A Model for Estimating Bike-Lane Demand

Aadarsh Vadakattu, Mohammad Amin Kuhail, and Harish Tata

Abstract—Unfortunately, bike-related accidents are very common. Some of these accidents can be deadly. Most of the times, these accidents happen due to the lack of a safer means of commute, a bike lane. Further, many citizens could potentially be cyclists if bike-lanes were installed. Driven by the idea of improving safety and convenience for cyclists, we contribute to a model that estimates bike-lane demand in the city. We argue that the demand for bike lanes increases, as the number of bike-related accident increases. Further, the demand increases as the number of popular businesses increases since some citizens commute to work and get around by bike. Our model estimates the demand for bike lane using accidents and ratings of businesses. Accidents are defined by features that represent the severity as well as the cause of the accident. Our model uses the Weight of Evidence algorithm to determine the significance of the accident features. Further, the model uses an algorithm that breaks down roads into equally sized sections based on the US addressing standards. The estimation of bike-lane demand is expressed via scores assigned to road sections.

Index Terms—Smart city, bike lane demand, bicyclist safety, weight of evidence.

I. INTRODUCTION

The number of cyclists in on rise. In the United States alone, there were 43 million cyclists in 2013. That number increased to 47.5 million in 2017 [1]. Many people use bike share services which also observed an increased use. Approximately 1 million people used these services in 2010, which increased to 35 million in 2017 [2]. Since many cyclists use roads without bike lanes, people have accidents that can be injuring or sometimes deadly. Out of the 35,000 road crashes in 2015, the number of cyclist fatalities is 818, which makes up to 2.3% of the total fatalities [3]. Among these crashes, nearly a third of them happen because they were being hit by a car [4]. In almost all of these cases, 39% of bicyclists felt threatened because the motorists drove very close to them [4].

Similar to how cyclists increased exponentially, the usage of motor-based vehicles also gradually increased over time [5], [6]. With the increase of the number of cyclists, it would be inefficient for cyclists to use regular roads that are designed for motorists. Since bike lanes are strictly used by cyclists, the probability of a crash with a car would be less. Thus, more bike lanes would increase safety for cyclists.

Apart from increasing safety, bike lanes have other benefits. More bike lanes motivate people living or working nearby to use them. This has a positive impact on their health. Increased bike usage also reduces reliance on fossil-fuel vehicles, which lowers pollution levels.

Motivated to improve safety and convenience for cyclists, we propose a model that analyzes accidents and businesses in a section of a road and estimates the demand for bike lane. The system computes estimation by considering safety and convenience for cyclists. Safety is determined by features from historical bike-related accident data, such as the severity and the type of the accident. Convenience for cyclists is determined by the existence of highly relevant businesses nearby. Historical bike-related accidents are described by several features. Each feature has several attributes involved and every attribute has different levels of impact. These factors are thus, weighted accordingly by the model. The final estimates depend on Severity factors determined by an expert, along with a defined Importance Factor. This factor is obtained by using underlying evidence with the help of the Weight of Evidence algorithm.

II. RELATED WORK

The related work can be divided into three sections: safety measures for cyclists, scoring models and estimations of bicyclist frequencies.

A. Safety Measures for Cyclists

As cities add bike lanes, the number of cyclists increases, and cycling becomes safer [7], [8]. There have been many safety measures taken in the past to protect and increase the safety of cyclists in bike lanes. For instance, the Context Sensitive Design (CSD) is an approach applied while designing bike lanes [9]. This approach considers all possible “contexts of improvement” to improve the safety of cyclists riding on bike lanes. The city of Portland, Oregon, implemented blue thermoplastic “Yield to Cyclist” signs that improved yielding at the most crash-prone spots [10]. Studies also proved that the existence of green colored or high-visible bike lanes encouraged lower levels of conflict [11]. While similar measures are implemented to improve safety for cyclists in bike lanes, there are few measures taken for cyclists’ safety on regular roads. A law from the state of California states that the motorists that violate a three-foot space distance from a cyclist are subject to fine [12]. Though these measures help, the safest path for a cyclist to ride is on a bike lane. According to a survey, 30% of cyclists wanted to have more bike lanes installed as it felt safer for them to ride [4]. Thus, the safety of cyclists and their confidence would be maximized if more bike lanes are installed.

Manuscript received May 17, 2019; revised August 1, 2019. This work was supported in part by the University of Missouri – Kansas City’s Department of Computing and Engineering and the School of Graduate Studies.

The authors are with the University of Missouri, Kansas City, Kansas City, USA (e-mail: avhmf@mail.umkc.edu, kuhailm@umkc.edu, avhmf@mail.umkc.edu).
B. Scoring Models

The scoring model is a weight-aggregating algorithm which is primarily used in evaluating worthiness of credit applicants. Bankers look at credit scores of applicants assigned by the scoring model to check if the applicant is worthy of a loan. These credit scores are calculated aggregations of weights which determine the historical information of the applicant. The model follows an evaluation procedure where different attributes are assigned with relevant weights. These weights are aggregated to finally form an interpretable score [13].

The scoring models implement the Weight of Evidence (WoE) algorithm to determine the predictive power of each attribute of all variables. The predictive power is determined by dividing the dataset into events and non-events and applying the following formula for all attributes:

$$WoE = \ln \left( \frac{\% \text{ of non-events}}{\% \text{ of events}} \right) \times 100$$  \hspace{1cm} (1)

Here, usually a non-event would be a good customer and an event would be a bad customer. Thus, the WoE will be positive if % of non-events is a larger number and vice versa. This model can be fine-tuned and applied to other problems similar to credit.

The second author designed an evidence-based personalized recommender system to recommend city residents to use bikes for transportation, when the historical trips of the user shows evidence for users using bikes in similar circumstances. The system used a scoring model that relied on statistical learning of user transportation habits and preferences [23].

C. Estimation of Bicyclist Frequencies

There have been multiple procedures to estimate bicyclist frequencies. Texas A&M’s Transportation Institute had estimated cyclist frequency by using technology such as inductance loop detectors fixed on bike paths, infrared sensors fixed on small poles, video cameras and video processing, and so on [21]. According to Turner’s research, bicycle traffic can be roughly estimated using simple calculations that depend on local housing and commercial units [22]. Though Texas A&M’s results are highly accurate, using Turner’s methodologies can result in an approximate bicycle traffic estimate without the need of expensive equipment. While all of these models help in bike traffic estimates, none of these approaches indicate the safety levels of cyclists or try to improve the safety standards. Further, these estimates don’t incorporate accident probabilities of cyclists.

III. METHODOLOGY

Our proposed model uses historical data to estimate the need of bike lanes in a city. The model assigns scores to sections of roads. The score is a number from 1 to 10 where 1 means the lowest demand, and 10 means the highest demand.

The scores the model generates rely on two factors: severity and importance of the accident. The severity factor is a score which is assigned by an expert to each feature of an accident. A government roads department official can be the expert. However, we assigned the severity scores based on our personal judgement, and these scores can be adjusted by an expert in future versions of the model. The severity scores are later multiplied with an Importance factor. The Importance factor is estimated using the Weight of Evidence Algorithm, which determines how predictive each attribute is in determining the final score estimate.

In the following sections, we will introduce the architecture of the model and how the individual components manipulate the data and return a score. The main components are - The Geocoder, the Address Sectionizer, the Accident Severity module, the Evidence Calculator and the Score Generator.

A. System Architecture

Fig. 1 shows the system architecture of the model. The entire implementation can be found at [20]. The historical accident and business data with latitude and longitude are sent to a Geocoder. The Geocoder generates standardized addresses for each combination of latitude and longitude. The data from the Geocoder is used by the Address Sectionizer, which groups the accident locations into road sections based on the street numbers. We estimate bike lane need to the sections defined by the Address Sectionizer. The Evidence Calculator uses a condition along with a tweaked Weight of Evidence algorithm. The Weight of Evidence helps in estimating the feasibility of each attribute. Meanwhile, an expert’s task is to identify the attributes of all the features of a bike accident and assign severity scores to the Accident Severity module. The Geocoder, the Address Sectionizer, the Evidence Calculator and the Accident Severity module send their data to the Score Generator, where all the processed data is aggregated to obtain a final determined score. The Score Generator normalizes the scores on a scale of 1 to 10, where a 10 would imply the road highly need a bike lane and a 0 means a bike lane is not needed.

In the following sections, a more-detailed explanation of all components is given.

B. Geocoder

We used two datasets to determine the bike-lane demand estimate. For the accident information, we used historical accident data from the state of North Carolina [15]. For the business’ information, we used a dataset from Yelp [16]. The
address data obtained from both the datasets needed to be cleaned, as there were many occurrences that had the same street names in different types of formats. For example, there were multiple occurrences of Park Place and Park Pl., which were the same road with different names. While correcting these issues, we also had to merge data from both the datasets and make the addresses unique for the same location. We also had to find a way to obtain missing addresses. For all of these purposes, we used the latitude and longitude of each accident occurrence and pushed them through an online geocoding service called Geocodio [17]. The Geocodio API returned a standardized address of a building that is the closest to the determined lat-lon pair. This simplified the cleaning process and also made it easy for getting unknown addresses.

The Geocoder effectively returned addresses with address numbers and their street names in the format shown below, for all of the records.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Street Name</td>
<td>Categorical</td>
<td>Describes the neighborhood. Examples include Residential, Commercial, etc.</td>
</tr>
<tr>
<td>Bikeinjury</td>
<td>Categorical</td>
<td>Determines the injury levels of the cyclist.</td>
</tr>
<tr>
<td>Drvinjury</td>
<td>Categorical</td>
<td>Determines the injury levels of the motorist.</td>
</tr>
<tr>
<td>Bikepos</td>
<td>Categorical</td>
<td>Determines the position of the bike during the crash.</td>
</tr>
<tr>
<td>Crashgrp</td>
<td>Categorical</td>
<td>Shorty describes the reason of the crash.</td>
</tr>
<tr>
<td>Crashloc</td>
<td>Categorical</td>
<td>Determines the position of the crash with respect to the road intersection.</td>
</tr>
<tr>
<td>Crashtype</td>
<td>Categorical</td>
<td>Describes the reason of the crash in further detail.</td>
</tr>
<tr>
<td>Development</td>
<td>Categorical</td>
<td>Describes the neighborhood. Examples include Residential, Commercial, etc.</td>
</tr>
<tr>
<td>Ambulancer</td>
<td>Boolean</td>
<td>Determines if there was an Ambulance assistance after the crash.</td>
</tr>
<tr>
<td>Bikedir</td>
<td>Categorical</td>
<td>Determines if the accident happened because of an ongoing vehicle, a facing vehicle or none.</td>
</tr>
</tbody>
</table>

**TABLE I: GEOCODER OUTPUT**

<table>
<thead>
<tr>
<th>Latitude</th>
<th>Longitude</th>
<th>Dataset Address</th>
<th>GC Address Num</th>
<th>GC Street</th>
</tr>
</thead>
<tbody>
<tr>
<td>35.29029</td>
<td>-80.790794</td>
<td>800 E SUGAR CREEK RD</td>
<td>800</td>
<td>E SUGAR CREEK RD</td>
</tr>
<tr>
<td>35.724439</td>
<td>-77.911446</td>
<td>GOLDENROD ST</td>
<td>126</td>
<td>Taylor Notion Rd</td>
</tr>
<tr>
<td>34.699022</td>
<td>-77.060245</td>
<td>TAYLOR NOTION RD</td>
<td>205</td>
<td>N Marshall St</td>
</tr>
<tr>
<td>36.099123</td>
<td>-80.248214</td>
<td>MARSHAL ST</td>
<td>424</td>
<td>1513</td>
</tr>
<tr>
<td>34.705645</td>
<td>-79.130441</td>
<td>SR 1513</td>
<td>3171</td>
<td>Everest Green Church Rd</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Street Name</td>
<td>Categorical</td>
<td>Describes the neighborhood. Examples include Residential, Commercial, etc.</td>
</tr>
<tr>
<td>Bikeinjury</td>
<td>Categorical</td>
<td>Determines the injury levels of the cyclist.</td>
</tr>
<tr>
<td>Drvinjury</td>
<td>Categorical</td>
<td>Determines the injury levels of the motorist.</td>
</tr>
<tr>
<td>Bikepos</td>
<td>Categorical</td>
<td>Determines the position of the bike during the crash.</td>
</tr>
<tr>
<td>Crashgrp</td>
<td>Categorical</td>
<td>Shorty describes the reason of the crash.</td>
</tr>
<tr>
<td>Crashloc</td>
<td>Categorical</td>
<td>Determines the position of the crash with respect to the road intersection.</td>
</tr>
<tr>
<td>Crashtype</td>
<td>Categorical</td>
<td>Describes the reason of the crash in further detail.</td>
</tr>
<tr>
<td>Development</td>
<td>Categorical</td>
<td>Describes the neighborhood. Examples include Residential, Commercial, etc.</td>
</tr>
</tbody>
</table>

The module would group the roads into sections of five hundreds as shown below. These are the sections which would be utilized by the score generator to assign the estimate scores. Thus, in the following example, a section of Camp Stewart Rd would start from the 30th street and would end at the 35th street.

**TABLE II: FLOOR AND CEIL OF ADDRESS NUMBERS**

<table>
<thead>
<tr>
<th>Street Name</th>
<th>Min (Street Number)</th>
<th>Max (Street Number)</th>
<th>Floor (Street Number)</th>
<th>Ceil (Street Number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adams Pl</td>
<td>3113</td>
<td>3112</td>
<td>3000</td>
<td>3000</td>
</tr>
<tr>
<td>Alphamate Rd</td>
<td>5173</td>
<td>5174</td>
<td>5000</td>
<td>5000</td>
</tr>
<tr>
<td>Baybrook Ln</td>
<td>9105</td>
<td>9103</td>
<td>9000</td>
<td>9000</td>
</tr>
<tr>
<td>Bradford Dr</td>
<td>2119</td>
<td>2118</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Camp Stewart Rd</td>
<td>2790</td>
<td>2791</td>
<td>6000</td>
<td>6000</td>
</tr>
</tbody>
</table>

D. Accident Severity Module

From the accident dataset, we selected the following nine features that would determine the cause of a specific accident.

All the features, with the exception of the first feature, are categorical. For the purpose of our model, we converted the categorical values using a scale from 0 to 5 to indicate the severity of the accident. The scale is interpreted as follows: 0
means the accident is the least severe or irrelevant for our purpose, and 5 means deadly. We assigned the scores based on our personal judgement. However, ideally an expert should assign the severity values to indicate higher scores to attributes that would be causing a severe bike accident on a road without bike lanes. Table V shows an example of severity scores for the Bikedir feature.

**TABLE V: SEVERITY SCORES FOR BIKEDIR**

<table>
<thead>
<tr>
<th>Bikedir</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facing Traffic</td>
<td>5</td>
</tr>
<tr>
<td>With Traffic</td>
<td>3</td>
</tr>
<tr>
<td>Unknown</td>
<td>0</td>
</tr>
<tr>
<td>Not applicable</td>
<td>0</td>
</tr>
</tbody>
</table>

### E. Evidence Calculator

The Evidence calculator evaluates the importance of each accident feature based on the predictive power of each attribute of every feature. To find the attribute importance, we designed the Evidence calculator so that it determines a value from 0 and 1 for all attributes, where 0 determines that the attribute is not important and 1 being highly important. This value is then multiplied by the accident severity scores in the Score Generator module. To add validity to our model, we wanted to weaken the accident features that were present in bike-related accidents that occurred in roads with bike lanes. To identify these importance values, we used the following formula:

$$WoE_{\text{attribute}} = \ln \left( \frac{\text{no. of attribute specific accidents w/o bike lane}}{\text{total no. of accidents w/o bike lane}} \right) \frac{\text{no. of attribute specific accidents with bike lane}}{\text{total no. of accidents with bike lane}} \tag{2}$$

We utilized the non-events as occurrences of bike accidents that happened on a road without bike lanes and the events as occurrences that happened in a road with bike lanes. Thus, a positive WoE means that there are accidents that happened on roads without bike lanes but did not happen on roads with bike lanes. On the other hand, a negative WoE means that the accident features occurred both on roads with and without bike lanes. Hence, we use the negative WoE to weaken the influence of that accident feature. Fig. 3 shows the counts and WoE for the feature “Crashgrp”:

$$WoE_{\text{attribute\ raw}} = \frac{\ln \left( \frac{\text{no. of attribute specific accidents w/o bike lane}}{\text{total no. of accidents w/o bike lane}} \right) \frac{\text{no. of attribute specific accidents with bike lane}}{\text{total no. of accidents with bike lane}}}{\text{max( |WoE| )}} \tag{4}$$

The final step is to calculate the importance factor, which is simply the normalized result of the WoE score. The value of the importance factor is between 0 and 1. We used the following formula to calculate the Importance Factor (IF) using the min-max terminology [19].

$$\text{Importance Factor (IF)}_{\text{raw}} = 1 - \left( \frac{|WoE|}{\text{max( |WoE| )}} \right) \tag{5}$$

Implementing the above formula to the tweaked WoE of “Crashgrp” shows the importance levels of attributes as shown:

$$\text{Importance Factor (IF)}_{\text{raw}} = 1 - \left( \frac{|WoE|}{\text{max( |WoE| )}} \right) \tag{6}$$

Thus, the Importance Factor (IF) helps in weakening the weights of accident features that happened even in the presence of bike lanes. We use this factor in the Score Generator to multiply with the severity scores determined by the expert.
F. Score Generator

The Score Generator calculates a score representing bike-lane demand for each road section. First, the Score Generator combines the severity scores with evidence from the evidence calculator by multiplying the Importance Factor with the severity scores. This yields us Attribute Weights specific for each attribute, which are dependent on their severity and evidence.

\[
\text{Attribute Weight} = \text{Severity score} \times \text{Importance Factor} \quad (5)
\]

Next, the start and end bounds of the address numbers are used from the Address Sectionizer to check with the address numbers of the individual accidents data. If the street numbers fall in the bounds of a specific street name, the attribute weights get added to the feature scores of that road section. Doing this for all accident occurrences will give us total scores for individual features of the accidents that occurred in a specific section of the road. Furthermore, we similarly matched the address numbers of businesses with the start and end street. After finding the matches, we added their average ratings to a business rating column. This meant that the score would be high if there are multiple businesses with a high average rating. We also made sure that there were at least a hundred reviewers for each business to be considered. The algorithm for the described process is as shown below:

```
Algorithm 1: compute_final_feature_scores
input: a road section (R) from the Address Sectionizer, records from the crash dataset with attributes replaced with attribute weights (C) and list of features (F)
output: Feature Scores (F) being updated
1. for each record in C do:
2.   if C[Street_Name]==R[Street_Name] do:
3.     if R[Start_Street_num]<=C[Street_Num]<=R[End_Street_num] do:
4.       for each feature in F do:
5.         F[feature]+=C[feature]
6.     end
7.   end
```

As described, a similar algorithm is implemented for the businesses’ rating as well. To obtain a final score for the sections, we added the scores of all the features of the section along with the business rating. 

\[
\text{final score} = \Sigma \text{feature scores} + \text{business rating} \quad (6)
\]

Using the above formula would yield our scores on undeterminable ranges. Fig. 6 shows a sample of the generated scores.

![Fig. 6. Final aggregated scores sample.](image)

To normalize the scores, we used a range of the scores in between 0 and 10, assuming 10 would definitely need a bike lane and vice versa. To fit the obtained scores in this range, we wanted to determine an upper bound that can be the worst possible score for a road section. Any score that is above this determined score would automatically get 10 as their score. We thought that attaining a score that is higher than the defined upper bound is equally bad as getting the score equal to the upper bound, as both scores would be definitely needing a bike lane regardless of their bad scores. We can determine the lower bound as a 0 directly, assuming the road section has no accidents and no businesses.

![Fig. 7. Section scores histogram.](image)

By looking at the histogram in Fig. 7, we recognized that the scores above 100 are outliers. Thus, we can consider 100 as our upper bound. With these bounds, we implemented the min-max algorithm [19] to obtain a score within 0 to 1. As we needed a score from 1 to 10, we simply multiplied this value with 10. We also determined that any score that crosses the upper bound would be replaced with the upper bound while applying the min max normalization. Thus, the normalized scoring formula is as follows:

\[
\text{normalized score} = \frac{\text{non_normalized score} - \text{min}}{\text{max} - \text{min}} \\
\text{max} = \text{determined by ignoring outliers and finding the maximum possible non-outlier} \\
\text{min} = 0, \text{a road section with no accidents and no businesses}
\quad (7)
\]

Thus, applying the above formula to the total scores, we get the final normalized scores which determine the need of a bike lane to the specific section of a road.

G. Data Independent Model

From the scores generated by the score generator, it is possible to generate a statistical model that would be independent of the data generated by the Evidence Calculator, the Accident Severity module and the Address Sectionizer every time to obtain a score. This would reduce the amount of dependency on the data to obtain the finalized scores. The model is shown in Fig. 8.

![Fig. 8. Data independent model architecture.](image)
The complete score data from the score generator is taken by the Data Independent Model Generator, which would apply one among the existing statistical models to the data. Before doing this, the module converts the scores into classes by taking a simple assumption. The assumption would split the scores into four equally divided classes. As the scores range from 0 through 10, the classification can be done follows: scores from 0 to 2.5 between do not need a bike lane at all, scores from 2.5 to 5 might not need a bike lane, 5 to 7.5 might need a bike lane and 7.5 to 10 would definitely need a bike lane. By making our data classifiable, we can fit classification models to our data. By looking at the existing data, we assumed Naïve Bayes, Logistic Regression, Support Vector Classifier and Random Forests to be good fits. We trained all the above models and obtained accuracy scores as shown in Figure 9:

![Fig. 9. Accuracy scores of statistical models.](image)

By looking at the accuracy scores, it can be understood that Support Vector Classifier and Random Forests work best for this scenario. Thus, The Data Independent Model can be any of either models. Thus, when the user gives different attributes of a road section as inputs, he/she can expect a class output instead of a score that would directly recommend a bike lane.

IV. RESULTS AND EVALUATION

A. Performance of the Estimates

To estimate the scores’ performance, we visualized them along with the counts of accidents and businesses. Figure 10 shows the graph with non-normalized scores. The same conclusions can be drawn from Fig. 11, a plot with normalized scores.

Upon observation, we can determine that the total counts of accidents and businesses are definitely responsible to the generated score. Moreover, it is understandable that more accidents in a section lead to a higher score. The fluctuations can be related to the attribute scores, evidences and the businesses ratings of every accident.

![Fig. 10. Non-normalized scores vs accident and business count.](image)

B. Visualized Estimates

For the better understanding of our results, we visualized the estimates of one road on a street map basing on a coloring scheme. Similar to how we split the scores for the Data Independent Model, we categorized the scores from 0 to 2.5 as green, from 2.5 to 5 as yellow, from 5 to 7.5 as orange and from 7.5 to 10 as red. The scores visualized for a few road sections of “Beatties Ford Rd” are as shown:

![Fig. 12. Visualized estimates of Beatties Ford Rd.](image)

By looking at the visualization, it is evident that the ends of our road section need not have a bike lane, but the sections in between the ending sections are suggested to have bike lanes to improve safety standards for cyclists.

C. Why This Model?

Instead of estimating the bike lane demand using scores and determined weights, it is also possible to estimate the demand directly, using statistical predicting algorithms. To achieve this, the expert must observe all the accident occurrences in a few road sections and must assign the need for bike lanes to these road sections. These few road sections can be given as a training set to a predicting algorithm of our choice. Though this is possible, there are several problems that might affect the efficiency of the model. Some of them are as listed:

1) The dataset of the model is too small to be reliant on predicting algorithms. If these few accidents are used to train the model, there might be possibilities of overfitting on the less amount of data that we have. Our model uses
the Weight of Evidence algorithm to determine the strength of prediction, which has less scope of overfitting when compared to statistical models.

2) Almost all statistical models are not very easy to understand. Comparatively, our model proposes simple aggregations along with an evidence calculating algorithm that can be easily understandable.

3) The predicting algorithms identify patterns and weigh the results, which cannot be easily determinable. Our model determines this importance using the Importance Factor, which could easily determine what specific attribute is more responsible in determining the final estimate. The Importance Factor is comparatively easy to understand than recognizing the patterns in a statistical model.

4) Though both approaches need the help of an expert, it would be difficult for the expert to estimate bike lane need on his own completely for a few road sections, which is to be given as a training set to a predicting algorithm. Comparatively, our model requires simple severity scores on a scale of 0 to 5 for all the attributes, where 5 determines the most severe.

5) There have been instances of applications which involved complex machine learning strategies to find creditworthiness, a problem that is similar in many ways to estimating bike-lane demand. The results showed very low accuracy increments compared to simpler scoring models and also, as these models are not easily understandable, they have a negative impact on interpretability [14]. Alternatively, the scoring model approach used by our algorithm has high interpretability and is simpler to execute. Thus, our model comparatively works better in several ways to estimate the need of a bike lane and is easier to understand.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced a model for estimating bike-lane demand, a data-driven model that can potentially help city officials in determining the demand of a bike lane based on its various features. Based on historical accident data, a specific section of a road is assigned a score based on accident features that are most responsible for the occurrence of a bike-related accident. Also, cyclist traffic and their convenience are estimated using business data, imagining of more businesses leading to more bike traffic. Along with the scores being based on intuition, the computed algorithms weaken scores of those accident features that correlated less with bike-related accidents. This evidence is extracted by comparing the occurrences of accidents that happened in the existence of a bike lane and the accidents that happened without their existence. This is done by utilizing a tweaked Weight of Evidence formula. In the future, we are planning to fit neural network models and evaluate the implementation accuracy of the data independent model. We also plan to increase the number of features, such as surrounding households, demographic information, etc., that can improve the estimates. We also look forward to use datasets from other parts of the world and modify the model to make it location independent. We are also planning to develop a simple application that would display the estimates based on the input location, making it easier for the user to use the model.

REFERENCES


Aadarsh Vadakattu belongs to Kakinada, Andhra Pradesh, India. He received his B.S. and M.S. degrees in computer science from the University of Missouri – Kansas City, USA, in 2018 and 2019 respectively.

He worked as a software engineering intern and a data scientist intern for approximately a year in two different startups in Kansas City, USA. His research interests include data analysis, data mining, machine learning and data science.

Mohammad Amin Kuhail received his M.Sc. in computer engineering from the University of York, York, UK in 2006 and Ph.D. in computer science from the IT University of Copenhagen, Copenhagen, Denmark in 2012.

He is an assistant professor at the University of Missouri - Kansas City, USA. He is an experienced computer scientist and software engineer with a diverse skill set that spans web development, object-oriented programming, algorithms, usability, and data science.

He is an accomplished researcher and educator with a proven record of publications, ABET & pedagogical training, and awards. He is also a seasoned professional who lived in different countries and connected with many people from various cultures and backgrounds. He is also a successful project manager with over four years of leading diverse teams and professionals and advising students.

Harish Tata belongs to Hyderabad, Telangana, India. He received his B.Tech in computer science from Gitam University in 2015. He is now pursuing his M.S degree from the University of Missouri Kansas City, Kansas City, USA in computer science and is expected to graduate in 2019.

He worked as a senior product developer for more than 3 years in Kony India Pvt LTD in India. His interests include data science, deep learning, data analysis and data mining.