International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 12, No. 1, April 2023, pp. 46~53
ISSN: 2252-8776, DOI: 10.11591/ijict.v12i1.pp46-53

Automated machine learning for analysis and prediction of vehicle crashes

Abhishek Saxena[1], Stefan A. Robila[2]
[1]Department of Data Science, Montclair State University, New Jersey, United States of America
[2]Department of Computer Science, Montclair State University, New Jersey, United States of America

Article Info

Article history:

Received Jun 2, 2022
Revised Jul 23, 2022
Accepted Aug 15, 2022

Keywords:

Machine learning
Open data
Support vector machines
Vehicular crash data
Visualization

ABSTRACT

This work discusses the study and development of a graphical interface and implementation of a machine learning model for vehicle traffic injury and fatality prediction for a specified date range and for a certain zip (US postal) code based on the New York City's (NYC) vehicle crash data set. While previous studies focused on accident causes, little insight has been offered into how such data may be utilized to forecast future incidents, Studies that have historically concentrated on certain road segment types, such as highways and other streets, and a specific geographic region, this study offers a citywide review of collisions. Using cutting-edge database and networking technology, a user-friendly interface was created to display vehicle crash series. Following this, a support vector machine learning model was built to evaluate the likelihood of an accident and the consequent injuries and deaths at the zip code level for all of NYC and to better mitigate such events. Using the visualization and prediction approach, the findings show that it is efficient and accurate. Aside from transportation experts and government policymakers, the machine learning approach deliver useful insights to the insurance business since it quantifies collision risk data collected at specific places.

Corresponding Author:

Stefan A. Robila
Department of Computer Science, Montclair State University
Normal Ave, Montclair, New Jersey, United States of America
Email: robilas@montclair.edu

1. INTRODUCTION

Despite over a century of continuous technological progress and countless safety innovations, vehicular road crashes continue to constitute a significant proportion of deaths and injuries globally, while at the same time generating losses estimated at close to two trillion US dollars/year [1]. While novel technologies such as driver assist, self-driving cars, traffic flow management, dedicated and traffic lanes, or enforcement campaigns may change the current trends, large scale analysis of data can lead to complementary insights [2]. Yet processing large data sets requires different approaches that combine geospatial-focused visualization with data mining and clustering. Meaningful results also require access to open, reliable, and renewable data streams.

Several large data sets are available. In US, the US fatal accidents dataset fatality analysis reporting system (FARS) is updated yearly by the national highway traffic safety administration as way to assist policy makers as well as provide consistent information to insurance companies [3]. Users can interact with the data by submitting queries on fatal accident frequency or download it for further analysis. Such data has been extensively used to test both ML models as well as derive accident trends. Li *et al.* [4] environmental factors such as road surface, weather, and light conditions, or human factors (such as alcohol consumption) were evaluated as indicators for increased fatality rate. Das *et al.* [5] focused the analysis on scooters and noted that within a recent five-year period fatal accidents for these vehicles increased 76% (compared to an overall 2% decrease in all accidents). Using cluster correspondence analysis, the authors identified compounding factors for the crashes and proposed their findings as input for rider and driver training and certification. Yuan *et al.* [6] used latent class clustering to identify three groups of truck drivers that correlated to different crash outcomes. Despite providing data that goes back close to 50 years, FARS is limited to fatal accidents; more data (including non-fatal accidents) would be needed to identify geographical risks [4]. Beyond nationwide statistics, many US states also provide their own data sets. As an example, the Maryland statewide vehicle crashes data set provides over 770 thousand records of accidents that occurred in the state since 2015 [7]. Xiong *et al.* [8] employed the data as input to a vulnerability exposure risk dashboard that is publicly accessible. Also using the same dataset, the Maryland State highway administration's chart program deployed fervor web-based visual analytics [6] tool that includes interactive maps, histograms, two-dimensional plots, and parallel coordinates plots that allow users to meaningfully visualize the data and efficiently find meaningful relationships among it features. Similar data sets can be identified in other countries. As example, Yuan *et al.* [6] describes the development of a data analytics platform for accident analysis based on the United Kingdom's road accidents and safety statistics data [9]. The system, formed of three parts uses Google Map application programming interface (API) for visualization and then allows for exploratory analysis of factors that lead to accidents, as well as predicts future accident trends. Data sets are not always publicly available or easily accessible, yet their use is revealed through scientific publications. Using 18,175 Lithuanian accident records extracted from government data, NHTSA [3] analyzed whether environmental factors, vehicle condition, and human factors can serve as predictors for the accident severity and showed that apart from factors that are traditionally associated with increased accident severity (such as time of data or weather conditions), other aspects are also highly correlated (males were twice as likely to be the authors of a fatal accident than females). Data are not always generated using governmental agency channels. Using commercially available traffic broadcast APIs (MapQuest traffic and Microsoft Bing traffic), a dataset of over 1 million records was built and then used to predict accident occurrence [10].

How to analyze traffic data is also of interest. Singh provides an analysis of India's accidents and compares it to other countries' statistics [11]. The findings show significant differences among geographical regions, age groups, time of day and weather and expose the complexity of traffic safety management for large countries. A cross country analysis (including accident reporting mechanism evaluation) is also described in [12]. Particularly concerning is one of the conclusions drawn by the authors that many current approaches still rely on out of data technology. The article also notes the need for renewed effort in the use of machine learning approaches for data analysis. In this context, an overview of predictive modelling, i.e., how such data can be used to predict accident risk can be found in [13]. Given the strong geographical connection, geospatial techniques are also heavily used, as are a variety of visualization approaches [14]. As examples, Richard and Ray [15] built a data analysis and processing pipeline using a big data spatial framework based on random forest classification models to predict if an accident had casualties. Volpi *et al.* [16] describes the development of Roma crash map, a visualization tool for vehicle accident data focused on open data sets. More importantly, the paper provides an evidence-based approach to the development of the tool that includes use of visualization and processing techniques that have been shown to be easy to use by the broader public. The iterative development of safe road maps (SRM) is described in [17]. SRM was designed to focus on transportation safety and provide a means of visual communication, with the potential to raise user awareness and impact driving behavior.

The objective of this project was to develop an interface and implement a machine learning model for vehicle traffic injuries and fatalities prediction for a given date range and for a specific postal (zip) code along with data analysis and visualization. As part of this project, an interface has been developed which will enable to gain insights into the data through Python visualization packages and implements a machine learning model to predict vehicle crash injuries and fatalities. This project involves the usage of machine learning framework, Python visualization packages and includes geospatial techniques as well. It includes additional data sets to further optimize the machine learning model for crash predictions.

2. METHOD

This work included the usage of several big data techniques such as data extraction, data processing, visualization, and vehicular crash prediction. Python libraries such as Pandas [18], Matplotlib [19], and Folium [20] were employed. The data set used for this project is from New York City's (NYC) traffic [20]. Additionally, Streamlit [20] was used to develop a framework which can be used as an app to visualize this data and ready to be deployed on a supporting cloud platform for public use. machine learning regression models were used along with other data transformation models such as encoder/decoder to perform prediction of the number of persons injured and number of persons killed on a specified date/date range and at a specific zip code. Figure 1 provides an overview of the platform. Each of the models used is described in the sections. The source code is available in GitHub [21].

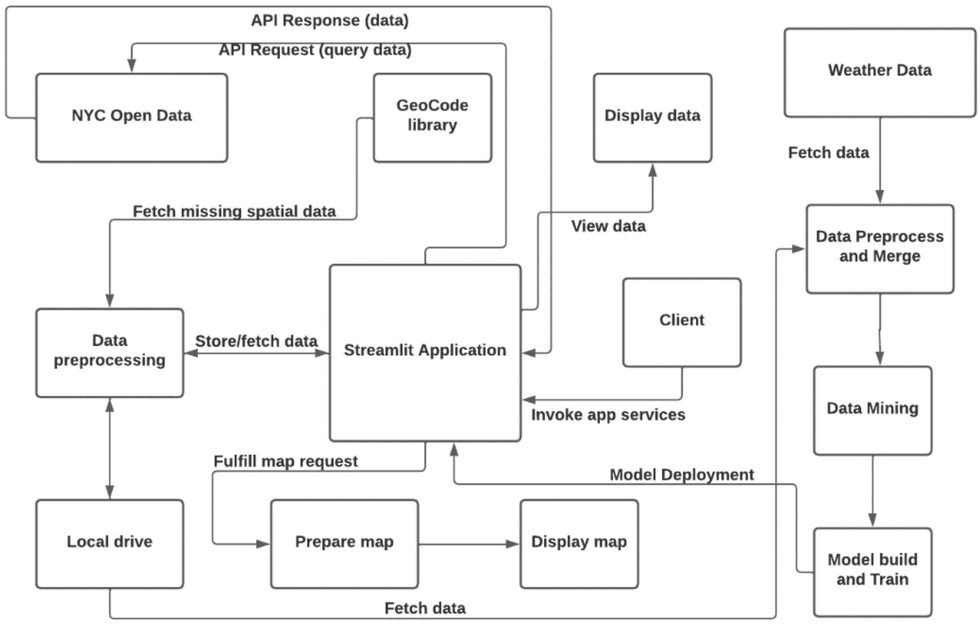

Figure 1. Interface design

2.1. Data preprocessing for model build

Before using a model, the weather data set is preprocessed and merged with the crash events data set discussed in the previous sections. In this section we will discuss some of the preprocessing steps that were taken specifically for priming the data set for model use and the corresponding data schema: i) data frame transformations–the weather data from the three weather stations (New York Central Park, John F. Kennedy International Airport (JFK), and LaGuardia Airports) is read into a Pandas [19] data frame. Before this data is merged with the crash events data, several data transformation steps are taken such as convert the data type from string to date time for the weather date field, convert temperature, precipitation from string to float. Later the Python pandas 'pivot' functionality is used to restructure the data and make it ready to be merged with the crash event data set discussed in the sections above, ii) rename and drop columns–the data set columns were renamed to give a more meaning full names for easy readability and redundant columns were dropped off from the weather data set, and iii) data filter–the weather data and the crash events data is filtered to train and test the machine learning models based on the data from year 2017 onwards.

2.2. Data mining

In brief, data mining is a computational field comprising of processes to find patterns, correlations and anomalies in a large data set and can be very useful in gaining meaningful insights or prepare the data for usage by the machine learning models. For this project, to research some interesting patterns in the combination of the crash events data and the weather data, association rule mining is performed using the two well-known algorithms–Apriori algorithm and frequent pattern (FP) growth. The Apriori algorithm is a pattern set mining and association rule mining algorithm mostly used in the area of market basket analysis. Similarly, FP growth is another association rule mining algorithm that represents a data set in form of a tree to derive the frequent item sets. There are some differences between these two algorithms such as Apriori uses bread-first search approach while FP growth uses depth-first search approach. Another difference is that Apriori generates the FP by making the item sets using pairing such as single item set, double itemset, triple itemset and FP growth generates FP tree for making the FP. Both algorithms are known to perform well over large data set and are popular in the area of pattern mining and market basket analysis. The PyPi Python libraries of the Apriori and FP growth were used to find interesting patterns within the data set. Association rule mining usually finds its application in market basket analysis, but it has been applied in this project to see if there's any interesting pattern that can be unraveled. A summary of the results is presented in Table 1.

Table 1. Data mining the performance comparison of algorithms

Algorithm	Event type	Pattern	Minimum support (%)	Minimum confidence (%)
Apriori	Injuries	{BROOKLYN} -> {1.0}, {QUEENS} -> {1.0}	20	60
Apriori	Fatalities	{BROOKLYN} -> {1.0}, {Night} -> {1.0}, {QUEENS} -> {1.0}	20	90
FP growth	Injuries	[{'QUEENS'},{'1.0'},0.7128786398280242], [{'BROOKLYN'},{'1.0'}, 0.7048750935462029]	20	90
FP growth	Fatalities	[{'Night'}, {'1.0'}, 0.9776785714285714], [{'QUEENS'}, {'1.0'}, 0.953416149068323], [{'BROOKLYN'}, {'1.0'}, 0.9792899408284024]	20	95

2.3. Gradient boosting regression model build

The gradient boosting algorithm works by sequentially adding predictors to an ensemble, each one correcting its predecessor. For this project, we use gradient boosted regression trees from Python scikit learn [22] library. The gradient boosting regression model is used to perform the injury and fatalities predictions. The Scikit [22] learn's 'train-test-split' module along with 10-fold cross validation is used to split the training and test data set and it's configured to set aside 30% of the data set as a test set. Tables 2 and 3 list the input data schema for this model. The fields were selected to best represent the events in terms of location, date and time and road and environment conditions. The field selection was also informed by the previous research described in the introduction.

The regression model was trained on the data set with the input schema mentioned in the Tables 2 and 3. There are two separate models created for the prediction of crash injury and fatalities at the zip code level. The hyperparameters selected are listed in Table 4.

Table 2. Crash injury model input schema

Field name	Description	Type
Month	This holds the month data such as 1 for January and 12 for December	Integer
Day_encoded	This is the encoded data for day. For example, Sunday could be encoded as 0	Integer
Timeofday_encoded	This is the encoded data for time of the day. For example, Morning could be encoded as 0	Integer
Borough_encoded	This is the encoded data for borough. for example, Brooklyn could be encoded as 0	Integer
Postalcode	This holds the zip code	Integer
PRCP	This holds the rainfall amount predicted for a day	Float
SNOW	This holds the snow amount predicted for a day	Float
TMAX	This holds the maximum temperature predicted for a day	Float
TMIN	This holds the minimum temperature predicted for a day	Float
Number of persons injuries	This holds the count of injuries reported	Integer

Table 3. Crash fatalities model input schema

Field name	Description	Type
Number of persons killed	This holds the count of fatalities reported	Integer
Month	This holds the month data such as 1 for January and 12 for December	Integer
Day_encoded	This is the encoded data for day. For example, Sunday could be encoded as 0	Integer
TmeOfDay_encoded	This is the encoded data for time of the day. For example, Morning could be encoded as 0	Integer
Borough_encoded	This is the encoded data for borough. for example, Brooklyn could be encoded as 0	Integer
PostalCode	This holds the zip code	Integer
PRCP	This holds the rainfall amount predicted for a day	Float
SNOW	This holds the snow amount predicted for a day	Float
TMAX	This holds the maximum temperature predicted for a day	Float
TMIN	This holds the minimum temperature predicted for a day	Float

Table 4. Model parameters and values

Model	Loss function	Learning rate	N_Estimators	RMSE
Crash_predictor	Mean squared_error	0.1	200	1.00
Injuries_predictor	Mean squared_error	0.1	200	1.005
Fatalities_predictor	Mean squared_error	0.05	200	0.788

2.4. Support vector machine model build

Support vector machines (SVM), are powerful and versatile machine learning algorithms, capable of performing linear and non-linear regression tasks. We use the SVM regression class from Python Scikit learn library for implementation in this project. The SVM models are built and trained for prediction based on the data schema discussed in Tables 2 and 3. 10-fold cross validation is performed on the training data and then the model fitment is done. The performance of each of the models is given in the Table 5. The performance of the SVM models show that since the root mean squared error (RMSE) score for the gradient boosting regression model is low, the model is selected for deployment to the interface.

Table 5. Model parameters and values

Model	Loss function	Learning rate	N_Estimators	RMSE
Crash_predictor	Mean squared_error	0.1	200	4.00
Injuries_predictor	Mean squared_error	0.1	200	2.005
Fatalities_predictor	Mean squared_error	0.05	200	1.788

3. RESULTS AND DISCUSSION

Figure 2 illustrates the overall visual user interface. The data set may be queried by users starting in 2017 for any time period. For respective intervals and regions in the city, maps are produced to display the accident frequency. The data can be utilized in several forms (i.e., persons injured, killed, pedestrians, and cyclists). Users can conduct more research from this first perspective of the location, vehicle, and time of day that are the most common accident causes [20] goes into further depth on the exploratory work that was done before the design and implementation of machine learning models. According to the data analysis and visualization presented in this study, the findings are explored. Based on the information from January 1 through December 11, 2021, they were created. According to the date range taken into account, Figure 2 reveals that the Brooklyn zip code 11,207 had the highest number of injuries.

Further insight can be extracted from the interface itself. Figures 2-4 are snapshots of the user interface. In Figure 3, the data distribution by borough shows that Brooklyn has the most recorded crash events and queens for the number of persons injured in the crash. Similarly, the data distribution for the top 5 crash reasons other than unspecified are driver inattention/distraction, failure to yield right-of-way, following too closely. Figure 4 shows that most injuries happened in early afternoon (12.00 PM to 3.00 PM) followed by evening (5.00 PM to 7.59 PM) and night (8.00 PM to 12.00 AM). In addition to this, the vehicle type sedan, station wagon/suv are top two types of vehicles involved in the injuries.

Finally, the interface also allows generation and prediction for possible injuries and fatalities (see Figure 5). For the date range between November 30, 2021 and December 3, 2021 with the time of day as early afternoon and zip code 11,217, the model predicts that there is a possibility of 2 crash injuries and 0 crash fatalities. This is with RMSE of 1.007 and 0.788 for crash injuries and fatalities respectively.

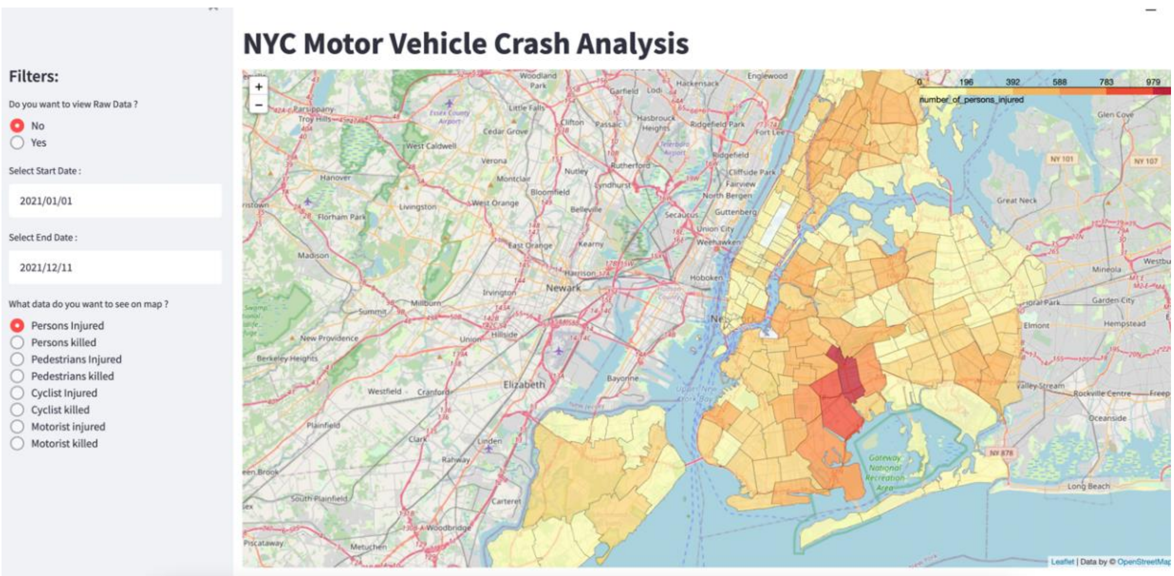

Figure 2. Overall interface view

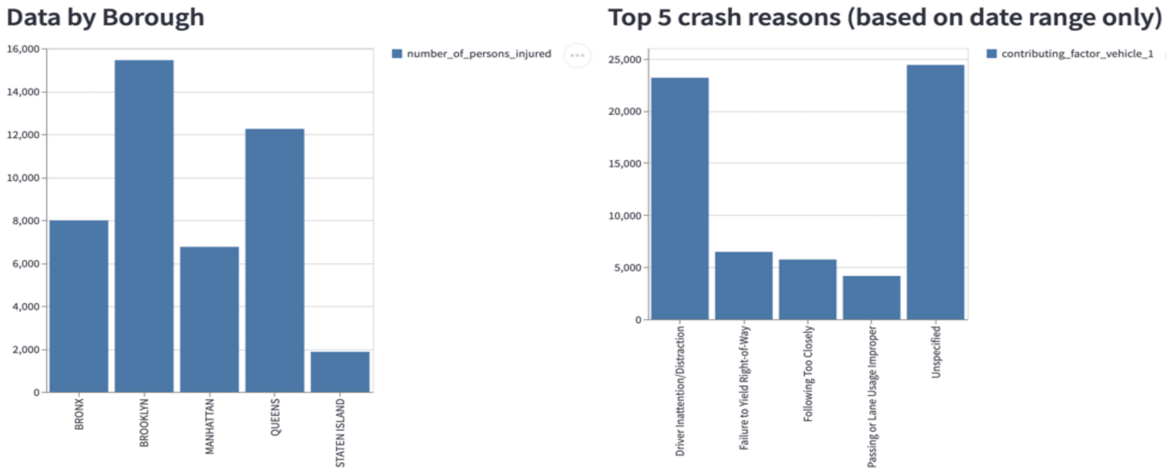

Figure 3. Results of injuries for borough and crash reasons (Jan 1, 2021 to Dec 11, 2021)

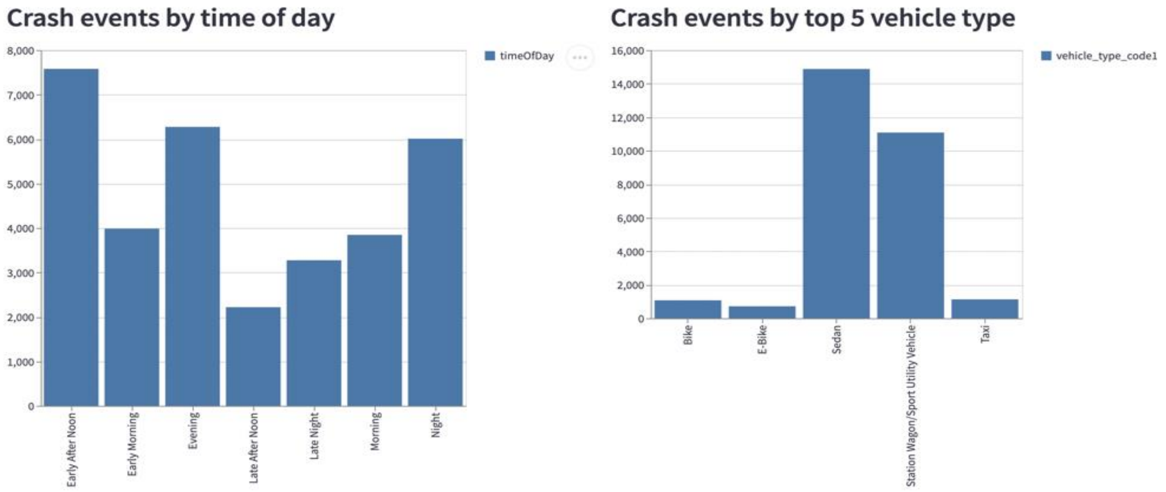

Figure 4. Results of injuries for time of day and vehicle type (Jan 1, 2021 to Dec 11, 2021)

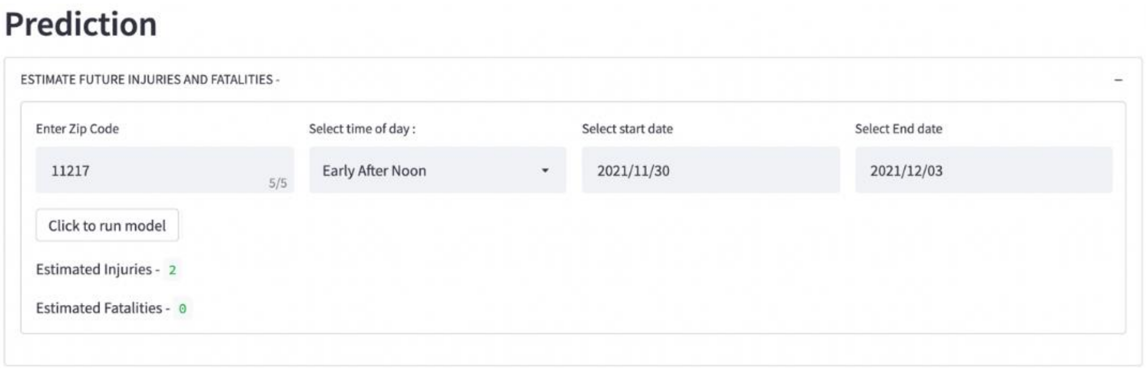

Figure 5. Results for model prediction of injuries

4. CONCLUSION

This paper described the research and development of a visualization interface and implementation of a machine learning model for vehicle traffic injuries and fatalities prediction for a given date range and for a specific zip (US postal) code based on the NYC vehicle crash data set. Using state of the art libraries, and integrating recent developments in machine learning, the interface is positioned as a tool for both practitioners and the public. While currently running on a local server, the tool can be deployed on a publicly facing server, thus providing broad access. The choice of popular programming languages and open source tools ensure that the data will continue being updated and the interface can be also expanded. The integration of weather data provides additional insight into accident characteristics and facilitates further predictions.

Despite its richness, the data used has some intrinsic limitations. The records are based on preliminary police reports and are not updated with further information. An example of such an amendment is that a person might be marked as "injured" in the report, but the severity of the injuries is unknown. This means that if there are any fatalities due to the severity of the injuries then that count is not reflected into the "persons killed" data. Consequently, the "persons killed" count can be considered as an undercount. Furthermore, the prediction models do not take in consideration road conditions (such as speed limit or traffic). As part of future work, a convolutional neural network-based model can be created to analyze the traffic conditions in real time and predict the likelihood of a crash and the resultant injuries or fatalities is proposed. Additionally, the prediction model can be reconfigured to return the zip codes where the count of injuries and fatalities is predicted to be one or more for a specific date range. This could include a geospatial representation as well. Finally, a broader integration with additional data sets may result a state or country level model.

REFERENCES

[1] WHO, "Global status report on road safety 2018," 2018.

[2] D. Mohan, "Traffic safety: rights and obligations," *Accident Analysis & Prevention*, vol. 128, pp. 159–163, Jul. 2019, doi: 10.1016/j.aap.2019.04.010.

[3] NHTSA, "Fatality analysis reporting system (FARS)," *National Highway Traffic Safety Administration*, 2021. [Online]. Available: https://www.nhtsa.gov/research-data/fatality-analysis-reporting-system-fars (accessed Apr. 29, 2022).

[4] L. Li, S. Shrestha, and G. Hu, "Analysis of road traffic fatal accidents using data mining techniques," in *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*, Jun. 2017, pp. 363–370, doi: 10.1109/SERA.2017.7965753.

[5] S. Das, M. M. Hossain, M. A. Rahman, X. Kong, X. Sun, and G. M. Al Mamun, "Understanding patterns of moped and seated motor scooter (50 cc or less) involved fatal crashes using cluster correspondence analysis," *Transportmetrica A: Transport Science*, pp. 1–21, Feb. 2022, doi: 10.1080/23249935.2022.2029613.

[6] Y. Yuan, M. Yang, Y. Guo, S. Rasouli, Z. Gan, and Y. Ren, "Risk factors associated with truck-involved fatal crash severity: Analyzing their impact for different groups of truck drivers," *Journal of Safety Research*, vol. 76, pp. 154–165, Feb. 2021, doi: 10.1016/j.jsr.2020.12.012.

[7] Maryland State Police, "Maryland statewide vehicle crashes," *Open Data Portal*, 2021. [Online]. Available: https://opendata.maryland.gov/Public-Safety/Maryland-Statewide-Vehicle-Crashes/65du-s3qu (accessed May 01, 2022).

[8] C. Xiong, J. Mahmoudi, W. Luo, M. Yang, J. Zheng, and C. Delion, *A data-driven safety dashboard assessing Maryland statewide density exposure of pedestrians, bicycles, and e-scooters*. Washington, DC: Maryland Department of Transportation, 2021.

[9] Department of Transport, "Road accidents and safety statistics," *Government UK*, 2021. [Online]. Available: https://www.gov.uk/government/collections/road-accidents-and-safety-statistics (accessed Apr. 26, 2021).

[10] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu, and R. Ramnath, "Accident risk prediction based on heterogeneous sparse data," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Nov. 2019, pp. 33–42, doi: 10.1145/3347146.3359078.

[11] S. K. Singh, "Road traffic accidents in India: issues and challenges," *Transportation Research Procedia*, vol. 25, pp. 4708–4719, 2017, doi: 10.1016/j.trpro.2017.05.484.

[12] A. A. Mohammed, K. Ambak, A. M. Mosa, and D. Syamsunur, "A review of the traffic accidents and related practices worldwide," *The Open Transportation Journal*, vol. 13, no. 1, pp. 65–83, Jun. 2019, doi: 10.2174/1874447801913010065.

[13] A. Mehdizadeh *et al.*, "A review of data analytic applications in road traffic safety. Part 1: descriptive and predictive modeling," *Sensors*, vol. 20, no. 4, p. 1107, Feb. 2020, doi: 10.3390/s20041107.

[14] G. Andrienko, N. Andrienko, W. Chen, R. Maciejewski, and Y. Zhao, "Visual analytics of mobility and transportation: state of the art and further research directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2232–2249, Aug. 2017, doi: 10.1109/TITS.2017.2683539.

[15] R. Richard and S. Ray, "A tale of two cities: analyzing road accidents with big spatial data," in *2017 IEEE International Conference on Big Data (Big Data)*, Dec. 2017, pp. 3461–3470, doi: 10.1109/BigData.2017.8258334.

[16] V. Volpi, A. Ingrosso, M. Pazzola, A. Opromolla, and C. M. Medaglia, "Roma crash map: an open data visualization tool for the municipalities of Rome," in *Information and Knowledge in Applications and Services*, 2014, pp. 284–295.

[17] M. B. A. Rabbani, M. A. Musarat, W. S. Alaloul, A. Maqsoom, H. Bukhari, and W. Rafiq, "Road traffic accident data analysis and its visualization," *Civil Engineering and Architecture*, vol. 9, no. 5, pp. 1603–1614, Aug. 2021, doi: 10.13189/cea.2021.090530.

[18] W. McKinney, "Pandas: a foundational Python library for data analysis and statistics," *Python for high performance and scientific computing*, vol. 9, no. 14, pp. 1–9, 2011.

[19] J. D. Hunter, "Matplotlib: a 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[20] A. Saxena and S. A. Robila, "Analysis of the New York city's vehicle crash open data," in *2021 IEEE International Conference on Big Data (Big Data)*, Dec. 2021, pp. 6017–6019, doi: 10.1109/BigData52589.2021.9672012.

[21] A. Saxena, “NYC_VehicleCrash_Analysis,” *GitHub*, 2021. [Online]. Available: https://github.com/abhi1188/NYC_VehicleCrash_Analysis (accessed Jun. 01, 2022).
[22] F. Pedregosa *et al*., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

BIOGRAPHIES OF AUTHORS

Abhishek Saxena is a technical manager in one of the leading Indian IT services companies. He completed his MS in Data Science (2022) at Montclair State University, in Montclair, NJ. He can be contacted at email: 4abhi1711@gmail.com.

Stefan A. Robila is Professor of Computer Science at Montclair State University, in Montclair, NJ. He completed his undergraduate studies in Computer Science at University of Iasi, Romania in 1997 and then continued his education with an MS in Computer Science (2000) and a Ph.D. in Computer Information Science (2002) at Syracuse University, Syracuse, NY. Dr. Robila’s main research interests are in computational sensing. Dr. Robila has worked extensively with collection and analysis of hyperspectral data, and the development and implementation of computationally efficient feature extraction algorithms that use high performance computing as well as social aspects of cybersecurity. This work has now expanded into more general research and applications for large data sets. He can be contacted at email: robilas@montclair.edu.