Data Driven Hourly Taxi Drop-offs Prediction using TLC Trip Record Data

Chathurika S. Wickramasinghe, Daniel Marino, Fatih Yucel, Eyuphan Bulut, and Milos Manic
Virginia Commonwealth University, Richmond, Virginia.
{brahmanacsw,marinodl, yucelf, ebulut}@vcu.edu, misko@ieee.org

***Abstract*—Crowdsourcing applications are proven to be a promising tool to gather valuable information, which can be used for a wide range of tasks, such as ensuring public safety. Traffic data collected using these applications have been used for efficient evacuation planning in large cities. In this paper, we propose to use regression-based machine learning methods to predict hourly taxi rides for a given location in a target day of week and month. The presented method can be used for the following purposes: 1) Predicting the number of taxi rides for a given location at a given time, 2) Identifying hot spots in a city, 3) Getting a rough count of the population density at a given location at a targeted hour, and 4) Planing evacuation routes for possible disasters. The presented approach has potential use for resource planning and evacuation in large cities. The Taxi and Limousine Commission (TLC) trip record data collected from 2017 to 2018 was used for this experiment. It was found that random forest regression can successfully predict hourly taxi drop-offs for a given taxi zone as well as for the entire city of New York.**

***Index Terms*—Crowdsourcing, Safety and Emergency scenarios, Traffic route prediction, Emergency response**

I. Introduction

Crowdsourcing relies on the wisdom of crowds to effectively achieve large scale and rapid data collection by distributing tasks among paid or voluntary workers through online platforms such as Amazon MTurk [1]. Probably the best example that demonstrates the true potential of crowdsourcing is Wikipedia [2], an enormous encyclopedia that owes its existence entirely to crowdsourcing.

The scope of crowdsourcing applications has largely expanded via developments in network accessibility and wide-reaching adoption of smartphones and other mobile devices which are capable of collecting and transmitting various types of sensor data (e.g., GPS, microphone, camera). Some prominent examples of crowdsourcing are taking pictures of specific locations [3], ride-sharing [4], food delivery [5], and traffic reporting [6].

A crucial application area of crowdsourcing is public safety and emergency scenarios [7], where the crowd can help maintain and protect the public safety proactively by informing the authorities of suspicious behavior they observe in their surroundings and reactively by recording and submitting eye-witness data on the events that are presently investigated by law enforcement agencies. This, in turn, enables the authorities to come up with a more befitting emergency response and effectively prevent/detect crime.

In such applications, detection of *hot-spots* (i.e., areas of significant activity) is generally the very first step as the precautions and reactions to be taken, respectively, before and during the emergency scenarios highly depend on the characteristics of the area. Since the number of taxi trips taken to/from a certain area is indicative of the activity within that area, one can, in fact, take advantage of such data, if available, in order to identify the hot-spots at the time of the event of interest.

Our goal in this paper is to compare predictive models that can be used to estimate the number of taxi pick-ups and drop-offs that will occur in different regions of the city using the historical taxi trip data. This approach has different advantages including identifying the hot-spots for enhanced emergency response. Our models can be beneficial to not only public safety and emergency applications, but also other crowdsourcing applications such as taxi dispatching [8].

Machine learning has been used to improve the performance of applications in many areas, including robotics, natural language processing, cyber-physical systems [9] [10], networking [11], and intelligent transportation systems. Regression is a widely used supervised machine learning technique, which is used to predict a continuous dependent variable from a number of independent variables [12]. In this work, we compared the performance of four different regression models on making predictions on traffic data.

The data set used for this experiment was the Taxi and Limousine Commission (TLC) Trip data records, which were collected during 2017 and 2018 [13]. Many researchers have performed prediction using TCL data set [14] [15]. However, most of them have focused on the data which were collected before 2016. Therefore, in this experiment, we used the data collected during 2017 and 2018. To the best of our knowledge, there was no much work performed using the data collected during 2017 and 2018. Thus, we are using supervised regression models to predict the following,

1) predict zone wise hourly taxi drop-offs for a given hour in a target day of the week and month
2) predict hourly taxi drop-offs for a given hour in a target day of the week and month, for the entire city

This paper is organized as follows; Section II discusses the related work. Section III presents the experimental setup while Section IV discusses the results of the experiment. Finally, section V presents the conclusions and future research directions.

978-1-7281-3980-7/19/$31.00 ©2019 IEEE

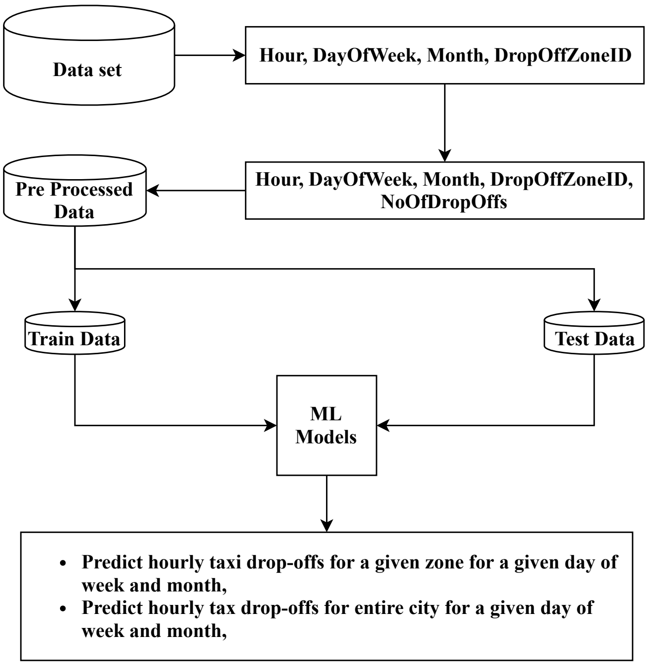

Data set
Hour, DayOfWeek, Month, DropOffZoneID
Pre Processed Data
Hour, DayOfWeek, Month, DropOffZoneID, NoOfDropOffs
Train Data
Test Data
ML Models
Predict hourly taxi drop-offs for a given zone for a given day of week and month,
Predict hourly tax drop-offs for entire city for a given day of week and month,

Fig. 1. Proposed Architecture

.

II. Related Work

In this section, we present an overview of some crowdsourcing applications which aim to improve public safety and emergency response mechanisms (a comprehensive survey on crowdsourcing can be found in [16]).

One of the key benefits of crowdsourcing based public safety applications is that they enable eyewitnesses to submit their evidence anonymously and much more effortlessly (without having to visit a government agency). In fact, when the authorities asked for eyewitness data to identify the perpetrators after the Boston Marathon attack in 2013, thousands of people have reportedly submitted the relevant digital media they had recorded during the event [17].

After several such successful applications of crowdsourcing, more generic public safety applications have been proposed [18–20]. For example, LiveSafe [21] is one of these applications, which is used by many universities in the US. It informs its users of nearby high-risk scenarios as well as allowing them to report anything they find suspicious in their vicinity. A comparative analysis of existing crowdsourcing based crime watch applications is presented in [22].

Lastly, exploiting taxi trip data to design and improve algorithms has also been considered in some crowdsourcing applications. For example, the authors in [8, 23] use taxi trip datasets in order to improve the performance of their taxi dispatching algorithm, and to predict the bike usage patterns, respectively.

III. Experimental Setup for Predicting Hourly Taxi Drop-offs

This section discusses the experimental setup of this work. First, we discuss the overview of this experiment, including data pre-processing steps. Then we discuss supervised machine learning algorithms that were used in this work. Finally, we discuss the evaluation matrices which were used to evaluate the performances of algorithms on the data set.

A. Overview

For this experiment, the NYC TLC trip record data provided by the NYC Taxi and Limousine Commission was used. The data set consisted of three types of trip records: Yellow, Green, and FHV.

Our experiments were performed using Yellow taxi trip records, which were collected during 2017 and 2018. We selected data collected on this period because they had similar features/fields capturing the following details: pick-up and drop-off dates/times, pick-up and drop-off locations/Zones, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts. Furthermore, the experiments showed that using earlier data records results in a significant reduction in prediction performance.

Figure 1 shows an overview of the experiment. First, four features namely hour, day of the week, month, and drop off zone, were extracted for each taxi ride. Then, for each zone, the number of rides per hour were counted (No of drop offs). Therefore, the processed data set consisted of five features: hour, day of the week, month, drop off zone, and no of drop-offs (rides).

These five features took different ranges of values. Machine learning algorithms prefer to work with features which have a similar range of values. Furthermore, some evaluation matrices cannot handle very large integers, which can result from such unprocessed (un-scaled) data. All the features were normalized using min-max scalar, which converts a range of values of a given feature to 0-1 range.

Finally, the pre-processed data set was divided into a train set and a test set. The train set consisted of all the data records that belong to the year 2017, whereas the test data set consisted of all the data that belongs to the year 2018.

B. Algorithms

The experiment was performed using regression methods. They are statistical methods which approximate the relationship between the input variables(features) and the target variables [24]. Unlike classification algorithms where the output is categorical, in regression, the output takes a set of continuous real values. Therefore, regression methods can predict a continuous dependent variable from a number of independent variables [24], [25]. For this experiment, we considered four widely used regression models which are described below.

1) Random Forest Regression: The Random Forest is a supervised learning algorithm which builds multiple decision tree models together, creating an ensemble of decision trees [26]. The basic idea of these ensemble methods is to increase the overall accuracy and stability of results compared to the use of a single decision tree model. Random forest can be used for both, regression and classification. Furthermore, they have the ability to model complex interactions of data. Other advantages of these models include: 1) no sensitivity to noise/outliers, 2) less sensitivity for over-fitting, 3) ability to run efficiently on large data sets, 4) ability to handle high dimensional data, and 5) fewer number of parameters compared to other machine learning algorithms such as Neural Networks [26]. The Random Forest has been used in many areas including remote sensing [27], network intrusion detection [28], and traffic flow prediction [29].

2) K-Nearest Neighbor (K-NN) Regression: The K-Nearest Neighbor is a supervised machine learning method which can be used for both regression and classification [30]. This algorithm uses the similarity between features of K nearest data samples to make predictions on new examples [31]. In K-NN classification, the most common vote among K nearest neighbors is assigned as the class label for a given data sample. In K-NN regression, the average value of K nearest neighbors is assigned as the predicted value of a given data sample. This technique has been widely used in pattern recognition [32], natural language processing [32], event recognition [32] and traffic flow forecasting [33].

3) Neural Network: Artificial Neural Networks are biologically-inspired programming paradigms in which, computers learn to perform tasks by analyzing a training data set [34] [35]. The basic building block of neural networks are neurons which have the ability to take inputs and produce an output. These neurons are organized as several layers and structures, resulting in powerful machine learning models with the ability to represent complex relationships of input data [34] [35]. Neural network based techniques have been used in a wide range of areas including process optimization, fault diagnosis [34], cyber-physical system security [10], and natural language processing [36].

4) Linear Regression: Linear regression is a statistical model which has the capability of finding linear relationships between two sets of continuous variables: a target variable and predictor variables [37]. There are two types of regression models: 1) Simple and 2) Multiple. The simple linear regression finds the relationship between two variables, whereas multiple linear regression deals with more than two variables. Linear regression models are widely used in biological, social and behavioral sciences to model relationships between data [37].

C. Evaluation Metrics

In this experiment, four evaluation matrices which are commonly used to measure the performance of regression models were used.

1) Mean Squared Error (MSE): Mean Squared Error is the average squared difference between the predicted values and the target (actual) values for a set of data records. Therefore this measure tells how close a trained regression model fit to a set of data records. This measure range from 0 to infinity, lower the value better the model is. MSE is calculated as follows where N is the number of data records, y is the target and $y^{'}$ is the prediction, i is the data record.

$$MSE = \frac{1}{N}\sum_{i=0}^{N}\left(y_i - y_i^{'}\right)^2 \tag{1}$$

2) Mean Squared Logarithmic Error (MSLE): Mean Squared Logarithmic Error measures the variation of the mean squared error. This error considers the relative difference between the actual and predicted value, avoiding large errors while penalizing small errors. Therefore, MSLE is specially used in cases where there is a wide range of target values. MSLE is a non negative value. Best possible MSLE value is 0.0. MSLE is calculated as follows,

$$MSLE = \frac{1}{N}\sum_{i=0}^{N}\left(\left(log\left(y_i + 1\right)\right) - log\left(y_i^{'} + 1\right)\right)^2 \tag{2}$$

3) Mean Absolute Error (MAE): Mean Absolute Error is the average absolute value difference between the predicted values and the target values for a set of data records. Compared to MSE, MAE is robust to outliers because it does not use the squared differences. MAE also results in a positive value (including 0). Better models take lower values. MAS is calculated as follows,

$$MAE = \frac{1}{N}\sum_{i=0}^{N}\left|y_i - y_i^{'}\right| \tag{3}$$

4) Explained Variance Score (EVS): The explained variance score computes the explained variance regression score, which is a mathematical model accounts for the variation of a given data set. Higher values of explained variance indicates a stronger strength of association, i.e., the trained model has the capability to make better predictions. Best possible EVS is 1.0, lower values are worse [38]. EVS is calculated as follows where Var is Variance (the square of the standard deviation). If y' is the estimated target output, y the corresponding (correct) target output, and Var is Variance, then the explained variance is estimated as follow:

$$EVS(y, y^{'}) = 1 - \frac{Var\left(y - y^{'}\right)}{Var\left(y\right)} \tag{4}$$

IV. Results and Discussion

This section presents the results obtained for this experiment. First, it discusses the prediction results obtained for zone-wise hourly drop offs count for a given day of the week and month. Then it discusses the results obtained for hourly taxi drop off predictions for the entire Manhattan NYC (all zones).

Table 1 shows the performance comparison between four models for the task of predicting the number of taxi drop-offs for a given zone at a given time. Out of the four models, random forest regression showed the best performance when predicting the number of taxi drop-offs for a given zone at a given hour (hourly drop offs count). It showed MSLE of

TABLE I

COMPARISON OF REGRESSION MODEL PERFORMANCES FOR ZONE WISE PREDICTIONS

Model	Evaluation Matrices				Model Details
	mean_squared_log_error	explained_variance_score	mean_squared_error	mean_absolute_error	
Random Forest Regressor	0.0000	0.9749	0.0001	0.0039	max_depth=30, n_estimators=200
MLP Regressor	0.0034	0.1061	0.0045	0.0359	hidden layer sizes= (50,), solver='adam', activation='relu', alpha=0.001,
K Neighbors Regressor	0.0013	0.6696	0.0016	0.0193	n_neighbors=2
Linear Regression	0.0036	0.0415	0.0047	0.0415	fit_intercept=True, normalize=False, copy_X=True, n_jobs=None

0, EVS of 0.97, MSE of 0.001 and MAE of 0.0039. The second best model was K-Neighbor regression. The worst performance was showed by linear regression. Different parameter values used for all the models were given in the last column of Table 1.

Figure 2 shows the true labels (actual no of drop offs per hour) plotted against the predicted labels (number of drop offs per hour, predicted by the model). An accurate model should produce a perfect 45°line, which corresponds to a perfect matching between predicted labels and true labels. It can be seen that the random forest regressor was able to show a precise linear relationship between the true value and the predicted value whereas K-Nearest regressor showed a moderately linear relationship. The MLP regression and the linear regression models performed poorly compared to random forest and K-nearest regressor. Therefore, it was observed that random forest regression can be used to make accurate predictions of hourly drop off utilizing TCL yellow taxi data.

Table 2 shows the performance comparison between four models for the task of predicting hourly taxi drop-offs for the entire city at a given day of week and month. The last column of the Table 2 represents the parameters used for each model. As similar to zone wise hourly drop offs, random forest regressor showed the best performance with highest EVS and lowest error, whereas K-Nearest regressor showed the second best performance. Linear Regressor showed highest error values and lowest EVS.

Figure 3 shows the true labels (values) plotted against the predicted labels for hourly drop offs for the entire city. It can be seen that the random forest and K-Nearest regressors showed a precise linear relationship between true values and predicted values compared to the other two regression models.

As overall, it was observed that the Random Forest regression can be used to make hourly drop off predictions for zone wise as well as for the whole city using TCL yellow taxi data very accurately.

V. CONCLUSIONS AND FUTURE WORK

Crowd-sourced applications can be used to gather valuable data such as data collected from taxi rides. Machine learning techniques can successfully be used to analyze these data and to extract valuable information which can be used in a wide range of tasks, including emergency evacuation planning. In this paper, we used TLC taxi data with supervised machine learning techniques in order to predict the number of taxi rides for a given hour. We found that random forest regression can successfully be used to make reliable predictions of number of taxi rides in a specific hour for a given taxi zone as well as for the whole city. The extracted information can be used in several applications such as: 1) Identifying hot spots in a city, 2) Getting rough count of the number of people available at a given location at a given hour, and 3) Planing evacuation routes for possible disasters. In future work, more analysis will be performed by expanding the data set as well as including more features for the prediction.

REFERENCES

[1] *Amazon mechanical turk*. [Online]. Available: https://www.mturk.com/..

[2] O. Nov, "What motivates wikipedians?" *ACM Communications*, vol. 50, no. 11, pp. 60–64, 2007.

[3] E. Horvitz and J. Krumm, *Microsoft spatial-crowdsourcing*, 2014. [Online]. Available: https://www.microsoft.com/en-us/research/project/spatial-crowdsourcing/.

[4] *Uber*. [Online]. Available: https://www.uber.com/.

[5] UberEats, 2018. [Online]. Available: https://www.ubereats.com/.

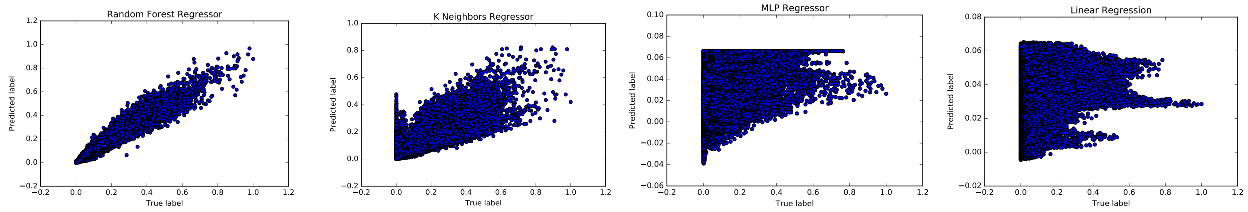
[6] Waze, 2018. [Online]. Available: https://www.waze.com/.

[7] LEEDIR, *Large emergency event digital information repository*. [Online]. Available: https://www.leedir.com/.

[8] J. Xu, R. Rahmatizadeh, L. Bölöni, and D. Turgut, "Taxi dispatch planning via demand and destination modeling," in *43rd IEEE Conference on Local Computer Networks, LCN 2018, Chicago, IL, USA, October 1-4, 2018*, 2018, pp. 377–384. DOI: 10.1109/LCN.2018.8638038. [Online]. Available: https://doi.org/10.1109/LCN.2018.8638038.

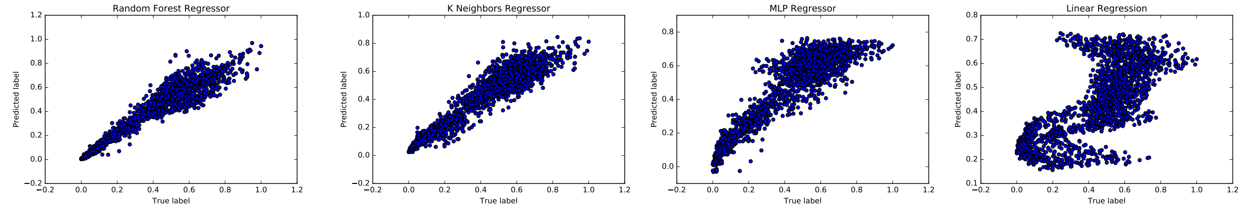
TABLE II
COMPARISON OF REGRESSION MODEL PERFORMANCES FOR THE WHOLE CITY

Model	Evaluation Matrices				Model Details
	mean_squared_log_error	explained_variance_score	mean_squared_error	mean_absolute_error	
Random Forest Regressor	0.0019	0.9105	0.0045	0.0464	max_depth=20, n_estimators=200
MLP Regressor	0.0048	0.8269	0.0105	0.0822	hidden layer sizes= (1000,), solver='adam', activation='relu', alpha=0.0001,
K-Neighbors Regressor	0.0026	0.8865	0.0058	0.0598	n_neighbors=8
Linear Regression	0.0137	0.4969	0.0258	0.1308	fit_intercept=True, normalize=False, copy_X=True, n_jobs=None



(a) (b) (c) (d)

Fig. 2. True Labels(values) vs Predicted Labels for Zone wise hourly taxi predictions (a) Random Forest Regression, (b) K-Neighbors Regression, (c) MLP Regression and (d) Linear Regression



(a) (b) (c) (d)

Fig. 3. True Labels(values) vs Predicted Labels for hourly predictions for the whole city (a) Random Forest Regression, (b) K-Neighbors Regression, (c) MLP Regression and (d) Linear Regression

[9] H. B. Pasandi and T. Nadeem, "Challenges and limitations in automating the design of mac protocols using machine-learning," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2019, pp. 107–112.

[10] K. Amarasinghe, C. Wickramasinghe, D. Marino, C. Rieger, and M. Manicl, "Framework for data driven health monitoring of cyber-physical systems," in *2018 Resilience Week (RWS)*, 2018, pp. 25–30.

[11] H. Barahouei Pasandi and T. Nadeem, "Poster: Towards self-managing and self-adaptive framework for automating mac protocol design in wireless networks," in *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, 2019, pp. 171–171.

[12] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012, ISSN: 1083-4419. DOI: 10.1109/TSMCB.2011.2168604.

[13] N. T. L. Commission. (). Tlc trip record data, [Online]. Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[14] X. Xu, B. Su, X. Zhao, Z. Xu, and Q. Sheng, "Effective traffic flow forecasting using taxi and weather data," in *Advanced data mining and applications*, ser. Lecture Notes in Artificial Intelligence, Springer, Springer Nature, 2016, pp. 507–519.

[15] A. Howard, T. Lee, S. Mahar, P. Intrevado, and D. Woodbridge, "Distributed data analytics framework for smart transportation," Jun. 2018, pp. 1374–1380.

[16] M. Yuen, I. King, and K. Leung, "A survey of crowdsourcing systems," in *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*, 2011, pp. 766–773. DOI: 10.1109/PASSAT/SocialCom.2011.203. [Online]. Available: https://doi.org/10.1109/PASSAT/SocialCom.2011.203.

[17] D. Arrindell and L. Lyster, *Crowdsourcing to solve crimes*, 2013. [Online]. Available: https://www.yahoo.com/news/just-explain-it--crowdsourcing-to-solve-crimes-190839795.html?ref=gs.

[18] I. P. Cvijikj, C. Kadar, B. Ivan, and Y. Te, "Towards a crowdsourcing approach for crime prevention," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, UbiComp/ISWC Adjunct 2015, Osaka, Japan, September 7-11, 2015*, 2015, pp. 1367–1372. DOI: 10.1145/2800835.2800971. [Online]. Available: https://doi.org/10.1145/2800835.2800971.

[19] S. Shah, F. Bao, C. Lu, and I. Chen, "CROWDSAFE: crowd sourcing of crime incidents and safe routing on mobile devices," in *19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2011, November 1-4, 2011, Chicago, IL, USA, Proceedings*, 2011, pp. 521–524. DOI: 10.1145/2093973.2094064. [Online]. Available: https://doi.org/10.1145/2093973.2094064.

[20] iGoSafely, *Personal security alarm application*. [Online]. Available: http://www.igosafely.com.

[21] LiveSafe. [Online]. Available: http://www.livesafemobile.com/.

[22] I. Ariffin, B. Solemoon, and M. L.W. A. Bakar, "An evaluative study on mobile crowdsourcing applications for crime watch," in *International Conference on Information Technology and Multimedia (ICIMU), IEEE*, 2014.

[23] D. Singhvi, S. Singhvi, P. I. Frazier, S. G. Henderson, E. O'Mahony, D. B. Shmoys, and D. B. Woodard, "Predicting bike usage for new york city's bike sharing system," in *Computational Sustainability, Papers from the 2015 AAAI Workshop, Austin, Texas, USA, January 26, 2015.*, 2015. [Online]. Available: http://aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10115.

[24] F. E. Harrell Jr., *Regression Modeling Strategies*. Berlin, Heidelberg: Springer-Verlag, 2006, ISBN: 0387952322.

[25] D. Muchlinski, D. Siroky, J. He, and M. Kocher, "Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data," *Political Analysis*, vol. 24, no. 1, 87103, 2016. DOI: 10.1093/pan/mpv024.

[26] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and qsar modeling," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[27] L. Wang, X. Zhou, X. Zhu, Z. Dong, and W. Guo, "Estimation of biomass in wheat using random forest regression algorithm and remote sensing data," *The Crop Journal*, vol. 4, no. 3, pp. 212 –219, 2016, ISSN: 2214-5141. DOI: https://doi.org/10.1016/j.cj.2016.01.008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214514116300162.

[28] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008, ISSN: 1094-6977. DOI: 10.1109/TSMCC.2008.923876.

[29] Y. Hou, P. Edara, and C. Sun, "Traffic flow forecasting for urban work zones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1761–1770, 2015, ISSN: 1524-9050. DOI: 10.1109/TITS.2014.2371993.

[30] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved k-nearest neighbor model for short-term traffic flow prediction," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 653 –662, 2013, Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013), ISSN: 1877-0428. DOI: https://doi.org/10.1016/j.sbspro.2013.08.076. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042813022027.

[31] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967, ISSN: 0018-9448. DOI: 10.1109/TIT.1967.1053964.

[32] N. Bhatia and Vandana, "Survey of nearest neighbor techniques," *CoRR*, vol. abs/1007.0085, 2010. arXiv: 1007.0085. [Online]. Available: http://arxiv.org/abs/1007.0085.

[33] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303 –321, 2002, ISSN: 0968-090X. DOI: https://doi.org/10.1016/S0968-090X(02)00009-8. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X02000098.

[34] C. S. Wicramasinghe, K. Amarasinghe, and M. Manic, "Deep self-organizing maps for unsupervised image classification," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019, ISSN: 1551-3203. DOI: 10.1109/TII.2019.2906083.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[36] C. S. Wickramasinghe, D. L. Marino, K. Amarasinghe, and M. Manic, "Generalization of deep learning for cyber-physical system security: A survey," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 745–751. DOI: 10.1109/IECON.2018.8591773.

[37] H. Tanaka, I. Hayashi, and J. Watada, "Possibilistic linear regression analysis for fuzzy data," *European Journal of Operational Research*, vol. 40, no. 3, pp. 389 –396, 1989, ISSN: 0377-2217. DOI: https://doi.org/10.1016/0377-2217(89)90431-1. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0377221789904311.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.