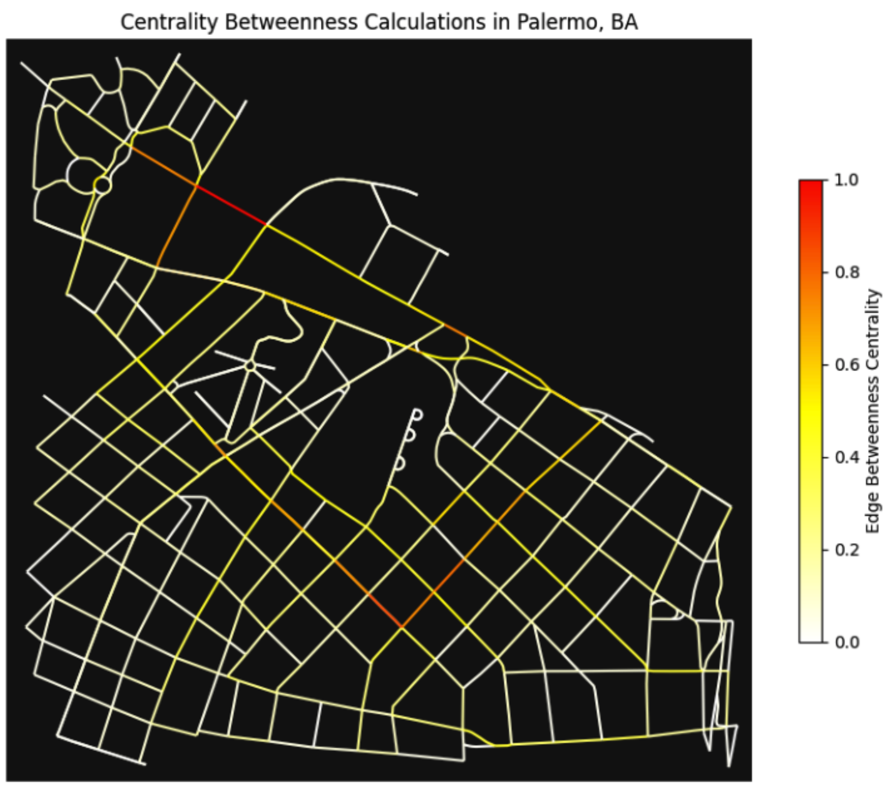

Figure 8. b. Heat mapping overlay of betweenness centrality in Palermo, near Junín 1930, C1113 CABA, Buenos Aires.

The reasoning for such clear results, in this case, can be observed by a key difference in this city network, as opposed to the previous network seen in Berlin: the interconnectivity of the system. The model simulates the congestion "bottlenecking" effect that occurs between the southern Recoleta district and the northern section constituting Palermo. There are many roads within each district, but those connecting the two are more infrequent. Thus, this "highway" road becomes central in connecting the two sections and its betweenness centrality accurately predicts the congestion it experiences.

Conclusion

By improving on, and modifying the traffic simulation initially provided by ChatGPT, a more realistic model and subsequent simulation has been created using traffic flow rules provided by Nagel and Schreckenberg (1992). Furthermore, this simulation has been run on two city networks–Berlin and Buenos Aires–and tested with the network metric of edge betweenness centrality. Through this comparison of empirical and theoretical results, we can conclude that the betweenness centrality metric can be accurate in polarized network structures where clusters are connected by a major roadway, such as highways. Through analyzing the city and road structures for this important metric, infrastructure planning can be conducted more thoughtfully with awareness of where congestion occurs, and solutions for improving traffic flow.

WORD COUNT: 3100 words

AI USAGE STATEMENT:

I used ChatGPT to implement my desired data visualizations, utilizing it to implement the interpolation of edges to provide real-time updates of the cars within the simulation.

References

Nagel, K., Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. *Journal de Physique I, 2*(12), 2221–2229.