NYC Car Crash Prediction: Influence of intersection condition and surrounding environment on car accident occurrence

Hai Yang, Jiale Li, Jingjing Ge, Ruoru Feng, Tao Liang, Xinyu Xu, Yichen Guo

Applied Data Science CUSP-GX 6001 2022 Spring

Stanislav Sobolevsky

5/3/2022

Abstract

Traditionally, car accidents are mainly investigated based on the characteristics of involved entities. However, characteristics of the location and surrounding environment will also affect the probability of a car accident. Focusing on New York City, this paper discusses the various factors affecting car collisions, by the intersection related features and their prediction performance in three machine learning models: Random Forest, XGBoost and Graph Neural Network (GNN). We will be exploring two categories of attributes influencing the level of intersectional car crashes from 2015 to 2019 using data from open sources. One category is road network information such as road types and number of lanes. The other category is neighborhood information such as population density and multi-type land usage. The results show that Random Forest and XGBoost predict with accuracy over 0.6, while the GNN model predicts with accuracy over 0.5. The feature importance from the Random Forest model shows that the average number of lanes has the highest influence, followed by the betweenness connectivity and the presence of stop signs.

Introduction

Car accidents account for a significant portion of the serious injuries and deaths reported each year. According to the *Motor Vehicle Traffic Crash Statistics* (New York State Department of Health, 2022), on average, 292 people die as a result of motor vehicle traffic-related injuries each year in New York. Considerable studies have found that intersections are among the most dangerous places on a road network. It is frequently difficult to determine what specific kind of road conditions beyond human factors cause car collisions, making it more sophisticated for government and legislation to address the level of severity in the context of car accidents. With general awareness of certain characteristics like road conditions and surroundings playing an important role, many questions remain unanswered. Which of these factors has more impact? Which machine learning model is more suitable for predicting car crash occurrence?

Literature Review

Our study aims to predict the risk level of different intersections. Abdel-Aty et al study analyzed the intersection features that correlated with car crashes. They found out that different intersection characteristics can cause different types of crashes. They adopted the tree regression methodology to better handle the multicollinearity and missing value (Abdel-Aty et al., 2019). Bao and Ukkusuri

used historical Manhattan crash data and other traffic-related data to predict the short-term car crash risk. The attributes they use in their study are taxi GPS data, road network attributes, land-use features, population data, and weather data (Bao et al., 2019). Their study compares two methodologies which are econometric models as well as machine learning models. These two methodologies they provide can serve different predicting needs. They found out the econometric models did a better job predicting the weekly car crash risk while the machine learning models performed better in predicting the daily car crash risk. Rahman, Abdel-Aty, Hasan, and Cai's study conducted a decision tree model on predicting pedestrian-bicycles accidents with traffic, roadway, and socio-demographic characteristics as attributes. Finally, their study applies bagging, random forest, and gradient boosting models to improve the prediction accuracy of the primary decision tree model crash count (Rahman et al., 2019). Inspired by these studies, we would like to predict car accidents by using machine-learning algorithms, which will use features related to intersection designs and surrounding environments.

Data Collection and Processing

As the study focuses on the crash that happened specifically at the intersection, we used the links data pulled from LION Single Line Street Base Map to get the whole street network in New York City. Although the linked data contains nodes that identify the origin and ends of a segment, these nodes may not represent the actual intersections but as the middle point of a street. Therefore, we used the network to remove any node connecting only two segments to make sure that each node is an intersection (connect more than two segments). Then, the cleaned network was spatially joined with crash count from 2015-2019 after creating a 50-feet radius buffer around each node. We used a similar approach to join other features for each node, including sign existence, the environment surrounding the node location, etc. Each row represents a street segment that has a start node and end node. The final dataset with all the features is listed in the appendix A.

a. Crash data

In 2020, the COVID-19 pandemic drastically changed the traffic volume in New York City, which made the crash data significantly different from previous years. Therefore, we chose to avoid any crash record after 2019. We downloaded New York City car crash records from 2015 to 2019 provided by the New York Police Department (NYPD). In total, there are 1658584 recorded

crashes with valid location information. To further aggregate them into intersection-level, we count the crash occurrences within a 50-feet radius of each intersection. If a crash occurred at a place within the 50-feet radius of more than one intersection, all intersections count this crash as theirs. Since the majority of the intersections in NYC are more than 100 feet away from each other, the problem of over counting is negligible in our case.

Since we are more interested in the likelihood of car accident occurrences, not the exact number of car accidents, we further categorized the crash counts into 4 levels (Figure 1). The lowest level group has one or no crash at each intersection. The second lowest group has 2 to 11 crashes. The second highest group has 12 to 32 crashes. Finally, all intersections having more than 33 crashes belong to the highest group. The percentage split of all groups is shown in table 1. Though group 1 occupies a higher split of the whole data, the weight among the four groups is relatively balanced.

Group #	Range	Percentage
Group 1	<= 1 crashes	36.8%
Group 2	2 – 11 crashes	22.8%
Group 3	12 - 32 crashes	19.9%
Group 4	>= 33 crashes	20.5%

Table 1: Crash Class

b. Network

We used the LION single line street base map to construct our network. There are 149989 unique undirected vehicle links and 107994 unique nodes in the most current version. However, there are many redundant nodes that break a single road into several segments. To capture only intersections, we deleted all the redundant nodes which only connect to 2 distinct links and merged the two links those nodes connect to. There were 66418 nodes left after the cleaning process. We then rebuilt the whole network with 215512 directed links. For all one-way roads, we created reverse-directional links with a length of 100,000 feet to make sure all nodes were connected by two links in both directions. This step ensures the creation of a symmetrical adjacency matrix, which is critical to model building steps (Figure 2).

c. Road Characteristics

Road characteristics describe road conditions at each intersection. We used four features to define road characteristics: truck route existence, bike route existence, average number of lanes, and average number of parking lanes. All these features are stored in LION. Truck route existence is a dummy variable, which describes whether there is any truck route dedicated to the roads passing through the intersection. Same logic is also applied to bike route existence. For the latter two features, we took simple averages of total lanes and parking lanes all linked roads have for each intersection.

d. Betweenness Centrality

To fully utilize the network information in our prediction model, we chose to use betweenness centrality to capture the connectivity of each intersection. Betweenness centrality is a shortest path-based centrality measure, which describes how important a node is inside the network (Figure 3). A node has higher betweenness centrality if a higher percentage of shortest paths pass through the node. To calculate the exact value of betweenness centrality of all nodes in the network, 66418*66417 = 4,411,284,306 shortest path enumerations are required, which would take too long to run. Therefore, when calculating each node's betweenness centrality, we used a sub-network with 20,000 randomly sampled nodes to do the approximation.

e. Demography and Land-Use Data

The demography data (e.g., population density, employment rate, vehicle ownership rate etc.) were derived from ACS 5 2018 at census tract level (Figure 4). We collected land-use data from PLUTO, and calculated the category ratio of each land-use type in each census tract:

$$F_i = \frac{n_i}{N_i}$$

$$C_i = \frac{F_j}{\sum F_j}$$

Where i is one land-use type, j is all land-use types, n is type i's count in one census tract, N_i is type i's count within the whole city. F_i is frequency density and C_i is the category ratio of a land-use type.

We then spatially joined the census tract boundaries within the 200-feet buffer of the intersection and assigned the average demography and land-use value to the corresponding node (Figure 5).

f. Traffic Signs

We filtered major traffic signs that may affect the car crash from the Street Sign Work Orders provided by the Department of Transportation, including stop signs, slow signs, yield signs, no left turn signs, and speed limit signs. We used the similar method as before in which we spatially join the location of each sign within the 50-foot buffer of the intersection (nodes) and create a dummy variable for each node. Each type of the sign is processed with 1 indicating sign existed near the intersection while 0 indicating no such sign existed.

Methods

In this project, we selected three models to conduct the prediction work: XGBoost, Random Forest, and Graph Neural Network (GNN). For XGBoost, we used the XGBoost library to construct the classifier model. For Random Forest, we used the Random Forest Classifier function in the Sklearn library to construct the model. For GNN, we used the Pytorch library to implement the full model.

We chose five hyper-parameters to tune the Random Forest model, the best parameter set is described below.

Hyper-parameter	Search Area	Optimal value
n_estimators	(10, 300)	91
max_depth	(1, 21)	6
min_samples_split	(2, 22)	19
min_samples_leaf	(1, 11)	3
max_features	(5, 30)	10

Table 2: Random Forest Tuning Result

To efficiently tune XGBoost hyper-parameters, we used the Bayesian Optimization (BO) library to fulfill the task. The BO library uses Bayesian Inference techniques to approximate the optimal set of hyper-parameters within a confined domain. Four hyper-parameters were chosen to be tuned: learning rate, number of estimators, maximum depth, and gamma. We ran 35 iterations of BO and

used the best parameter set as our final hyper-parameter input. Search areas of selected hyper-parameters, as well as the optimal value obtained from BO are shown in Table 3.

Hyperparameter	Search Area	BO Optimal value
Learning Rate	(0.01, 0.05)	0.03
Number of Estimators	(100, 1000)	292
Maximum Depth	(3, 10)	4
Gamma	(0, 5)	4

Table 3: Xgboost Tuning Result

To construct and implement the GNN model, we first transformed the data to graph form. Demographics, building environment, and network characteristics data - which represent nodes and constitute node's features, and distances between intersections - which represent edges and constitute edges' features, were used to establish the network mobility matrix for the model. Three main hyper-parameters were chosen to be tuned: learning rate, weight decay, and percentage of nodes masked. We trained the GNN model with the best parameter set to perform inference on data described by graphs.

Hyperparameter	Search Area	Optimal value
Learning Rate	[0.0001, 0,001, 0.01]	0.01
weight decay	[10e-3,10e-4,10e-5]	10e-4
percentage of nodes masked	(0,5, 0,9)	0.9

Table 4: GNN Tuning Result

Results

Model Accuracy

We used the full node dataset to run our XGBoost and Random Forest model, which contains 66418 rows and 30 columns. While for the GNN model, we initially utilized the full network data set, which contains 215512 rows and 29 columns because of the exclusion of betweenness connectivity. However, the process of adjacency matrix construction exceeded our RAM capacity. To make the dataset manageable, we reduced the network size and chose to use the Manhattan

network to run our GNN model. Therefore, our final input data for the GNN model only included 19682 rows and 29 columns. The performance of both XGBoost and Random Forest were both measured by 10-fold cross validation accuracy. For the GNN model, accuracy of test data was the final performance measure. Table 5 shows the performances of the three models.

	Random Forest	XGBoost	GNN
Accuracy	10-fold CV: 0.6058	10-fold CV: 0.6066	0.5010

Table 5: Model Results

Feature Importance

Figure 6 shows the feature importance sorted from high to low of the Random Forest model. Obviously, the "Number_Tot_int", which means average number of lanes at each node, is the top contributing feature in our model, followed by the "Betweenness" (only used in Random Forest and XGBoost) and "stop-dummy" (stop sign present or not). The feature importance of "Number_Par_int" (average number of parking lanes at each node) ranks fourth, almost 50% lower than the previous one, followed by "Pop_D" (gross population density on unprotected land) and "Res_D" (Gross residential density on unprotected land) at almost the same importance level. The table also shows land-use related features didn't show much significance on the likelihood of car collision occurrence.

Discussion

Random forest and XGBoost did a decent job in predicting long-term car crash risk. Accuracy in both mentioned models is slightly over 0.6, while the accuracy in GNN model reached 0.5010. However, improvements can be made by taking actions below:

- Take crash incidents related factors into account. For example, distraction features such as the number of dogs along the street could potentially increase the crash likelihood. Other data such as traffic jams, traffic flow, commercial density, employment density, distance to city center, major arteries, and more detailed demographics also have the possibility of improving the model performance.
- Access more resources to run more complex models such as GNN. As mentioned in previous sections, our current RAM resource would not fully utilize the whole network dataset when running our GNN model. We finally used only the Manhattan area network. With higher

resource accessibility, the performance of the model could be improved with larger datasets and more locational information in different areas.

- Some features used can have underlying correlation such as population density, employment density and residential density. This leads to collinearity and may undermine the statistical significance of the independent variables we are interested in.

Conclusion

In model performance, Random Forest and XGBoost both provide decent accuracy in predicting the level of car accidents happening in NYC after avoiding the impact of the Covid-19 pandemic on traffic volume. Theoretically GNN should have a decent performance using the network data, but due to limited resources, our GNN model was costly running on large networks data. If time permitted, the performance of the GNN model could be improved with enough resources. According to the feature importance of the Random Forest model, traffic volume related features are most significant when predicting the car accident occurrence, while the land-use related features play less important roles. Future improvements can be made by adding other features, accessing advanced resources and preprocessing data for better interpretability.

Tables and Figures

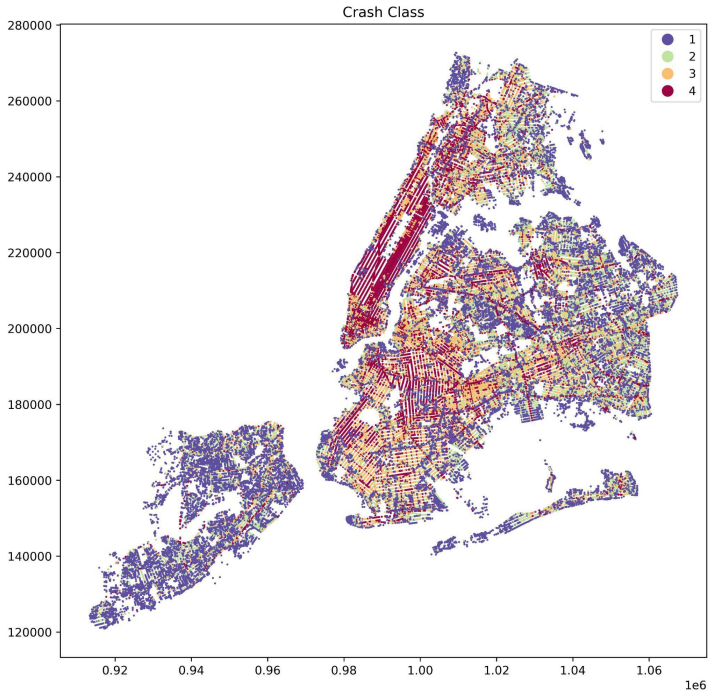

Figure 1: Car Accident Level

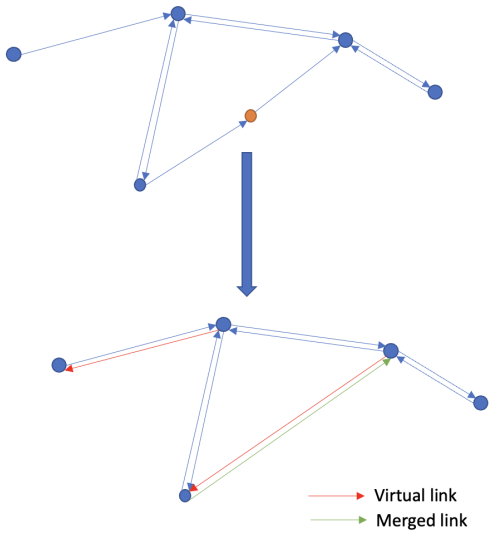

Figure 2: Network Cleaning

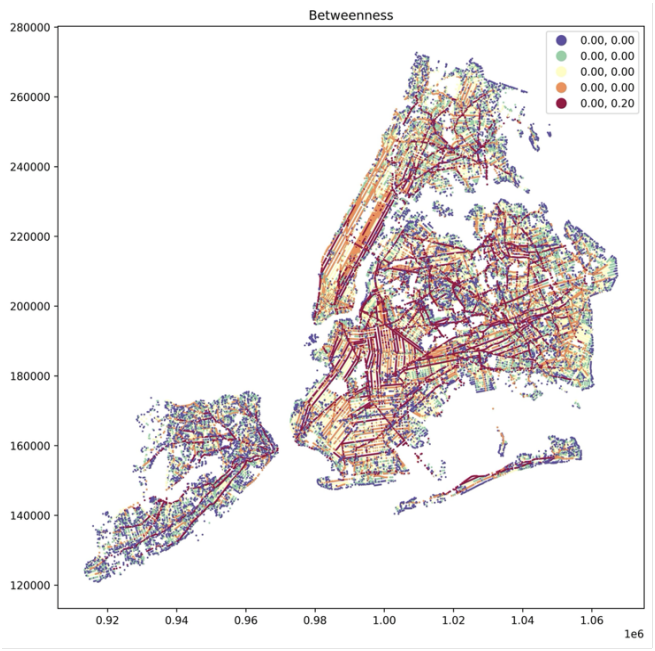

Figure 3: Betweenness

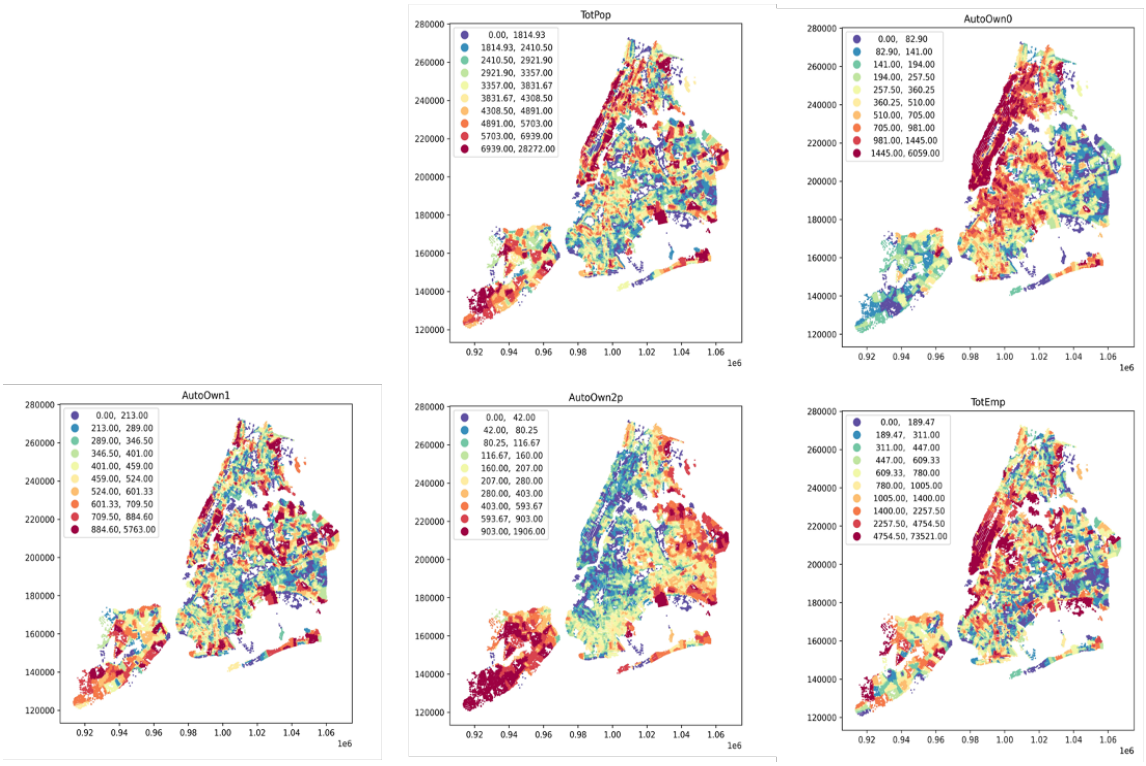

Figure 4: Demographic Data

Figure 5: Land-Use Data

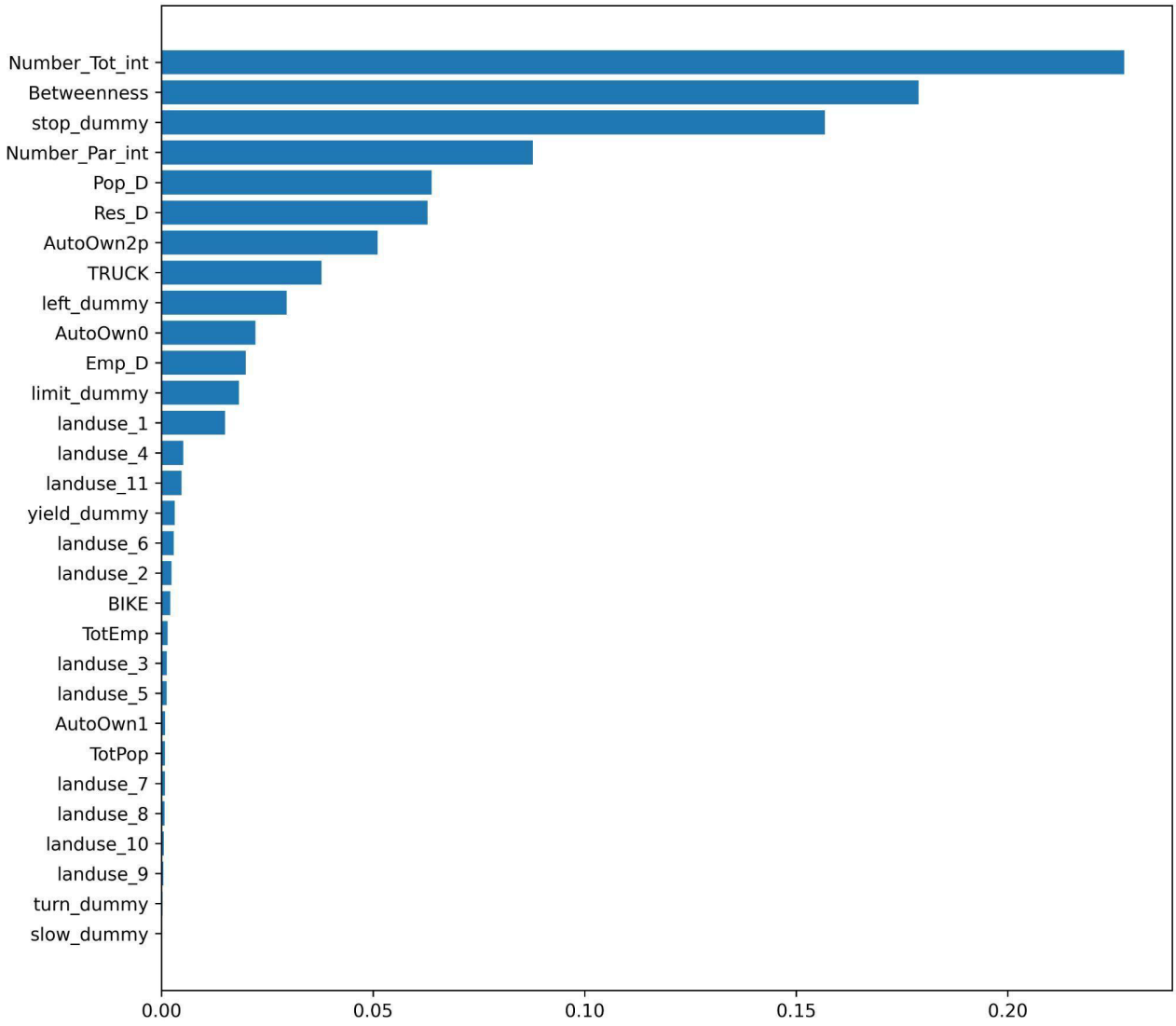

Figure 6: Feature Importance (RF)

Reference

Abdel-Aty, M., Keller, J., & Brady, P. A. (2005). Analysis of types of crashes at signalized intersections by using complete crash data and tree-based regression. Transportation Research Record, 1908(1), 37-45.

Abdulhafedh, A. (2017). Road Crash Prediction Models: Different Statistical Modeling Approaches. Journal Of Transportation Technologies, 07(02), 190-205. doi: 10.4236/jtts.2017.72014

AG, M. T. (n.d.). Satellite raster tiles for North America openstreetmap tiles, geodata and opendata maps. Satellite raster tiles for North America OpenStreetMap Tiles, GeoData and OpenData Maps | MapTiler Data. Retrieved May 2, 2022, from https://data.maptiler.com/downloads/dataset/satellite/north-america/#1.01/60.4/-92.6

Bao, J., Liu, P., & Ukkusuri, S. V. (2019). A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data. Accident Analysis & Prevention, 122, 239-254.

Bureau, U. S. C. (n.d.). Explore census data. Retrieved May 2, 2022, from https://data.census.gov/cedsci/

Environmental Protection Agency. (n.d.). Interactive maps and data for measuring location efficiency and the built environment. EPA. Retrieved May 2, 2022, from https://www.epa.gov/smartgrowth/smart-location-mapping

Henrick, C. (n.d.). NYC Crash Mapper, by Chekpeds. NYC Crash Mapper. Retrieved May 2, 2022, from https://crashmapper.org/#/

New York State Department of Health. (2022). Motor Vehicle Traffic Crash Statistics: New York State Residents. Retrieved 2 May 2022, from https://www.health.ny.gov/statistics/prevention/injury_prevention/traffic/county_of_residence.htm

Police Department (NYPD). (2014, April 28). Motor Vehicle Collisions - Crashes. NYC.

Appendixes

DATA		DESCRIPTION (SOURCE)
Crash_Count		Quantile count of collisions in New York per NodeID from 2015 to 2019 (NYPD)
Network Characteristics	**NodeID**	Default ID assigned to each node in LION
	Target	End NodeID of each link (LION)
	weight	Distance between two NodeIDs (ft, LION)
	betweenness	Only used in Random Forest and XGBoost, betweenness connectivity of all nodes with subsample of 20000
	Number_Tot_int	Average number of lanes at each node (LION)
	Number_Par_int	Average number of parking lanes at each node (LION)
Build Environment	**turn_dummy**	1: signs present that indicate the lane is ending or turning, dead end or U turn; 0: signs not present (DOT)
	yield_dummy	1: yield sign present; 0: not present (DOT)
	left_dummy	1: no-left-turn sign present; 0: not present (DOT)
	limit_dummy	1: speed limit sign present; 0: not present (DOT)
	stop_dummy	1: stop sign present; 0: not present (DOT)
	slow_dummy	1: slow sign present; 0: not present (DOT)
	BIKE	1: bike way present; 0: no bike way present (LION)
	TRUCK	1: truck route present; 0: no truck route present (LION)
Demographic Data	**TotPop**	Population (ACS 5)
	AutoOwn0	Number of households in Census Tract that own zero automobiles (ACS 5)
	AutoOwn1	Number of households in Census Tract that own one automobile (ACS 5)
	AutoOwn2p	Number of households in Census Tract that own two or more automobiles (ACS 5)
	TotEmp	Total employment (ACS 5)
	Res_D	Gross residential density (household/acre) on unprotected land (ACS 5)
	Pop_D	Gross population density (people/acre) on unprotected land (ACS 5)
	Emp_D	Gross employment density (jobs/acre) on unprotected land (ACS 5)
	landuse_1	Land use for one- & two-family buildings (PLUTO)
	landuse_2	Land use for multi-family walk-up buildings (PLUTO)
	landuse_3	Land use for multi-family elevator buildings (PLUTO)
	landuse_4	Land use for mixed residential & commercial buildings (PLUTO)
	landuse_5	Land use for commercial & office buildings (PLUTO)
	landuse_6	Land use for industrial & manufacturing (PLUTO)
	landuse_7	Land use for transportation & utility (PLUTO)
	landuse_8	Land use for public facilities & institutions (PLUTO)
	landuse_9	Land use for open space & outdoor recreation (PLUTO)
	landuse_10	Land use for parking facilities (PLUTO)
	landuse_11	Land use for vacant land (PLUTO)

Appendix A: All features used in model running

Authors Contribution

Hai Yang:
Data hunting, aggregation and cleaning, final data check, model design, RF and XGBoost models running, report editing, PowerPoint & presentation

Jiale Li:
Idea generation, data aggregation and cleaning, model design, GNN models running, feature and data selection

Jingjing Ge:
Data hunting, aggregation and cleaning, data merge, feature and data selection, report editing

Ruoru Feng:
Idea generation, notetaking, data hunting, citation check, report editing

Tao Liang:
Data hunting, processing and merge, model design, RF and XGBoost models running, feature selection, PowerPoint

Xinyu Xu:
Team operations and management, data arrangement, report editing, PowerPoint & presentation, video

Yichen Guo:
Data hunting, aggregation and cleaning, data merge, GNN models running, report editing, PowerPoint & presentation