Network Analysis of Traffic Lights and the Correlation With Traffic Crashes

Nadeem Hussein & Layla Shalabi

Introduction

Our project aims to analyze patterns between traffic light locations and speed-related traffic crashes in Chicago. Both traffic light locations and speed camera locations were points of interest. We created a unimodal network graph in which each node represents a traffic light and each undirected, weighted edge represents a straight line path between the lights. That is, for a given traffic light, it is connected to any other traffic light along a straight line path (straight line paths are defined as the straight line paths of the streets).

The intuition behind representing straight line paths as edges comes from the idea of speeding and running through red lights. Generally speaking, it is much easier for a driver to run through a red light if they are continuing straight through the light. It is much harder to speed through when they need to turn right or left because they must first slow down. As such, the edges represent the closer relationship we want to highlight between traffic lights along the same straight line path and explore if this relationship is at all connected to traffic crashes caused by speeding. The edges are weighted by a normalized distance metric.

We then overlay locations of traffic crashes and speed cameras on top of our network graph in an effort to illustrate any evidence of further correlation among these variables. We hoped that a network graph of this nature would demonstrate some correlation between crash locations, traffic light paths, and speed cameras. Additionally, we hoped this modeling of the problem would provide insights on the distribution of speed-related car crashes in Chicago and perhaps aid in future predictive analysis of likely crash sites.

Data

We used three datasets in total for this project: Traffic Crashes - Crashes (Chicago Police Department, 2024), Map - Speed Camera Locations (City of Chicago, 2024), and Traffic Lights Locations (OpenStreetMap, 2024). These three datasets contain traffic crash data (including location and cause), locations of speed cameras, and locations of traffic lights – all within the City of Chicago – respectively. The Traffic Crashes and Speed Camera Locations datasets were both collected from the City of Chicago Data Portal. The Traffic Crashes dataset is maintained and updated by the Chicago Police Department daily. The Speed Camera Locations dataset is maintained by the City of Chicago.

The Traffic Crashes dataset has over 818,000 instances. We sampled this dataset by filtering it to only include crashes that had a primary or secondary contributing cause related to speed. This reduced our subset of the dataset to around 68,000 instances. From the remaining crashes we removed all entries that did not have a latitude and longitude, or had an invalid latitude and longitude (eg. (0, 0)). This process removed an additional 201 entries. No preprocessing was done on the speed cameras dataset.

The traffic light locations were queried from OpenStreetMap, an open source map with all types of geographical data. We queried specifically for traffic light locations in the City of Chicago, giving us around 4,000 lights to work with. From these lights, we removed all the ones tagged as a “ramp-meter” as it wasn’t the type of traffic light we were interested in and it interfered with the street collection in the next step. We then used a reverse geocoding API to extract the names of the streets that the intersection lay on. Because of the limitations of the API, it would only return an address with a single street instead of an intersection. To counteract this problem, we queried 5 times per light. The first query was at the location of the light, and the other queries were nudges in one of the 4 directions (North, East, South, and West). This gave us a good indication of all the streets the light lay on. This information was then added to the dataset.

Our overall data process followed the plan we had outlined in our data description deliverable.

Project Process

After collecting our data, the first thing we did was create a connected network graph of all the traffic lights in the city of Chicago. We connected any 2 lights that had a road in common and weighted the edge based on a normalized pythagorean distance metric. One issue we realized is that many roads will switch names at some point along the road, for example from “South State St.” to “North State St.”. This essentially splits the road up into 2 distinct roads. This distinction made sense for some roads such as “Michigan Avenue” which has a distinct concrete barrier between its North and South streets. Because of this, fixing the issue wasn’t as simple as removing East/West and North/South pairs. Ultimately we decided to leave the N/S and E/W distinction for the roads because it stayed consistent throughout the dataset, and attempting to fix it would require time consuming hand-sifting through the data for no guaranteed benefit.

The distance metric used was the Pythagorean distance between the lights divided by the largest distance to normalize between 0 and 1. We then assigned each node a weight equal to the average weight of all its edge connections. This metric was utilized to characterize the connectivity of each light. For example, in the heart of downtown Chicago, most of the traffic lights have very short straight line paths, whereas some traffic lights on the edge of the city have some very long straight line paths that reach the other end of the city. The weight of the node

attempts to characterize the closeness to other traffic lights, a sort of modified clustering metric. Our final network graph is displayed below.

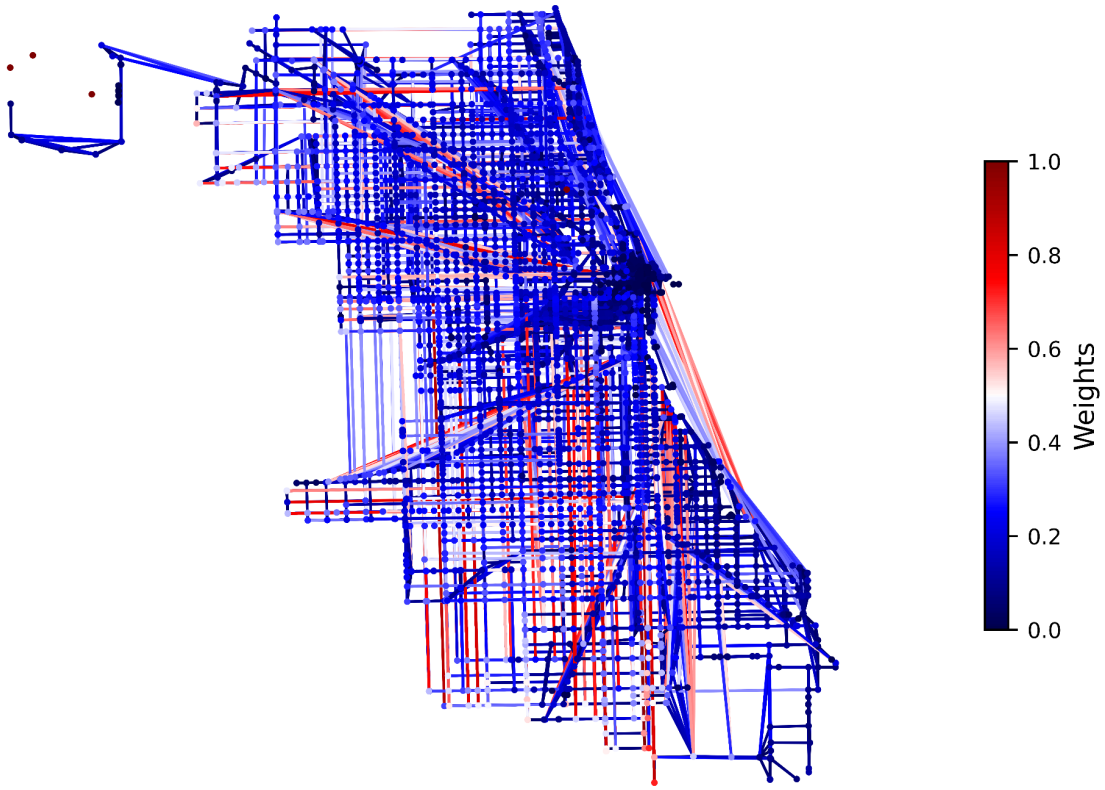

Figure 1. Network Model: Traffic Lights on Straight Line Paths

The next step was to overlay the speed camera locations over the traffic crashes to explore any correlations or patterns between traffic crashes and speed camera locations. An issue we faced here was with displaying this data in a readable format because there are so few camera locations in comparison to traffic crashes. We overcame this issue by adjusting the size and color of the camera location nodes and the traffic crashes nodes. The overlaid graph is displayed below with the gray nodes representing traffic crashes and red nodes representing speed cameras. An interesting note is how plotting the crashes clearly defines the streets and outlines the City of Chicago. This indicates that over the past several years car crashes have occurred almost everywhere in Chicago where you can drive, and this is just speed-related crashes.

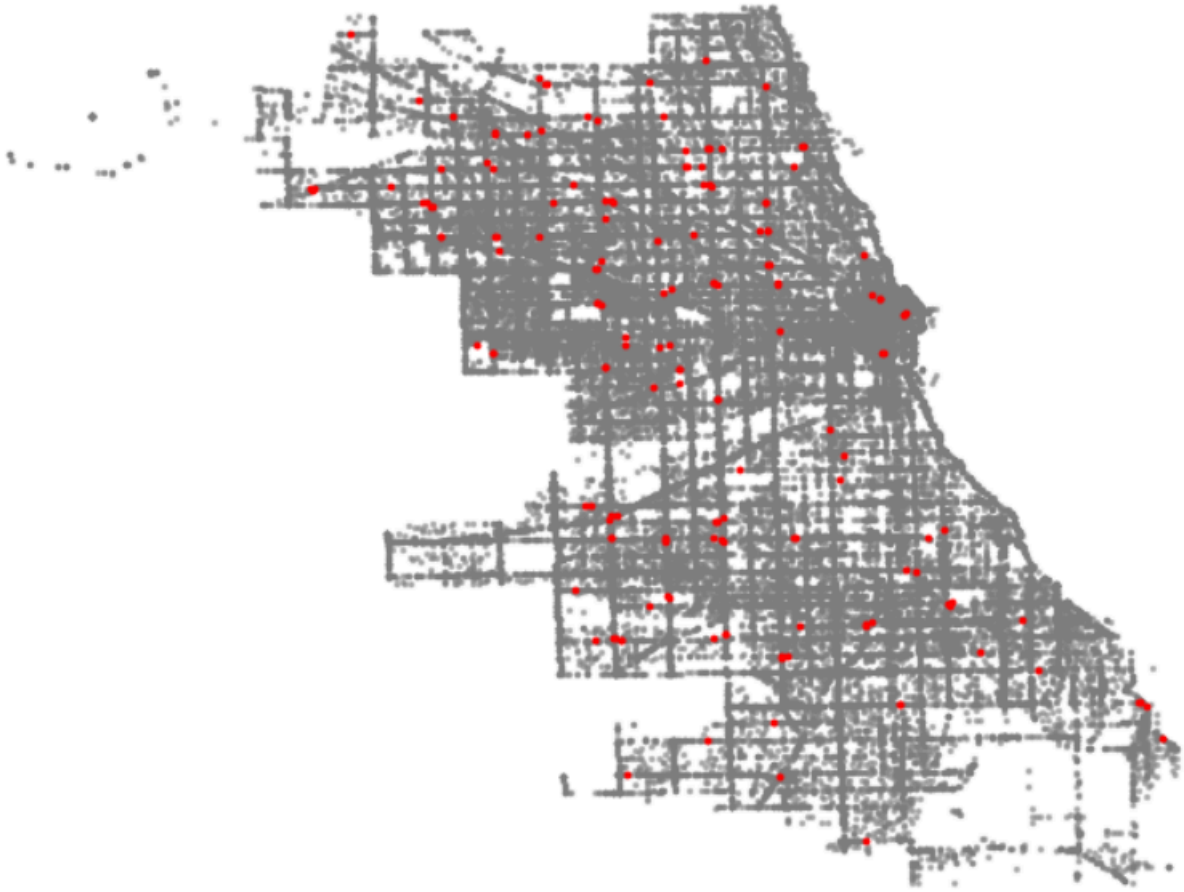

Figure 2. Overlay of Speed Cameras over Traffic Crashes

Before we can begin analyzing correlations, we need to calculate graph metrics for our nodes. We used NetworkX to calculate the degree centrality, betweenness centrality, closeness centrality, and clustering coefficient for each node (i.e. each traffic light). We faced one significant challenge, and that was ensuring that each value was associated with the correct node. When we calculated the metrics for each node they were stored in separate dictionaries. The issue is that dictionaries in Python are not guaranteed to conserve order so we couldn't simply combine the data together. Instead we had to query each dictionary individually for a specific node and then place them in an array where order is conserved, a process that took several hours.

We then calculated a crash density metric for each node that quantifies how many crashes occurred within a 1-block radius of the light. This allowed us to plot the metrics against crash density to visualize any correlations between the variables.

Lastly, we explored the effect of speed light cameras on the rate of car crashes. We did this by looking at the average crashes per year before and after the go-live date for the speed cameras.

An important note is that the Chicago Police Department did not start collecting all the crash data in Chicago until September 2017. There are several records from beforehand such as 2015, but they do not represent all the crashes from that time. We decided to filter out all crashes before September 2017 in order to have a fair comparison and avoid any biases with data collection before the CPD officially started collecting crash data.

We also had to sample speed camera lights that had at least a year of data before and after the go-live date to give a fair representation of how the speed camera affected the rate of crashes. This limited us to go-live dates between 01/01/2018 and 01/01/2023, giving us 13 speed cameras to analyze. We then counted the amount of crashes within a 1-block radius and calculated the average number of crashes per year before and after the go-live date.

Project Results

We used 5 metrics in total in our visual analysis: degree centrality, betweenness centrality, closeness centrality, clustering coefficient, and node weight. Each metric was plotted against the crash density on a scatter plot. The figures below show the results.

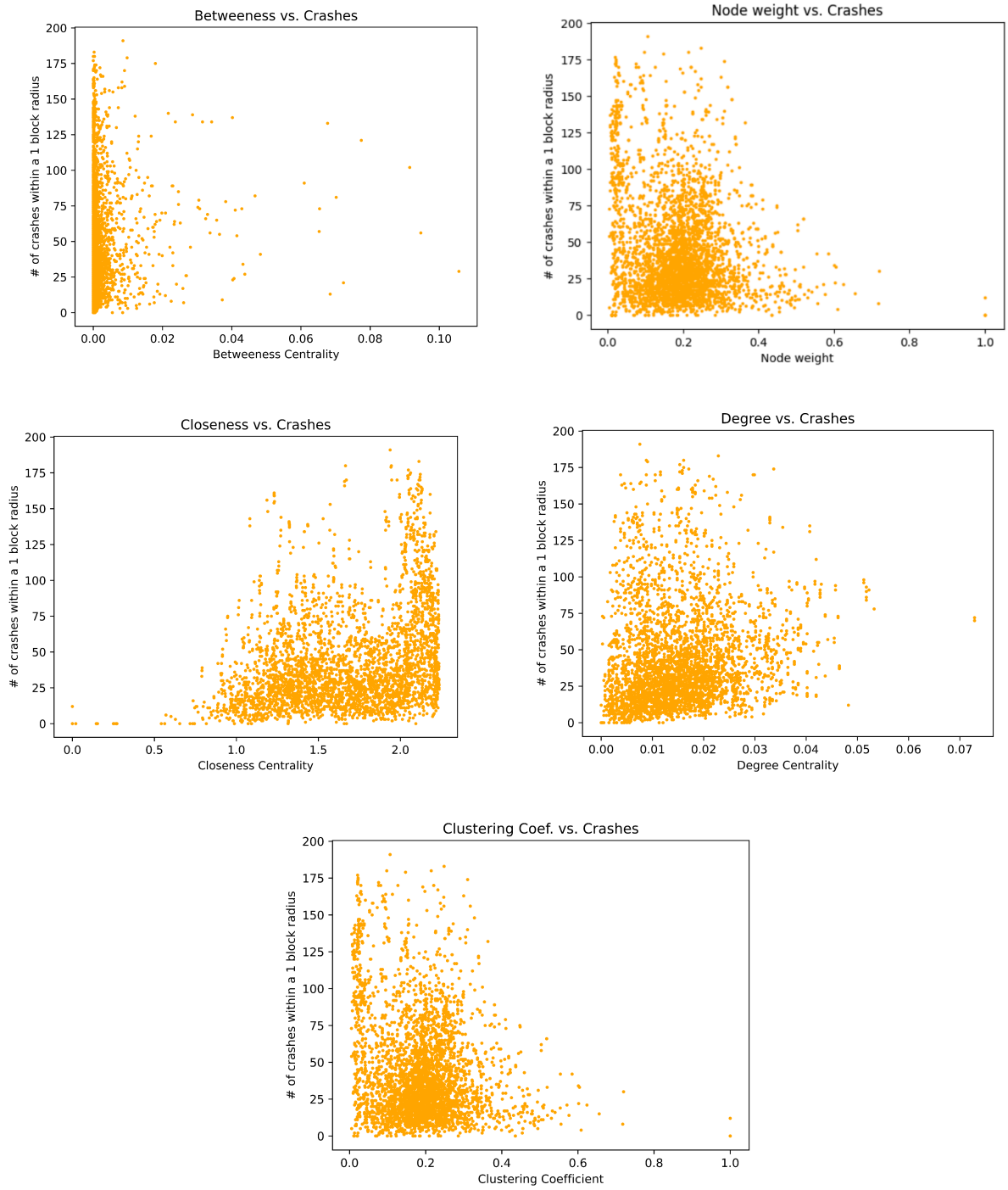

Figure 3. Various metrics plotted against crash density

We see that the data is pretty dispersed and random, there doesn't appear to be any clear correlations. The betweenness doesn't tell us much as the majority of the nodes have a betweenness less than 0.01. Degree centrality is also pretty random, although there does seem to

be a drop-off in the amount of crashes after a degree 0.03. However this is also a minority of nodes, and the majority of nodes seem to have a degree under 0.03 without any clear correlation.

One interesting thing to note is that the node weight and clustering coefficient plots look strikingly similar. Remember that our node weight is the average of all edge weights in order to characterize how close it is to other lights, a type of cluster characterization. They look so similar because both metrics measure clustering which indicates that our interpretation of our graph is consistent.

Now the clustering and closeness plots are also pretty random, but they have a characteristic spike. For clustering, the spike happens at the beginning, and for closeness the spike happens towards the end. This indicates that there is a correlation between low clustering/high closeness and a higher probability of having large numbers of crashes. This can be interpreted as traffic lights that are further away from other traffic lights increase the probability of having crashes. The symmetry between the closeness and clustering plots also make sense because a higher closeness indicates a longer distance to other nodes, in other words the opposite of clustering. Overall this can be interpreted as traffic lights that are far away from other traffic lights increase the probability of crashes in the area.

Next we analyzed the rate of crashes before and after speed cameras were installed. Here's a case study on the selected sample of cameras.

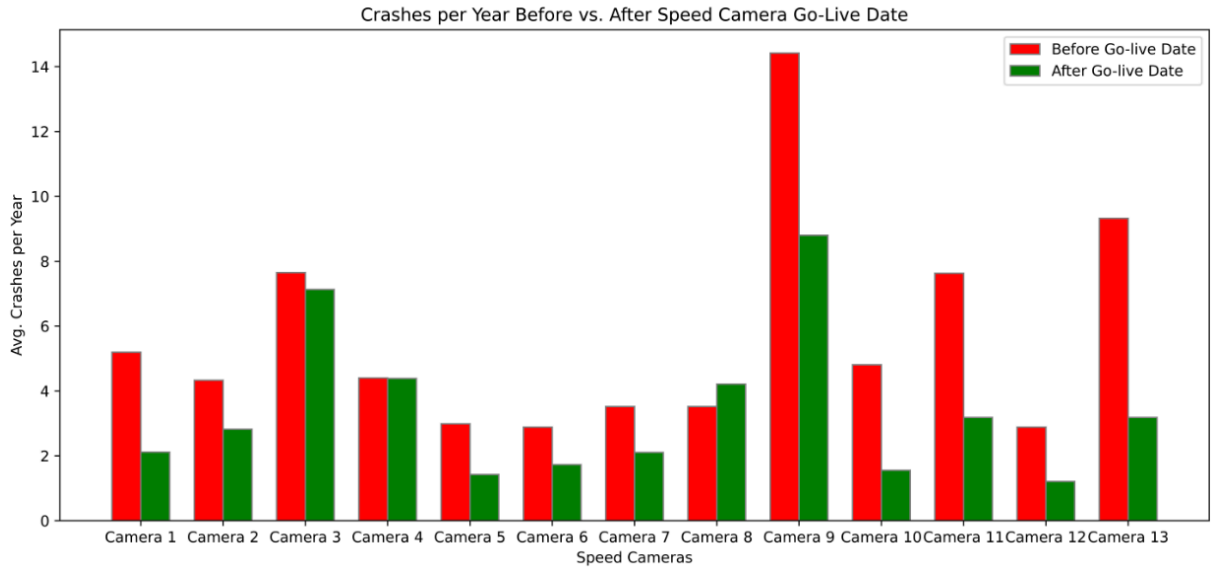

Figure 4. Rate of crashes before and after speed camera go-live date

Some cameras did not make a significant difference in the crash rate such as cameras 3, 4, and 8. However for the majority of cameras you can see a noticeable dip in the amount of speed-related car crashes after they were installed. Cameras 8, 11, and 13 see the most significant drop. Overall this histogram shows convincing evidence that speed cameras are effective in reducing the amount of crashes. This provides a good foundation for future analysis regarding speed lights.

Discussion of Project

Overall we see the project as successful, we were able to successfully create and analyze a graph of traffic light data and integrate it with the crash and speed camera datasets. There were many challenges in trying to collect and format the data for analysis, but we were able to successfully overcome them and continue with our initial plans.

One issue we ran into was the runtime for some of the processing steps. Some steps would take a few seconds per node to complete, and since we had thousands of nodes this process took several hours. One thing we'd do differently is use some type of cloud computing resource such as AWS to perform these lengthy and computationally expensive tasks. What takes several hours on our laptops may only take an hour using cloud computing resources. This would free up our time to perform more analysis as we're not stuck waiting several hours for results before moving on.

Overall we learned a lot about modeling. A big challenge we had was modeling our data as a connected network graph. Figuring out what we wanted the nodes and edges to represent taught us a lot about the considerations and design challenges that go into graph modeling. It also taught us how we can use centrality metrics on the graph to find correlations with other data. Overall we learned about new techniques to analyze data by constructing graphs.

Future Work

There is a lot of further exploration we can perform on these datasets. We can explore different graph representations to see if they provide any stronger correlations with crash datasets. We can also expand our scope beyond speed-related crashes and consider fatal crashes or even all crashes in general.

In terms of speed-camera analysis, we can do further exploration to see where speed cameras are most effective. We can also add the red-light camera dataset as well and perform the same analysis with car crashes. Additionally we can add historical traffic data to explore how traffic conditions and the quantity of cars in a given area, along with the other metrics we calculated, affects the rate of car crashes in the area.

References

Chicago Police Department. (2024). *Traffic Crashes - Crashes* [Dataset]. City of Chicago. https://data.cityofchicago.org/Transportation/Traffic-Crashes-Crashes/85ca-t3if/about_data

City of Chicago. (2024). *Map - Speed Camera Locations* [Dataset]. https://data.cityofchicago.org/Transportation/Map-Speed-Camera-Locations/7ajp-yjhe

Github. (2024). *CS579 Project 2* [Repository]. https://github.com/BlueOceanWave/CS579-Project-2

OpenStreetMap. (2024). - *OpenStreetMap Wiki.* https://wiki.openstreetmap.org/wiki/Main_Page

Overpass Turbo. (2024). - *Traffic Light Locations* [Dataset]. https://overpass-turbo.eu/