# CycleSafe: Safe Route Planning for Urban Cyclists

Mehdi Shah[1], Tianqi Liu[2], Sahil Chauhan[1], Lianyong Qi[3], and Xuyun Zhang[1](✉)

[1] Department of Electrical, Computer and Software Engineering, University of Auckland, Auckland, New Zealand
{msha563,scha676}@aucklanduni.ac.nz, xuyun.zhang@auckland.ac.nz

[2] Northeastern University, Boston, USA
liu.tianq@husky.neu.edu

[3] Qufu Normal University, Jining, China
liangyongqi@gmail.com

**Abstract.** Cyclist numbers in major cities are constantly increasing whilst traffic conditions continue to worsen. This poses a major issue for cyclists who attempt to share congested roads with motor vehicles. This paper shows that there is not enough work being done to improve the safety of cyclists on the road, and proposes a solution to this problem in the form of a route planning application. Current cyclist route planning applications do not take safety factors like traffic, rain or visibility into account when providing cycle routes. We use Auckland city as a case study to explore our solution. The traffic and weather data in Auckland are acquired by using Google, Bing and Wunderground APIs. An evaluation of our solution shows that our system successfully implements a route planning application that routes users away from unsafe traffic conditions, thus improving cyclist safety.

**Keywords:** Traffic data fusion and analytics · Mobile computing · Cloud services

## 1 Introduction

With the increase in popularity of cycling, cyclist safety is becoming a much more important concern to society. For example, Auckland Transport statistics show that there has been a 62% increase in all day cycle trips in the city center compared to 2013 [4]. There are many awareness campaigns to warn drivers that they must safely share the road with cyclists, however, research shows there is not enough work done to ensure the safety of cyclists on the road. We believe that awareness campaigns alone are not enough to improve cyclist safety. Whilst cycling has become much more common, with more cyclists on the road than ever before, traffic congestion has also increased. This poses a much higher risk for cyclists as there are more of them on the road but there are also more cars,


X. Zhang et al. (Eds.): CloudComp 2019/SmartGift 2019, LNICST 322, pp. 312–327, 2020.
https://doi.org/10.1007/978-3-030-48513-9_26

the risk of injury is very high for cyclists who share the road with motor vehicles. This problem of cyclist safety will be addressed in this paper and a viable route planning solution will be presented. This solution takes traffic conditions into account when planning routes and will adjust the route accordingly to ensure that cyclists are on the safest route possible.

## 2 Problem Background

The major problem that is examined is the lack of any applications or systems that directly improve cyclist safety. Statistics show that cyclist numbers are increasing over time, however, there is not an increased effort put into improving the safety of these cyclists. Awareness campaigns are put into place to educate the public in terms of sharing the road with cyclists, however, they are simply not enough as most people do not pay them any heed. Whilst infrastructure is finally being put into place to accommodate for cyclists, it cannot be implemented fast enough to compensate for the increasing number of cyclists. In order for us to find a comprehensive solution to this issue, we must first look at the root cause of the problem and figure out what it is that makes cyclists unsafe on the roads. It was clear that sharing the road with motorists was the primary cause of injury or death on the roads. Because cyclists have to share the road with motor vehicles there is an ever-present danger of injury or death. In 2016, 6% of the total number of casualties from police reported crashes were from cyclists [6]. This statistic only looks at police reported crashes too, there are likely numerous unreported crashes that occur. There are several factors which apply to cyclists, that increase their risk on the road, but do not apply to the vehicles around them. Firstly, they have less protection in the event of any kind of accident, as well as no protection against weather conditions. In a city like Auckland, which has worsening traffic and can undergo many weather changes in a single day, it is very important to ensure that cyclists do not travel in unsafe traffic and weather conditions. The cycling increase statistics do not take into account the bike-sharing increase in recent times. In late 2017, the bike-sharing application ONZO was launched in New Zealand. These ONZO bikes are placed all around Auckland city. Users are able to go to one and using the application, unlock it and rent it for a short or long trip. The rise of bike sharing poses new problems for cyclist safety. It not only increases the amount of cyclists in general but it also allows for less experienced cyclists to share the road with vehicles, these inexperienced cyclists are at an even greater risk as they have not been cycling in traffic conditions and may not the best ways to do so safely. ONZO users also do not always wear helmets whilst riding the bikes. This obviously increases the risk to their lives drastically as helmets, whilst offering little protection compared to vehicles, can still save lives in a crash. Sharing the road with cars during high levels of congestion is a huge problem for cyclists as it reduces the space between cyclists and vehicles, thus, increasing the risk of accidents. This is one of the two major risks in sharing the road with vehicles, with the other being the speed of the vehicles. Vehicles travelling faster are obviously more dangerous for cyclists. A thorough analysis of the Auckland Traffic Count data spreadsheet shows that traffic trends continuously worsened from July 2012 to February 2018 [5]. Because traffic conditions are only worsening as time goes on, cyclists need a way to travel that can allow them to avoid unsafe traffic congestion. Our proposed solution to this issue is to give cyclists a way to travel safely and circumvent congested roads which increase road usage risks.

## 3 Related Work

Because we are looking into a route planning application that can be used by cyclists to improve their safety, the main area of research covered in this paper is cycle route applications. The biggest issue with existing cycle route applications is that they do not focus on cyclist safety. They do not take traffic congestion into account when planning routes for these cyclists so the routes often prove to be unsafe.

### 3.1 Cyclist Route Planning Applications

Whilst there are many route planning applications, from research conducted we found that there are not many which cater primarily for cyclists. Also, after an in-depth analysis, we found that there are no route planning applications which are designed to improve cyclist safety. Several applications were examined thoroughly and then narrowed down to the best 3 applications. A summary of the notable features of these 3 applications can be found in the Table 1 below.

**Table 1.** Comparing existing applications.

| Existing solution | Shortest path | Support cycleways | Elevation tracking | Traffic data | Weather checks |
|---|---|---|---|---|---|
| Google Maps | Yes | Yes | Yes | No | No |
| Flattest Route | No | No | Yes | No | No |
| Open Cycle | No | Yes | No | No | No |
| Proposed | Yes | Yes | Yes | Yes | Yes |

Out of the 3 applications that were examined, the very best one was Google Maps. Open Cycle Map simply provides a map which shows users the locations of cycle paths [8], it does not have any route planning aspect. Flattest Route is a route planning application that is aimed at finding the flattest route between two points [7]. These two applications are overshadowed by Google Maps which can outperform them in every aspect, thus we will be comparing our application to Google Maps in our results and evaluations.

The main focus for Google Maps is to determine the fastest route to a destination [3]. This application contains many cyclist friendly features such as elevation information and cycle path support for routes and is very efficient at finding the shortest path. It does not, however, take traffic congestion into account when plotting these routes. Thus, the safety of cyclists is not a consideration, this further proves the need for our research and application. There is also no consideration for weather conditions.

All these existing route planning applications contain a concerning lack of focus on the safety of their users. Even Google Maps, which is the best route planning application on the market, does not take traffic or weather conditions into account. These are the most important factors in determining cyclist safety and must be considered in order to ensure that cyclists have the safest route possible.

### 3.2 Data Collection Techniques

This section covers the research done to look into traffic data collection techniques. For this paper we required real-time traffic data as that can be used in our route planning application to improve cyclist safety by avoiding areas which show unsafe traffic congestion.

Traditional methods of traffic data collection utilize static sensors such as roadside cameras with image recognition [2] and underground & radar sensors. Whilst, these sensors are effective in the long term they lack redundancy, have a limited sample size and are fairly expensive to implement. They also sometimes require specific installation which can result in road closure and thus an increase in traffic congestion in that area. These systems require a vast amount of static sensors to ensure accurate readings and provide an accurate representation of traffic congestion.

For this paper, time constraints meant that we could not implement static sensors and collect historical data manually. We also wanted to use real-time data to ensure that traffic is updated in our application in real-time. We felt that this would ensure the safest possible travel for our users. After further research, we found that the most accurate method of real-time data collection is through the use of floating car data. This method utilizes the phones of road users to provide an easily obtainable, accurate source of real-time data [9]. We obtained this data through the use of Google and Bing REST APIs as these sources were found to be the most accurate and easiest to use.

The weather data that was used in the application was sourced using Weather Undergrounds REST API. After careful study into this part of the paper, we found it most beneficial to provide our users with warnings if they intend on cycling in unsafe weather conditions. We also looked into weather prediction data, however, found that this can be very inaccurate and is difficult to use in our application. By using real-time data, we can provide users with accurate warnings as they plan their route. Perhaps, once weather prediction becomes more accurate and available this can be implemented in our paper, but for now real-time data proved to be the most appropriate.

## 4 Development Methodology

The goal of this paper is to implement an application that can be used by cyclists to plan routes which will improve their safety. After several ideas, it became clear that the easiest way to implement this is through the use of a phone application. We aim to ensure that the safety of our users is a priority by taking traffic data into account when providing routes. The application also takes weather information into account and provides the user with warnings if conditions are unsafe. We want this application to also provide users with other cyclist specific features that can increase their quality of life as well as add more depth to the application.

### 4.1 Commute Mode

Because the main focus of this paper was the safety applications for cyclists, this was the core functionality that was implemented first. This part of the application was called commute mode. The route planning was implemented using Google & Bing APIs. The information taken from these APIs was plugged into our own algorithm which found the safest route between two points. Once this part of the application was implemented, it could serve as a base to the extra features that we wanted to add and any future work can be built off of this.

### 4.2 Weather Warnings

Next, we focused on getting the weather warnings to be functional as we felt that this is another aspect of cyclist safety that can be improved. Weather data is taken from the Weather Underground Wunderground REST API. This API was chosen as it provides accurate easy to use data that is also readily available. We realized that all we can do in terms of the weather is provide warnings during unsafe travel periods. We cannot physically route cyclists away from unsafe weather if it is raining throughout Auckland, we can only provide users with information and warnings.

### 4.3 Exercise Mode

The next part of the application was implemented as the exercise mode. This mode allows users to plan exercise routes based off of a distance that they input. This mode is based off of the previously developed algorithm. Thus, the route provided will be the safest possible route and take traffic conditions into account. This mode was added to ensure that our application can provide tangible benefits and an increased quality of life for not only those who use cycling to commute but those who use it as an exercise form too.

### 4.4 Machine Learning

We also decided to add a prediction feature to our application. This would allow the traffic levels on common routes to be predicted and shown to our users. The information can be viewed in the application in the form of a graph, it plots the level of congestion against time of day. This feature is useful for cyclists as they can see in advance if their route will be safe to travel or not (Fig. 1).

## 5 System Architecture

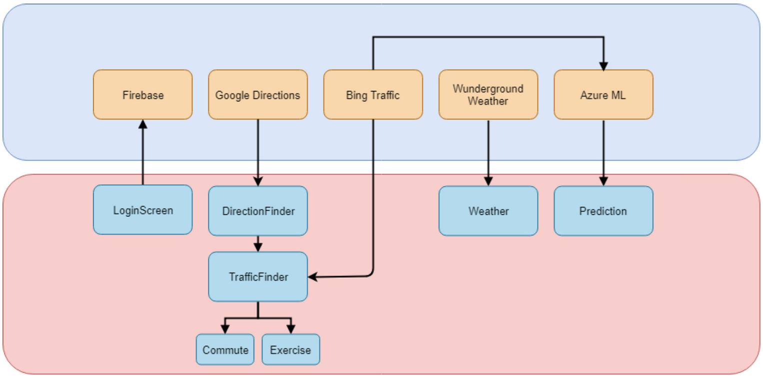


**Fig. 1.** System architecture. (Color figure online)

Our system is split into two main areas, the cloud end and the front end mobile part. The route planning is done locally on the user's mobile device, whilst the cloud is used to pull all the data that we are using. For the following sections, please refer to the above system architecture diagram. The back end (cloud) services are in the large blue box with the front end modules in the large red box. The system is comprised of several modules, with each one having a specific purpose. Our modular system allows for easy expansion and easy integration of more data sources. The Direction Finder module is used to call the Google Directions API and process the JSON data that it receives from Google. The processed data is then fed into our Traffic Finder module. The Traffic Finder module receives its data from Direction-Finder. It calls the Bing Traffic API and finds the traffic along the routes that it receives from Direction Finder. Traffic Finder passes data into the Azure Machine Learning studio to train it and allow for prediction, it is also used to find the safest route. The Weather module is called after a route has been found. It calls the Wunderground API and processes the data that is received. The Prediction Module acquires data from the Azure Machine Learning studio which trains our prediction model. The login Screen module handles logins and was intended to communicate with Google

Firebase to store information. However, the login section of this application was not necessary to ensure functionality and we had to focus on the safety aspect of the paper. Thus, the login database has been added to future work.

# 6 Implementation

Before starting the implementation of this paper there was much planning that had to be done. After conducting sufficient background research, we began to formulate our own solution which would address the problem and its lack of current solutions.

## 6.1 Resources

This section covers the resources that we used and the reasons behind using them.

1) Android Studio: The application was implemented in Android Studio using Java. It was initially discussed to implement the paper using the Ionic framework, however, after further consideration Android Studio was chosen for multiple reasons. Firstly, we have much more experience in object-oriented Java programming than in typescript. Additionally, we found that Android Studio has much better integration with the Google Maps sdk, thus allowing for easier implementation.
2) Google Directions & Google Places APIs: We used the Google Directions API in order to acquire route data between two points. This API takes a start and end location then returns multiple possible routes between the two points. The directions API is the best of its kind and can be very easily integrated with our application. The places API is used in our autocomplete text view to ensure that user experience is of a high quality. This API allows users search for locations more easily.
3) Bing Traffic API: The bing traffic API looks at a segment of road and returns the amount of congestion on it. This API uses traffic data to provide congestion information, from a comparison with other traffic APIs this was found to be the best and easiest to use.
4) Azure Machine Learning Studio: We use the Azure Machine Learning (ML) studio to implement a prediction algorithm and predict traffic along commonly travelled routes. A specified date and time can be passed in and a prediction for traffic congestion is returned. The Azure ML studio was used as it was relatively easy to use and allowed for easy integration with our application.

During this paper, due to time constraints and the fact that we could not get many users to use the application, it was not possible to input the user's common routes into the machine learning algorithm. Instead we ran a python script that called the Google and Bing APIs to find traffic data for specific routes, this data was stored and plugged into the machine learning algorithm. Our prediction model uses this data to make predictions rather than data taken from users who use the application. Thus, our prediction model was more of a proof of concept for now and will need to be worked on in the future to incorporate data from the users themselves.

### 6.2 Modules

The following section covers the modules that were implemented and provide insights into their functionality. 1) Direction Finder: This module uses the Google Directions API to find all the possible routes between two points. It takes the origin and destination coordinates and then calls the Google API. After processing the results, it obtains a list of all the possible routes. From here, it needs to check which route is the safest for our users. This is done by splitting each route into steps and then checking the traffic conditions along each step of the route. The traffic checking is done by the Traffic Finder module. After the Traffic Finder module is run, each route is assigned a total congestion score based on our algorithm. Direction Finder then chooses the route with the lowest congestion score and takes this as the safest route. This route is then sent to the fragment for our current mode (commute or exercise) and plotted onto the map. This module also provides information to our Weather Finder module. It provides the weather module with two sets of coordinates which can allow it to find weather data in that specific area. 2) Traffic Finder: This module is used to find traffic congestion levels. It is given a segment of a route and uses the Bing API to find the current traffic conditions on that segment. This module uses our safety algorithm in order to give each segment of road, thus each route, an overall traffic congestion score. Safety Algorithm: Our algorithm takes specific safety factors into account when looking at traffic conditions. A complete lack of traffic is obviously the safest way to travel for cyclists, so we give this a congestion score of 0. Next, light congestion is assigned a congestion score of 1 with medium being a higher congestion score of 3. High traffic levels are considered the most unsafe and thus have a congestion score of 4, however, very high/standstill traffic is considered to be safer as the vehicles are essentially not moving whereas cyclists can go between cars and be quite safe. Thus, this has a congestion score of 3. This can be easily seen in the Table 2 below:

**Table 2.** Safety algorithm.

| Traffic conditions | Safety |
|---|---|
| No traffic | High |
| Low congestion | Medium |
| High congestion (Slow moving) | Low |
| Very high congestion (Standstill) | Medium |

3) Weather Finder: This module handles all the weather information in our application. It takes two sets of coordinates as an input, which are provided by Direction Finder, and uses this area to call the Wunderground API and acquire weather information. It displays rain info, wind conditions and visibility conditions using a dialog pop up box after finding the safest route.

# 7 Evaluation

The main goal of this paper was to improve cyclist safety. This was done by providing cyclists with a novel route planning application that takes their safety into account whilst still planning efficient routes. The first method of evaluation that was carried out is a direct route comparison with Google Maps. We felt that Google Maps is currently the best route planning application that cyclists have at their disposal. We evaluate against Google Maps to assess the efficiency and improved safety of the routes that our application provides. The next evaluation method is an evaluation of our application itself and its usability. This was done by providing surveys to users of our app. The following section will discuss the evaluation of our results in detail.

## 7.1 Google Maps Comparison

For each of the following results, a route was calculated using cycling mode on Google Maps and using our application on the same day at the exact same time. The traffic conditions along the routes are also shown in images which follow each route. This was done by manually plotting the routes on Google Maps using driving mode, as this is the only way to see traffic along the route, so the safety of our routes can be assessed. For each of these routes we are looking at an efficiency comparison between our route and the Google route. We also want to judge our algorithm to see if we are successfully avoiding unsafe traffic conditions and providing safer routes than Google Maps. This was repeated for different routes on different days at various times to provide a wide range of results.

1) Grange Rd - Bassett Rd: In the following figure (Fig. 2) we can see a route from Grange Road in Mount Eden to Bassett Road in Remuera. Our application is displayed on the left with Google Maps on the right as a comparison.

Our route is vastly different from the one provided by Google Maps. In terms of efficiency, ours is 5.3 km long while the Google route is 4.4 km it also takes five extra minutes of travel time. So our route is slightly less efficient than the one Google recommends.

Safety Comparison: Fig. 3 utilises Google Maps' driving mode to show the traffic congestion along our route and the Google Maps cycling route.

As can be seen in Fig. 3 above, our route (left) avoids a lot of the unsafe congestion that is present (circled) on the Google route. There is slight congestion along our route (red arrows), however, we feel that this is negligible compared to the amount that we avoid in comparing with the Google route. It should also be

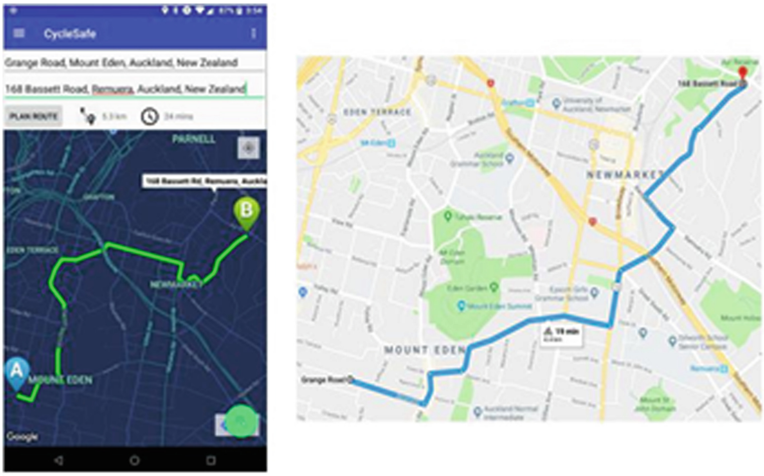


**Fig. 2.** Grange Rd. - Bassett Rd. route comparison

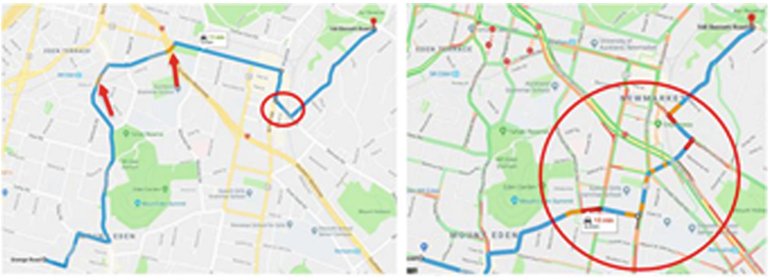

**Fig. 3.** Grange - Bassett: traffic comparison. (Color figure online)

noted that the red circle in our route shows the driving mode differing from our actual route. This is because our cycling route goes through a park at this point which cannot be done in the driving route, this is taken as being safe as there cannot be any traffic congestion in the park. From this result, it appears that our initial assessment of Google Maps was correct. Google does not take traffic conditions into account when planning routes for cyclists, instead the application just plots the shortest path between two points. The results which follow will further prove this assessment.

2) Paice Ave - OGGB (UOA): Fig. 4 below shows the cycling route from Paice Avenue in Sandringham to the Owen Glenn building in the University of Auckland as given by our application (left) compared to Google Maps (right).

Both these routes are very similar, they both end up taking the Northwestern & Grafton Gully cycle paths. Because the routes are so similar they are almost identical in length, with ours being 200 m longer and taking 2 min longer in

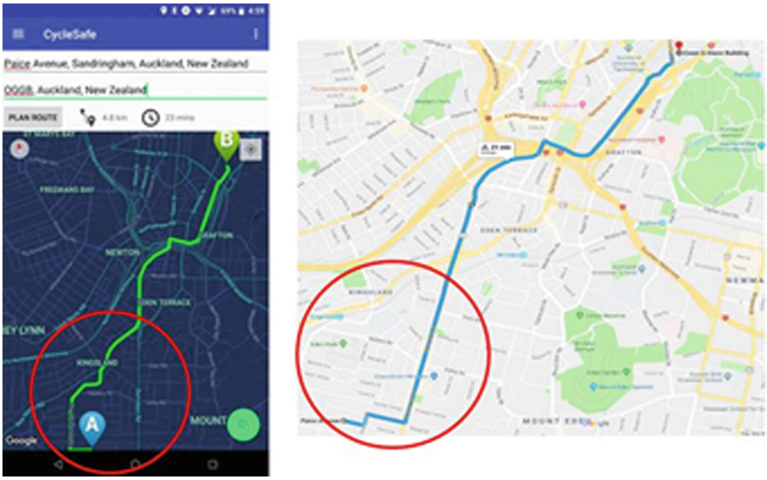


**Fig. 4.** Paice Ave - OGGB: route comparison

travel time. The main difference between the two routes has been circled and will be assessed in terms of safety.

Safety Comparison: Fig. 5 shows the traffic conditions along the parts of the routes that differ (circled in the previous images).

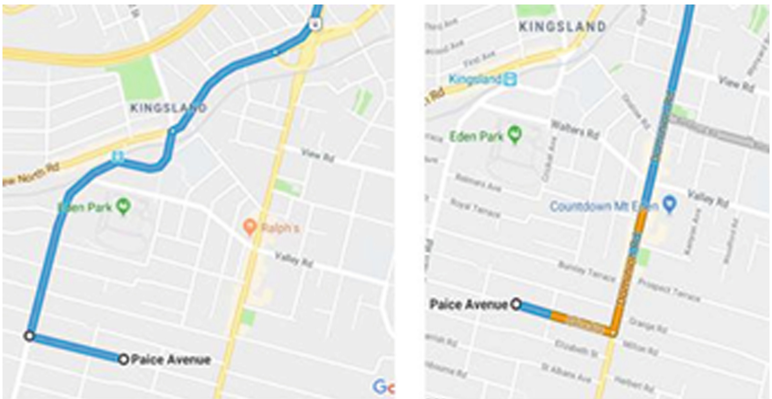


**Fig. 5.** Paige - OGGB traffic comparison

The very start of the two routes is the only point of difference. Our route avoids slight congestion on dominion road and has almost the same time and distance as the Google Maps route. This shows that our algorithm is working as intended as it routes our cyclists away from unsafe congestion conditions.

Whereas, Google Maps' route only acts as the shortest path and routes cyclists through Dominion road which contains some unsafe traffic congestion conditions.

3) Larchwood Ave - George Street: Figs. 6 shows the cycling route from Larchwood Avenue in Westmere to George Street in Mount Eden.

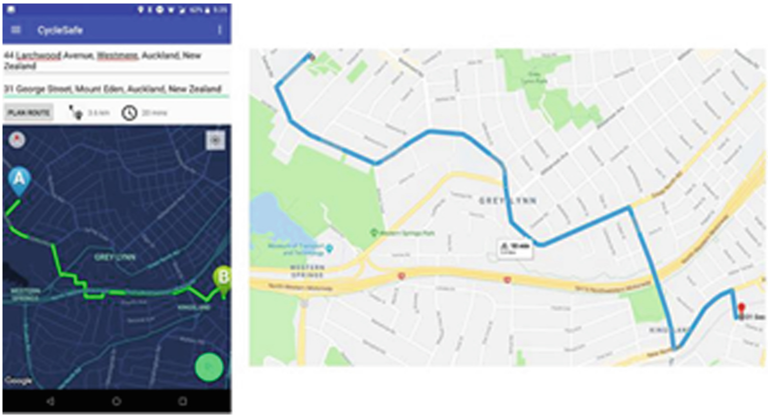


**Fig. 6.** Larchwood Ave - George street route comparison

In this scenario, our route once again differs quite a lot compared to the Google route. This time, however, our route is 300 m shorter than the Google route but takes 2 extra minutes to travel. The main difference is that our route takes a cycle path while the Google route is more of a direct route to the destination.

Safety Comparison: Fig. 7 shows the traffic conditions along our route and the Google Maps route for the time at which these screen shots were taken.

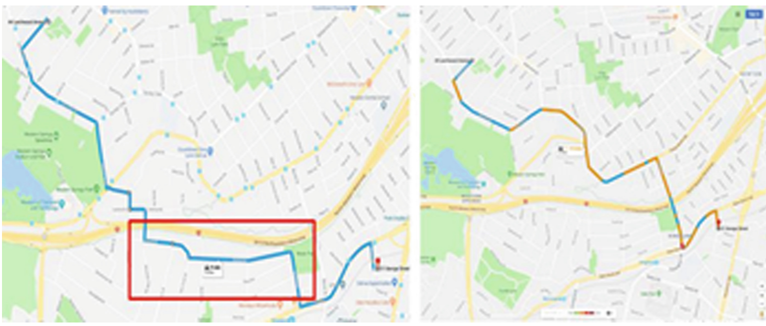

**Fig. 7.** Larchwood - George traffic comparison (Color figure online)

From Fig. 7, we can see unsafe congestion all along the Google Maps route (red and orange parts), our route avoids this congestion and takes the cycle path.

Resulting in an overall shorter & safer path, with a 2 min increase in travel time. The red box shows where this route differs from our actual cycling route. This is because our route takes the cycle path at this point and then cuts across a park which cannot be done in driving mode. This part of the route is considered very safe as there can be no unsafe traffic conditions on cycle paths or in parks. Our route can be seen to be much safer than Google's route as it avoids congestion and prioritizes the cycle path.

All of our results were directly compared with Google Maps as we felt that this is the best and the most popular current route planning application that exists for cyclists. From the results that were shown, our application plots routes that avoid unsafe traffic whereas Google Maps' routes are simply the shortest path. This proves that our research into Google Maps was accurate and it does not take the safety of users into account.

Because our routes avoid so much unsafe traffic, we can conclude that our algorithm is working as intended and provides a much safer path than Google Maps. In certain cases, the routes provided by our application are slightly longer than the Google routes, however, at the most our routes were no more than an extra five minutes of travel time. We feel that this tradeoff is worth it when looking at the extra safety that is provided by our application. The results show that our application takes traffic conditions into account and provides users with safer routes than any other current application.

### 7.2 User Surveys

In order to assess the functionality and user friendliness of our application we decided to carry our several user surveys. We wanted to see if cyclists who used our application felt that their safety was improved whilst travelling. The survey asked specific questions about the length of the routes and impact on their safety. It also inquired about the app in general such as its ease of use and any suggestions that they wanted to make about it. The surveys were handed out after exhibition day.

The survey results were quite promising. 80% of the users said that the app was easy to use and all participants said that the lengths of the routes were quite reasonable compared to other applications. 94% of users said that the routes provided by our application helped them to feel safer. A common suggestion that we encountered was a request for audio queues, many users felt that being able to hear directions whilst travelling will enhance their experience with the application. The survey results show that our solution successfully accomplishes our goal of improving cyclist safety through a route planning application. The suggestion of audio queues has been added to the future work section in response to the user survey.

# 8 Future Work

## 8.1 Algorithm Improvements

Whilst, the results and evaluation show that our algorithm can successfully route our users away from unsafe traffic conditions, we would like to work more on this algorithm in the future and improve it in any way we can.

The first way that we'd like to improve our algorithm is by integrating more data sources. HERE map data can be easily integrated with our current system as another data source. HERE maps provides another accurate real-time data source that we can use [1]. They are also a proven source of data and are widely used in conjunction with Garmin technologies. Integrating this with our current application can increase the accuracy of our results as it will provide us with more information about current traffic conditions. In the future we would also like to integrate Taxi GPS data with our application. Taxi data can provide us with more valuable traffic information. Also, if we acquired a big data set then we would be able to improve our prediction algorithm as our machine learning model will have much more data to work with.

Finally, we need to work further on our algorithm for exercise mode and on our prediction model. Both of these modes were not worked on as much as the commute mode and the weather functionality as we felt that these were the core parts of this paper. The implementation of exercise mode is entirely based on the algorithm for commute mode with minor tweaks, this can be changed in the future to have its own specific algorithm. We feel that having its own algorithm would significantly improve our exercise mode algorithm.

Our prediction model also needs to be changed to incorporate data from the cyclists who use the application. This could not be done earlier as the core safety aspects of the application had to be mostly complete as we did not want cyclists using an unfinished and unproven safety application.We only gave our application our for use to cyclists after exhibition day so users could respond to the surveys.

## 8.2 Quality of Life Changes

We would like to implement multiple quality of life upgrades to our application to improve usability and user experience. These changes will not affect our safety algorithm, instead they will just enhance the application and make it more appealing for cyclists to use.

Firstly, we would like add GPS tracking and audio direction queues for users whilst they travel the route. The GPS tracking will allow the application to know the users current location on the route. This tracking can then be used to provide helpful audio queues to the users. This way the user can just listen to the queues and understand which direction they need to go next instead of having to memorize the route beforehand or put their phone on a mount.

We feel it could potentially be useful for users if they had the option to look at a list of directions that they must follow on their route. This way, they can just look at a handy list of instructions and memorize them instead of trying to discern them from looking at the plotted route on the map.

Another quality of life change that we would like to implement in the future is stat tracking for users. In order to implement this we must first implement our login database. We intended on having a database for logins, but instead had to move that to future work as other parts of the application required more immediate attention. This database can be used to store exercise statistics and provide the user with useful information such as distances travelled, speeds travelled and calories burnt. It can also remember common trips that users take and suggest these trips to them when the app is opened.

## 9 Conclusions

In this paper, we proposed to develop a mobile route planning application that can improve cyclist safety. We used Google and Bing APIs to acquire route and traffic data for Auckland, New Zealand. Using this data, we provided a route planning application that can take traffic conditions into account to ensure that cyclists are not routed through unsafe areas. We also implemented an exercise mode so the users of our application can be routed safely when planning workout routines. Our machine learning algorithm can be used by users to plan their future trips and ensure they are travelling during safe times.

The results show that our application provides a vast improvement in safety compared to current cyclist route planning applications. We compared our application against Google Maps as we felt that this is the best application that cyclists have at the moment. The results show that our application takes unsafe traffic conditions into account and successfully routes users away from these unsafe areas whilst still providing an efficient route. Google Maps does not do this and only looks at the shortest path for users. Thus, our application has successfully addressed the problem and can be seen to improve cyclist safety.

In the future, we would like to firstly improve our algorithm by adding more data sources to improve accuracy. We also need to look further into our exercise mode and prediction algorithms as there is some room for improvement there. Finally we would like to add several quality of life improvements for our users in order to increase the usability of our application.

**Acknowledgement.** This work was supported in part by the New Zealand Marsden Fund under Grant No. 17-UOA-248, and the UoA FRDF fund under Grant No. 3714668.

## References

1. HERE: Here map data. https://www.here.com/en/products-services/map-content/here-map-data
2. Jain, V., Sharma, A., Subramanian, L.: Road traffic congestion in the developing world. In: Proceedings of the 2nd ACM Symposium on Computing for Development, p. 11. ACM (2012)

3. Machay, J.: How does google detect traffic congestion?. http://smallbusiness.chron.com/google-detect-traffic-congestion-49523.html
4. Auckland Transport: The auckland cycling account. https://at.govt.nz/cycling-walking/research-monitoring/
5. Auckland Transport: Traffic counts 2012 to 2018. https://at.govt.nz/about-us/reports-publications/traffic-counts/
6. Ministry of Transport: Ministry of transport - cyclist crash facts 2017. https://www.transport.govt.nz/resources/road-safety-resources/crashfacts/cyclistcrashfacts/
7. Weinstock, Z.: Flattest route. https://www.flattestroute.com/
8. OpenStreetMap Wiki: Open cycle map. https://wiki.openstreetmap.org/wiki/OpenCycleMap
9. Xu, L., Yue, Y., Li, Q.: Identifying urban traffic congestion pattern from historical floating car data. Procedia Soc. Behav. Sci. **96**, 2084–2095 (2013)