Predicting Crash Occurrence and Injury Severity at Intersections in Texas: MLE vs ML Models

Theodore Charm
Haoqi Wang

December 2021

Descriptive Statistics

This paper obtained the data from various sources. Crash records from 2010 to 2019 were acquired from the Texas Department of Transportation (TxDOT) Crash Records Information System (CRIS, 2020). The CRIS system collects crash reports occurring in intersections from the police all 254 Texas counties. In addition, this paper employed the TxDOT Roadway Inventory database to acquire intersection-specific attributes.

The CRIS data was spatially matched with land use, population, employment, median income, median age, precipitation, and other location attributes (hospitals, schools, and transit stops). Census tract-level variables (population, employment, median income, and median age) were obtained from the American Community Survey dataset (ACS, 2020). The 2015-2019 ACS 5-year estimates were used in the analysis. This paper also used the annual rainfall data (1981 to 2010) from the Texas Water Development Board (2014) to obtain county-level average yearly precipitation.

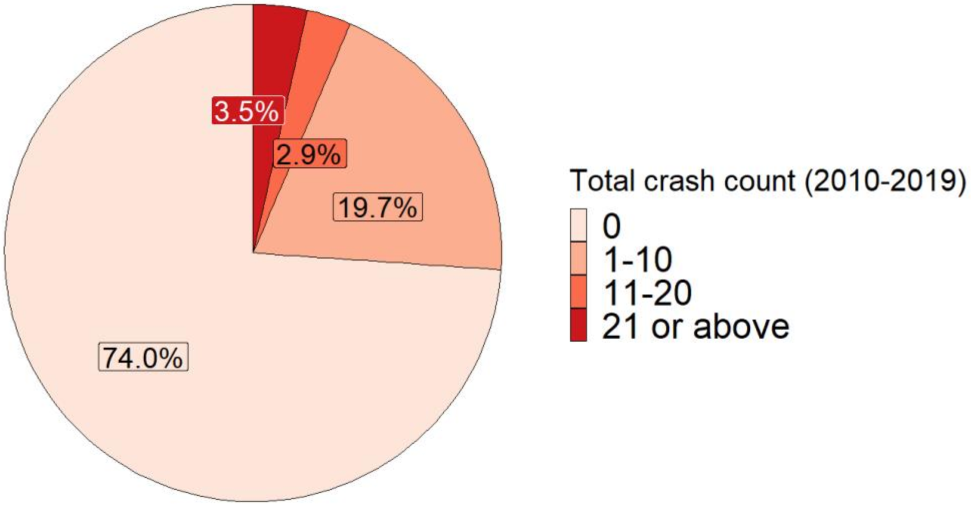


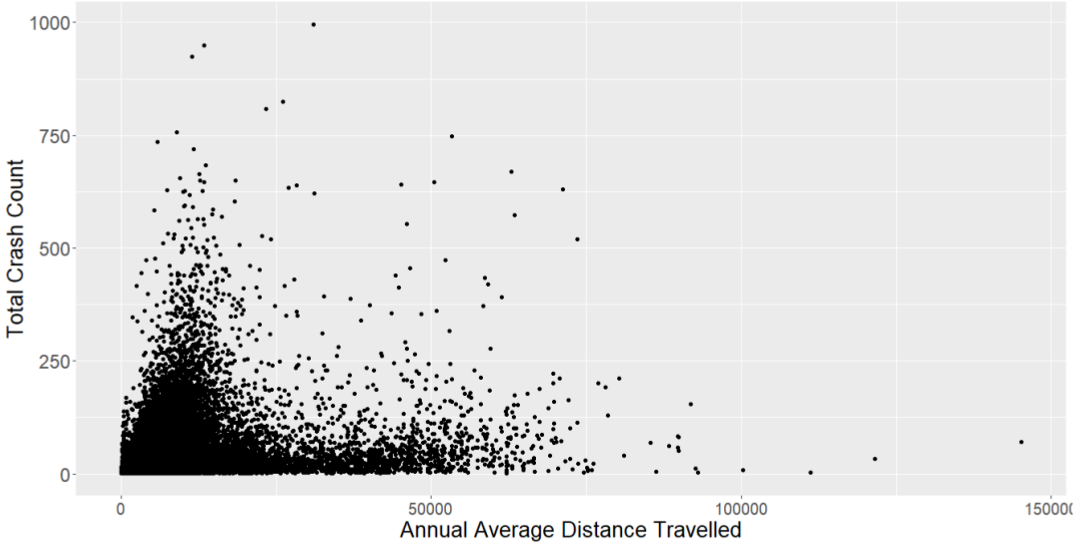
Figure 1: Total crash occurrences at Texas intersections in 2010-2019 (no. of observations = 707,161)

Total crash counts for 707,161 intersections statewide from 2010 to 2019 were obtained. Among all intersections, 522,933 (74%) had 0 crashes recorded during the period. 19.7% had 1 to 10 crashes, 2.9% had 11 to 20 crashes, and less than 4% had 21 or more traffic crashes. Figure 1 illustrates the distribution of the crash counts. Table 1 presents the summary statistics of the variables at the intersection and census-tract levels.

Table 1: Summary statistics for the variables

Variable	Mean	Std. Dev	Min	Median	Max
Total police-recorded crashes from 2010 to 2019	3.18	15.62	0	0	996
Length of sidewalk within 150 ft of intersection centroid	10.81	63.72	0	0	1092
Number of lanes major approach	2.23	0.72	1	2	8
Number of lanes minor approach	2.03	0.25	0	2	8
Presence of median on the major approach	0.014	0.12	0	0	1
Presence of median on the minor approach	0.0021	0.046	0	0	1
Intersections located on the TxDOT system	0.16	2.14	0	0	1
Median width major approach (ft)	0.56	7.70	0	0	519
Median width minor approach (ft)	0.085	3.35	0	0	519
Lane width major approach (ft)	10.5	2.11	0	10	49
Lane width minor approach (ft)	9.85	1.26	0	10	49
Shoulder width major approach (ft)	0.72	2.34	0	0	38
Shoulder width minor approach (ft)	0.065	0.70	0	0	32
Annual average daily traffic (AADT) major approach	1,141	3,208	0	188	142,733
Annual average daily traffic (AADT) minor approach	221	607	0	136	62,054
Percentage of truck in the major approach	4.85	5.43	0	3.2	95.8
Percentage of truck in the minor approach	3.44	2.25	0	3.2	93.3
Walk-miles traveled per area	326	454	0	155	15,339
Walk-miles traveled per capita	0.14	0.035	0.094	0.13	0.40
Walk-miles traveled	772	484	0	675	4,443
Speed limit major approach (mph)	57.02	6.50	10	58.88	85
Speed limit minor approach (mph)	58.54	3.03	10	58.88	85
Local major approach	0.67	0.47	0	1	1
Local minor approach	0.93	0.25	0	1	1
Collector major approach	0.18	0.38	0	0	1
Collector minor approach	0.052	0.22	0	0	1
Arterial major approach	0.14	0.12	0	0	1
Arterial minor approach	0.015	0.12	0	0	1
Unknown major approach	0.0067	0.082	0	0	1
Unknown minor approach	0.00090	0.030	0	0	1
Rural (pop: <5,000)	0.27	0.44	0	0	1
Small urban (pop: 5,000-49,999)	0.12	0.32	0	0	1
Urbanized (pop: 50,000-199,999)	0.11	0.31	0	0	1
Large urbanized (pop: 200,000+)	0.50	0.50	0	0	1
Signalized intersection	0.02	0.15	0	0	1
Number of approaches arriving in the intersection	3.19	0.68	0	3	5
Distance to nearest school (miles)	1.41	2.28	0	0.55	18.64
Distance to nearest hospital (miles)	5.10	5.16	0.017	2.83	18.64

Transit presence within 0.25 miles of intersection centroid	0.021	0.14	0	0	1
Count of transit stops within 0.25 miles of intersection centroid	0.067	0.62	0	0	26
Population density (per acre)	3.51	3.92	0	2.18	96
Job density (per acre)[1]	2.71	3.07	0	1.62	65.66
Median income (in USD)[2]	32,370	13,792	2,499	29,025	124,355
Median age[3]	37.25	6.72	18.8	36.5	73.7
Average yearly precipitation (1981 to 2010) (inches)[4]	36.62	11.18	9.85	37	59.59



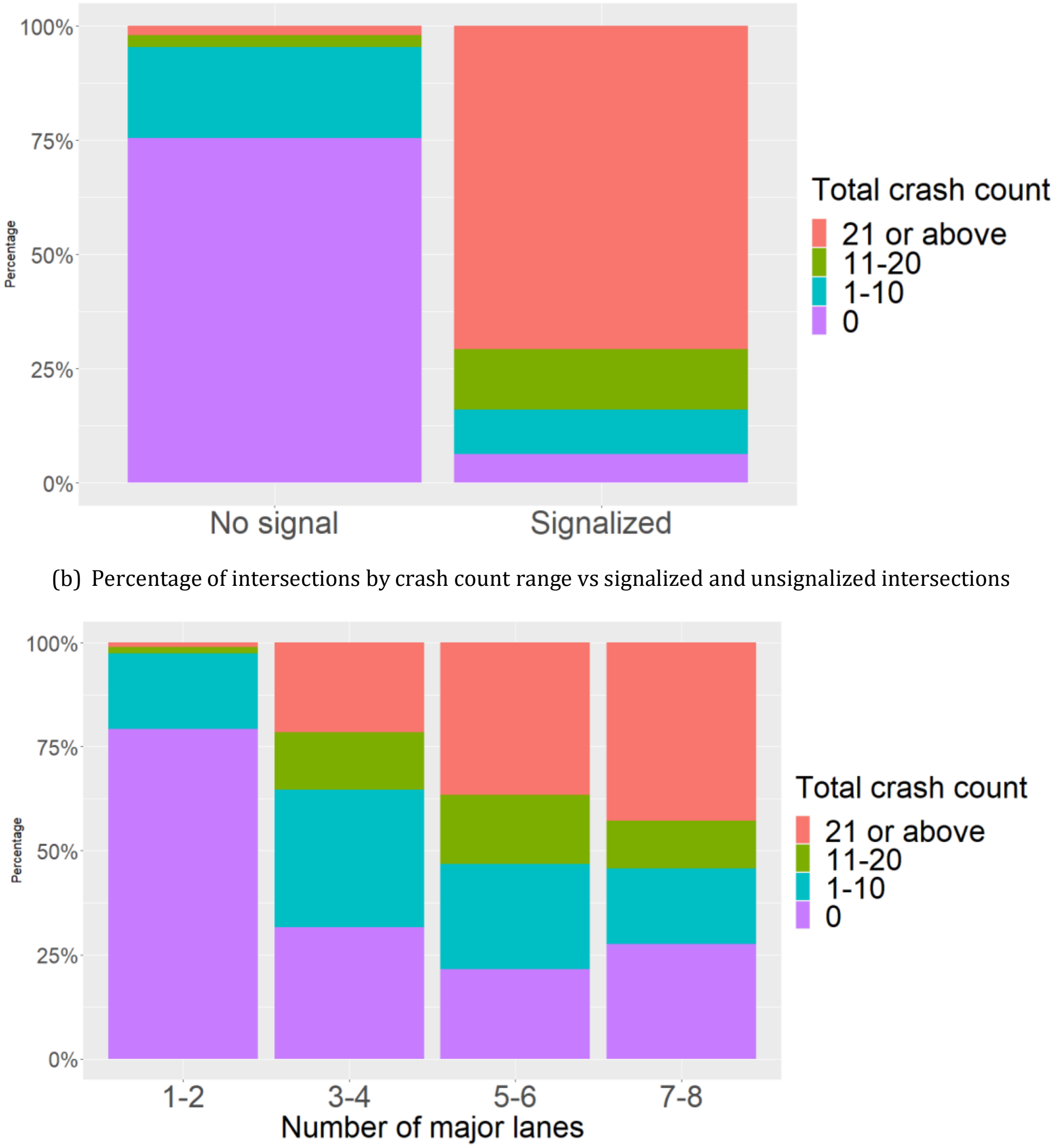
(a) Scatter plot for crash counts vs AADT

[1] Population and employment densities were calculated by the total population (or jobs) divided by the areas (in acres) of each census tract, using the 2015-2019 ACS 5-year estimate.

[2] This was measured by the median household income of each census tract, using the 2015-2019 ACS 5-year estimate.

[3] This was measured by the median age of each census tract, using the 2015-2019 ACS 5-year estimate.

[4] This was measured by the average yearly precipitation of each county, from 1981 to 2010, using the Texas Water Development Board precipitation data.

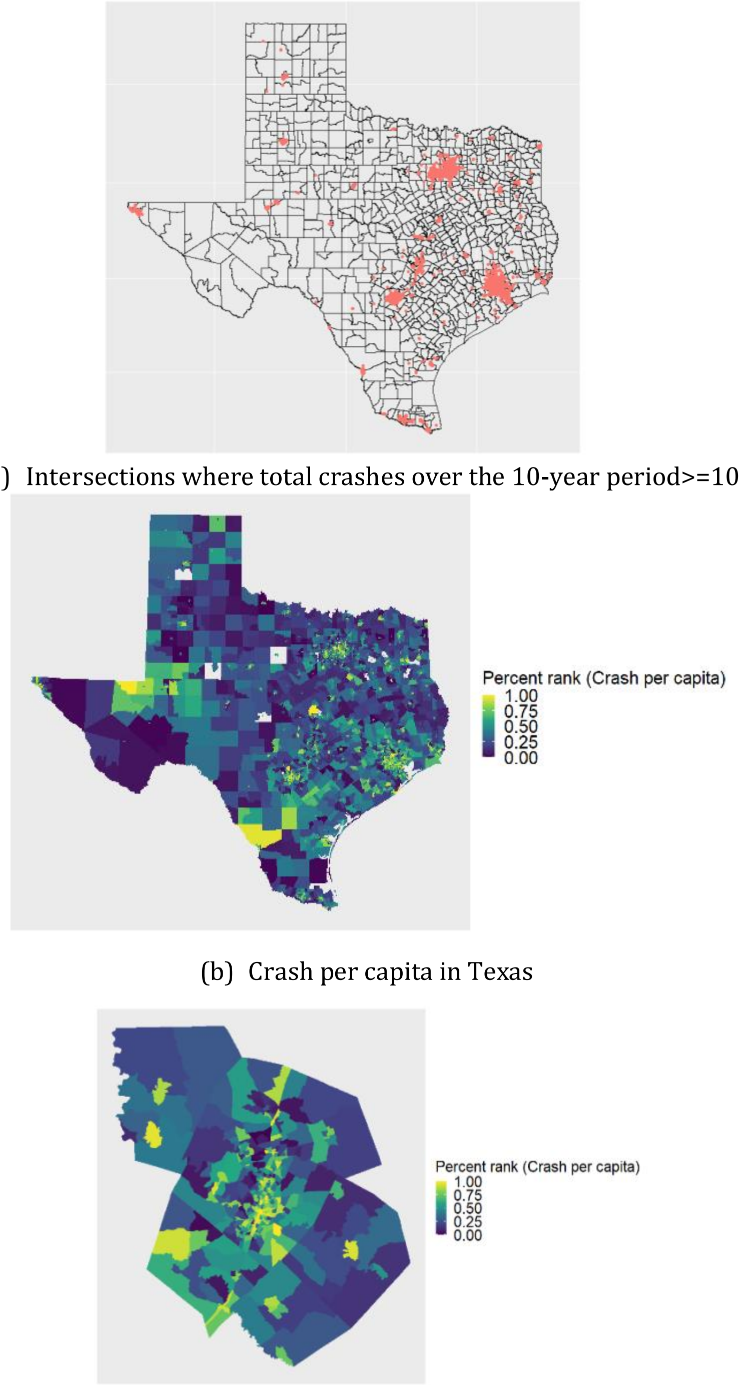


(b) Percentage of intersections by crash count range vs signalized and unsignalized intersections

(c) Percentage of intersections by crash count range vs lane count

Figure 2

The association between crash counts and a number of explanatory variables is illustrated in Figure 2, in particular annual average daily traffic (AADT), signalized intersection, and number of lanes at major approach. The sum of AADTs for the major and minor approaches was computed, and the crash counts against the sum of AADTs is plotted in Figure 2a. It shows that intersections with frequent crashes tend to have higher-than-average AADTs. Figure 2b shows that for intersections with no signals, most of them had very few numbers of crashes. Nonetheless, for the signalized intersections, a high proportion of them had relatively high number of crashes. In particular, about 70% and 40% of signalized intersections had more than 20 crashes and 50 crashes from 2010 to 2019 respectively. Figure 2c illustrates that when the number of lanes at the major approach increases, the proportion of low-crash intersections decreases while the high-crash intersections increase its share. For example, most intersections with 1 or 2 major lanes had no crashes, but for intersections with 5 to 6 major lanes, about 37% had 21 or above crashes. For intersections with 7 to 8 major lanes, about 40% had more than 50 crashes. The graphs provide some evidence that intersection crashes are positively correlated with AADT, signalized intersections, and number of lanes at major approaches.


Percent rank (Crash per capita)
1.00
0.75
0.50
0.25
0.00
Percent rank (Crash per capita)
1.00
0.75
0.50
0.25
0.00

(a) Intersections where total crashes over the 10-year period>=100

(b) Crash per capita in Texas

(c) Crash per capita in Austin, Texas

Figure 3: Census-tract level variables

This paper also provides visualization of the crash counts at the census-tract level. Figure 3a locates the intersections that had over 100 crashes count over the 2010-2019 period (i.e. over 10 crashes per year on average). Such intersections are represented by the red dots in the map. Figure 3b illustrates the crash per capita for each census tract. Crash per capita was computed by dividing the total number of crashes by the population within the census tract. The percent ranks for each value are represented by different colors. Higher percent ranks are closer to the yellow end of the color spectrum, while lower percent ranks are closer to the purple end. The yellow spots are concentrated in large cities in Texas, where the census tracts have higher population densities. That indicates that large cities are more likely to have higher average crash counts than the rural counterparts. Specifically, it suggests there is an association between population density and crash counts at the census tract level. To better capture the relationship, this paper examines the Austin metropolitan area as an example, which includes the counties of Bastrop, Burnet, Caldwell, Hays, Travis, and Williamson. Figure 3c illustrates that the more densely populated census tracts tend to have higher average crash counts, in particular Travis County and the areas along the IH-35 corridor. In the next section, statistical models were used to study the association of crash counts with the explanatory variables.

Regression Models

As a baseline, zero-inflated negative binomial models were conducted to appreciate the effects of various explanatory variables on the total (10-year) crash counts at each of Texas' 707,161 intersections. Since over 70% intersections had 0 crashes recorded, this paper used a zero-inflated count model. As the standard deviation of the outcome was significantly higher than the mean, the data was over-dispersed. In light of this issue, this paper used a negative binomial model instead of a Poisson model. Therefore, zero-inflated negative binomial models were employed for regression analysis. This paper presents the full model, which included all explanatory variables in the analysis. Since the variables were measured on different scales, this paper standardized all explanatory variables to make the values comparable. It first subtracted the means from the original values. It then divided by the standard deviation to obtain the standardized values.

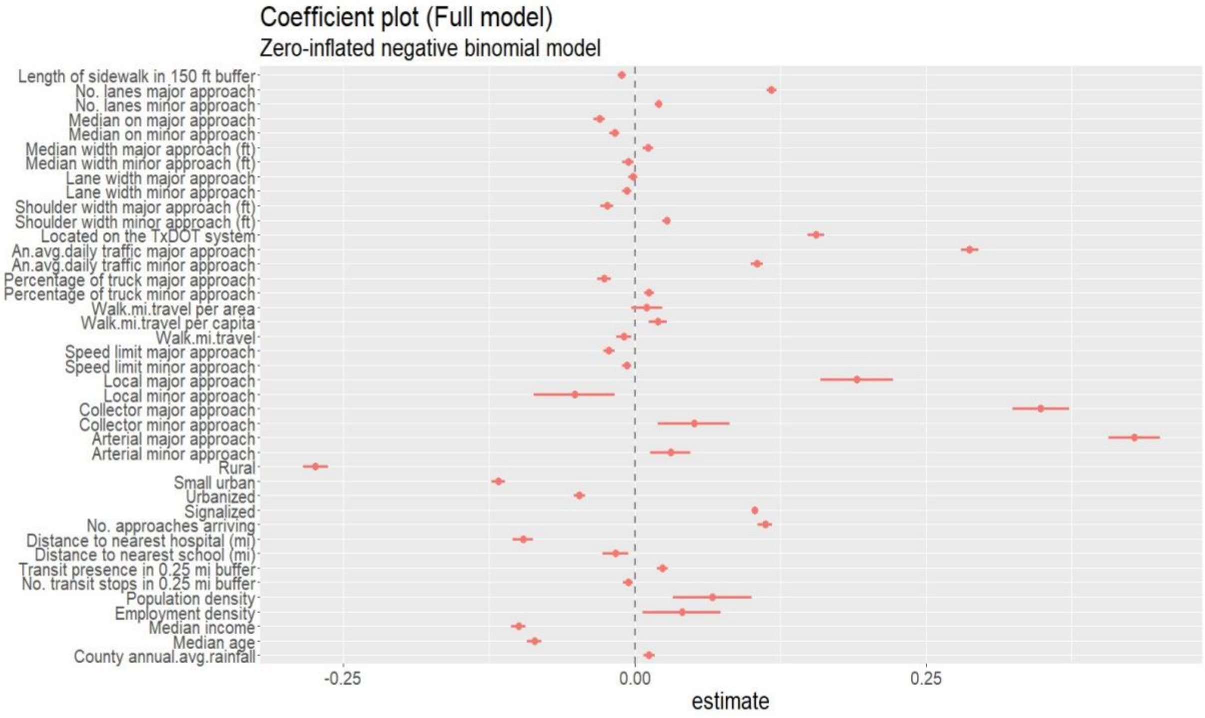


Figure 4

Figure 4 presents the coefficient plot for the zero-inflated negative binomial model. One may interpret the coefficients as follows: “For one unit change in the predictor, the difference in the logs of expected counts of the outcome variable is expected to change by the respective regression coefficient, given other predictors are held constant”[5]. The results in Figure 4 are to a large extent consistent with Figure 2. Crash counts are positively correlated with AADT, signalized intersections, and number of lanes in the major approach. Population density also has a positive effect on traffic crashes. Regarding other road-specific attributes, presence of median on approaches shows negative effect. Lane widths of approaches show negative effects. Walk-miles traveled per capita increases crash counts, while speed limits at the approaches tend to decrease crash counts. Number of approaches arriving in the intersection has positive effect. Other location features also show significant effects. Distance to the nearest hospital reduces crash counts, but the presence of transit within 0.25 miles of the intersection centroid increases crash counts. As to census-level attributes, population density and average annual rainfall demonstrate positive effects, while median income and median age show negative effects.

Tree-based Machine Learning Models

Next, various tree-based ensemble machine learning (ML) models were used to predict crash occurrences at intersections across Texas, including random forest, extreme gradient boosting (XGBoost), light gradient boosting (Light GBM), and Bayesian additive regression trees (BART). The models had 42 features in total. This paper evaluated the performance of the models in the predictions. The procedures were as follows: (1) randomly split the data into 75% training and 25% test sets; (2) fit the model on the training data and generate predictions; and (3) evaluate model performance with various metrics, namely R-squared and root mean squared error (RMSE).

Random Forest

A random forest regression consists of decision trees generated by “splitting each node using the best among a subset of predictors randomly chosen at that node with a different bootstrap sample of the data” (Zhao et al., 2021). The random forest method computes the final prediction value based on the average results of each decision tree (Liaw and Wiener 2002; Li and Kockelman 2021). The number of trees was set to 500 in the random forest regression. This paper used the squared error to measure the quality of the split and considered all features when looking for the best split.

XGBoost

Chen and Guestrin (2016) devised the XGBoost method as a scalable ML system for gradient tree boosting. XGBoost constructs consecutive small trees with each tree correcting the net error from the previous trees (Zhao et al., 2021). XGBoost is trained in a forward ”stage-wise” manner, aiming to minimize the sum of squared errors by tuning the parameters continuously (Li and Kockelman, 2021). ”The first tree is split on the most predictive feature, and then the weights are updated to ensure that the subsequent tree splits on whichever feature allows it to correctly classify the data points that were misclassified in the initial tree. The next tree will then focus on correctly classifying errors from that tree, and so on. The final prediction is the weighted sum of all individual predictions” (Zhao et al., 2021). In the XGBoost training model, the maximum depth of the trees was set to 6, the number of rounds for boosting was 500, and eta (learning rate) was 0.1.

[5] This was cited from the UCLA Statistical Group website. See the bibliography.

Light GBM

The Light GBM method is particularly useful for large datasets (Ke et al., 2017). It incorporates gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) (Li and Kockelman, 2021). The GOSS algorithm keeps all the instances with larger gradients while randomly drops those instances with smaller gradients (Li and Kockelman, 2021). Light GBM speeds up the training process, thus reducing the computational time significantly. In the Light GBM model, the maximum number of leaves was set to 6, number of boosting iterations was 1000, and learning rate was 0.1.

BART

BART is a Bayesian non-parametric approach that fits a model using an influential prior distribution (Chipman et al., 2010). BART is a Bayesian "sum-of-tree" model in which "each tree is constrained by a regularization prior to be a weak learner" (Chipman et al., 2010). It performs iterative fitting and inference through conducting the back-fitting Monte Carlo Markov Chain (MCMC) that generates samples from a posterior. BART is robust to hyperparameter settings and addresses uncertainties with a Bayesian approach (Zhao et al., 2021). However, the method requires a lot of memory and time for computation. The number of trees was set to 100 for the model's training.

Neural Networks

Neural networks were implemented to solve three problems:

1. Predict the sum of 10 years of total crash count.

2. Predict the sum of 10 years of crash count by severities and total crash count.

3. Predict crash count by severities and total crash count in year 2019

As data preparation for neural network, the data was split into the training and testing data, 80% (565,440 examples) and 20% (141,260 examples), respectively. Numerical variables were standardized, in which case the values were centered around the mean with a unit standard deviation. Categorical variables were one-hot encoded. The model had 53 features in total, including geometric features (provided by Natalia) and demographic features (provided by Theodore).

The models were trained with processor Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz, RAM 32 GB, GPU NVIDIA Quadro T2000.

The code is available at GitHub. The parameters for training neural networks are listed below unless otherwise specified:

- Device: cuda

- Loss function: mean squared error

- Optimizer: Adam (Kingma and Ba, 2017)

- Batch size: 100

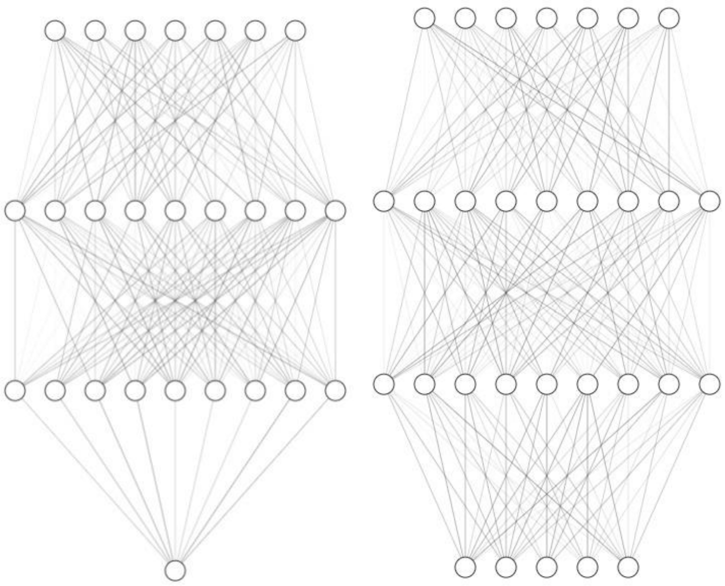
- Number of epochs: 10
- Learning rate: 0.0001
- Number of hidden layers: 3
- Hidden layer dimension: 100

After the model is trained, the model inputs testing data and acquires predicted values. It then compares the predicted values and ground truth to calculate root mean squared error (RMSE) and R^2. Training data is also evaluated to indicate if the model is over trained.

Problem 1: Total Crash Counts in 10 Years

Here, the goal is to predict the sum of 10 years of total crash count, given an intersection with 53 features. This paper builds neural networks with fully connected layers as shown in Figure 5a. Training and evaluation take about 3 minutes in total.

The total loss is decreasing as the epoch increases. Evaluation of training data gives RMSE of 10.756 and R^2 of 0.514. Evaluation of testing data gives RMSE of 11.163 and R^2 of 0.516. The evaluation metrics indicate the model is not overtrained and the neural network explains about 51% variability in data.



(a) Neural network 1 (b) Neural network 2

Figure 5: Neural network structure for problems 1, 2

Problem 2: Crash Count by Severities

Here, the goal is to predict the sum of 10 years of crash count by severities and total crash count, given an intersection with 53 features. The crash by severities is listed as followed:

- Low = not injured + unknown

- Incapacitated

- Nonincapacitated

- Possible

- Killed

This paper has crash count data by severities from year 2010 to 2019 of different intersections. It aggregates the count by severities over 10 years. Severity of not injured, unknown is added as low severity. The total crash count is the sum of all the crash count by severities.

The model is almost the same as the model in problem 1. The only difference is that the number of nodes in output layer is 6 instead of 1 as shown in Figure 5b. The total loss is decreasing as the epoch increases. Evaluation of training data gives RMSE of 3.765 and R^2 of 0.367. Evaluation of testing data gives RMSE of 3.913 and R^2 of 0.369. The evaluation metrics indicate the model is not overtrained and the neural network explains about 37% variability in data.

Problem 3: Predicting Crash Count by Severities in Year 2019

Here, this paper aims to predict crash count by severities and total crash count in year 2019, given 53 features mentioned above and crash count by severities and total crash count from 2010 to 2018. In other words, this paper intends to use previous time series data as predictor. The data is the same as that in problem 2. Missing years indicate no crash and this paper manually fills missing years with 0s.

A deep neural network integrating gated recurrent unit (GRU) (Cho et al., 2014) network is proposed for predicting the crash count in the year of 2019 as shown in Figure 6. The proposed neural network includes two components: the GRU and fully connected layers. The proposed model can integrate the spatial and temporal dependencies of the of variables. Temporal data from 2010 to 2018 is input to GRU layer, which is developed to capture the temporal dependencies and extract the temporal features. Merging temporal features with other features is an important part. This paper proposes two ways of merging and one baseline model without temporal features. In order to train the models faster, the hidden layer dimension becomes 50 instead of 100 as aforementioned. The bath size is changed to 1.

1. The processed temporal features pass through a fully connected layer. Then the output is merged with geometric and demographic features by matrix multiplying. After another fully connected layer, the model output the predictions of crash count by severities in year 2019. See Figure 6a.

2. The processed temporal features are concatenated with geometric and demographic features. After two fully connected layer, the model output the predictions of crash count by severities in year 2019. See Figure 6b.

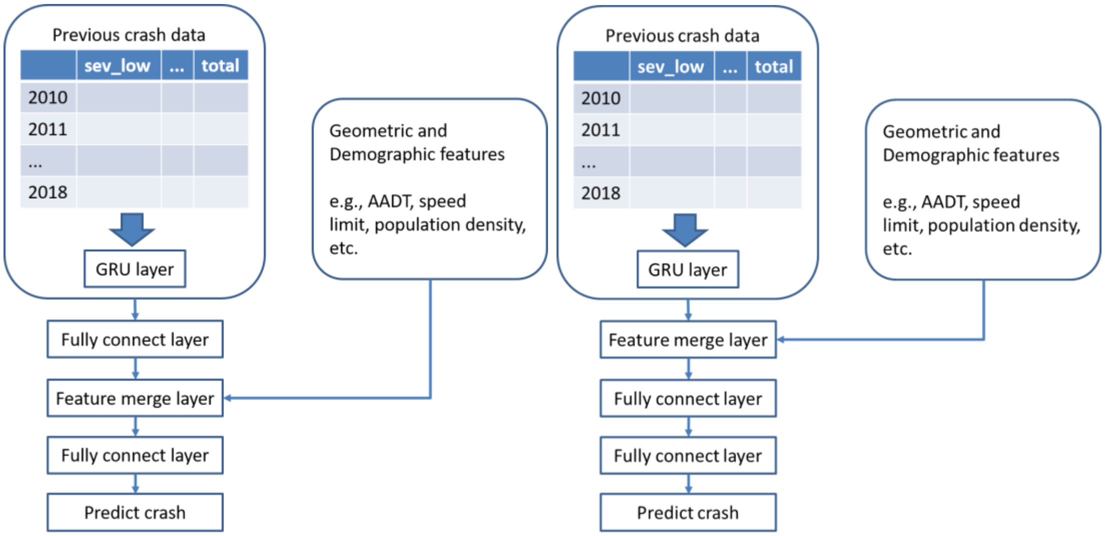
3. Baseline model. The model structure is the same as the neural network in problem 2 as Figure 5b.

   The output is changed to the crash count by severities in the year 2019.

The methods used are briefly explained as follows.

Temporal features extracted from GRU

Artificial neural network usually cannot capture temporal features for time-series data. To solve this problem, this paper proposes the recurrent neural network (RNN), which successfully handles sequence data, whereas traditional RNNs have the drawbacks of gradient vanishing or exploding (Pascanu et al., 2013).



(a) Architecture 1

(b) Architecture 2

Figure 6: Neural network structure for problems 3

Therefore, GRU neural network was developed by introducing gates into traditional RNNs, which can deal with long sequences. The structure of GRU is shown as Figure 7, more details can be found at http://dprogrammer.org/rnn-lstm-gru.

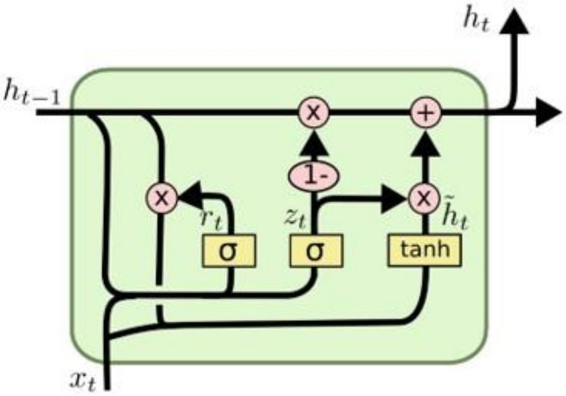


Figure 7: GRU layer detail

For each time step, the gates will be iteratively calculated by equations 1, 2, 3 and 4.

$$z_t = \sigma\ (W_z \cdot [h_{t-1},x_t] + b_z) \tag{1}$$

$$r_t = \sigma\ (W_r \cdot [h_{t-1}, x_t] + b_r) \tag{2}$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h) \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{4}$$

where h_t = hidden layer vectors, x_t = input vector, b_z, b_r, b_h = bias vector, W_2, W_r, W_h = parameter matrices and σ, tanh = activation functions.

GRU cell maps the input vector sequence *x* to a hidden vector *h* by iterations. In this context, *x* are input previous crash by severities data, the number of iterations is 9, which indicates the previous 9 years, and *h* is the estimated value.

Result

The evaluation results are shown in Table 2. The result of merging method 1 is similar to baseline and only RMSE is slightly better than it. The result of merging method 2 is much better than the baseline. Its RMSE decreases by 26.6%, R^2 increases by 66.0% compared with baseline. This indicates the first merging method fails to incorporate features. The second merging method can use temporal features and improve the prediction.

Table 2: Model results

	RMSE	R_2	Training time
Merging method 1	0.579	0.24	4 hrs
Merging method 2	0.428	0.4	2.6 hrs
Baseline	0.583	0.241	2 mins

Balanced and Unbalanced Data

In the crash dataset, there were 522,933 (74%) zero-count intersections and 184,228 (26%) non-zero-count intersections. That made the data highly imbalanced. To address this issue, the dataset was resampled by implementing the *ovun.sample* function of the *ROSE* package in R, which is a "bootstrap-based technique that helps the task of binary classification in the presence of rare classes" (Lunardon et al., 2021). *Ovun.sample* generated synthetic balanced samples through a combination of randomly oversampling the minority class (intersections with non-zero crashes) and undersampling the majority class (intersections with zero crashes). In particular, it used bootstrapping to draw synthetic samples from the feature space neighborhood around the minority class to create new rows of new data for the minority class. It also randomly selected a set of majority class observations and removed those observations from the dataset (He and Garcia, 2009). After resampling, the numbers of zero-crash and non-zero crash intersections were approximately equal (zero crash: 353,813 and non-zero crash: 353,113), thus the balance of the dataset was adjusted. The modified sample was denoted as balanced data.

Signalized vs Unsignalized Intersections

Signalized intersections and AADTs exerted disproportionately high weights on the model predictions (This will be explained further in Section 5.4). As a result, other features were not well accounted for in the predictions. To deal with this problem, this paper subsetted the data into signalized intersections and unsignalized intersections. It also only included the intersections where the sum of AADTs of the incoming links exceeded 500 (i.e. excluding the low-volume sites). After subsetting the data, this paper found that there were 15,222 signalized intersections and 235,822 unsignalized intersections. Among the unsignalized intersections, 121,983 had zero crashes and 113,839 had non-zero crashes.

R-square and RMSE

Table 3 presents the summary of the model performances, in terms of R-square and root mean square error (RMSE). R-square and RMSE are commonly used metrics to evaluate model fit and performances for ML models (Li and Kockelman, 2021). Using the original (or imbalanced) data, it found that the R-squares of the ML models were not particularly high. Zero-inflated negative binomial model produced the worst predictions, as it yielded the lowest R-square and highest RMSE among all models. Concerning the five ML models, random forest regression resulted in the highest R-square (0.534) and XBART had the lowest R-square (0.508). The RMSE ranged from 10.64 to 19.71. Light GBM yielded the lowest RMSE, followed by random forest. The RMSEs indicated unsatisfactory predictions of the models. There were two possible reasons regarding this issue. First, the data contained a high proportion of zero-crash intersections. Second, there were a number of extremely values. For example, the maximum number of crashes was 996. Model predictions are likely to be affected by the extreme values. Resampling the data led to better predictions for some of the models. The R-squares increased across the models, with random forest reaching a R-square of above 0.8. RMSEs, on the other hand, only showed improvement for three models. RMSEs for the random forest, XGBoost, and ZINB models decreased by 2.07, 0.19, and 7.36 respectively after resampling. Random forest's RMSE saw the most significant improvement. However, the RMSEs for Light GBM and XBART increased by 1.92 and 7.82 respectively. That indicated poorer predictions for the two models, especially BART.

Table 3: Comparison of model performance: Imbalanced vs balanced data

	Imbalanced data		Balanced data	
	R-square	RMSE	R-square	RMSE
ZINB	-1.442	59.03	-5.979	51.67
Random Forest	0.534	10.66	0.832	8.59
XGBoost	0.527	10.69	0.753	10.50
Light GBM	0.531	10.64	0.647	12.56
BART	0.508	19.71	0.602	27.53
Neural Network	0.516	11.16		

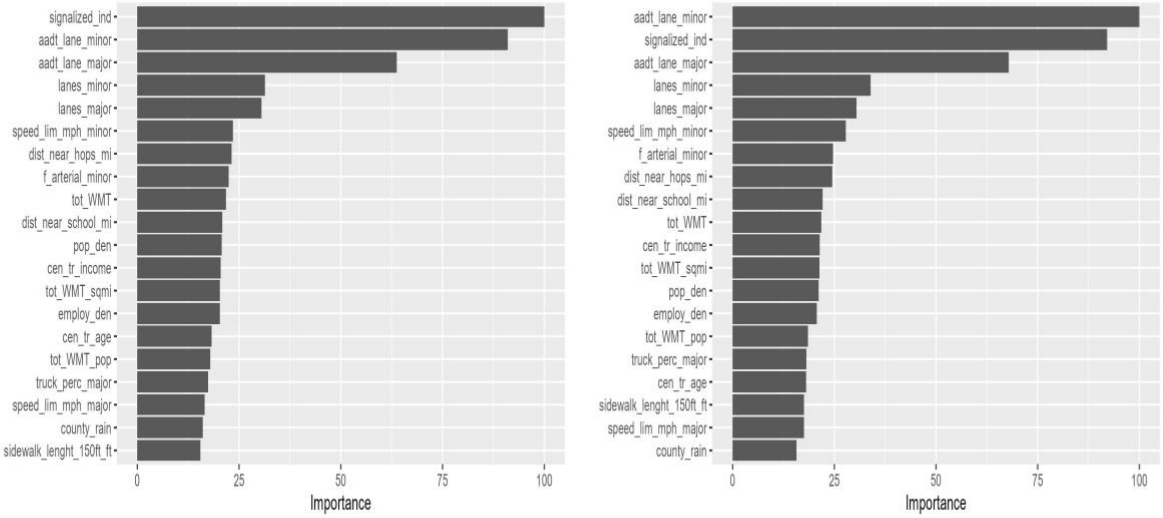
Table 4 compares the performances of the ML models between signalized and unsignalized intersections. It shows that the R-squares are comparable across the two groups, but the RMSEs are much higher for signalized intersections. This is partly due to the higher variation of crash counts, in particular the higher number of extreme values, at signalized intersections. Unsignalized intersections had many more zero crash counts, thus yielding lower RMSEs that are comparable to Table 3. Considering model performances, one can see that the random forest model yielded the best model performance overall.

Table 4: Comparison of model performance: Signalized vs unsignalized intersections

	Signalized		Unsignalized	
	R-square	RMSE	R-square	RMSE
Random Forest	0.245	63.12	0.287	10.47
XGBoost	0.243	63.66	0.241	10.67
Light GBM	0.261	62.87	0.232	10.74

BART	0.203	83.69	0.194	14.27

Feature Importance

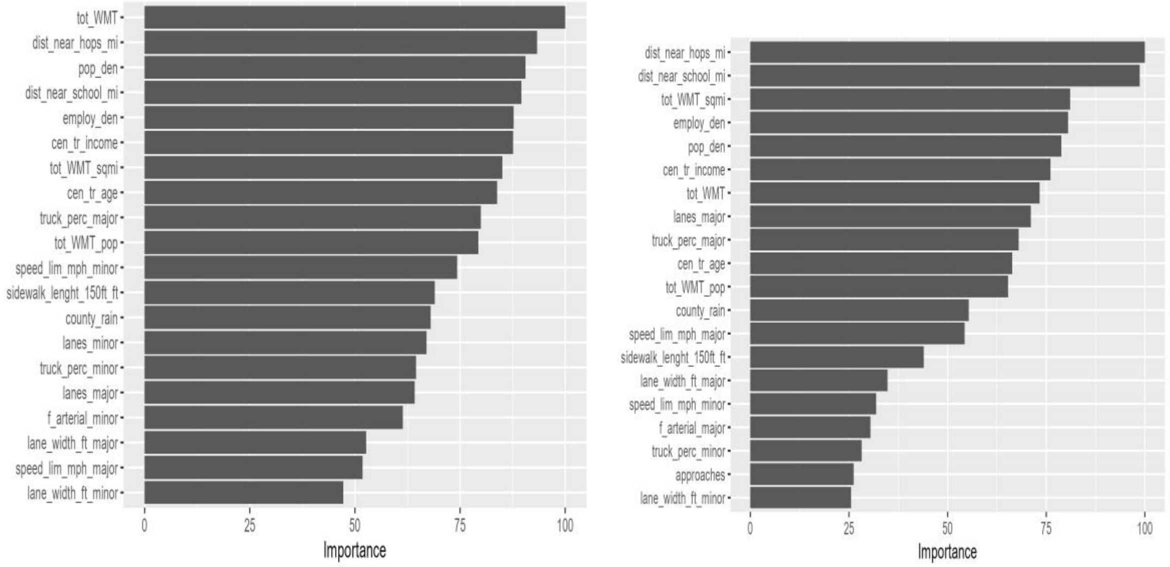


(a) Unbalanced data

(b) Balanced data

Figure 8: Feature importance for the Random Forest models

Given random forest had the best model predictions in the analysis, this paper is interested in the relative importance each feature has on the predictions of the random forest model. Figures 8a and 8b illustrate the feature importance of individual features using the imbalanced and balanced data respectively. The graphs include the top 20 features in terms of importance. This paper scaled all measures of importance, such that the top feature had a maximum value of 100. The figures show that signalized intersections are the top feature, followed by AADTs, number of lanes of the approaches, and speed limit of the minor approach. Other important features included distance to the nearest hospital, distance to the nearest school, arterial minor approach, and walk-miles traveled. A number of census-tract level attributes were also important, including population density and median income. It is noteworthy that signalized intersections, and to some extent AADT at minor approach and AADT at major approach, had exceptionally high feature importance compared to other features. The three features exerted disproportionately high weights on the random forest model predictions.



(a) Signalized intersections

(b) Unsignalized intersections

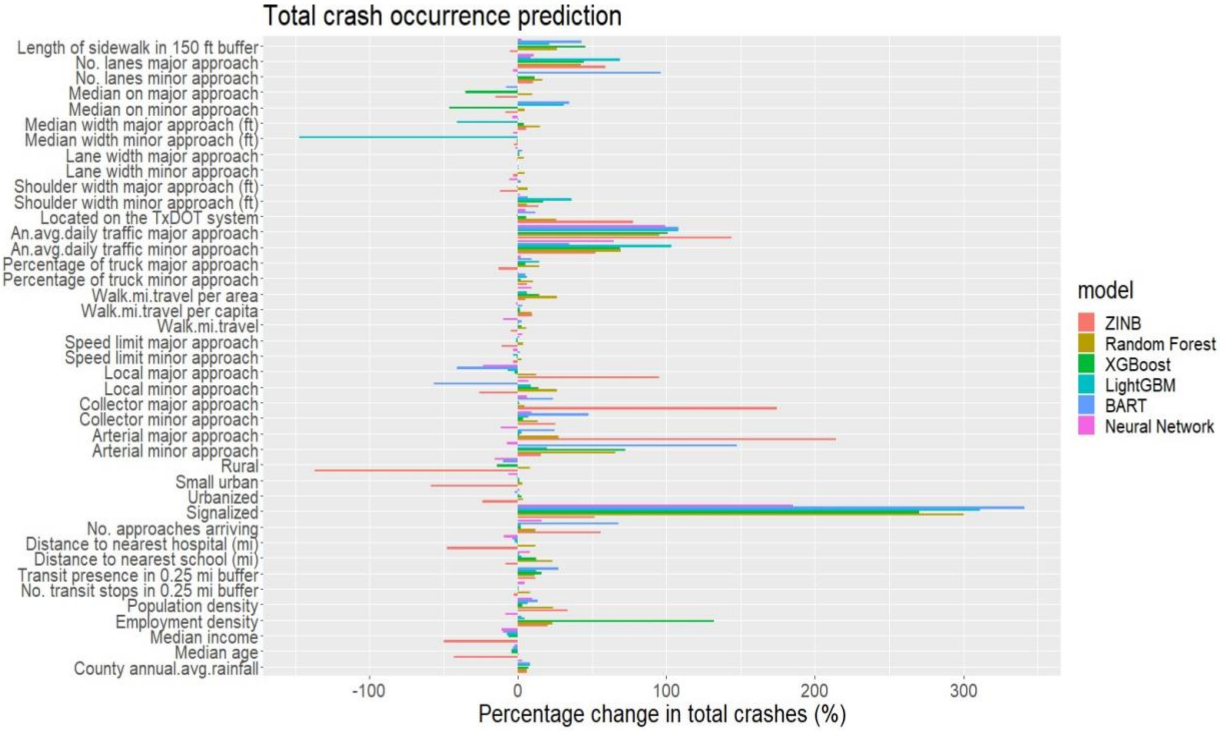
Figure 9: Feature importance for the Random Forest models for signalized and unsignalized intersections with Sum(AADTs) ≥ 500

As explained in Section 5.2, the data was subsetted to focus on features other than signalized intersections and AADTs. Analyzing only the high-volume intersections where the sum of AADTs exceeded 500, Figures 9a and 9b illustrate the feature importance for signalized and unsignalized intersections respectively. They found that total walk-miles traveled, distance to the nearest school, distance to the nearest hospital, population density, and employment density were the most important features to the model predictions, although the five features were ranked differently between signalized and unsignalized intersections.

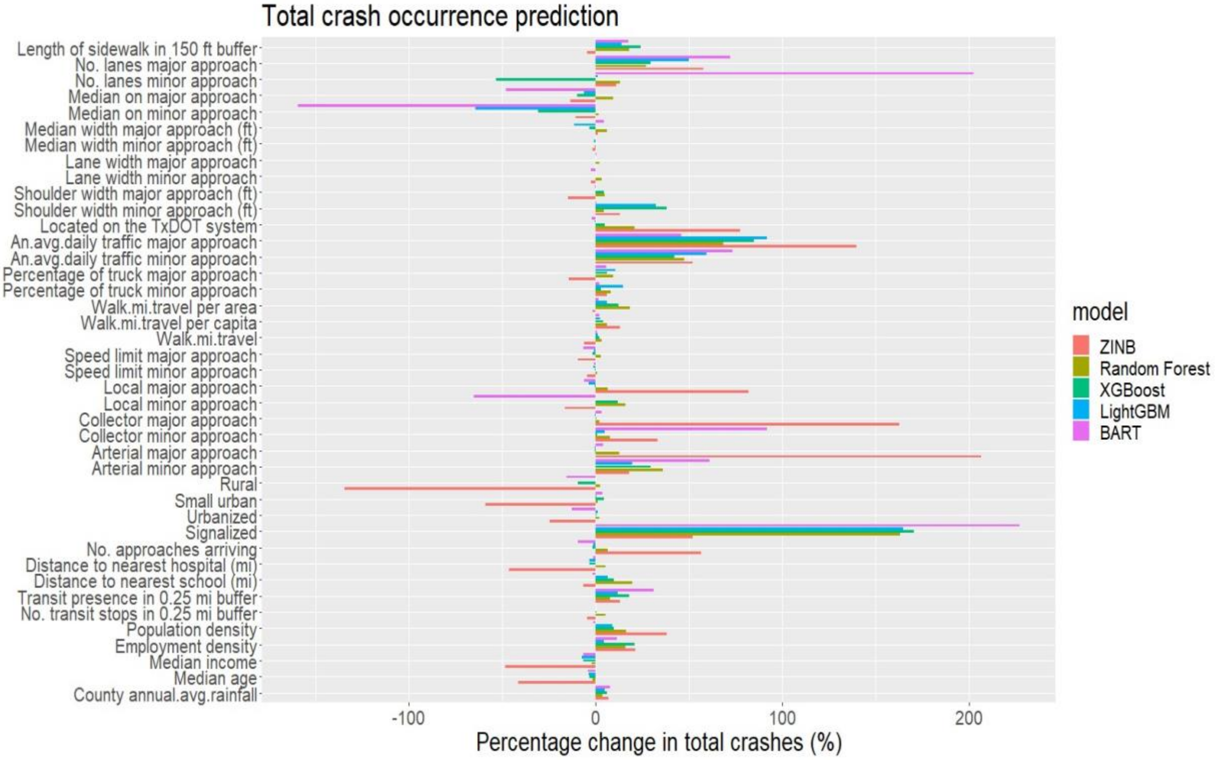
Sensitivity analysis of total crash count

While regression and ML models excel at capturing relationships between features and outcome variables, the results may not be easy to interpret. Specifically, one may find it difficult to quantify the substantive effects of each feature. Following Zhao et al. (2021), this paper employed a sensitivity analysis that captured the contribution each variable had on the model's predictions. Let X be the set of features. The procedures of evaluating the sensitivity of variable X_i were as follows: (1) train the model on X and compute y as the prediction vector; (2) generate a new set X* where a transformation is performed on variable X_i; (3) generate prediction on X* and define $y*$ as the prediction vector, and (4) compute the percentage change in the prediction mean, denoted as $\frac{\overline{y*}-\bar{y}}{\bar{y}} * 100\%$ (Zhao et al., 2021). Following Li and Kockelman (2021), the transformation was as follows: (1) increase one standard deviation for continuous features; (2) binary change (0 to 1) for dichotomous features. Essentially, one standard deviation or binary change was implemented on each observation (Li and Kockelman, 2021). The new prediction was computed using the modified variables, and the difference between the mean of new predictions and original predictions represented the contribution of each feature (Zhao et al., 2021).

Figure 10 illustrates the sensitivities of each X_i by computing percentage changes in the outcome after performing transformation on each X_i. Specifically, it computed the percentage changes in the outcome variable, averaged across all 707,161 intersections, after one standard deviation change or binary change in each X_i. The higher the percentage changes, the higher contribution of a given variable on the model's predictions.



(a) Total crash occurrence prediction (imbalanced data)



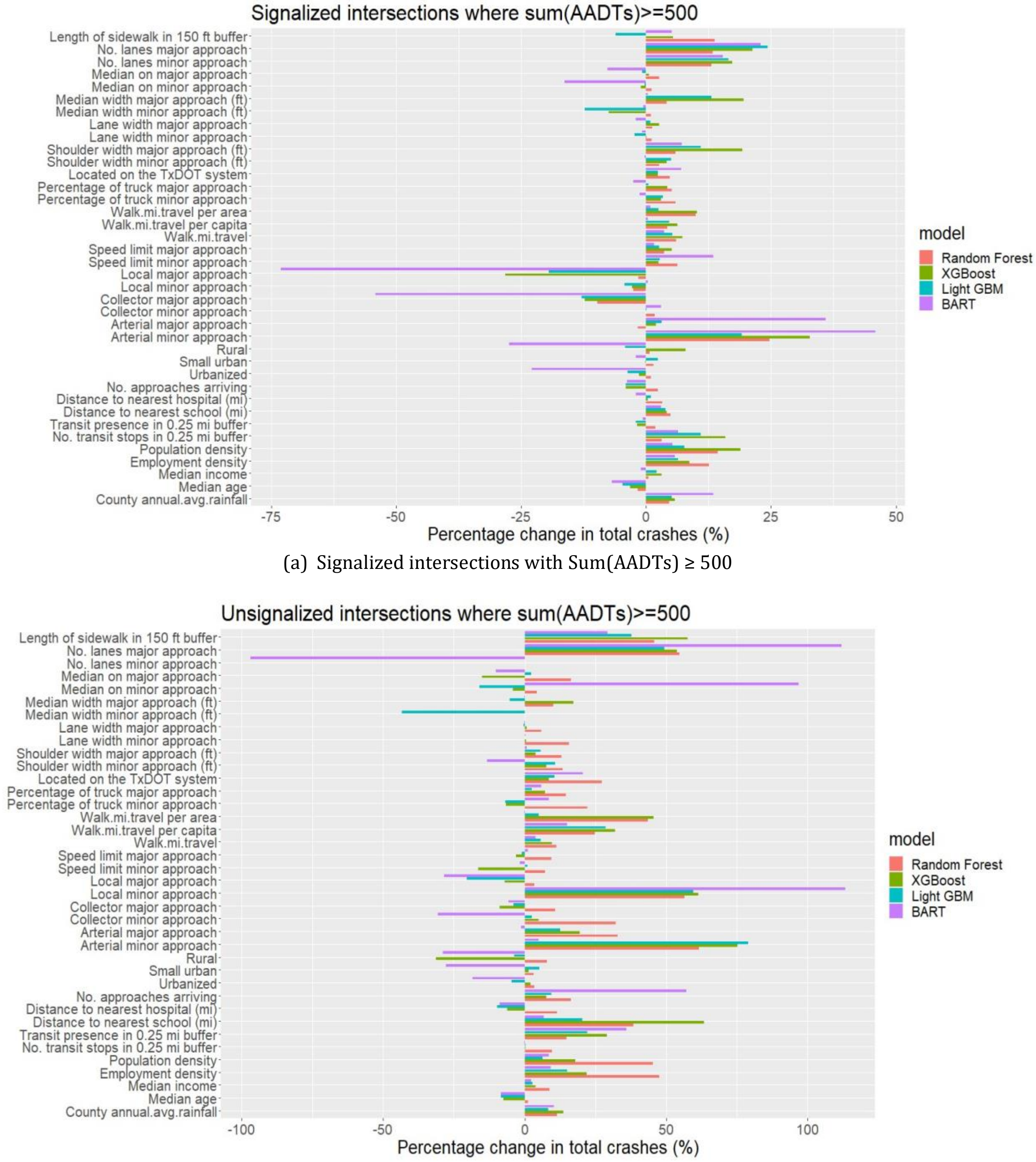
(b) Total crash occurrence prediction (balanced data)

Figure 10: Sensitivity analysis for crash occurrence predictions

Figure 10 illustrates the sensitivities for the ZINB models and ML models for imbalanced and balanced data respectively. One can see that the effects of several variables show different directions across different models. Considering the more important features, number of lanes at the minor approach, speed limits at the major and minor approaches, and distance to the nearest hospital show different directions in Figures 10a and 10b. This was possibly due to the fact that different ML models interpreted the significance of the features differently (Zhao et al., 2021). Therefore, it is vital that one chooses the best performing model when one evaluates the metrics and examines feature importance with the optimum model. Comparing the performances of the ML algorithms, this paper placed more weights on the results of the random forest models when drawing inferences.

For the ZINB model, the land use characteristics (local, collector, and arterial approaches) increased the outcome by a large percentage. In particular, arterial major approach had the most significant impact on total number of crash occurrence. A binary change on arterial major approach could lead to 214% increase in crash occurrences per intersection. The percentage changes for land use characteristics were smaller in the ML models. For example, a binary change on arterial major approach resulted in less than 30% increase in crash counts for all ML models. For ZINB models, intersections in rural areas, small urban, and urbanized areas decreased crash counts by 137%, 59%, and 24% respectively compared to large urban areas. In the ML models, the percentage changes pointed to different directions for different urban-rural classifications. For XGBoost and BART models, rural areas decreased crash counts while for random forest model, rural areas increased crash occurrences. Small urban and urbanized areas increased crash occurrences for most ML models. These contrasted with the ZINB results.

Regarding road design variables, the number of lanes and AADTs at the major and minor approaches had significant impact on crash occurrence in the ZINB models. In the ZINB model, one standard deviation increase in the number of lanes at major approach led to 59% increase in crash counts. In the random forest ML model, the percentage change decreased to 42%. In the ZINB model, one standard deviation increase in AADTs at major and minor approaches contributed to about 144% and 52% increase in crash occurrences respectively. In the ML models, AADTs at major and minor approaches also showed increases in crash counts. The percentage increases ranged from 35% to 108% on imbalanced data, and from 42% to 92% on balanced data. Signalized intersections also contributed to a large increase in the outcome for both ZINB and ML models. A binary change on signalized intersections contributed to 300% and 163% increases in crash counts in the random forest model on imbalanced and balanced data respectively. As to census-tract level attributes, one standard deviation in population density, employment density, and precipitation increased crash counts by 33%, 20%, and 6% respectively, while one standard deviation in median income and median age reduced crash occurrence by 50% and 43% respectively. ML models showed the same directions in terms of percentage changes.



(a) Signalized intersections with Sum(AADTs) ≥ 500

(b) Unsignalized intersections with Sum(AADTs) ≥ 500

Figure 11: Sensitivity analysis for crash occurrence predictions

This paper compared the sensitivity analysis results of the ML models for signalized and unsignalized intersections in Figures 11a and 11b respectively. Comparing the two figures, most features showed similar directions in percentage changes. Focusing on the most important features, it found that the percentage changes were mostly consistent across the ML models. Consider the top five features. When one standard deviation was increased to total walk-miles traveled, distance to the nearest school, population density, and employment density one at a time, this paper found positive percentage changes in crash occurrences for all models. The only exception was distance to the nearest hospital. Among signalized intersections, all models showed positive percentage changes except BART, whereas among unsignalized intersections, all models demonstrated negative percentage changes except random forest. It is noteworthy that random forest yielded relatively large percentage changes for the top five features.

Sensitivity analysis of crash count by severities

The sensitivity results can be found in Figures 12, 13, 14, 15, 16.

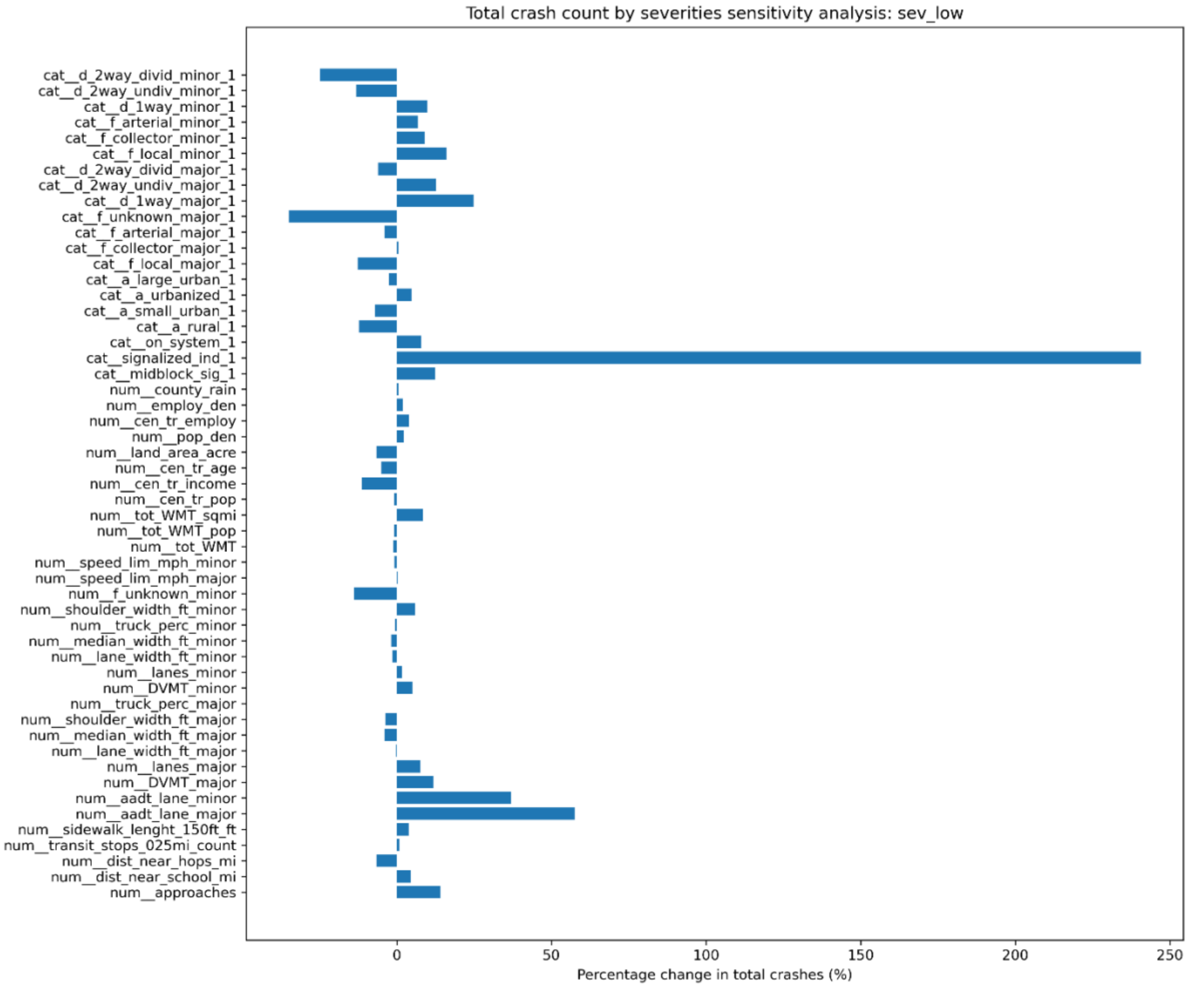


Figure 12: Sensitivity analysis of severity low

The important features were similar to the previous analyses. An interesting finding was that across different severities, ”killed” was the most sensitive to any feature changes.

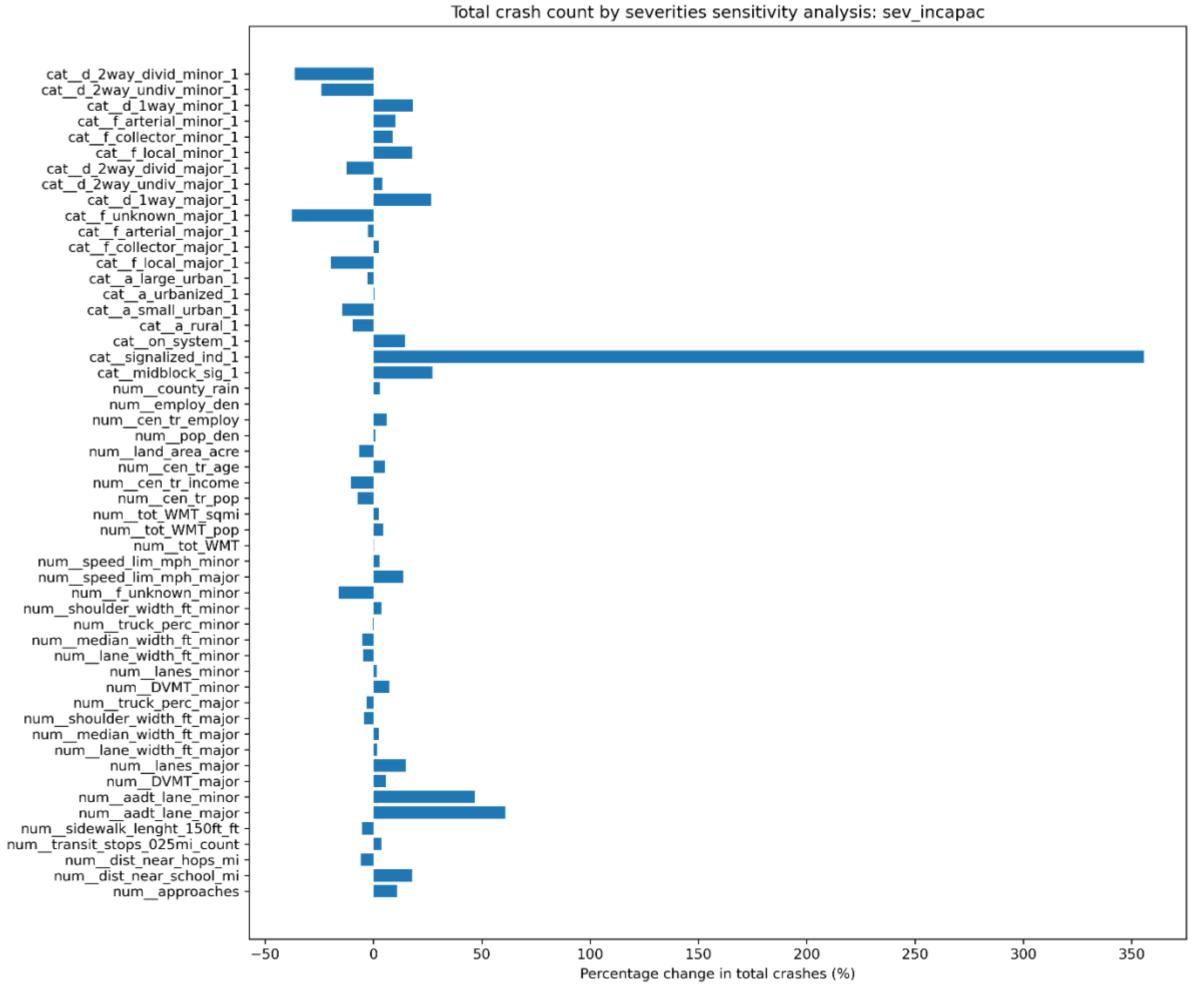


Figure 13: Sensitivity analysis of severity incapacitated

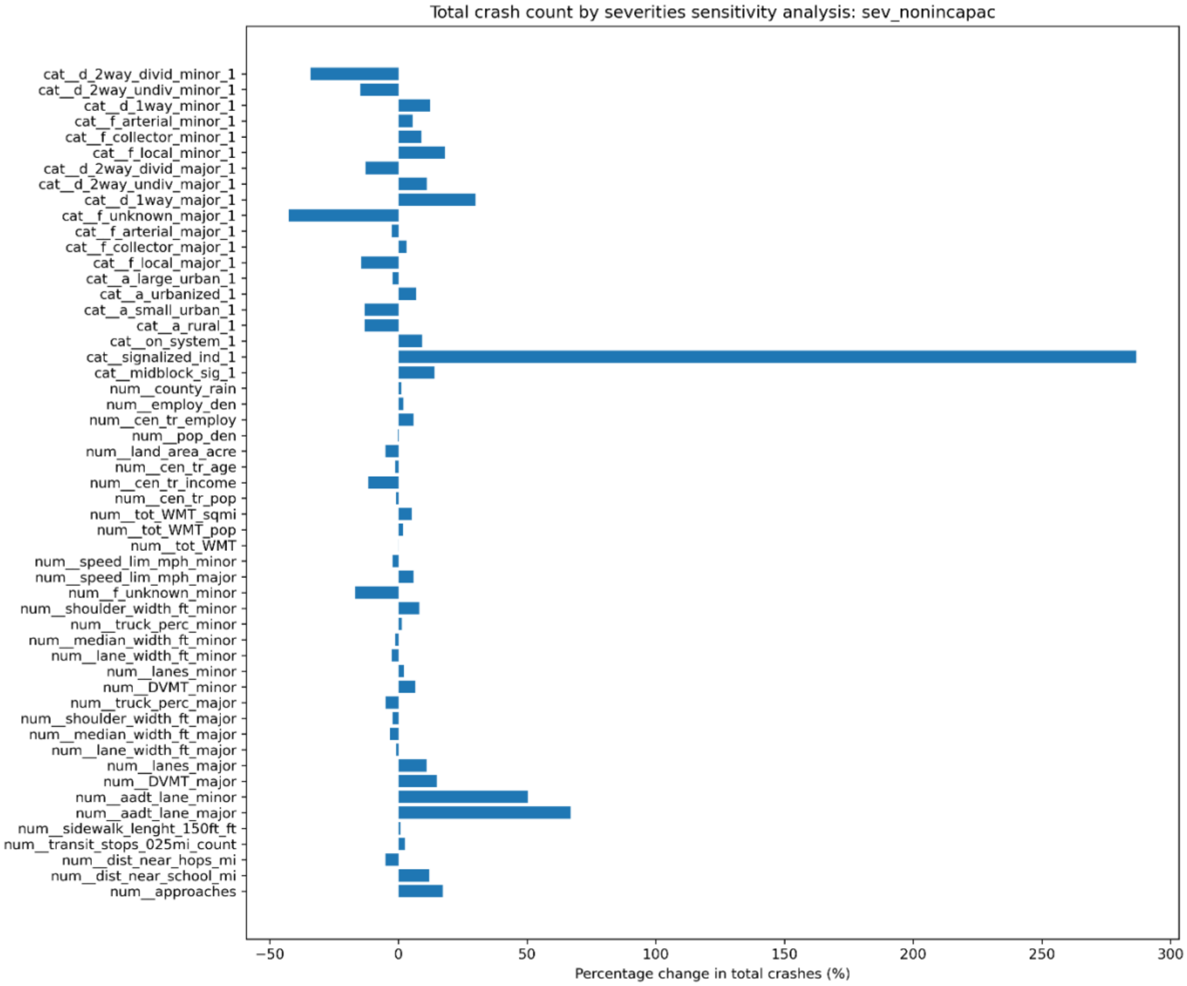


Figure 14: Sensitivity analysis of severity nonincapacitated

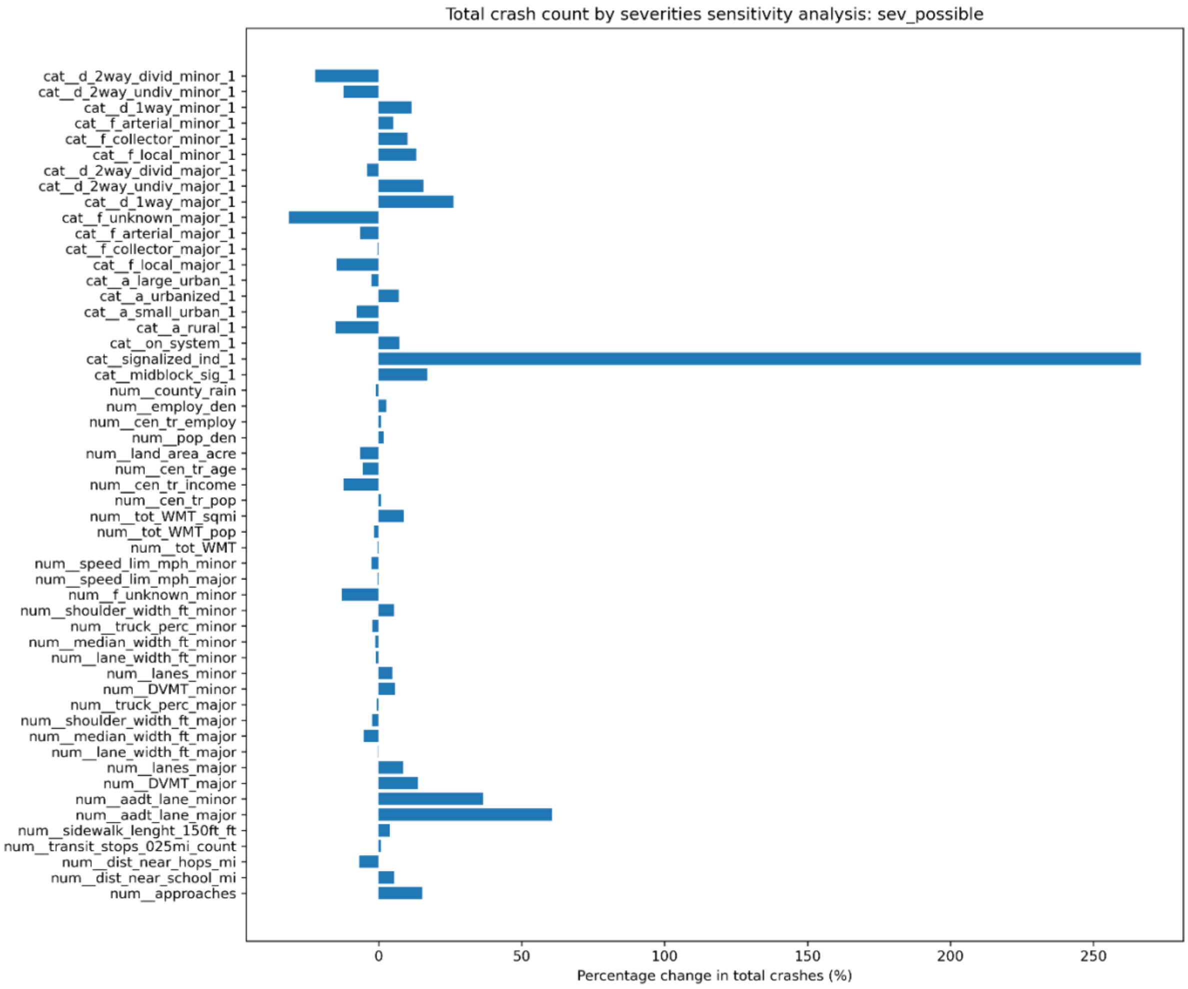


Figure 15: Sensitivity analysis of severity possible

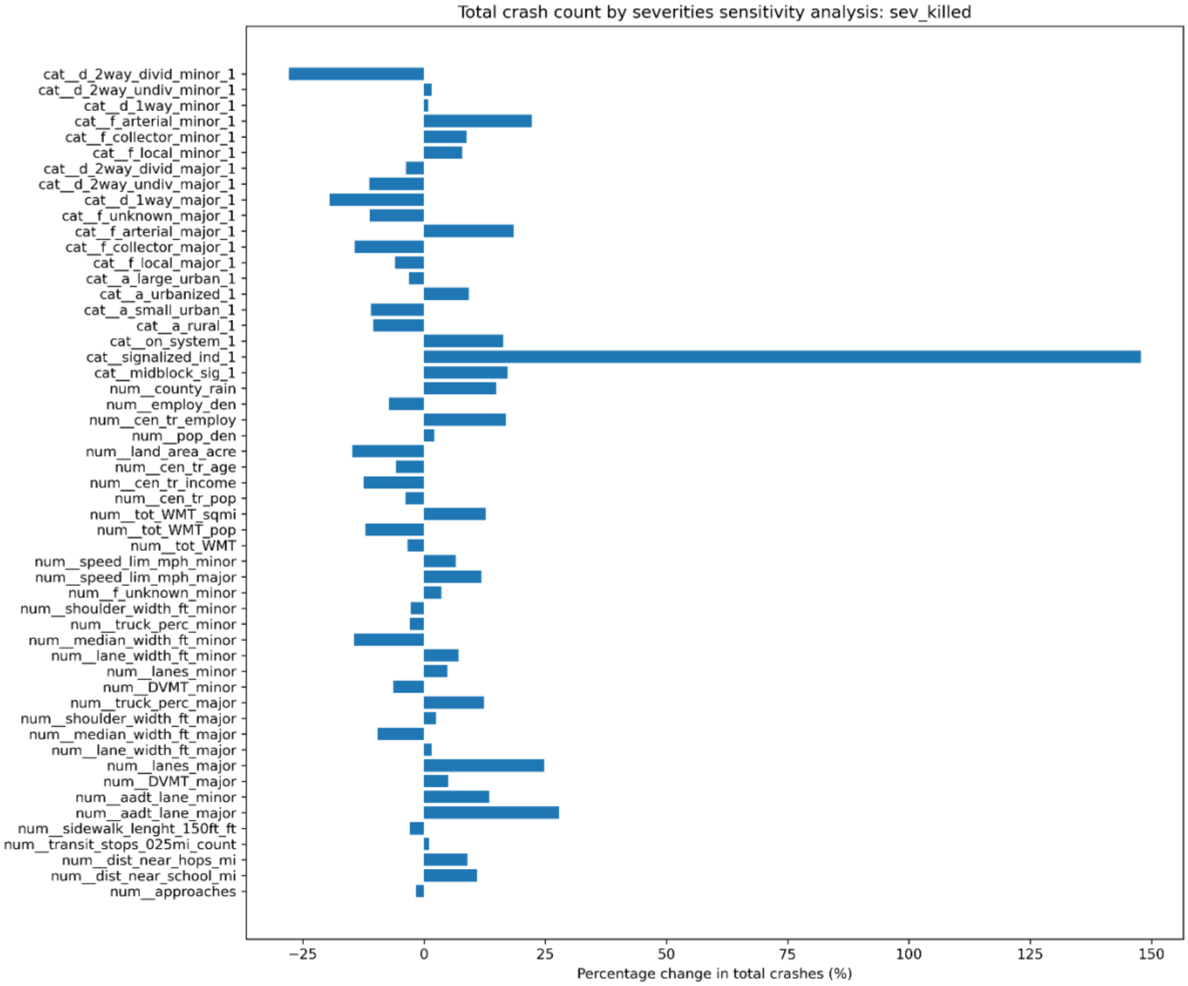


Figure 16: Sensitivity analysis of severity killed

References

American Community Survey (2020). American Community Survey Data via API. Available at: https://www.census.gov/programs-surveys/acs/data/data-via-api.html See codes from the "get_census_tract.R" replication file.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics,* 4(1):266-298.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* pp. 1724-1734.

He, H., and Garcia, E.A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263-1284.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Proceedings of the 31st Conference on Neural Information Processing Systems*.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference for Learning Representations, San Diego, 2015*.

Li, W. and Kockelman, K. M. (2021). How does machine learning compare to conventional econometrics for transport data sets? a test of ml versus mle. *Growth and Change*, pages 1–35.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Lunardon, N., Menardi, G., and Torelli, N. (2021). Random Over-Sampling Examples. Retrieved from https://cran.r-project.org/web/packages/ROSE/ROSE.pdf

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia,* pp. 1310–1318.

C.R.I.S., Texas Department of Transportation (2020). C.R.I.S. Query. Retrieved from https://cris.dot.state.tx.us/public/Query/app/home

Texas Water Development Board. (2014). GIS data: Texas precipitation. Retrieved from https://www.twdb.texas.gov/mapping/gisdata.asp

Negative Binomial Regression: Stata Annotated Output. UCLA: Statistical Consulting Group. Retrieved from https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-how-do-i-cite-web-pages-and-programs-from-the-ucla-statistical-consulting-group/

Zhao, B., Zuniga-Garcia, N., Xing, L., and Kockelman, K. M. (2021). Predicting pedestrian crash occurrence and injury severity in texas using tree-based machine learning models. *Under Review for Presentation at the 101st Annual Meeting of the Transportation Research Board and for publication in Transportation Research Record*.