Context-aware Mobility Analytics and Trip Planning

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

Mohammad Saiedur Rahaman

M.Sc (Computer Science)

B.Sc (Computer Science and Engineering)

American International University Bangladesh (AIUB)

School of Science

College of Science, Engineering, and Health

RMIT University

Melbourne, Victoria, Australia

March, 2018

This work is dedicated to the secret behind my success

...my beloved wife ***Farzana Afrin***

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Mohammad Saiedur Rahaman

March 29, 2018

Acknowledgement

All praise and thanks to my God for granting me the opportunity to undertake this research and giving me the strength, knowledge, and ability to persevere and complete satisfactorily.

I am greatly indebted to my supervisors A/Prof. Margaret Hamilton and Dr. Flora Salim for their guidance, support, encouragement, constructive criticism and continuous effort to improve my skills throughout this thesis. I would like to thank Dr. Yongli Ren and Dr. Yi Mei for their valuable mentoring. I thank my guru Dr. Ashfaqur Rahman for his important suggestions at the beginning of my candidature. I acknowledge my colleagues and management of AIUB and my teachers of Dhaka Residential Model College.

I remember the sacrifice of my parents Sharif Hossain and Gulara Banu who always provided me courage and prayed for my success and good health. I acknowledge the prayer and appreciation of my in-laws Habibur Rahman and Mansura Habib. I sincerely thank my brothers Touhidour Rahman, Wahidur Rahman and Tanzilur Rahman for their supports. My gratitude to my wife Farzana Afrin for her unfailing support and care. I would like to express my thanks to our daughter Rida Safeerah Rahman for being such a good girl always cheering me up.

Thanks to my friends Mahfuz and Azim for their extraordinary support on my arrival at Melbourne. Life in Melbourne would have been far less interesting and very different without Himel, Saif, Tasin, Halim, Debajyoti, Ripon and Zakaria. My appreciation to Tasnim Nasrin, Nadia Hasin, and Shahnaz Pervin for their exceptional care to my family. I thank the members of CRUISE_oldies Irvan, Jonathan, Amin, Wei, Hui, and Rumi who have provided friendship and support, and with whom I have shared laughter, frustration and companionship.

I would like to acknowledge the Australian Research Council (ARC) and RMIT University for providing me scholarships, resources and environment to support my research.

Credits

Portions of the materials used in this thesis have previously appeared or under consideration in the following scientific publications:

- M.S. Rahaman, Y. Mei, M. Hamilton, and F.D. Salim, CAPRA: A Contour-based Accessible Path Routing Algorithm. *In: Information Sciences*, Volume: 385, pp. 157–173, December 2016. **(Impact Factor: 4.83, SJR: Q1) - Chapter 4**

- M.S. Rahaman, M. Hamilton, and F.D. Salim, Predicting Imbalanced Taxi and Passenger Queue Contexts in Airport. *In: Proceedings of the* 21^{st} *Pacific Asia Conference on Information Systems (PACIS 2017)*, pp. 172, Langkawi, Malaysia, July 2017. **(CORE Rank: A) - Chapter 2**

- M.S. Rahaman, M. Hamilton, and F.D. Salim, Queue Context Prediction Using Taxi Driver Intelligence. *In: Proceedings of the* 9^{th} *International Conference on Knowledge Capture (K-CAP 2017)*, Article no. 35, Austin, Texas, United States, December 2017. **(CORE Rank: A) - Chapter 3**

- M.S. Rahaman, M. Hamilton, and F.D. Salim, CoAcT: A Framework for Context-Aware Trip Planning Using Active Transport. *In: Proceedings of the* 16^{th} *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Athens, Greece, March 2018. **- Chapter 4**

- M.S. Rahaman, M. Hamilton, and F.D. Salim, Using Big Spatial Data for Planning User Mobility. *In: Encyclopedia of Big Data Technologies*, pp. 1–6, February 2018. **- Chapter 1**

- M.S. Rahaman, Y. Ren, M. Hamilton, and F.D. Salim, Neighborhood Identification from Heterogeneous Contextual Features for Taxi Driver Queue Wait Time Prediction at Airports, 2018 (Under review). **- Chapter 3**

This research was supported by the Australian Research Council (ARC) through the provision of a linkage grant (project number LP120200305) for the project titled *"Integrated Smart Airport Services"* partnered with ARUP. Participations and travels to the conferences were supported by RMIT School of Graduate Research and RMIT School of Science in the provisions of conference travel, research support, and publication grants.

Contents

List of Figures

List of Tables

Abstract

The study of user mobility is to understand and analyse the movement of individuals in the spatial and temporal domains. Mobility analytics and trip planning are two vital components of user mobility that facilitate the end users with easy to access navigational support through the urban spaces and beyond. Mobility context describes the situational factors that can influence user mobility decisions. The context-awareness in mobility analytics and trip planning enables a wide range of end users to make effective mobility decisions. With the ubiquity of urban sensing technologies, various situational factors related to user mobility decisions can now be collected at low cost and effort. This huge volume of data collected from heterogeneous data sources can facilitate context-aware mobility analytics and trip planning through intelligent analysis of mobility contexts, mobility context prediction, mobility context representation and integration considering different user perspectives. In each chapter of this thesis such issues are addressed through the development of case-specific solutions and real-world deployments.

Mobility analytics include prediction and analysis of many diverse mobility contexts. In this thesis, we present several real-world user mobility scenarios to conduct intelligent contextual analysis leveraging existing statistical methods. The factors related to user mobility decisions are collected and fused from various publicly available open datasets. We also provide future prediction of important mobility contexts which can be utilized for mobility decision making. The performance of context prediction tasks can be affected by the imbalance in context distribution. Another aspect of context prediction is that the knowledge from domain experts can enhance the prediction performance however, it is very difficult to infer and incorporate into mobility analytics applications. We present a number of data-driven solutions aiming to address the imbalanced context distribution and domain knowledge incorporation problems

for mobility context prediction. Given an imbalanced dataset, we design and implement a framework for context prediction leveraging existing data mining and sampling techniques. Furthermore, we propose a technique for incorporating domain knowledge in feature weight computation to enhance the task of mobility context prediction.

In this thesis, we address key issues related to trip planning. Mobility context inference is a challenging problem in many real-world trip planning scenarios. We introduce a framework that can fuse contextual information captured from heterogeneous data sources to infer mobility contexts. In this work, we utilize public datasets to infer mobility contexts and compute trip plans. We propose graph based context representation and query based adaptation techniques on top of the existing methods to facilitate trip planning tasks. The effectiveness of trip plans relies on the efficient integration of mobility contexts considering different user perspectives. Given a contextual graph, we introduce a framework that can handle multiple user perspectives concurrently to compute and recommend trip plans to the end user.

This thesis contains efficient techniques that can be employed in the area of urban mobility especially, context-aware mobility analytics and trip planning. This research is built on top of the existing predictive analytics and trip planning techniques to solve problems of contextual analysis, prediction, context representation and integration in trip planning for real-world scenarios. The contributions of this research enable data-driven decision support for traveling smarter through urban spaces and beyond.

Chapter 1

Introduction

The users of urban spaces need to travel from one place to another for various reasons such as work, leisure, and freight distribution. The study of user mobility describes this movement of individuals which consists of sequences of trips using different modes of transports [2]. As cities are becoming more and more complex day by day [3], the concerns over providing decision support for user mobility are also growing [4, 5, 6, 7]. In response to growing concerns over user mobility in many modern cities, we have witnessed worldwide growth of urban sensing infrastructures [8] enabling us to capture user movement and various factors related to user mobility in the cityscape [9, 10]. *Context-awareness* is the key to deliver effective user mobility decision support as it allows us to analyse and predict various situational factors related to user mobility and to compute tailored mobility solutions based on different user perspectives [11, 12, 13]. Furthermore, it contributes to the application domain of urban computing [14, 15, 16] utilizing big spatial and temporal data [17, 18].

Mobility analytics arise from conducting intelligent analysis and prediction on associated spatio-temporal factors [19, 20, 21] and diverse mobility contexts [22] that influence the users' mobility decisions. The context-aware mobility contexts can better describe users situations for making mobility decisions [16, 23]. Therefore, context-aware mobility analytics enables the provision of intelligent analysis on mobility contexts [24] considering different user perspectives. The success of many applications such as transport management and location recommendation requires the discovery of valuable knowledge through extensive analysis of related

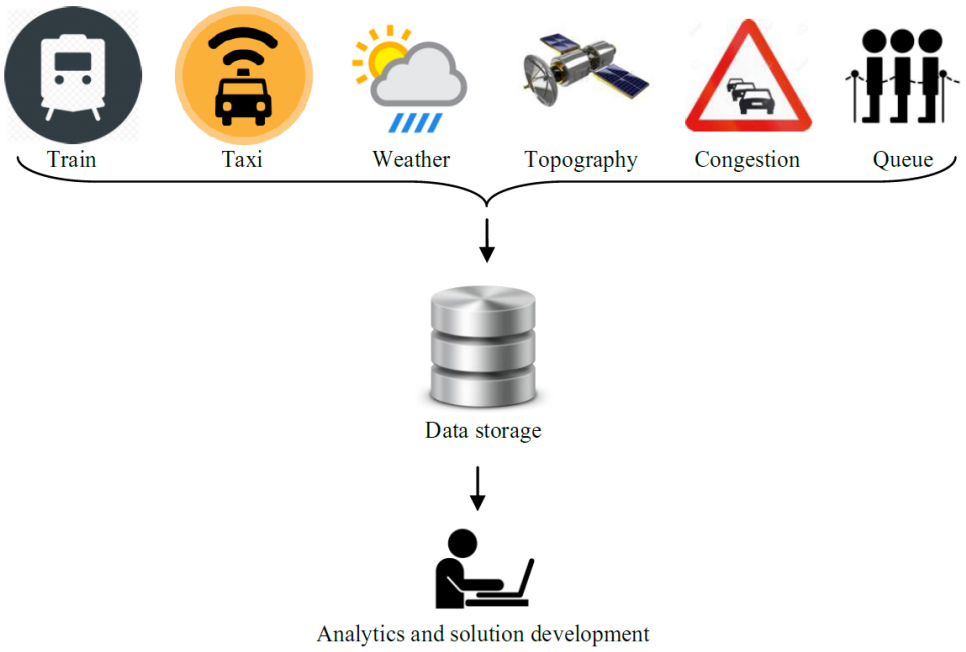


Figure 1.1: Ubiquitous Data Sources Facilitating User Mobility Solutions.

factors [25, 26]. Figure 1.1 depicts various heterogeneous data sources from where important mobility-associated factors for user mobility decisions can be collected and stored to be used for context-aware user mobility analytics and mobility solution development. These data can aid knowledge discovery which can further be used in future mobility context prediction. The prediction of future mobility contexts [20, 25, 27] is necessary in many application areas for providing seamless mobility decision support given the fact that the user mobility contexts can change over time and situation. For example, an airport can be regarded as the first and last impression of a city. Since a longer passenger wait time for a taxi ride can diminish the satisfaction rating of an airport [26], the authorities try hard to maintain a higher customer satisfaction rating by providing various mobility services such as easy and comfortable airport transfer to the city using taxicabs. However, the demand-supply equilibrium of taxis is highly dependent on the taxi drivers' decisions to make airport trips. The ubiquitous data can help with managing the mobility of airport users by detecting different mobility contexts (i.e. situations of the concurrent queues related to passengers and taxis) [28, 26]. The intelligent analysis

and prediction [26] of different mobility contexts can help with making mobility decisions for airport passengers and taxis [29] at different times of the day. The mobility context prediction also can help airport authorities to alleviate possible chaos due to lack of taxis at the airport by ensuring the demand-supply equilibrium of taxis.

Trip planning is an important component of user mobility which provides navigational support to users by integrating trips using single or multiple transport modes [30, 31, 32, 33] such as public transport, private car, taxi, Uber and active transport. Context-awareness in trip planning provides the opportunity to integrate trips based on different user perspectives [34, 35] of mobility contexts. Moreover, an effective trip plan must have the capability to consider multiple mobility contexts simultaneously during the computation of trip plans [16]. For example, the list of recommended plans in active transport (i.e. walking) trip planning may include mobility contexts such as distance and steepness of the routes simultaneously. The effectiveness of recommended trip plans varies along with the variation in user perspectives of these mobility contexts. The inferred contextual information collected from ubiquitous data sources has the potential to aid user-specific trip planning [34]. Furthermore, the growing need for efficient urban intelligence in user mobility has led us to conduct context-aware mobility analytics and trip planning.

1.1 User Mobility and Motivating Scenario

User mobility is defined as the movement of individuals between two meaningful places [2]. For example, the user mobility of 'home to office' and 'office to home' can be termed as user mobility since both home and office are two meaningful places [22]. User mobility can be decomposed into small segments called 'trips'. Essentially, a sequence of trips connecting two arbitrary places defines user mobility between two meaningful places [2]. Each trip is accomplished using a mode of transport such as bike, car, taxi, public transport or walking. Let us consider an example of user mobility between two meaningful places p_1 and p_5 which consists of a sequence of inter-connected trips using different modes of transport. The selection of different transport modes by a user can result in different route recommendations for user mobility between p_1 and p_5. Three different routes from p_1 to p_5 are marked with three different line

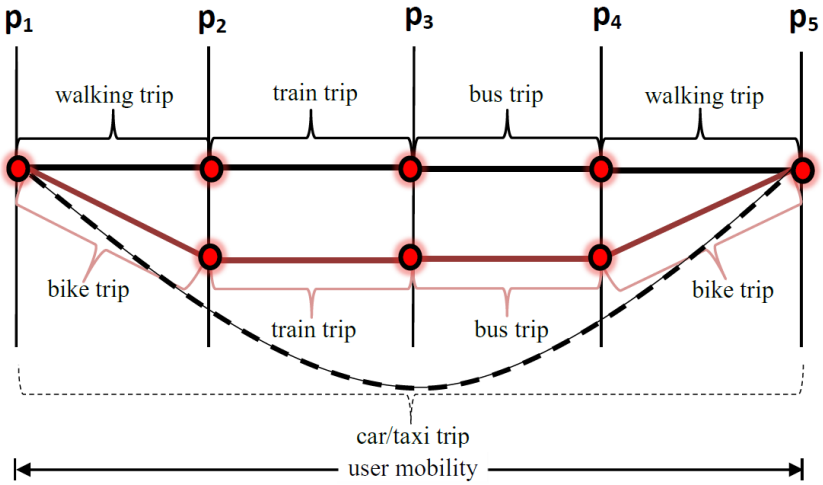


Figure 1.2: Example of User Mobility Options Connecting Several Trips.

styles as shown in Figure 1.2. Each of these routes has a different combination of transport modes. The first two routes (i.e. black and brick red) integrate trips with public transport modes and the corresponding sequences of transport modes from p_1 to p_5 are (walk, train, bus, walk) and (bike, train, bus, bike), respectively. These two routes have some segments in common, as seen between p_2 and p_4. The third route (black dashed) employs only car or taxicab to travel between p_1 and p_5.

Trip planner systems facilitate user mobility by selecting and integrating several trips in the trip plans [34, 36, 37]. Usually, a trip planner integrates trips on the basis of user preference chosen from a predefined list of options which may include shortest time, shortest distance, number of transport mode changes and the preferred mode of transport [16]. However, the consideration of different user perspectives of mobility contexts during trip planning can significantly improve the effectiveness of recommended trip plans. The reason is because different user perspectives of mobility contexts in trip planning can better describe the user situation. This enables a wider range of users to make appropriate mobility decisions. However, the diversity of user perspectives provides a challenge to the current trip planning systems [26, 16, 13, 38].

Let us consider two scenarios to discuss two different users and their perspectives on mobility contexts. We discuss how mobility contexts describe the situations of users and influence the user mobility decisions. The first user is a person with limited mobility in a wheelchair,

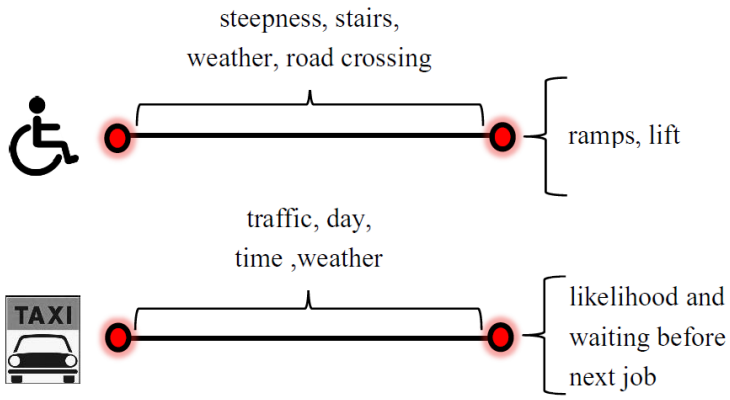


Figure 1.3: Example of Diverse Situational Factors and User Perspectives of Contexts.

and the second person is a taxicab driver. We can see from Figure 1.3 that the trip planning for the first person may consider mobility contexts such as distance and accessibility of the route. The accessibility of the route can be quantified by many external associated factors such as steepness along the route [16], weather, and road crossings. This user may also prefer information about other associated factors including ramps and lifts at the trip end location to ensure comfortable mobility. The inference of accessibility from the associated factors is vital for this user's mobility decisions. However, the inference and integration of mobility contexts (i.e. accessibility, distance) in trip planning is challenging due to the diversity in user perspectives about mobility contexts. Intelligent data fusion and computation can be applied to address this challenge. In contrast, Figure 1.3 shows the trip plan that a taxicab driver may consider for information about mobility contexts such as the likelihood and wait time for getting a passenger. The prediction based techniques can be used to infer these mobility contexts. Moreover, these mobility contexts can be influenced by many associated factors including traffic congestion, weather [39], hotness of the passenger pick-up spots [40] and the spatio-temporal trip lucrativeness [41]. Therefore, predicting the mobility contexts in the presence of many associated factors is an important task as they contain the potential to provide better decision making about user trips.

Context-awareness in mobility analytics [42, 43] and trip planning [44, 45, 46, 16, 13, 38] has gained significant attention in recent years especially in the field of urban computing. It requires

dealing with large spatio-temporal complex datasets [47] which opens many research challenges in the field to intelligent contextual analysis and mobility context prediction, representation and integration in trip planning. The main objective of this work is to conduct intelligent analysis and prediction of mobility contexts from large datasets. We are interested in designing a model that can be used to process imbalanced mobility context datasets [26]. We examine the process of domain knowledge incorporation [48, 49, 50] in context prediction tasks. We also aim to represent and integrate mobility contexts based on user perspectives during trip planning. Considering the diversity of user perspectives, we are interested in designing a conceptual framework that can infer and represent mobility contexts to be integrated in trip planning algorithms. The objective of our study also includes testing and evaluating our developed models and techniques with real-world user mobility scenarios including airport ground transport management [51] and active transport trip planning [16, 52].

1.2 Research Challenges

In this thesis, we aim to address key challenges related to context-aware user mobility analytics and trip planning. The rapid growth of urban sensing tools and techniques facilitates the collection of diverse spatio-temporal factors associated to user mobility. This has the potential to provide valuable information for making user mobility decisions. The ubiquity of these associated factors and the heterogeneous nature of the data sources has raised various research challenges [53] in areas such as the analysis of mobility contexts and their prediction, representation and integration in user specific trip plans.

For intelligent analysis of mobility contexts, cross-domain data fusion is required since it can better represent the effects of mobility-associated factors for user mobility management and planning. However, it is challenging due to the diversity of applications [54]. In user mobility applications, the mobility contexts and user perspectives of mobility contexts are also diverse in nature. The context-aware mobility analytics consider different user perspectives concurrently to provide intelligent analysis and context prediction. For example, the airport ground transport managers send requisition to the taxi service providers by estimating the taxi demand in a future time window. This is to ensure the seamless passenger mobility from

the airport using taxis. However, the flawed manual estimation can lead to enormous queue waiting times for passengers as well as taxis.

It is challenging to deal with such scenarios since it requires some key related issues to be addressed. Different machine learning techniques can be used to predict future mobility contexts (i.e. queue situations). The machine learning techniques are trained using historical mobility context data for the prediction task. However, the predictive analytics suffer from mobility context imbalance problem because some mobility contexts occur very infrequently compared with others making the predictive analytics more challenging. Many researchers have addressed this issue in different application areas [55] but they have not been adapted to user mobility analytics where many parties with different perspectives can be involved in a common user mobility scenario. Therefore, our aim is to design a common framework for user mobility analytics which can provide intelligent analysis and prediction for enhanced user mobility management and planning considering different user perspectives.

Some researchers have shown that the incorporation of knowledge from experts can improve the prediction accuracy in many application areas [48, 49]. However, it is difficult in user mobility scenario due to the presence of diverse mobility-associated factors. An investigation on how expert knowledge can be inferred and incorporated into this problem domain is required. We aim to investigate and model multiple associated factors for user mobility and to design an approach to incorporate the inferred expert-like knowledge in mobility context prediction.

Another important component of user mobility is to provide effective trip plans to support user navigation. However, the effectiveness of trip plans depends on appropriate consideration of user mobility contexts. This is a challenging issue to deal with since there are many diverse user perspectives. Several algorithmic approaches have been proposed to handle user-defined contextual preferences but they mainly require a user to select a preference from pre-defined list of preferences and hence cannot serve the purpose where the aim is to consider multiple mobility contexts from a single user and different user perspectives of contexts during trip planning. Trip planning also suffers from context sparsity requiring specific contexts to be inferred before designing trip planning algorithms. The inferred contexts need to be represented in such a manner to ensure efficiency of the trip planning task. Our aim is to design a model

that can infer and represent sparse mobility contexts and provide context-aware trip planning considering multiple mobility contexts of a user and different user perspectives of a specific mobility context.

In summary, the core challenges in effective context-aware mobility analytics and trip planning are as follows:

- Fusion and representation of collected large mobility-associated factors from heterogeneous data sources.
- Providing intelligent analysis and prediction on mobility contexts.
- Incorporating expert knowledge for enhanced mobility context prediction.
- Inferring sparse mobility contexts from heterogeneous data sources.
- Representing inferred mobility contexts for efficient computation of trip plans.
- Handling user trip planning queries by considering multiple mobility contexts simultaneously and different user perspectives of a specific mobility context.

1.3 Research Questions

In order to overcome the research challenges, we define the following research questions (RQs) to achieve context-aware mobility analytics and trip planning.

RQ-1. How to provide context-aware mobility analytics from heterogeneous large datasets?

This research question addresses the challenges related to data fusion, representation and context imbalance for intelligent analysis and prediction of mobility contexts. The significance of dealing with these issues is primarily to ensure that we can handle various datasets of mobility-associated factors collected from heterogeneous data sources to provide context-aware mobility analytics using a common approach.

RQ-2. How to model different associated factors to improve the mobility context prediction task?

This research question is designed to address the challenges related to modelling the associated factors to enhance the mobility context prediction tasks in RQ-1. Specifically, the aim of this research question is to investigate diverse associated factors and incorporate knowledge from domain experts during mobility context prediction in different scenarios.

RQ-3. How to integrate multiple mobility contexts in context-aware trip planning?

The solutions from RQ-1 and RQ-2 can be used for prediction based mobility context inference. However, there are some mobility contexts which cannot be inferred using a prediction based method. This research question addresses the issues regarding context sparsity and multiple mobility context integration in trip planning considering different user perspectives. The framework presented here provides fusion based inference of sparse mobility contexts. By utilizing the inferred mobility contexts, a technique is developed that can consider different user perspectives of a specific mobility context and integrate multiple mobility contexts during the computation of context-aware trips.

1.4 Research Contributions

To address the aforementioned research questions, the contributions of this thesis are as follows:

1. **Context-aware mobility analytics using heterogeneous data**

   To provide context-aware mobility analytics, large heterogeneous datasets of user mobility-associated factors are required to be fused and analyzed. These datasets are also a great source for inferring different mobility contexts. The prediction of future mobility contexts is very important for better user mobility management and planning. In this thesis, we take airport ground transport management as a case scenario of user mobility where different parties including passengers, taxi drivers, and ground transport managers are

involved to ensure seamless airport ground transport operation. Another reason to choose this scenario is that the user mobility situation at the airport is influenced by a diverse set of associated factors and hence is a complex issue faced by almost every busy airport in the world.

We use large publicly available datasets of associated factors to infer various mobility contexts. We present a framework to predict future mobility contexts with imbalanced distribution leveraging existing data mining and sampling techniques. We also conduct intelligent analysis of mobility contexts to provide useful information to the parties involved in our scenario.

2. **Modelling associated factors for mobility context prediction**

   In relation to the first contribution, we further investigate various mobility-associated factors and present a technique to model factors for mobility context prediction. Since research has shown that the domain-specific knowledge can improve the prediction performance, we develop and present a technique to extract and incorporate domain knowledge in terms of feature weights for mobility context prediction. Specifically, we explore similarity based data mining techniques (i.e. neighborhood based methods) to incorporate domain knowledge during the mobility context prediction task. This technique is useful for making mobility decisions.

3. **Inferring and integrating mobility contexts in context-aware trip planning**

   There are various user perspectives of a specific mobility context that need to be considered during trip planning using active transport (i.e. walking, biking, wheeling). Some mobility contexts are sparse in nature and the development of new technique is required to infer them. Moreover, existing trip planners are not built to consider different user perspectives for the same specific mobility context and hence cannot serve the purpose. In this thesis, new algorithms are designed to tackle the above challenges. The new trip planner presented here can satisfy the perspectives of a wide range of users using active transport and can be extended to trip planning using other transport modes.

1.5 Thesis Organization

The organization of this thesis includes the following chapters as below:

- **Chapter 2: Context-aware mobility analytics using heterogeneous data.**

  A framework for mobility context prediction is presented in this chapter illustrating the problem of predicting imbalanced taxi and passenger queue contexts at the airport. The solution presented here is in relation to *RQ-1*. The technique to infer mobility contexts that are useful for airport transport managers, taxi drivers and passengers is also presented. A number of intelligent analyses of mobility contexts are also highlighted.

  Copyright/ credit/ reuse notice: The contents of this chapter have been taken and revised as needed from our paper published as:

  M.S. Rahaman, M. Hamilton, and F.D. Salim, Predicting Imbalanced Taxi and Passenger Queue Contexts in Airport, In: proceedings of the 21^{st} Pacific Asia Conference on Information Systems, Langkawi, Malaysia, 16–20 July 2017, PACIS 2017 Proceedings. 172.

  DOI: http://aisel.aisnet.org/pacis2017/172

- **Chapter 3: Modelling associated factors for mobility context prediction.**

  In relation to RQ-2, this chapter presents the modelling of mobility-associated factors for mobility context prediction. In addition to the scenario demonstrated in Chapter 2, we introduce another real-world scenario where the mobility of taxi drivers depends on the queue wait times before a passenger pickup from airport. We predict different situations of taxi and passenger queues along with the queue wait times for taxi drivers by introducing a domain knowledge incorporation technique as the means of feature weighting scores. Specifically, the similarity based data mining techniques are explored in this chapter for incorporating domain knowledge for prediction of these mobility contexts.

  Copyright/ credit/ reuse notice: The contents of this chapter were taken and revised as needed from two of our papers published as:

M.S. Rahaman, M. Hamilton, and F.D. Salim, Queue Context Prediction Using Taxi Driver Intelligence, In: proceedings 9^{th} International Conference on Knowledge Capture, 35, Austin, Texas, United States, 4–6 December 2017.

DOI: http://dx.doi.org/10.1145/3148011.3154474

M.S. Rahaman, Y. Ren, M. Hamilton, and F.D. Salim, Neighborhood Identification from Heterogeneous Contextual Features for Taxi Driver Queue Wait Time Prediction at Airports, 2018 (Under review).

- **Chapter 4: Inferring and integrating mobility contexts in trip planning.**

  In this chapter, new algorithms are presented in relation to the challenges stated in RQ-3. The solutions associated with sparse mobility context inference and integration of multiple mobility contexts considering different user perspectives during context-aware trip planning are introduced. A real-world case study is illustrated where we consider active transport trip planning. The developed techniques are deployed in several locations around the world.

  Copyright/ credit/ reuse notice: The contents of this chapter has taken and revised as needed from two papers published as:

  M.S. Rahaman, Y. Mei, M. Hamilton, and F.D. Salim, CAPRA: A Contour-based Accessible Path Routing Algorithm, In: Information Sciences, Volume: 385, pp. 157–173, December 2016.

  DOI: https://doi.org/10.1016/j.ins.2016.12.041

  M.S. Rahaman, M. Hamilton, and F.D. Salim, CoAcT: A Framework for Context-Aware Trip Planning Using Active Transport, In: Proceedings of the 16^{th} IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Athens, Greece, March 2018.

- **Chapter 5: Conclusion.**

  This chapter concludes the thesis by summarizing the main contributions, key findings and limitations of the proposed methods. In addition, the significance of this research and potential future directions are also discussed.

In short, the succeeding core chapters (Chapter 2–4) of this thesis contribute to a number of key research questions on context-aware mobility analytics and trip planning with a concluding summary and future research directions. Note that the core chapters appear in a self-contained and self-explanatory manner which includes real-world scenarios. Different contexts require different types of reasoning. Therefore, the relevant contexts and content including discussions on related work, developed models, datasets, experimental setups and evaluation metrics are presented in each of these chapters separately.

Chapter 2

Context-aware Mobility Analytics Using Heterogeneous Data

As discussed in Chapter 1, intelligent analysis and prediction of mobility context are important for making mobility decisions. The proliferation of urban sensing technologies facilitates the collection of large volumes of heterogeneous data related to user mobility. This can provide mobility management and planning considering different user perspectives through intelligent analytics. However, it requires cross-domain data fusion to better represent the effects of mobility-associated factors. To predict different mobility contexts, prediction algorithms are employed and trained using historical mobility context data. This requires dealing with imbalanced context distribution problem since some mobility contexts are very infrequently occurring as compared to the others. Many techniques have been reviewed to address this issue in different application areas. In relation to *research question* (RQ-1), this chapter presents a framework that integrates the solutions of the above issues to provide mobility analytics.

We consider the taxi-passenger queue context prediction scenario at the airport. The taxi and passenger queue contexts indicate various situations of queues related to taxis and passengers (i.e. taxis are waiting for passengers, passengers are waiting for taxis, both are waiting for each other, none is waiting). The queue contexts are good examples of mobility contexts and the prediction of these queue contexts in a future time is very important for better airport ground transport operations by considering different user perspectives including taxi

drivers and air passengers. We choose this scenario to demonstrate that the queue contexts are influenced by many diverse mobility-associated factors and to explain our developed a framework to predict imbalanced queue contexts.

We begin with investigating different mobility-associated factors including time, frequency of taxi trips, passenger arrivals and weather for queue context prediction. We predict the queue contexts by following the detailed approach of our developed framework. Specifically, we generate a queue context dataset by fusing three real-world datasets including taxi trip logs, passenger arrivals and processing times, and weather condition at a major international airport. The following sections of this chapter show a number of experimental results and analyses to demonstrate our queue context prediction framework.

2.1 Motivation and Contribution

Taxis are considered to be the most convenient transportation option for passenger mobility between the airport and the city. Hence the management of passenger and taxi queues plays an important role in the running of an airport. Either taxis or passengers can experience unexpected wait times for each other causing disruption and chaos for passengers and taxi drivers at the airport if either queue becomes too long. A queue context describes who is waiting in a given time window (i.e. taxi driver or passenger or both or none) [28]. Therefore, predicting different queue contexts at different times of the day can help to improve the airport satisfaction rating by providing timely taxi and passenger mobility management. This can help the taxi drivers within the airport vicinity by providing timely information about the passenger queues in different passenger terminals as well.

The queue context prediction at the airport is challenging for many reasons. First, the formation nature of both taxi and passenger queues is dynamic and influenced by factors such as flight arrivals, passenger processing, frequency of taxi trips and weather conditions. Second, the queue contexts are imbalanced i.e. some of the queue contexts occur far less frequently than other queue contexts. The problem of queue prediction in terms of wait time has been studied in other applications by employing various machine learning algorithms [1, 56, 57]. However, the existing approaches are not sufficient to provide prediction and corresponding

analyses of imbalanced queue contexts since there are several stakeholders involved with the two simultaneously occurring queues at the airport. This is due to the fact that a single technique serves the purpose of one party at a time. Therefore, the development of a common framework is required to address all these integrated issues.

In this chapter, we develop a framework that provides step by step procedures to predict the taxi and passenger queue contexts and provides contextual analysis on the prediction outcomes considering different user perspectives. We also extract various factors and patterns from the associated datasets fused for queue context prediction. Specifically, our framework employs a suite of sampling and machine learning techniques to predict the imbalanced queue contexts related to taxi and passenger queues. Our experiments with the dataset generated for the JFK (John F. Kennedy) airport demonstrates the reasonableness of our developed framework. The reason to choose the JFK airport for our experiment is that it is one of the busy airports in the U.S. The taxi rank called the central taxi holding (CTH) area at the JFK is far away from the passenger terminals. Any taxis planning for an airport passenger pickup job must join a waiting queue at the CTH area before picking a passenger up. The taxi dispatch managers at the JFK are responsible for dispatching taxis from this CTH based on the demand at several passenger terminals [51]. Another airport in New York City is the LaGuardia airport. However, this airport is not covered by the border control facility and hence no passenger arrival and wait time information is available. Therefore, we take the JFK international airport as our case location to prepare our dataset of taxi-passenger queue contexts at the airport. The contributions of this chapter include the followings:

- Fusion of three real-world heterogeneous contextual datasets for the research of imbalanced mobility context prediction (i.e. taxi and passenger queue context prediction): taxi trip data, airport passenger wait time data and weather condition data.
- Extraction and analysis of heterogeneous mobility associated factors and patterns from the airport queue context dataset.
- Development of a framework to provide step by step procedures for queue context prediction and analysis.

Table 2.1: Description of Four Queue Contexts at Airport

Queue Contexts	Description of Queue Contexts
TQ	Only taxis are waiting in the queue (taxi rank) for passengers.
PQ	Only passengers are waiting in the queue (terminal) for taxis.
TPQ	Both taxis and passengers are concurrently waiting in their respective queues for each other.
NoQ	No taxis or passengers are waiting in their respective queues.

2.2 Related Work

The airport provides the impression of a city and the air-passengers' satisfaction about an airport depend on the availability of taxis at the terminals. Several recent works analyze the demand-supply equilibrium of airport taxicabs [58, 59]. However, both taxi drivers and the passengers at the airports can experience long wait times [60, 61] in their respective queues for many reasons such as flawed manual taxi demand estimation. Moreover, the taxi drivers' decisions about making future airport trips are influenced by the current and speculated situations of these queues [29]. Hence it is important to analyze and predict the different situations of both taxi and passenger queues at the airport for providing better mobility management of taxis and passengers.

The queue contexts describe the existence of any one of the four states of taxi and passenger queues [28] which include 'taxi queue only' (TQ), 'passenger queue only' (PQ), 'both taxi and passenger queues' (TPQ) and 'no queue' (NoQ). Table 2.1, describes the four queue contexts that are observed in the airport. The airport trips are lucrative for taxi drivers. However, the presence of too many taxis exceeds the actual demand can cause long waiting times for taxi drivers at the airport taxi rank. On the other hand, the lack of taxis at the taxi rank can cause long passenger queues waiting for taxis.

Many researchers have investigated citywide taxi trips with a view to providing recommendations for the taxi drivers and passengers. The recent works focus on finding profitable taxi cruising routes for passenger pickup [62, 63, 64]. A profitable taxicab parking location analysis

framework to help taxi drivers was proposed by [65] which utilizes the knowledge of passengers' mobility patterns and taxi drivers' pick-up behaviors inferred from taxi GPS trajectories. Taxi GPS trajectory data have been widely used to predict city-wide traffic conditions and to analyze the behavior and movement patterns of the population [66, 67, 68, 69, 70, 71, 72]. The frequency of city-wide taxi passenger pickup is predicted using machine learning techniques and associated factors are investigated in [73]. A recommendation system for finding vacant taxis and passengers is presented by [74]. A spatio-temporal factor analysis model to find the best passenger pickup location is presented in [75]. The four factors considered by this model include distance, wait time, fare, and cluster transition probability. However, airports are often located in a designated area and supported by various transport modes. There is another research project that aims to estimate the passenger waiting time before a taxi ride by observing the behavior of vacant taxis [76]. A real-time taxi trip information system is proposed in [77] where passengers are able to know their estimated trip time and fare before their trip. A technique presented in [78] recommends pickup points to avail a taxi ride. The pickup points within a specified distance are ranked based on potential wait time. A passenger wait time prediction model from historical taxi trajectories is presented in [79]. The model is built by considering the arrival and departure events of taxis at a given location. A context aware system for spatio-temporal traffic prediction in different road segments is proposed in [80]. A system for monitoring taxis at a pickup location by mining GPS trajectories is presented in [81]. It also provides real time information about any taxi stand and surrounding traffic condition via RESTful web services. A taxi and passenger queue context detection framework is presented by [28] which utilizes taxi traces and taxis' mobile data terminal logs in Singapore. However, these techniques are effective for citywide taxi operations and cannot be applied directly to predict airport taxi or passenger queue contexts since the regulations for airport taxi operation is different than the citywide taxi operation. In the airport, any taxi must join the airport taxi rank queue and wait to be called by the ground transport manager to pickup a passenger from the passenger terminal.

There are very few research papers that deal with managing airport taxi operations. In [29], logistic regression is used to model the taxi drivers next pickup decision for an airport

trip. The model uses binary decisions of airport pick-up or cruising for customers at the end of each trip. Research projects related to the airport taxi operations mainly focus on taxi queue modeling [58, 82] to direct taxi drivers to the terminal with passenger queues. Airport taxi and passenger queue context prediction is different and challenging since queues at airports form dynamically. Also, the airport taxi and passenger queue context data suffers from the imbalanced queue context problem which compromises the learning and prediction performance of prediction algorithms. Different sampling techniques can be used to deal with this issue [55]. In the rest of this chapter, we introduce the formal problem definition and develop a framework that illustrates the step by step procedures for queue context prediction and analysis considering user perspectives.

2.3 The Queue Context Prediction Framework

2.3.1 Problem Definition

Let, $C_Q = \{TQ, PQ, TPQ, NoQ\}$ be the set of four possible queue contexts corresponding to an hourly time window where TQ, PQ, TPQ, and NoQ indicate 'taxi queue only', 'passenger queue only', 'both taxi and passenger queues' and 'no queue' queue contexts respectively. Let, each sample instance (time slot), x in the training data be described by a d-dimensional vector of attributes R^d and a queue context label $c(x) \in C_Q$. Therefore, the instance x can be written as $\langle a_1(x), a_2(x), ..., a_d(x), c(x)\rangle$ where a_i is the i^{th} attribute of x and $i = 1$ to d. If $f(.)$ is the queue context prediction function then for a set of d-features corresponding to a query time slot x_q, $f(.)$ predicts $\hat{c}(x_q)$ such as $f(x_q) : R^d \rightarrow \hat{c}(x_q)$ where $\hat{c}(x_q)$ is the predicted queue context of the query time slot x_q.

In this section, we present our developed queue context prediction framework. The framework provides step by step procedures to be followed for heterogeneous data fusion, queue context inference, and queue context prediction. The framework can also be used for intelligent contextual analysis. The queue context prediction framework utilizes a set of classifier and sampling techniques. To satisfy different user perspectives, the framework identifies the best pairs of sampling techniques and prediction algorithms. There are three main compo-

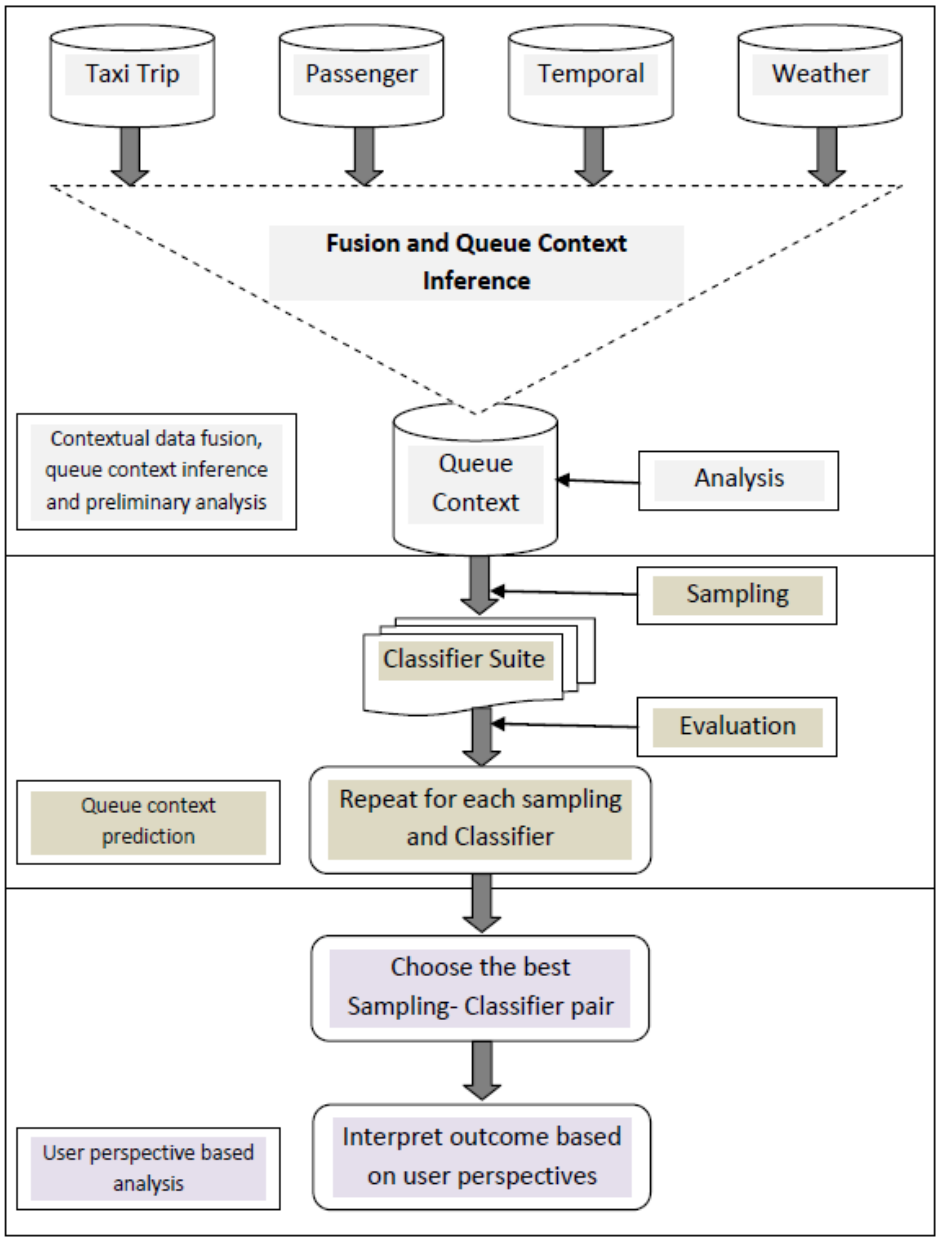


Figure 2.1: Queue Context Prediction Framework

nents of the queue context prediction framework known as i) data fusion, context inference and preliminary analysis ii) queue context prediction and iii) user perspective based analysis. The components of the framework are illustrated in Figure 2.1.

2.3.2 Contextual Data Fusion, Context Inference and Preliminary Analysis

In this module, different contextual data such as time, taxi trips, passenger arrivals and weather conditions are fused together and different queue contexts are inferred to generate the taxi and passenger queue context dataset. An instance in the queue context dataset is an hourly timewindow with a queue context label. A preliminary analysis is conducted to learn the influence and patterns of different associated factors. In this study, we prepare the taxi queue context dataset for the JFK (John F. Kennedy) airport by adapting a state of the art queue context inference algorithm.

2.3.2.1 Contextual Data Fusion

Our framework fuses temporal information and three other real-world contextual datasets from the New York City (NYC) and represents on the basis of hourly time windows. It facilitates to capture the influence of extracted associated factors in our scenario. The fused datasets are: i) the taxi trip log data, ii) the JFK airport passenger wait times data and iii) the JFK weather condition data.

Taxi Trip Logs: This dataset is a real-world dataset from New York City containing taxi trip logs. The NYC taxi trip dataset is available through the Taxi & Limousine Commission [83]. In NYC, 13 thousand taxis generate 0.5 million trips on an average per day totaling 175 million trips per year. Each record in this dataset represents one taxi trip. A taxi trip record is described by its start and end geo-location with corresponding time-stamps, trip distance, passenger count, fare type and mount, tip amount, and taxi's medallion number. Table 2.2 describes the fields for each record in the NYC taxi trip dataset. In this chapter, we process all the taxi trips made during the year of 2013 in NYC.

Airport Passenger Wait Time Data The passenger wait time data is available through the U.S. Customs & Border Protection [1]. This dataset provides information about passenger wait times at different U.S. airports. Additional features include hourly frequencies of flight

[1] http://awt.cbp.gov/

Table 2.2: Fields in the NYC Taxi Trip Dataset

Fields	Description
Medallion	Taxi identification number.
Pickup date-time	Start time and date of the trip.
Drop-off date-time	End time and date of the trip.
Pickup lat/long	Start location of the trip.
Drop-off lat/long	End location of the trip.
Trip distance	Distance traveled during the trip.
Rate type	Type of fare (i.e., fixed or not).
Payment type	How the payment was made.
Tip amount	The amount of tips.
Passenger count	Number of passengers in the trip.

and passenger arrivals with the numbers of passengers processed at the passenger processing booths. It also provides the hourly passenger wait times at different terminals at the JFK airport.

Weather Condition Data The weather condition data for JFK airport is collected from Weather Underground [2]. This dataset provides historical weather condition information including precipitation, temperature, wind speed, dew point, weather events (e.g., normal, rain, snow, rain-snow) and weather conditions (e.g., clear, overcast, mostly cloudy) at JFK.

2.3.2.2 Queue Context Inference

To infer the taxi and passenger queue contexts, it is required to estimate the taxi arrival rate and job wait time in a given hourly time window. However, the taxi arrival rate and job wait times cannot be estimated directly for all the taxi trips that start from the JFK airport. The reason is because a large volume of empty taxis arrive anytime without any pre-booking request. These trips with no passengers are not stored in the taxi trip dataset and the taxi arrival times are unknown.

[2] http://www.wunderground.com

To overcome this problem, we rely on the findings of survey results presented in recent literature [65]. This research reveals that experienced drivers use their own expertise to choose nearby parking places to wait for their next passenger pickup rather than cruising randomly after a passenger drop-off. Motivated by this fact, we choose those taxi drivers who join the airport taxi rank for their next passenger pickup after an earlier passenger drop-off at the airport. Another reason to choose these taxis is that we know their arrival and departure times to and from the passenger pickup queue. We assume that these trips are able to provide insights about the average taxi waiting times in a time slot. We consider the trips that start or end at the JFK airport for our experiments in this chapter. To separate these trips we consider the latitude/longitude bounding box for the JFK airport given in [84]. For a given a time slot, we design Algorithm 1 to separate the JFK airport trips.

Algorithm 1: Separating airport trips from trip dataset $\mathbf{T_x}[...]$

Input: taxi trip dataset $\mathbf{T_x}[...]$, airport latitude/longitude bounding box using $(minLat, maxLat, minLong, maxLong)$
Output: Airport taxi trip dataset $\mathbf{A_x}[...]$

```
// Initialization
```
$\mathbf{A_x}[...]$=NULL, minLat=$minLat$, maxLat=$maxLat$, minLong=$minLong$, maxLong=$maxLong$;
function separateAirportTrips($T_x[...]$,minLat,maxLat,minLong,maxLong)
foreach $t_x \in T_x[...]$ **do**
if $minLat < t_x{\cdot}Trip\_end\_lat() < maxLat$ ***and*** $minLong < tx{\cdot}Trip\_end\_long() < maxLong$ **then**
    $A_x[...] \leftarrow t_x$;
    `// Separate` t_x `from` T_x `and insert in to` A_x
else
    if $minLat < t_x{\cdot}Trip\_start\_lat() < maxLat$ ***and*** $minLong < tx{\cdot}Trip\_start\_long() < maxLong$ **then**
        $A_x[...] \leftarrow t_x$;
        `// Separate` t_x `from` T_x `and insert in to` A_x
    end
end
return $A_x[...]$;

We also design two other algorithms to compute the hourly average taxi wait times and the passenger pickup rates. These are used by the queue context inference algorithm to infer

Algorithm 2: Computation of Hourly Average Wait Time

Input: An hourly time window of a day $(\mathbf{t_i}, \mathbf{t_j})$, $\mathbf{A_x}[...]$
// t_i and t_j are the start and end time of the time window
Output: Hourly average taxi queue wait time $\bar{\tau}$
// Initialization
$trip\_count = 0$;
function hourlyAverageTaxiQueueWaitTimes($\mathbf{A_x}[...]$,$\mathbf{t_i}$, $\mathbf{t_j}$)
foreach $t_x \in A_x[...]$ **do**
if $t_i < t_x{\cdot}Trip\_end\_dateTimes() < t_j$ **then**
    if $minLat < t_x{\cdot}Next\_Trip\_start_lat() < maxLat$ ***and*** $minLong < t_x{\cdot}Next\_Trip\_start\_long() < maxLong$ **then**
        $T_{arr} \leftarrow t_x{\cdot}Trip\_end\_dateTime()$;
        $T_{dep} \leftarrow t_x{\cdot}Next\_trip\_start\_dateTime()$;
        $w \leftarrow T_{arr} - T_{dep}$;
        // taxi queue wait time
        $trip\_count++$;
    end
end
return $\bar{\tau} \leftarrow sum(w)/trip\_count$;

Algorithm 3: Computation of Hourly Pickup Rate

Input: An hourly time window of a day $(\mathbf{t_i}, \mathbf{t_j})$, $\mathbf{A_x}[...]$
// t_i and t_j are the start and end time of the time window
Output: Hourly passenger pickup rate ρ
// Initialization
$pick\_count = 0$;
function passengerPickupRate($\mathbf{A_x}[...]$,$\mathbf{t_i}$, $\mathbf{t_j}$)
foreach $t_x \in A_x[...]$ **do**
if $t_i < t_x{\cdot}Trip\_start\_dateTimes() < t_j$ **then**
    $pick\_count++$;
end
return $\rho \leftarrow pick\_count/|(t_i, t_j)|$;
// $|(t_i, t_j)|$is the length of time window $(\mathbf{t_i}, \mathbf{t_j})$ in minutes

taxi-passenger queue contexts for any given hourly time window. For calculating the average taxi queue wait times and passenger pickup rates we use Algorithms 2 and 3. First, an hourly time window is selected. Then all the passenger drop-off times of the airport taxi trips within that hour are stored to calculate the time difference with their next passenger pickup times. Similarly, the hourly passenger pickup rate is calculated considering all the taxi trips that

initiated from the airport within that hour. The taxi trips that only started from the airport without a precedent airport drop-off are also considered in this case.

For queue context inference, we utilize the Queue Context Disambiguation (QCD) algorithm proposed by [28] which performs well with the taxi GPS trip traces and taxi mobile data terminal (MDT) log data and needs adaptation in our scenario where none of these are avaialble. However, the QCD algorithm relies on two core assumptions. First, a time slot is labeled as 'taxi queue only' when the average waiting time ($\bar{\tau}$) of taxis is more than a waiting time threshold (τ). This is not valid for the time slots during airport off-peak hours in our case. Some of the drivers willingly join the taxi waiting queue during this time and wait for the arrival of the first flight in the early morning. These drivers may experience long wait times which do not constitute the existence of taxi queue context. We remove off-peak hours from our dataset and set $\tau = 90$ minutes for inferring the taxi queues. Second, a time slot is labeled as the 'passenger queue only' when taxi arrival rate of empty taxis or overall passenger pickup rate (ρ) is very high. The arrival rate indicates the number of taxi arrivals per minutes while the pickup rate indicates the number of passenger pickups by taxis per minute. However, these assumptions cannot be used directly in the scenario of an airport. The most common reason is due to the large volume of empty taxis arriving anytime without any pre-booking request from airport passengers. Moreover, a passenger queue also can occur for a low passenger pickup rate if there is a shortage of taxis for a very high demand.

Unlike the QCD algorithm, we use a new threshold pickup rate which combines the upper ($\rho - up$) and lower ($\rho - low$) bounds of the pickup rate. This new threshold labels a time slot as 'passenger queue only' when $\rho - up < \rho < \rho - low$. Note that this assumption is not valid for airport off-peak hours since there are very few or no passenger pickup events observed. We arbitrarily set $\rho - low = 2.5$ and $\rho - up = 8.0$ to infer the corresponding queue context labels for all the hourly timestamps except airport off-peak hours. In the final dataset, each record represents a time stamp of one hour duration. A timestamp in this queue context dataset is described by the extracted associated factors from the three real-world datasets discussed below in Section 2.3.2.3 (Extraction of Associated Factors and Preliminary Analysis).

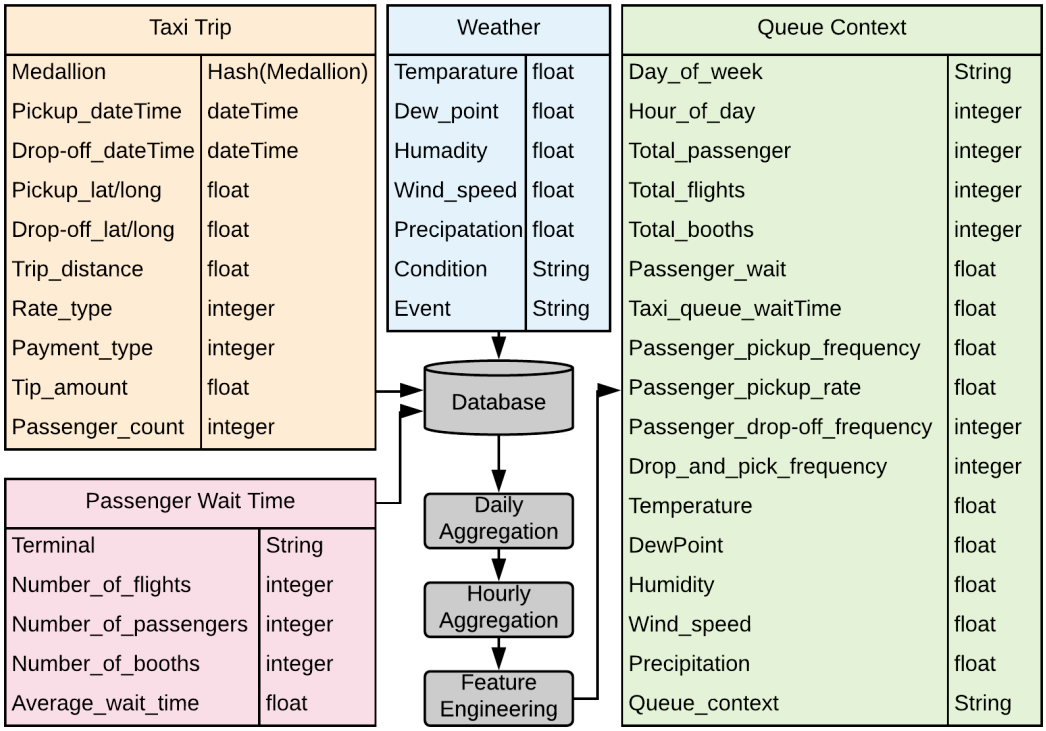


Figure 2.2: Data Fusion and Extracted Features for the Queue Context Dataset

2.3.2.3 Extraction of Associated Factors and Preliminary Analysis

We extract various features to generate the queue context dataset by fusing three real-world datasets and temporal information. Note that all these features correspond to the current hour and we extract and compute the values of the similar features in the previous and next hours as well. We also consider temporal features including the hour of the day, the day of the week, and the week number of the year. In total each record is described by 44 features and one of the four queue context labels. Figure 2.2 shows the steps to generate the final 'queue context' dataset. In data fusion stage, the daily and hourly aggregation is performed to ensure that all the data points are computed under the same time window length. In feature engineering, additional features including drop-off and pickup frequencies, pickup rates, and wait times are calculated. The extracted features are summarized as below:

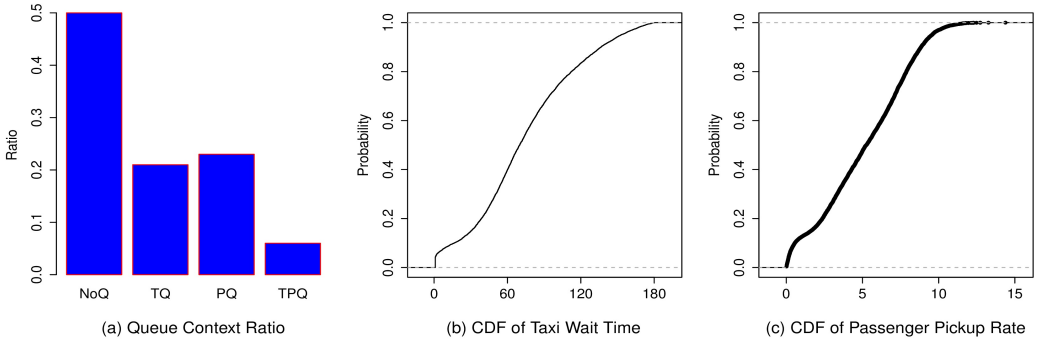


Figure 2.3: (a) Proportion of Queue Contexts. (b)-(c) CDF of Taxi Wait Times and Passenger Pickup Rate at the John F. Kennedy (JFK) airport.

- NYC taxi trip data: The extracted features include hourly taxi queue wait time, passenger pickup and drop-off frequencies, and frequency of taxi trips that start subsequent to an airport passenger drop-off.

- Passenger wait time data: The features include the frequency of hourly flight and passenger arrivals, number of passenger processing booths and hourly average passenger wait times at the passenger processing booths.

- Weather condition data: The features are hourly precipitation, temperature, dew point, wind speed, weather events (i.e. rain, snow, normal), and different weather conditions (i.e. clear, fog, cloud).

Further, we conduct a preliminary analysis to identify the proportions of different queue contexts. From Figure 2.3(a), we can see that the proportion of NoQ context is very high compared to other three while the proportion of TPQ is low. This may lead to poor prediction performance. On the other hand, Figure 2.3(b) illustrates the cumulative distribution function (CDF) of taxi queue wait times. We can see that some taxis wait in the queue for more than one hour for 60% of the instances. Another analysis from Figure 2.3(c) shows that about 60% cases the pickup rate (ρ) indicates an existence of passenger queues since passenger queues exist when $\rho - up < \rho < \rho - low$. This is the reason why it is so important to provide an accurate prediction for the taxi and passenger queue contexts for smooth running of the airport.

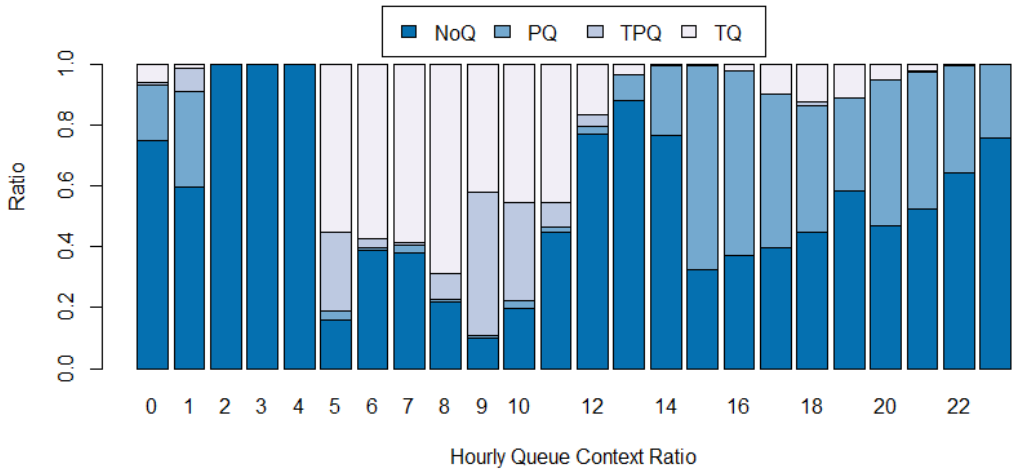


Figure 2.4: Hourly Proportions of Four Queue Contexts From Midnight of a Day to Midnight of the Next Day.

We also conduct an analysis from the perspective of time. Figure 2.4 shows the hourly ratio of the four queue contexts. We can see that the ratio of TQ dominates the other three queue contexts during the morning hours and PQ is the minority queue context during this time. As we approach the afternoon, we can see that the PQ and NoQ become the majority where TPQ is the minority context of all. We also note that the off-peak hours are dominated by NoQ context. We also examine the three heat maps in Figure 2.5 that represent the hourly taxi wait time, passenger pickup and passenger arrivals. Here, x-axis represents the days of the year in 2013 and y-axis represent 24 hours of a day. In the heat maps, the red color indicates higher values while blue color indicates lower values. From Figure 2.5(a), we can see that the taxi drivers mostly wait longer roughly between 03:00 and 13:00. Figure 2.5(c) shows that a large number of passengers arrive roughly between 11:00 and 22:00. This clearly indicates that a major portion of the taxi drivers sit idle in the airport while waiting for a passenger. Also, it is clear from Figure 2.5(b) that only a small portion of these taxi drivers are able to pick a passenger between 05:00 and 09:00 in the morning. This clearly indicates the taxi demand-supply imbalance in most of the occasions during the year of 2013 at the JFK airport.

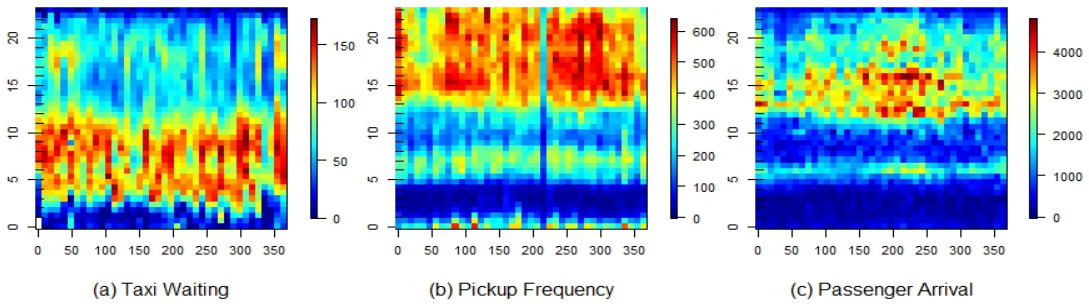


Figure 2.5: Heat Maps of Hourly a) Taxi Wait Times b) Passenger Pickup Frequency and c) Passenger Arrivals at the John F. Kennedy (JFK) airport. Note: x-axis represents the days of the year in 2013 and y-axis represents 24 hours of a day from midnight of a day to midnight of the next day. For example, 50 in the x-axis corresponds to 50^{th} day of the year of 2013 while 10 in the y-axis corresponds to 10:00 am.

Hence it is very important from the perspective of passengers, taxi drivers and airport ground transport managers to predict the queue contexts by analyzing various queue context features.

Furthermore, we analyze Pearson's Correlation Coefficient between the hourly taxi queue wait times and all other features of the context dataset. We observe a negative correlation with total flight arrivals (-0.30), total passenger arrivals (-0.26), total flight processing booths (-0.23), passenger pickup frequency in the previous hour (-0.49) and passenger pickup frequency in the current hour (-0.44). This implies that the more the flights, passenger arrivals, passenger processing booths and passenger pickup frequencies, the less the queue wait time for taxis.

2.3.3 Queue Context Prediction

The queue context prediction module of our developed framework is to provide prediction on future queue context by identifying the best pair(s) of sampling technique and prediction algorithm to satisfy different user perspectives. The queue context dataset suffers from the queue context imbalance problem. The queue context prediction module overcome this problem by applying various sampling techniques before employing a prediction algorithm from a suite of classifiers. The sampling techniques applied in our experiments are as below:

- Oversampling (OS) adjusts the class (i.e. queue context) distribution of a dataset by increasing the number of minority classes.
- Under sampling (US) adjusts the class (i.e. queue context) distribution of a dataset by decreasing the number of majority classes.
- Joint Sampling (JS) adjusts the class (i.e. queue context) distribution of a dataset by simultaneously increasing and decreasing the number of minority and majority classes respectively.
- No Sampling (NS) No adjustment of the class (i.e. queue context) distribution is performed.

The sampled dataset is used to train the prediction algorithms. We use a (60%-40%) split of the sampled dataset to train and test the prediction algorithms. To test the performance of different prediction algorithms we employ a classifier suite. The classifier suite contains 7 algorithms which includes the naïve bayes (NB), decision tree (J48), random forest (RF), decision table (DT), PART decision rule, support vector machine (SVM) and k-nearest neighbor (k-NN). We use the Weka [85] implementation of these classifier. Note that we choose $k = 5, 10, 15, 20, 25$ to consider different variants of the k-NN. Different evaluation metrics are applied to find the best sampling technique and the best set of prediction algorithms. Finally, we analyze the in-depth prediction performance from two user perspectives: the taxi drivers and the airport passengers.

2.3.3.1 Identifying Sampling-Classifier Pair(s)

We evaluate different sampling and prediction techniques applied to our queue context dataset. We begin with no sampling (NS) of the dataset. Then we employ under sampling (US), two variants of joint sampling (JS1 and JS2) and oversampling (OS). Note that in our queue context dataset, the 'NoQ' and 'TPQ' contexts are the majority and minority queue context labels respectively along with two other queue context labels (i.e. 'TQ' and 'PQ') as illustrated in Figure 1(a). In the under sampling stage, we randomly under sample all other queue contexts up to the number of 'TPQ' context.

Table 2.3: Performance Evaluation Under Different Sampling Techniques

	NS		US		JS1		JS2		OS	
	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
Accuracy(%)	74.12	50.05	80.34	25.88	84.51	49.72	85.71	55.09	**96.96**	62.62
Precision	0.74	0.25	0.81	0.38	0.85	0.58	0.86	0.67	**0.97**	0.63
Sensitivity	0.74	0.50	0.81	0.26	0.85	0.50	0.86	0.55	**0.97**	0.63
F-Score	0.73	0.33	0.81	0.18	0.84	0.41	0.86	0.52	**0.97**	0.60
AUC	0.90	0.50	0.95	0.51	0.97	0.67	0.97	0.70	**0.99**	0.86
AUPRC	0.81	0.35	0.88	0.25	0.92	0.46	0.93	0.50	**0.99**	0.66

During oversampling, we randomly select instances and repeat to increase their number until it becomes equal to the 'NoQ'. On the other hand, we oversample 'TPQ' and under sample 'NoQ' and 'PQ' contexts up to the number of 'TQ' contexts in our first joint sampling (JS1).In our second joint sampling (JS2), we under sample 'NoQ' and oversample 'TQ' and 'TPQ' contexts up to the number of 'PQ' contexts. Specifically, we pick one sampling technique at a time for our dataset and employ our classifier suite. Then we evaluate these sampling techniques under different performance metrics. The metrics include the predictive accuracy, sensitivity, F-Score, area under the ROC curve (AUC), and the area under the precision-recall curve (AUPRC). Specifically, we note the best metric score given by the classifier suite under each sampling technique. Table 2.3 summarizes the maximum and minimum metric score produced by our classifier suite under different sampling techniques.

First we observe the maximum predictive accuracy produced by our classifier suite under a specific sampling technique. We can see from Table 2.3 that the oversampling (OS) produces the maximum predictive accuracy over other sampling techniques. However, better prediction accuracy cannot reflect a good performance in our case. The reason is that the large number of majority queue context labels present in the dataset may degrade the prediction performance of minority queue contexts. Therefore, we further analyze the sensitivity of the prediction task. In binary classification, the sensitivity score tells us about how many relevant items are selected. Let us assume, there are two classes called 'positive' and 'negative'. The number of positive instance classified as positive is called true positive (TP) and number of positive instances classified as negative is called false negative (FN). Then the sensitivity score is given

by $TP/(TP + FN)$. In this way, we will be able to know the weighted average of prediction performance for minority classes. We can see from Table 2.3 that the oversampling (OS) gives the maximum sensitivity score. We also analyze the performance from different user perspectives. The F-score is the harmonic mean of sensitivity and precision where precision is a measure of the amount of retrieved instances that are relevant. The other two are very well-known in binary classification knows as the area under the ROC curve (AUC) and the area under the precision-recall curve (AUPRC). The AUC plots the true positives against false positives while the AUPRC plots the precision against the sensitivity. Table 2.3 shows that our classifier suite produces peak performances under all the performance metrics when oversampling is applied to the imbalanced data.

Next, we select the best classifiers from the classifier suite after applying the oversampling technique and analyze the confusion matrix they produce. For selecting the best classifier we rely on the AUPRC values in our research. First, we take a base classifier and observe significance of the AUPRC score of other classifiers. Specifically we perform paired t-test with a significance score of 0.05 between the AUPRC values of the base classifier and the other classifiers. In next iterations, we remove the previous base classifier from the classifier suite and choose a new base classifier that has the lowest AUPRC score. Note that we select this new base classifier after removing those classifiers with insignificant AUPRC score if there is any. After several iterations we get the support vector machine (SVM) and the Random Forest (RF) as the best two classifiers from our classifier suite. Note that the RF uses an ensemble learning technique that fits the training data into a number of decision tree classifiers to produce the final prediction. On the other hand, the SVM uses an imaginary hyper plane to discriminate between the training data to produce prediction for a query instance.

2.3.4 Evaluation and User Perspective Based Analysis

The evaluation metrics discussed above treat the queue context prediction as a binary class problem which actually cannot provide the true picture of the prediction performance. Therefore, we analyze the confusion matrices produced by the classifiers from a multi-perspective point of view. To visualize the performance of these selected algorithms, the confusion matrices

Table 2.4: Confusion Matrix-SVM

		Predicted as $\hat{c}(x_q)$			
		NoQ	TQ	PQ	TPQ
Actual	NoQ	1	0	0	0
	TQ	0.05	0.95	0	0
	PQ	0.07	0	0.93	0
	TPQ	0	0	0	1

Table 2.5: Confusion Matrix-RF

		Predicted as $\hat{c}(x_q)$			
		NoQ	TQ	PQ	TPQ
Actual	NoQ	0.75	0.1	0.15	0
	TQ	0.01	0.99	0	0
	PQ	0.02	0	0.98	0
	TPQ	0	0	0	1

for SVM and RF are illustrated in Tables 2.4 and 2.5 respectively. The airport taxi-passenger queue manager is responsible for proper management of queue contexts related to taxi and passenger at the airport by considering the perspectives of both taxi drivers and passengers.

To avoid long queues of taxis and passengers, the queue manager regulates incoming flow of the taxis at the taxi rank based on the predicted queue context in a future time slot. The taxi drivers are satisfied when they are able to avoid long queue wait times before a passenger pickup. If any future time slot is predicted as 'TQ' or 'NoQ', no taxis should enter the taxi rank area to avoid long queue wait times for taxis. Moreover, the 'TQ' and 'NoQ' contexts should not be predicted as 'PQ' since in such cases more taxis will enter the taxi rank unnecessarily and experience unwanted queue wait times. We can see from 'blue' filled cells of Tables 2.4 and 2.5 that the SVM performs better compared to the RF in this regard and there are no actual 'TQ' and 'NoQ' instances which are classified as 'PQ' by the SVM in contrast to 15% of 'NoQ' instances predicted as 'PQ' by the RF. On the other hand, the airport passengers expect a taxi as soon as they arrive at the terminal curbside. Therefore, the 'PQ' instances

should not be misclassified as it would restrict the taxi drivers from entering the airport taxi rank and eventually, it would cause long queue wait times for passengers waiting for taxis. The 'green' filled cells of Tables 2.4 and 2.5 indicate that some 'PQ' contexts are mis-classified as 'NoQ'. We can see that the prediction error produced by the RF (0.02) is lower compared to the SVM (0.07). Given Tables 2.4 and 2.5, the airport manager could decide the number of taxis required in a future time window more accurately to avoid unnecessary wait times for both taxis and passengers. Note that the optimal decision making was out of scope for this research.

2.4 Conclusion

In this chapter, we addressed the queue context prediction problem in the presence of imbalanced queue contexts related to taxi and passenger at the airport. We integrated three real-world datasets to study and analyze the problem of queue context prediction. We developed a framework that provides a step by step procedures to predict different contexts of the queues that are important to manage passengers and taxis at the airport. Moreover, our queue context prediction framework provides steps to identify the best prediction models and sampling techniques. It also provides intelligent analysis considering two different points of views. The experimental results show the effectiveness of our approach for queue context prediction at a busy international airport. We observe that the Support Vector Machine (SVM) performs better from the taxi drivers point of view while Random Forest shows better results from the point of view of the airport passengers to predict different queue contexts in a given future time stamp. The research presented in this chapter predicts four different queue contexts which can be applied to any location not only airports but also shopping malls, ferry platforms.

There is further scope for improvement by providing the information about lengths of queues in real time along with these queue contexts. The two thresholds used to calculate the queue contexts were chosen arbitrarily. The selections of thresholds are sensitive and need domain adaptation. The optimal thresholding was not considered for this thesis and can be addressed by future research. In future, more contextual data sources such as local events, traffic congestion can be incorporated with our queue context dataset. It would be

interesting to study further the effects of other features extracted from these datasets. We also plan to apply our approach to other airports when the data becomes available. For future deployment, the taxi regulations would need to be considered carefully. For example, at JFK airport, taxicabs are not allowed to pick-up passengers after a passenger drop-off at the airport terminal curbside. The taxi driver must join the taxi rank queue before being called by the passenger terminal. Moreover, the JFK airport maintains two types of queues (i.e. short and long trip queues) for taxi drivers. The future research may investigate the influence of one queue over another.

In summary, this chapter develops and presents a mobility context prediction framework considering queue context as an example of user mobility context. Considering two different user perspectives, we provide analysis of different prediction outcomes produced by selected prediction techniques.

Chapter 3

Modelling Associated Factors for Mobility Context Prediction

Previously in Chapter 1, we identified mobility-associated factors which can the making of mobility decisions. In relation to RQ-2, this chapter presents the modelling of mobility-associated factors since they enhance the performance of prediction algorithms. This is achieved through the identification and incorporation of expert-like knowledge by modelling the heterogeneous mobility-associated factors from the historical data. Specifically, a scheme is introduced by combining the expert-like knowledge and probability theory to estimate the weights (i.e. importance scores) of heterogeneous mobility-associated factors selected for predicting a specific mobility context.

We consider two mobility context prediction scenarios. The first scenario is about taxi-passenger queue context prediction at the airport where, taxi and passenger queue contexts indicate various situations of two concurrently occurring queues related to taxis and passengers. The second scenario is to predict taxi drivers' queue wait times at the airport taxi rank. An instance in the wait time dataset is labelled with numeric target scores of queue wait times. The reason for choosing these two scenarios is to develop a way of predicting both categorical and numerical mobility contexts for real scenarios which are influenced by many diverse mobility-associated factors. To demonstrate the reasonableness of our technique for enhancing the mobility context prediction task, we conduct our experiments on two target mobility contexts:

queue context and queue wait time. Specifically, we incorporate our technique with different neighborhood-based prediction methods. For our experiments, we utilize the queue context dataset presented previously in Chapter 2. We conduct feature engineering to extract more features. We also construct a dataset for queue wait time by replacing the target label (i.e. queue context) of the queue context dataset with hourly queue wait time. The process of calculating the hourly queue wait time was described in Algorithm 2. This chapter highlights our motivation and contributions by explaining how we model mobility-associated factors to develop our feature importance score calculation scheme for mobility context prediction.

3.1 Motivation and Contribution

As discussed is Chapter 2, taxis are regarded as a convenient mode of transport for transfer between the airport and the city. Taxis and passengers at the airport must join respective queues for each other and wait their turn before being served. The efficient management of the passenger and the taxi queues plays an important role in the smooth running of an airport. It creates disruption and chaos for passengers and taxi drivers at the airport if either queue becomes too long. The passengers may remain in long queues waiting for taxis when there is a shortage of taxis at the airport taxi ranks. On the other hand, long queue wait times at the taxi rank may influence taxi drivers' decisions about not making an airport trip in future. Long taxi queues also cause traffic congestion and wasted land use while taxis wait for pickup jobs at the airport.

Prior knowledge about which queue (i.e. taxi, passenger, both, or none) is going to experience unusual waiting time in a future time window could provide timely management of these two concurrent queues. So it is a very important mobility context to predict different queue contexts related to taxi and passenger queues. It is also important to provide prediction on the queue wait times of taxi drivers so that they can better plan their airport trips. Queue wait time is an important mobility context for taxi drivers. However, it is challenging to estimate the taxi-passenger queue contexts and queue wait time for taxis at the airport taxi rank since these situations are highly affected by many heterogeneous mobility-associated factors including the weather, and the dynamic taxi, passenger and flight arrivals.

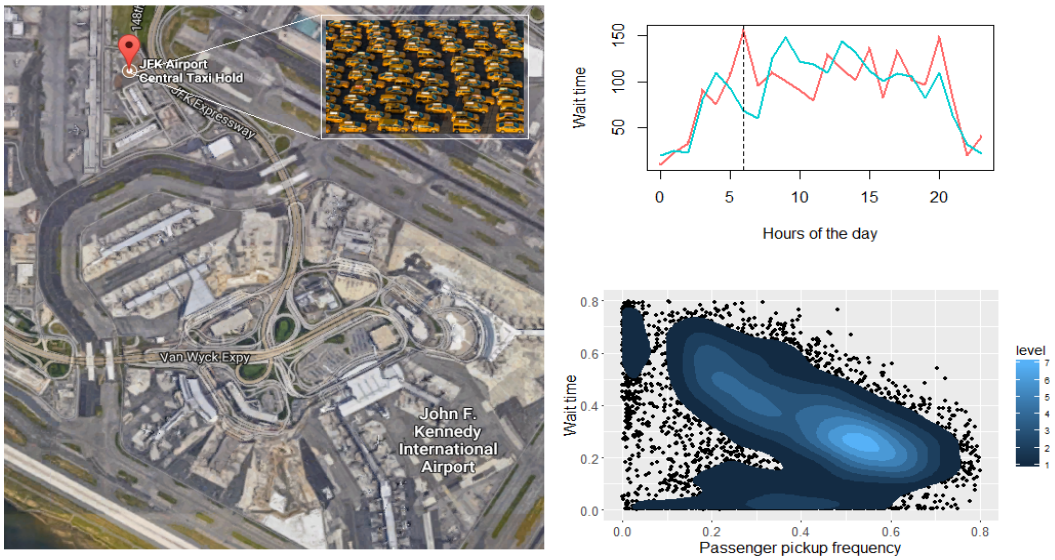


Figure 3.1: Taxis Waiting at The Central Taxi Holding Area at the JFK International Airport in New York City (left). The drivers experience different waiting times on two Mondays over two different weeks of May 2013 (top right). The Density Map (bottom right) shows the variation of wait times w.r.t. current passenger pickup frequency.

Figure 3.1 (above left) shows the taxis waiting at the central taxi holding area at the JFK international airport in New York City. The JFK airport is one of the busy airports in the U.S. All the taxis must enter and join the queue in this area and await their turn before being called from a passenger terminal for passenger pickup job. However, the drivers of these taxis experience different wait times while waiting in the taxi rank. Figure 3.1 (top right) also shows a comparison of average queue wait times for two different Mondays in April 2013. We can see that the wait times varies throughout the day. Also, the variation in densities of the taxi drivers' queue wait times with respect to current passenger pickup frequency is well scattered as illustrated with a 2- dimensional density plot in Figure 3.1 (bottom right). We can see a highly dense area with larger current pickup frequency results in low queue wait times while it is not sufficient to conclude that a low pickup frequency always results in a higher wait times as some highlighted dense areas can also be seen for low passenger pickups. This indicates that it is an important and complex problem to identify the importance of related contextual features for taxi driver queue wait time and taxi-passenger queue context prediction.

The problem of queue context and queue wait time prediction has been studied in many application areas [26, 86], and various machine learning techniques have been examined by [56, 1]. A recent attempt on queue wait time prediction is based on k nearest neighbor-based method (kNN) [1] which utilizes only three temporal factors. However, no existing research investigates the effectiveness of using a large number of external heterogeneous mobility-associated factors as mentioned above, which in reality have direct influence on mobility contexts such as the queue wait times of taxis and taxi-passenger queue contexts. For example, the bad weather may cause big demands of taxis, and the delayed flight arrivals may cause long taxi or passenger queues. The dynamic and heterogeneous nature of these factors makes the prediction of different mobility contexts at the airport complex. Moreover, it is known that the prediction accuracy of kNN methods is dominated by the identified neighborhood and proper selection of factors [87]. We have found that an improvement in the quality of identified neighborhood can further improve the prediction accuracy [88].

In this chapter, we re-investigate the queue context prediction scenario presented in Chapter 2 and solve the taxi drivers' queue wait time prediction problem. We model mobility-associated factors for enhanced prediction performance using neighborhood-based methods. Our objective is to model heterogeneous mobility-associated factors in the identification of quality neighbors and hence improve the prediction performance by addressing the following problem:

How to identify a dense quality neighborhood for kNN-based methods to predict taxi-passenger queue contexts and taxi queue wait times by considering heterogeneous factors, e.g. time, weather, flight information and taxi trips?

We begin by providing a summary of related literature followed by the development of methodologies and experiments. Using two scenarios, we model heterogeneous contextual features and incorporate taxi driver intelligence with probability theory to develop a feature weighting scheme to identify dense quality neighborhood for mobility contexts (i.e. queue context and queue wait time) prediction. The experiment results demonstrate that the modelling of heterogeneous contextual features together with the drivers' intelligence can improve

the quality of the identified neighborhood, thus significantly boosting the prediction accuracy. The contributions of this chapter are as follows:

- Extraction, and analysis of heterogeneous mobility-associated factors and patterns related to taxi-passenger queue contexts and taxi queue wait times.
- Identifying factors that can influence the problem of taxi-passenger queue contexts and taxi queue wait times predictions.
- Inferring expert-like knowledge by investigating passenger pickup decisions of taxi drivers.
- Modelling mobility-associated factors to propose a weighting scheme that identifies dense high quality neighborhoods for mobility context prediction using kNN-based methods.

3.2 Related Work

Mobility context prediction requires extraction and analysis of different features and pattern associated with user mobility. A number of research papers aim to extract and analyze different mobility-associated factors to provide timely information for user mobility decision support. Aiming to provide location recommendation, a probabilistic analysis of spatio-temporal factors is presented in [65]. A smart city application for location recommendation is presented in [79] where the authors investigate factors such as wait time and events from historical taxi trajectories. Another spatio-temporal factor analysis model is developed by [75] which aims to provide mobility decision support for the taxi drivers. A pickup location recommendation for taxi users using multiple feature set is proposed in [78]. Feature extraction from user movement trajectories to estimate the transportation modes during users' mobility is presented in [89]. The authors consider both handcrafted feature engineering and automated feature engineering using deep neural network. The feature extraction from trajectory data is conducted for driver risk profiling in [90]. The article also provides a spatio-temporal analysis of extracted features. A model to find and analyze interesting and unexpected patterns from taxi trajectories is proposed by [91]. Aiming to detect and describe the mobility of vehicles, urban traffic patterns are analyzed using taxi trajectories in [92]. Another data driven model to analyze taxi drivers'

airport pick-up decisions is presented by [29]. The authors extract and analyse various spatio-temporal and external factors. A city-wide bicycle mobility has been analyzed by [93]. The spatio-temporal pattern of urban shared bicycle mobility is discussed in [94].

There are several papers which discuss research conducted for detecting and predicting queue context and queue wait times in various means and different application areas [56, 1, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104]. In [101], a model for estimating crowd density using deep convolutional neural network is proposed. Another crowd density estimation technique is proposed in [102] which utilizes bluetooth-based sensing with mobile phones. A traffic congestion forecasting technique based on the changes in driving behavior is proposed in [103]. Another prediction model for traffic congestion during special event is presented in [104]. The first research paper on predicting a user's residence time using a non-linear time series analysis is proposed by Scellato et al [105]. The residence time is the time spent by the user when they visit their most important locations. Wu et al [81] have developed a system which continuously monitors each taxi stand and takes account of the numbers of taxis queuing and passing the taxi stand, as well as the traffic conditions in the area around the stand. Zhang et al [56] have recommended sensing the fuel consumption of taxi drivers with a view to minimizing queues at petrol stations and ultimately predicting strategic placement of petrol stations. Qi et al [106] have considered the passenger wait times at taxi ranks with a view to discovering the flow of people through the city and optimizing the transportation network as a whole. On the passenger side, Anwar et al [58] have also considered passenger movement through an airport, with a view to sending taxis this information for the demand so they can service the longest queue first. The "OpenStreetCab" app of Salinikov et al [107] provides users with information on the cheapest available cab in the city with a view to providing competition to Uber and demonstrates how yellow taxis can provide a cheaper option to UberX using publicly available data. It provides an example of how big datasets that become public can improve urban services for consumers. In our application we consider this in relation to integrating weather and taxi data.

The nearest neighbor regression is an effective machine learning approach for prediction on a numeric scale in various applications. This is because of it's simple implementation and

performance guarantees [108]. It also has shown its effectiveness in shop queue wait time prediction [1]. However, neighborhood identification is still a challenging area for nearest neighbor regression. Various approaches have been adopted such as distance weighting [109, 110, 111] of nearest neighbors. Feature weighting has shown its effectiveness in regard to increased prediction performance in many application domain [112, 113, 114, 49, 115, 116]. In [1], feature importance score is calculated by building a linear regression model. This approach is effective with incomplete and non-uniform data. Another feature weighted distance measure for k-NN is proposed in [117]. It is based on the mutual information between a feature and the class value. The mutual neighborhood information is used to boost the performance of nearest neighbor classification by [118]. In [119], a categorization framework for feature weighting approaches is proposed. A brief survey is conducted which refers to the use of 'domain-specific information' for feature weighting.

Although the nearest neighbor regression has been used effectively in real-world wait time prediction application with a small number of features[1], taxi driver queue wait time prediction at the airport using nearest neighbor regression is a challenging issue. This is because of the presence of various heterogeneous contexts such as weather, flight arrival, and flight processing. Many features can be extracted from these heterogeneous contexts. Since these features are heterogeneous in nature, the identification of relationships between those and the queue wait time is a complex task. Therefore, it is required to identify the influence of these features in queue wait time prediction considering their heterogeneity.

Research has shown that the use of expert knowledge is able to increase the prediction accuracy [48]. Also the expert drivers use their expertise to go to a place for passenger pickup rather than cruising randomly [74]. We take this note and use the taxi drivers' intelligent moves for feature weighting to predict taxi queue wait time using nearest neighbor regression.

3.3 Datasets

For experiment setup and analysis, we use two datasets that represent two mobility context scenarios (i.e. queue context and queue wait time) at the airport.

The Queue Context Dataset: We utilize the queue context dataset used in Chapter 2 and presented in [26] which was generated to examine different queue contexts at the JFK airport during the year of 2013.Each instance in this dataset is a time slot of one hour in duration and described by a feature vector of 44 elements. The features can be categorized under four types:

- *Temporal* includes hour of the day, day of the week.
- *Taxi* includes number of passenger pickups, number of passenger drop-off, number of pickups with a precedent airport drop-offs which the driver makes and the average queue wait times of taxis.
- *Passenger* includes number of flights, number of passengers, number of flight processing booths, average passenger waiting time.
- *Weather* includes precipitation, wind speed, temperature, dew point, humidity, weather conditions.

Note that all these features correspond to the current hour of the day. The queue context dataset also contains the same features for the previous and next hourly time window for passenger and weather related features except average passenger waiting time. Also the taxi related features are available only for the previous hourly time window together with current hourly time window since these feature values are not available in the next hourly time window. In this research, we compute and include one additional feature against each available feature in the queue context dataset. These additional features are calculated by taking any feature and computing the deviation from its mean feature score. The final dataset has a total of 66 features for queue context prediction.

The Queue Wait Time Dataset: We utilize the same queue context dataset presented in Chapter 2. However, we extract hourly queue wait time for taxis which is the target score for our prediction task. The developed technique for hourly queue wait time calculation is given by Algorithm 2. To construct the queue wait time dataset, we replace the target label

(i.e. queue context) of queue context dataset with the hourly queue wait time. In this way the feature hourly queue wait time becomes the target score for prediction task. In the final dataset, each record describes an hourly time stamp totaling 8760 data points during the year of 2013.

3.4 Neighborhood Identification Using Feature Weight Score for Mobility Context Prediction

In this section, we present a technique to enhance the queue context prediction using neighborhood based methods. We model associated factors for expert-like knowledge acquisition to be used for feature weight calculation. Note that we use the similar technique for feature weigh calculation in both of our scenarios but with a little variation in factor modelling. We show that even with a variation in factor modelling, the developed technique enhances the mobility context prediction performance. Table 3.1 lists the notations used in this chapter.

Table 3.1: List of Notations

Notation	Description
T_i	An hourly time window.
T_Q	An hourly query time window.
$\overline{w}(T_i)$	Queue wait time during T_i.
$\overline{\overline{w}}(T_Q)$	Predicted queue wait time during T_Q
F_c	Set of contextual features.
$d(T_Q, T_i)$	Distance between T_Q and T_i.
a_j	The j^{th} contextual feature.
ω_j	Weight of j^{th} feature.
DI	Driver Intelligence.
MI	Mutual information.
$I(a_j; \overline{w}(T_i)\|DI)$	Driver intelligence-biased MI.
$TDID$	Temporal Driver intelligence Deviation(TDID)
$I(a_j; c(T_i)\|TDID)$	TDID-biased MI

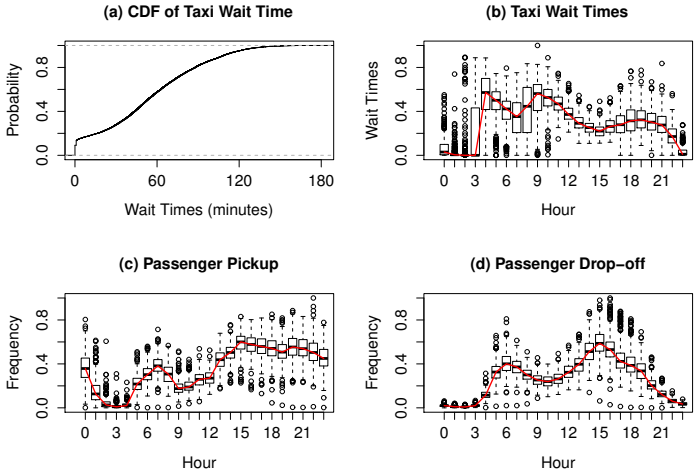


Figure 3.2: (a) Cumulative Density Function of Taxi Wait Times and (b) Taxi Wait Times, (c) Frequency of Passenger Pick-up, (d) Frequency of Passenger Drop-off

3.4.1 Scenario 1: Queue Wait Time Prediction

Let, T_i be an hourly time window. In our taxi queue wait time dataset, each instance T_i is described by a set of contextual features F_c. The hourly taxi queue wait time during T_i is denoted as $\overline{w}(T_i)$ which is the taxi drivers' average time spent in the queue from the time of arrival during T_i until the next passenger pickup. Given a query time window T_Q and corresponding set of contextual features F_c, we predict the hourly taxi queue wait time as: $(T_Q, F_c) \rightarrow \overline{\overline{w}}(T_Q)$.

We aim to find the quality dense neighborhood for nearest neighbor regression. We also provide a comprehensive analysis on the relationship between contextual features and the queue wait times is conducted. Based on the analysis, we introduce the driver intelligence-biased weighting scheme to improve the quality of identified neighborhood, so as to improve the accuracy of queue wait time prediction.

3.4.1.1 Contextual Analysis

We first conduct a preliminary analysis on taxi queue wait time from the perspective of *time*. We observe the hourly and daily patterns of the taxi queue wait times, taxis' passenger pickup and drop-off frequencies, passenger arrivals, and passenger wait times. Figure 3.2 (a) - (d) and

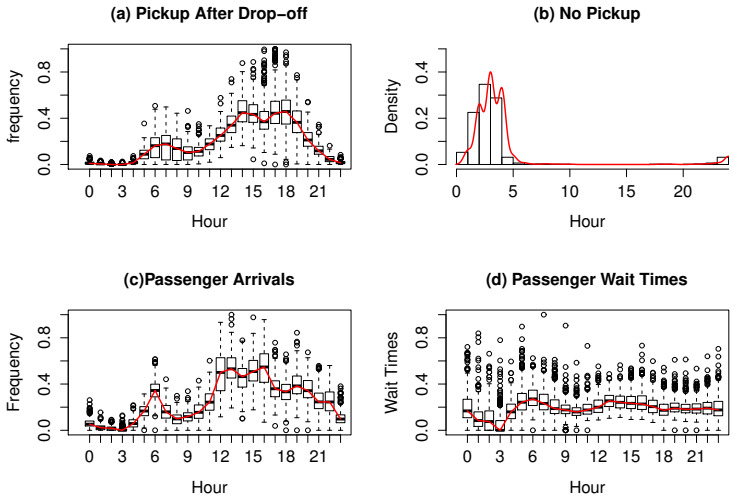


Figure 3.3: (a) Frequency of Passenger Pickup After Subsequent Passenger Drop-off by a Taxi, (b) Density of Zero Passenger Pickups After subsequent Passenger Drop-offs by a Taxi, (c) Frequency of Passenger Arrivals, (d) Passenger Wait Times.

Figure 3.3 (a) - (d) show the hourly patterns along with the Empirical Cumulative Distribution Function (ECDF) of taxi queue wait times. The ECDF in Figure 3.2(a) shows that 80% of the data points have wait time equal or below 90 minutes. Figure 3.2(b) shows the hourly patterns for normalized taxi queue wait times. We can see that the highest wait times are observed during 04:00 am and 09:00 am. Figure 3.2(c) and (d) show the hourly patterns in passenger pickup and drop-off by taxis. A spike is observed during 15:00 pm for passenger drop-off and after that it reduces till mid-night. The reason may be due to the high volume of departing flights in the afternoon. On the other hand, two clear spikes are observed for passenger pickup frequencies: one appears at 07:00 am in the morning while the other starts at 15:00 pm and continues until mid-night. We also observe the hourly frequency of taxi trips started from JFK after a subsequent passenger drop-off at JFK in Figure 3.3(a). It is observed that a large number of taxis decide to pickup their next fare from the airport between 13:00 pm and 19:00 pm. Figure 3.3(b) shows the density of taxis where they decide to leave the airport after a passenger drop-off. The hours between 01:00 am and 03:00 am share almost all of the densities. This is expected because these hours are the off-peak hours at JFK. We also observe from Figure 3.3(c) that the passenger arrivals maintain a similar trend line to the

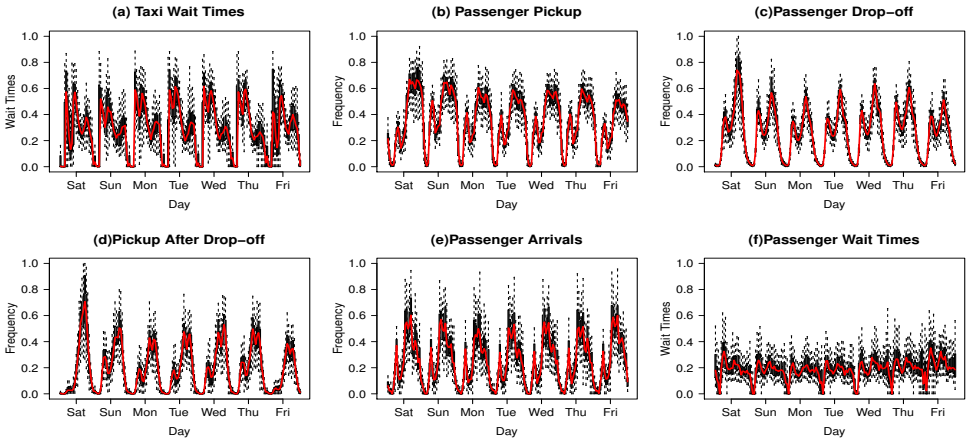


Figure 3.4: Contextual Analysis of Taxi Wait Times with Daily Patterns of (a) Taxi Wait Times, (b) Frequency of Passenger Pick-up, (c) Frequency of Passenger Drop-off, (d) Frequency of Passenger Pickup After Subsequent Passenger Drop-off by a Taxi (e) Frequency of Passenger Arrivals, (f) Passenger Wait Times.

passenger pickup frequency. However, a difference of an hour is observed in the rise and fall of the spikes. Figure 3.3(d) shows the hourly average passenger wait times. We can observe almost a similar trend in wait times throughout the day except a sharp rise from 03:00 am when the flights start to arrive.

Figure 3.4 illustrates the daily patterns for different extracted features. Specifically, Figure 3.4(a) shows that taxi wait times follow a uniform trend during the week except Sunday. Figure 3.4(b) shows a decreasing trend in passenger pickup from Saturday to Friday while maximum passenger drop-offs are observed during Saturdays (Figure 3.4(c)). The maximum number of taxi drivers deciding to pickup their next fare from the airport occurs during Saturdays (Figure 3.4(d)). Figure 3.4(e) and (f) show the daily pattern for passenger arrivals and passenger wait times. Similar trends are observed throughout the week for both.

Furthermore, we analyze correlation statistics (Table 3.2) between the hourly taxi queue wait times and all the other features extracted from three heterogeneous contextual datasets, including *passenger*, *trip*, and *weather*. We compute the Pearson's Correlation Coefficient to measure the relationship between wait times and the other features. Specifically, a negative correlation is observed with total flight arrivals in the previous hour, total passenger arrivals in the previous hour, total flight processing booths in the previous hour, passenger pickup

Table 3.2: Pearson's Correlations between Average Queue Wait Times and all the Contextual Features Extracted From Three Heterogeneous Contextual Datasets, sorted by *Passenger*, *Trip*, and *Weather*.

Contexts	Features	Correlation
Passenger	Total passenger in previous hour	-0.26
	Total passenger in current hour	-0.09
	Total passenger in next hour	+0.13
	Total flights in previous hour	-0.30
	Total flights in current hour	-0.13
	Total flights in next hour	+0.10
	Total booths in previous hour	-0.23
	Total booths in current hour	-0.05
	Total booths in next hour	+0.16
	Average passenger waiting in previous hour	-0.15
	Average passenger Waiting in current hour	-0.06
	Average passenger Waiting in next hour	+0.12
Trip	Passenger pickup frequency in previous hour	-0.49
	Passenger pickup frequency in current hour	-0.44
	Passenger pickup frequency in next hour	-0.21
	Passenger drop-off frequency in current hour	+0.12
	Drop and pick frequency in current hour	-0.07
Weather	Temperature (°C) in previous hour	-0.02
	Temperature (°C) in current hour	+0.01
	Temperature (°C) in next hour	+0.03
	Dew point in previous hour	-0.01
	Dew Point in current hour	-0.01
	Dew Point in next hour	-0.01
	Wind speed(Kmph) in previous hour	-0.03
	Wind speed(Kmph) in current hour	-0.01
	Wind speed(Kmph) in next hour	+0.02
	Precipitation(mm) in previous hour	-0.05
	Precipitation(mm) in current hour	-0.04
	Precipitation(mm) in next hour	-0.04
	Snow in current hour	+0.03
	Rain in current hour	-0.08

frequency in the previous hour and passenger pickup frequency in the current hour. This implies that the more flights, passengers, passenger processing booths and passenger pickup frequencies, the less the taxi queue wait time.

Table 3.3: Statistically Significant Features Computed From Time Contexts and the Three Heterogeneous Contextual Datasets, including *Passenger*, *Trip*, and *Weather*.

Contexts	Features	95%CI
Time	Day of the week	(-0.00593, -0.00241)
	Hour of the day	(0.005447, 0.006999)
	Week number of the year	(-0.00099, -0.00049)
Passenger	Total passenger in current hour	(-0.12461, -0.06930)
	Total passenger in next hour	(0.012649, 0.06580)
	Average passenger waiting in current hour	(-0.22139, -0.13371)
	Average passenger waiting in next hour	(0.019088, 0.105692)
Trip	Passenger pickup frequency in previous hour	(-0.32382, -0.25338)
	Passenger pickup frequency in current hour	(-0.46570, -0.38555)
	Passenger drop-off frequency in current hour	(0.265628, 0.344074)
Weather	Temperature in previous hour	(-0.68603, -0.45740)
	Temperature in next hour	(0.488688, 0.714952)
	Precipitation in previous hour	(-0.34068, -0.11975)
	Precipitation in next hour	(-0.24090, -0.01995)

3.4.1.2 Feature Selection

We perform a feature selection to enhance the prediction performance. For this purpose, we build a multiple regression model to predict the queue wait times. If $\hat{Y}$ is the target score, we write the multiple regression model using n number of features as follows:

$$\hat{Y} = \beta_1 a_1 + \beta_2 a_2 + \beta_3 a_3 + ... + \beta_n a_n \tag{3.1}$$

Here, a_i is the i^{th} feature and β_i is corresponding feature coefficient of a_i; $i = 1, 2, 3, ..., n$. Then we investigate the importance of each features to the research problem. For simplicity, we randomly select a subset of $n = 15$ features and build the regression model. Then we examine the coefficients of all these features within the model by using a 95% confidence interval. Specifically, we examine if 0 is within this interval. If so, it indicates that the coefficient can have a value of 0 thus the feature has no or less effect to predict the target score (queue wait times). We consider such features as unimportant. In every iteration, we leave one unimportant feature out and include a new one for next round. Finally, we find a total of 14 statistically significant features as shown in Table 3.3 which are and associated with the queue wait times. We consider the features from the *time* context as well along with *passenger*, *trip*, and *weather*.

Table 3.4: Driver Intelligence (DI)-biased Mutual Information

Contexts	Features	$I(a_j; \overline{w}(T_i)\|DI)$
Time	Day of the week	0.050
	Hour of the day	0.829
	Week number of the year	0.094
Passenger	Total passenger in current hour	0.400
	Total passenger in next hour	0.348
	Average passenger waiting in current hour	0.108
	Average passenger waiting in next hour	0.109
Trip	Passenger pickup frequency in previous hour	0.228
	Passenger pickup frequency in current hour	0.385
	Passenger drop-off frequency in current hour	0.696
Weather	Temperature(°C) in previous hour	0.058
	Temperature(°C) in next hour	0.049
	Precipitation(mm) in previous hour	0.003
	Precipitation(mm) in current hour	0.004

3.4.1.3 Feature Weight Calculation Scheme

In this section, we calculate feature weights for queue wait time prediction. The recent literature [74] shows that the experienced drivers prefer not to randomly cruise after a passenger drop-off. Instead, they usually go to the place they know well for picking up new passengers. We assume that this is also applicable in our scenario. Motivated by this fact, we consider the hourly frequency of taxi trips that are initiated from the airport vicinity after a precedent passenger drop-off at the airport. We call this frequency the Drivers' Intelligence (DI).

The mutual information is a measure of the mutual dependence between two random variables. Therefore, we can calculate the amount of mutual dependence between the queue wait time and all other features available in our queue wait time dataset after feature selection. Specifically, we calculate the conditional mutual information where we use the drivers' intelligence as a condition. We call this mutual information as the driver intelligence-biased mutual information. We calculate the driver intelligence-biased mutual information as follows:

$$I(a_j; \overline{w}(T_i)|DI) = -\sum_{a_j, \overline{w}(T_i), DI} p(a_j, \overline{w}(T_i)) \log \frac{p(a_j, \overline{w}(T_i)|DI)}{p(a_j|DI)p(\overline{w}(T_i)|DI)} \tag{3.2}$$

Here, a_j is any contextual feature; $\overline{w}(T_i)$ is the target (taxi queue wait time) and DI is the drivers' intelligence. Table 3.4 lists the corresponding driver intelligence-biased mutual

information scores for different features. In the next step, we normalize these scores of driver intelligence-biased mutual information between 0 and 1 to be used as feature weights ω_j. Here, ω_j is the feature weight of j^{th} feature.

3.4.1.4 Formulation of *k*-NN Methods

Given a set of training samples, we formulate the problem of predicting a corresponding target score using k-NN regression. Each sample T_i in the training data is described by a d-dimensional vector of contextual features and a target score $\overline{w}(T_i)$ as follows:

$$\langle a_1(T_i), a_2(T_i), a_3(T_i), ..., a_d(T_i), \overline{w}(T_i)\rangle,$$

To predict the target score of query instance T_Q, the distances between T_Q and all the training samples T_i are calculated as follows:

$$d(T_Q, T_i) = \sqrt{\sum_{j=1}^{d}[a_j(T_Q) - a_j(T_i)]^2}, \tag{3.3}$$

where $a_j \in F_c$ is the j^{th} contextual feature of T_i and T_Q; $j = 1, 2, 3, ..., d$.

Note that the basic k-NN regression treats each feature equally during this distance calculation. However, the contribution of each feature can be taken into account by rewriting Eq. 3.6 as follows:

$$d(T_Q, T_i) = \sqrt{\sum_{j=1}^{d}\omega_j * [a_j(T_Q) - a_j(T_i)]^2}, \tag{3.4}$$

where, ω_j is the weight of j^{th} feature.

Next, the k-nearest neighbors are identified by T_Q by sorting the values of $d(T_Q, T_i)$ in ascending order. If $T^{NN} = \{T_1^{NN}, T_2^{NN}, T_3^{NN}, ..., T_k^{NN}\}$ be the set of k-nearest neighbors of T_Q. If $\overline{w}(T_i^{NN})$ is the target score of T_i^{NN}, the predicted target score $\overline{\overline{w}}(T_Q)$ of the query instance T_Q is calculated by averaging the target scores of k-Nearest Neighbors as follows:

$$\overline{\overline{w}}(T_Q) = \sum_{i=1}^{k}\overline{w}(T_i^{NN}/k) \tag{3.5}$$

Note that the key here is to estimate the weights in Eq. 3.4 for each features appropriately so as to get a better neighborhood for higher prediction accuracy.

3.4.1.5 Experiments and Results

To evaluate the effectiveness of the proposed driver intelligence biased weighting scheme, we designed two sets of experiments:

1. *Taxi Queue Wait Time Prediction*: We predict the taxi queue wait time and compare with several weighting methods, including

   - *Baseline* ([1]): The baseline Nearest Neighbor Estimation (NNE) approach employs a regression based optimization for feature weighting. It considers only three attributes for queue wait time prediction (time of the day, day of the week, and week number of the year), and weights the features based on the co-efficients obtained from a linear regression model.
   - *LR-trained weights*: Unlike the *Baseline* ([1]), all 14 significant contextual features from Table 3.3 are considered. Then the feature weights are calculated by normalizing the co-efficients obtained from a trained linear regression (LR) model which are to be used for the Nearest Neighbor Estimation.
   - *Equal weights*: In this approach, all 14 significant contextual features from Table 3.3 are considered with equal weights for the Nearest Neighbor Estimation.
   - *MI-based Weights*: The MI (Mutual Information)-based weights includes all the significant contextual features from Table 3.3. Then the pure mutual information between each feature and the target (taxi queue wait time) is calculated and normalized between 0 and 1 to be used as feature weights for Nearest Neighbor Estimation.
   - *DI-biased weights*: The DI (Driver Intelligence)-biased feature weighting is the proposed weighting scheme of contextual features. The scores for each features obtained from Eq. 3.2 are normalized to be used as feature weights for Nearest Neighbor Estimation.

2. *Neighborhood Density/Quality*: we evaluate and compare the density and quality of neighborhood between the baseline and our proposed approach.

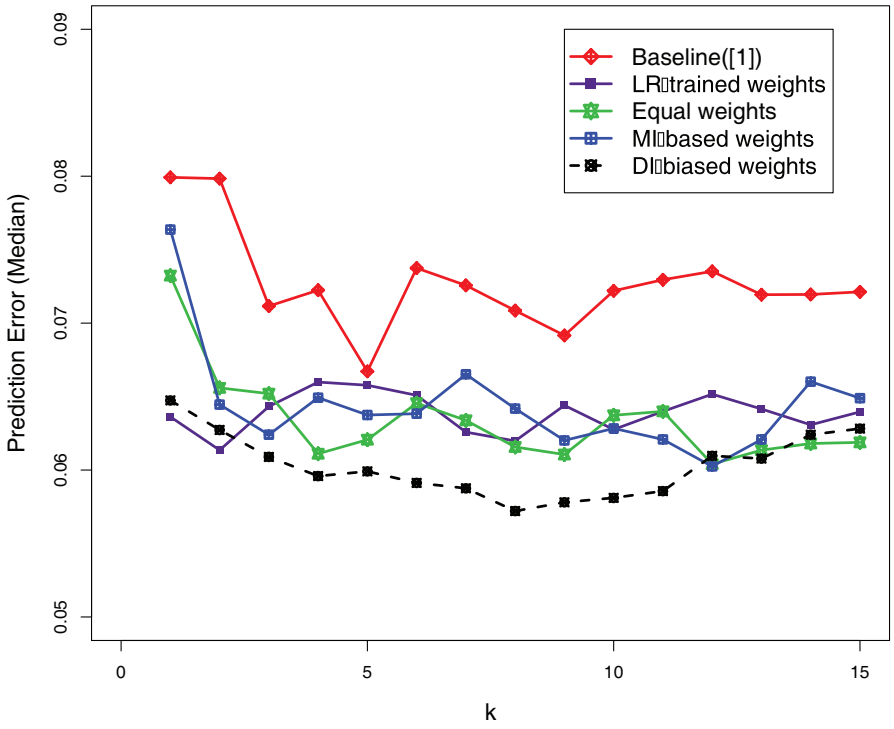


Figure 3.5: Percentage of Median Prediction Errors Using Various Feature Weighting Techniques for Varying k-values Between 1 and 15. This shows that the driver intelligence-biased feature weighting scheme gives the least amount of median errors.

We use a random 30-40-30 split to conduct our experiments with different feature weighting schemes for Nearest Neighbor Estimation. The first part contains 30% of the samples which were used for feature selection and feature importance calculation. The second part contains 40% of the instances which were used for training purpose, and the third part contains 30% of the instances which were used to test the performance of Nearest Neighbor Estimation. For performance evaluation, we consider the median and mean prediction errors for different k values between 1 and 15.

Queue Wait Time Prediction: Figure 3.5 shows the comparison of median prediction errors among all the methods. The comparison of mean prediction errors among all the methods is

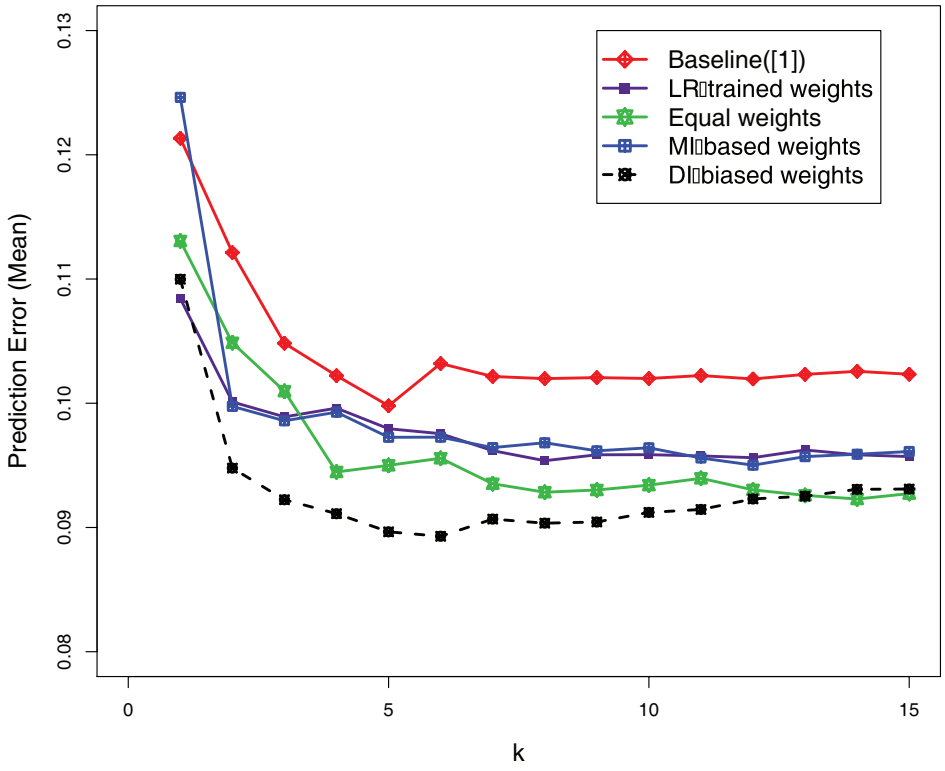


Figure 3.6: Percentage Mean Prediction Errors Using Various Feature Weighting Techniques for Varying k-values Between 1 and 15. This shows that the driver intelligence-biased feature weighting scheme gives the least amount of mean errors.

shown in Figure 3.6. We can see that the proposed feature weighting method *DI-biased weights* and its three variants (*LR-trained weights*, *Equal weights* and *MI-based weights*) outperform the *Baseline* ([1]) for different k values between 1 and 15 since they produce less prediction errors.

Next, we examine the statistical significance of this improvement. Specifically, a paired t-test is conducted to examine whether the improvement in prediction errors is statistically significant when comparing with *Baseline* ([1]). A paired t-test can determine whether the mean differences between two sets of paired samples differs from 0. The mean difference 0 indicates that the paired samples are similar. In our case, the first sample is the prediction errors produced by the *Baseline* ([1]) approach while the second sample is set in turn for the

Table 3.5: Paired t-Test of Prediction Errors Between Different Feature Weighting Techniques. This shows that the DI-biased weighting scheme provides the most significant improvement compared to other techniques.

Methods	Metrics	t	p
LR-trained weight *vs* Baseline ([1])	median	08.25	<0.001
	mean	09.04	<0.001
Equal weights *vs* Baseline ([1])	median	14.46	<0.001
	mean	17.97	<0.001
MI-based weights *vs* Baseline ([1])	median	09.63	<0.001
	mean	06.54	<0.001
DI-biased weights *vs* Baseline ([1])	median	17.89	<0.001
	mean	21.41	<0.001
DI-biased weights *vs* LR-trained weights	median	05.33	<0.001
	mean	07.21	<0.001
DI-biased weights *vs* Equal weights	median	04.39	<0.001
	mean	04.03	<0.001
DI-biased weights *vs* MI-based weights	median	05.30	<0.001
	mean	07.55	<0.001

prediction errors produced by the *LR-trained weights*, *Equal weights* and *MI-based weights*. Note that the pair-wise prediction errors are taken in to consideration for varying k-values between 1 and 15. Table 3.5 lists the statistics obtained from the paired t-test. We can see that the improvement in terms of prediction errors are statistically significant between the proposed method *DI-biased weights* (including its variants) and the *Baseline* ([1]) since the values of t-test statistics (t) differ significantly from 0 and p values < 0.001 on this small sample size of 15 supports about this significance.

Neighborhood Analysis: Here, we analyze the ECDFs (empirical cumulative distribution function) and the densities of inter neighbor $NN(p, q)$ distances where $p = 1$ to 14, $q = p+1$. We compare *LR-trained weights*, *Equal weights*, *MI-based weights* and *DI-biased weights* with the *Baseline* ([1]) feature weighting approach. Figure 3.7 shows the ECDFs of distances between consecutive neighbors. From the plotted ECDFs, we can see that there are some jumps in the ECDFs for the *Baseline* ([1]) approach, which means that the subsequent neighbors are not dense which results in a sparse neighborhood. On the other hand, the ECDFs for all of the other four approaches show smooth trend lines which imply the existence of a dense quality neighborhood. We also can see that our *DI-biased weights* shows the most smoothness compared to the rest.

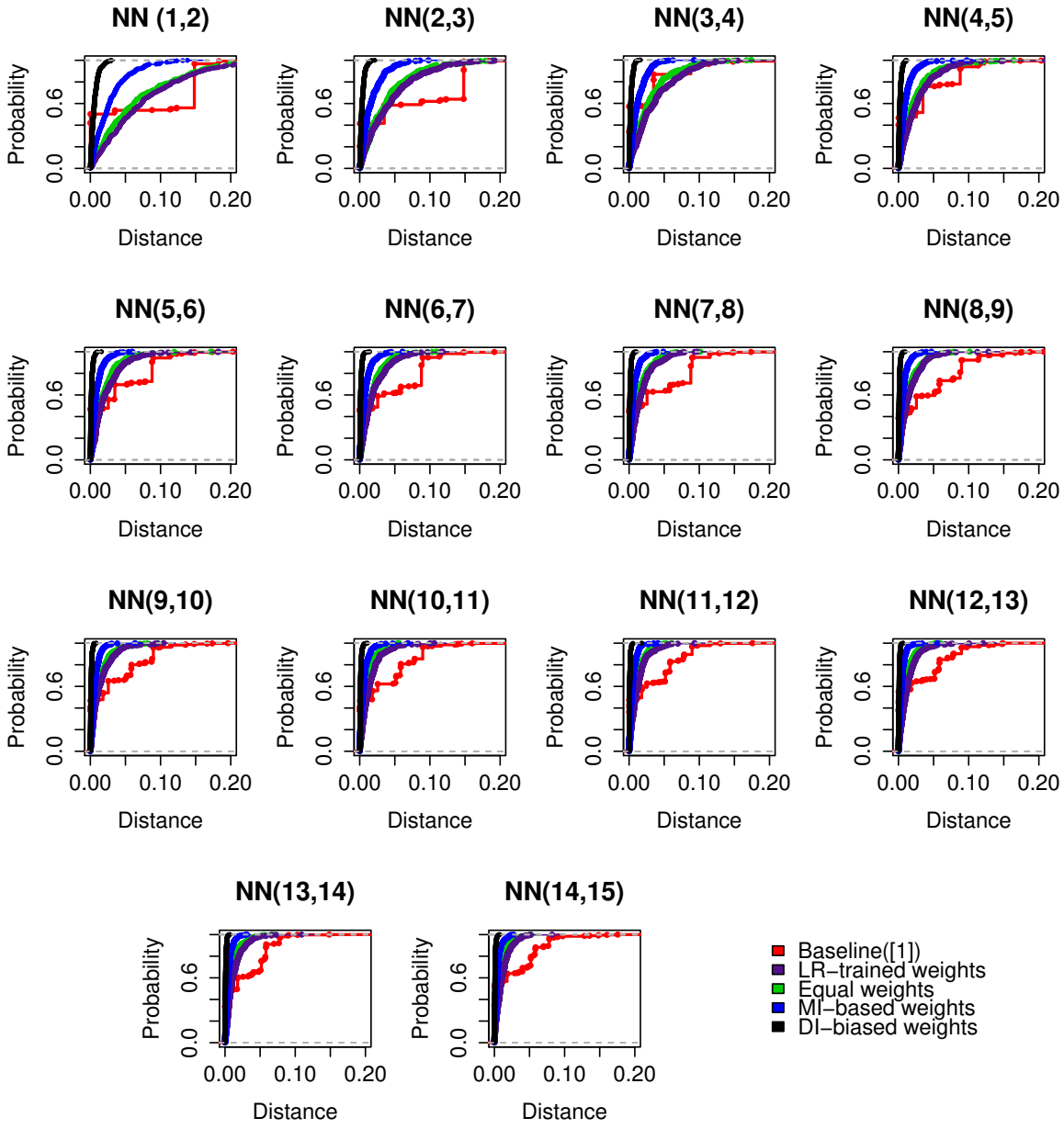


Figure 3.7: Empirical Cumulative Distribution Functions (ECDFs) of Inter Neighbor Distances: Comparison Among *LR-trained weights*, *Equal weights*, *MI-based weights*, *DI-biased weights* and the *Baseline* ([1]).

Also the density plots of inter neighbor distances support our claim. We can see that there is mostly one peak in density plots for *LR-trained weights*, *Equal weights*, *MI-based weights* and

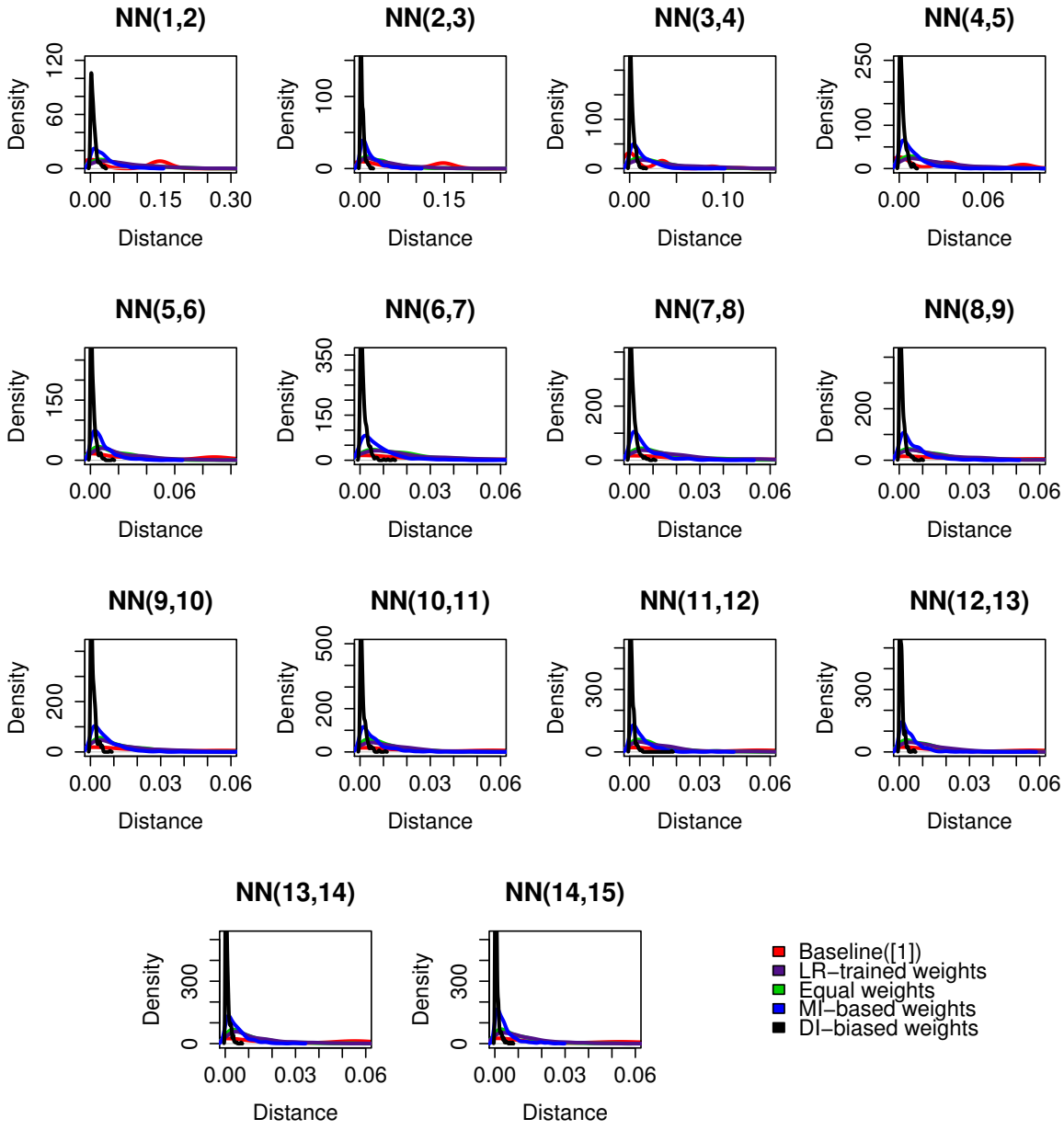


Figure 3.8: Densities of Inter Neighbor Distances: Comparison Among *LR-trained weights*, *Equal weights*, *MI-based weights*, *DI-biased weights* and the *Baseline* ([1]).

DI-biased weights as shown in Figure 3.8. This trend also remains the same when we consider the two furthest neighbors 14 and 15. On the contrary, we can see that there are more than one peak in first four density plots for the *Baseline* ([1]) approach. As it moves towards the

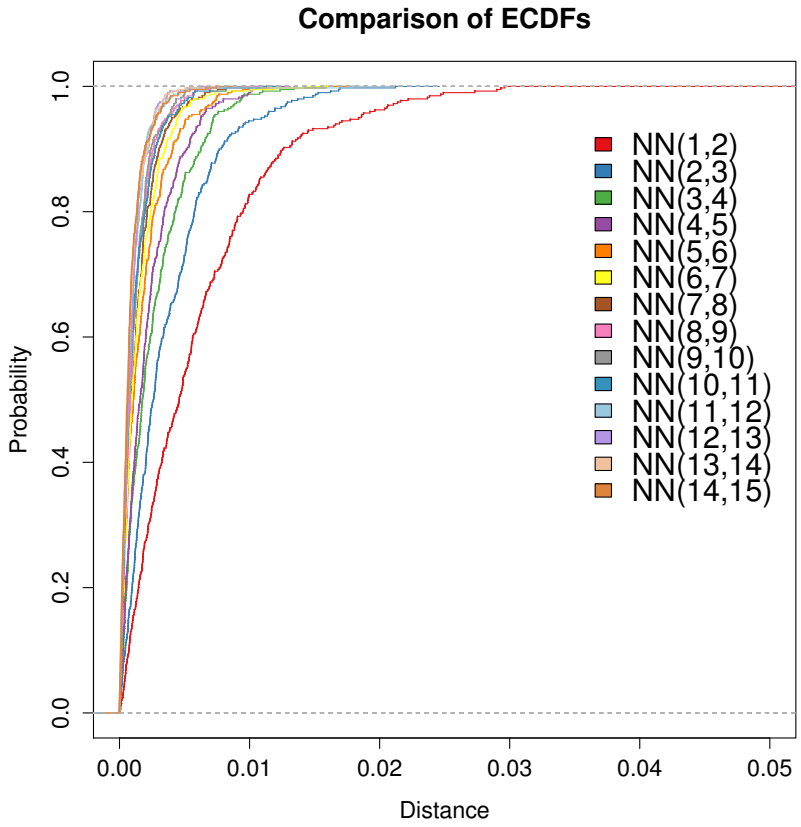


Figure 3.9: Empirical Cumulative Distribution Functions (ECDFs) of Inter Neighbor Distances Using *DI-biased weights* Showing the Improvements Achieved for Increasing k.

furthest nearest neighbors, we can see a long tail in the density distributions. This implies a sparse neighborhood using the baseline approach. From the density plots, we also can see that the proposed approach *DI-biased weights* is able to achieve the most dense neighborhood.

Next, we examine the robustness of our approach *DI-biased weights*. Specifically, we examine the changes of inter-neighbor distances with the change in k values. We plot the ECDFs and densities of all the distances between two consecutive neighbors using *DI-biased weights* only. From Figure 3.9, we can see that the first two nearest neighbors are the least dense compared to the next two, and so on. However, as we increase the size of the neighborhood, a more dense distance between two furthest neighbors is seen. This indicates that *DI-biased weights* is able to find the dense neighborhood when we increase the value of k.

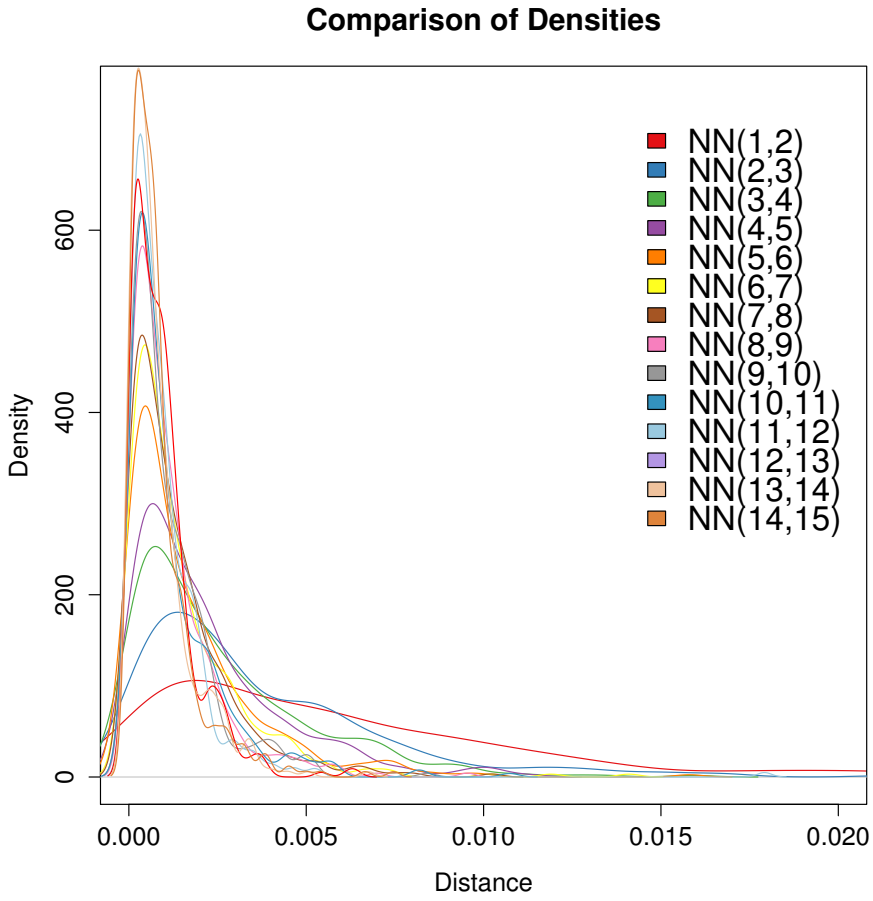


Figure 3.10: Density of Inter Neighbor Distances Using *DI-biased weights* Showing the Improvements Achieved for Increasing k.

Furthermore, from the density plots in Figure 3.10, we can see that the furthest two neighbors are more dense compared to the closest two. This implies that the *DI-biased weights* approach shows its robustness in identifying dense neighborhood with any neighborhood size between 1 and 15 in this study with taxi queue wait time dataset.

We also examine the relationship between the identified neighborhood and the taxi queue wait time prediction. Specifically, a K-S (Kolmogorov-Smirnov) test is deployed. The K-S test measures the difference between ECDFs of distances among identified neighborhoods by applying the *DI-biased weights* and *Baseline* ([1]) method respectively in terms of D-value which is the maximum difference between these two. We examine the corresponding p-values

Table 3.6: Relationships Between the Identified Neighborhood and the Taxi Queue Wait Time Prediction Errors Using K-S (Kolmogorov-Smirnov) Test which Shows the Existence of Correlations.

*ECDF*s of distances between $NN(p,q)$	D (K-S Test)	p	k	*d_error* (Median)	*d_error* (Mean)	Correlation $[D, d\_error(\text{Median})]$	Correlation $[D, d\_error(\text{Mean})]$
(1,2)	0.50	<0.001	2	0.017	0.017		
(2,3)	0.46	<0.001	3	0.010	0.013		
(3,4)	0.55	<0.001	4	0.012	0.011		
(4,5)	0.45	<0.001	5	0.007	0.010		
(5,6)	0.47	<0.001	6	0.015	0.014		
(6,7)	0.46	<0.001	7	0.014	0.011		
(7,8)	0.45	<0.001	8	0.014	0.012	0.484	0.393
(8,9)	0.50	<0.001	9	0.011	0.012		
(9,10)	0.47	<0.001	10	0.014	0.011		
(10,11)	0.47	<0.001	11	0.014	0.011		
(11,12)	0.48	<0.001	12	0.013	0.010		
(12,13)	0.47	<0.001	13	0.011	0.010		
(13,14)	0.48	<0.001	14	0.010	0.009		
(14,15)	0.45	<0.001	15	0.009	0.009		

to know the statistical significance of this difference. As shown in Table. 3.6, the D-value is around 0.45 and p-value < 0.001, which means the neighborhoods are statistically different for different k-values and the difference would be statistically as large or larger than the observed ones. Then, let $d\_error(Median)$ and $d\_error(Mean)$ denote the improvement shown by the *DI-biased weights* method over the *Baseline* ([1]) method in terms of median and mean of prediction errors respectively for different k-values. Finally, the correlation between the corresponding D-values and the prediction errors ($d\_error(Median)$ and $d\_error(Mean)$) are measured to show the relationship between the improvement in dense quality neighborhood and the improvement of prediction accuracy. As shown in the last two columns of Table 3.6, the Pearson correlation scores of 0.484 (with median) and 0.393 (with mean) are obtained which is a positive correlation. It indicates that the improvement in terms of dense quality neighborhood is correlated with the improvement in terms of prediction accuracy.

In total, the experiment results demonstrate that the heterogeneous contextual factors together with the driver intelligence (DI) can improve the quality of identified neighborhood significantly, which leads to a significant improvement in taxi queue wait time prediction.

3.4.2 Scenario 2: Queue Context Prediction

Let T_i be an instance which represents the current hourly time window in the queue context dataset. Each instance T_i is described by a set of contextual features and a queue context $c(T_i)$. Given a query time window T_Q which represents the next hourly time window and corresponding set of features F_c, we predict the taxi-passenger queue contexts as: $(T_Q, F_c) \rightarrow \hat{c}(T_Q)$ where $\hat{c}(T_Q)$ is the predicted queue context. We aim to perform feature selection and compute feature weight for good quality neighborhood calculation. We begin with the formulation of kNN methods. Then we present our technique for feature weight calculation and experimental results. The prediction steps are described in the following subsections.

3.4.2.1 Formulation of k-NN Methods

Let each sample T_i in the queue context dataset is described by a d-dimensional vector of relevant features and a target context label: $\langle a_1(T_i), a_2(T_i), a_3(T_i), ..., a_d(T_i), c(T_i)\rangle$ where, $c(T_i)$ is the queue context label and $c(T_i) \in \{TQ, PQ, TPQ, NoQ\}$.

To predict the target queue context for any query instance T_Q, the distances between T_Q and all the training samples T_i denoted as $d(T_Q, T_i)$ are calculated as follows:

$$d(T_Q, T_i) = \sqrt{\sum_{j=1}^{d} [a_j(T_Q) - a_j(T_i)]^2} \tag{3.6}$$

where $a_j \in F_c$ is the j^{th} contextual feature of T_i. and T_i:

Unlike the basic k-NN method which treats each feature equally during this distance calculation, the contribution of each feature can be taken into account by multiplying with the feature importance score. If ω_j is the feature importance score of j^{th} feature, we rewrite the Eq. 3.6 as follows:

$$d(T_Q, T_i) = \sqrt{\sum_{j=1}^{d} \omega_j * [a_j(T_Q) - a_j(T_i)]^2} \tag{3.7}$$

Next, the k-nearest neighbors of T_Q are selected by observing the values of $d(T_Q, T_i)$ and sorting them in ascending order. Let us assume $\{T^{NN} = T_1^{NN}, T_2^{NN}, T_3^{NN}, ..., T_k^{NN}\}$ is the set

of k-nearest neighbors of T_Q based on k smallest $d(T_Q, T_i)$. The predicted target score $\hat{c}(T_Q)$ of the query instance T_Q is calculated by applying majority voting technique within the target context labels of k-nearest neighbors as follows:

$$\hat{c}(T_Q) = \arg\max \sum_{i=1}^{k} \delta(c, c(T_i^{NN})) \tag{3.8}$$

Note that the key is to compute the appropriate feature importance score in Eq. 3.7 to achieve a higher prediction accuracy through identification of good quality neighborhood of size k.

3.4.2.2 Feature Weight Calculation Scheme

In this section, we calculate the feature importance score for queue context prediction. To do so, we compute the driver intelligence by following the process described in Section 3.4.1.3. Then we calculate deviation of this 'driver intelligence' from the hourly mean frequency for each hour. We call this number as the 'Temporal Drivers-intelligence Deviation' (TDID). Note that we calculate the hourly 'driver intelligence' deviation for each instance in the queue context dataset. Then we employ the notion of mutual information to calculate the feature importance score. The mutual information is a measure of the mutual dependence between two random variables. In this research, we use the TDID as a conditional variable for calculating TDID-biased mutual information between any feature and the queue context. We calculate the TDID-biased mutual information, $I(a_j; c(T_i)|TDID)$ as:

$$I(a_j; c(T_i)|TDID) = - \sum_{a_j, c(T_i), TDID} p(a_j, c(T_i)) \log \frac{p(a_j, c(T_i)|TDID)}{p(a_j|TDID)p(c(T_i)|TDID)} \tag{3.9}$$

Here, a_j is the j^{th} queue context feature and $c(T_i)$ is the taxi-passenger queue context and $TDID$ is the temporal driver-intelligence deviation. Next, we normalize these values of $I(a_j; c(T_i)|TDID)$ to be used as feature importance score.

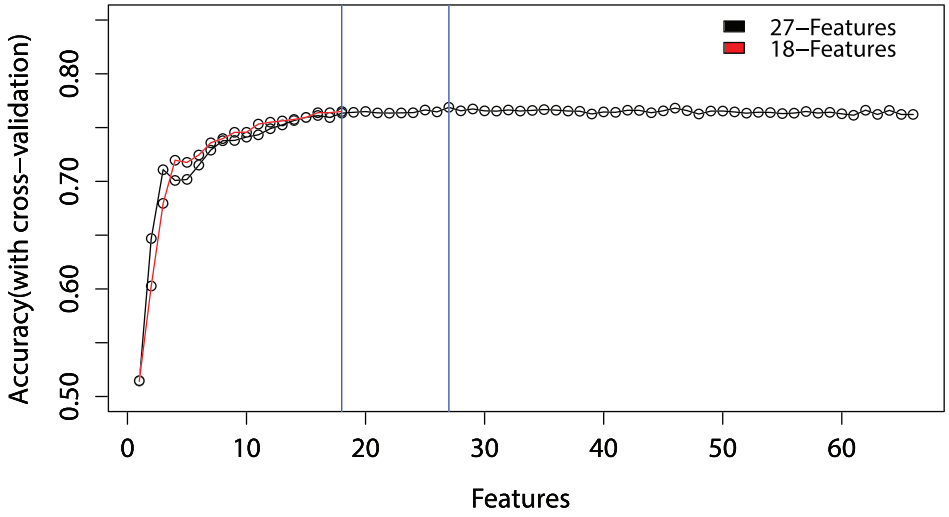


Figure 3.11: Accuracy (%) vs Number of Features

3.4.2.3 Experiments and Results: Feature Selection

As we extract more features by computing the deviations of all feature values from its hourly mean along with the current features of the queue context dataset [26], it is necessary to check the relevancy of all features. The reason is that the use of all these features may degrade the prediction performance significantly due to the inclusion of some irrelevant and redundant features. Therefore, to build an accurate model, it is required to identify the relevant features from this large feature set of 66 features. The automatic feature selection techniques are considered as an effective tool in this scenario. We use a well known automatic feature selection technique called Recursive Feature Elimination (RFE) [120]. In each iteration of REF, a Random Forest algorithm is employed and the model is evaluated. All possible subsets of the features are considered. Figure 3.11 shows that a subset of 27 features produces the maximum accuracy (76.58%).

Next, we take this subset of 27 features and check for feature redundancy. Specifically we examine if this subset contains attributes that are highly correlated with each other. To remove this problem we generate and analyze a correlation matrix between all 27 attributes. Then the highly correlated attributes are identified based on a cut-off threshold. We remove

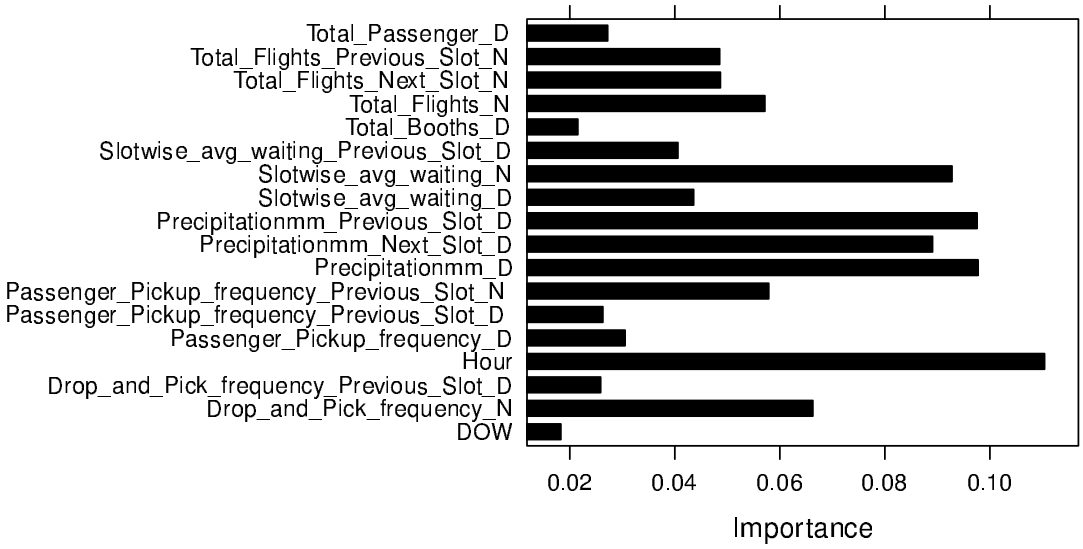


Figure 3.12: Feature Importance Scores Based on TDID-biased Mutual Information

9 queue context features with an absolute correlation of 0.75 or higher and obtain 18 relevant features for the queue context prediction. These 9 features include normalized passenger pickup frequency, total booths, and total passenger corresponds to the current hourly time window; normalized slotwise avg waiting, drop and pick frequency, total booths, and total passenger corresponds to the previous hourly time window; and normalized total passenger, and total booths corresponds to the next hourly time window. Then we apply the recursive feature elimination technique again within the dataset of 18 features to confirm that no more feature is selected to be removed and the maximum accuracy of 76% is obtained using the subset of all of those 18 features. Since this feature reduction shows no significant change in prediction accuracy, we keep both of the datasets with 18 and 27 features respectively for the purpose of comparison. We apply our feature importance calculation technique in both datasets and observe the prediction performance.

3.4.2.4 Experiments and Results: Feature Weight Calculation

After feature selection, we calculate the feature importance scores for each feature in the queue context dataset by normalizing the values of $I(a_j; c(T_i)|TDID)$ between 0 and 1. Figure 3.12 illustrates the feature importance scores for our dataset with 18 features. We perform the

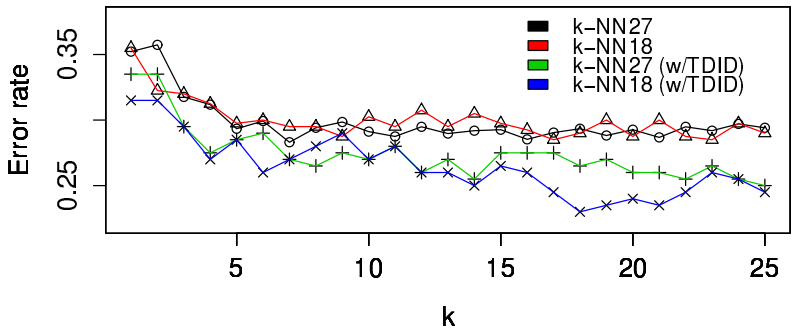


Figure 3.13: Comparison of Error Rates (%)

same for our another dataset with 27 features. Then we use these normalized scores during the distance calculation of k-NN methods. Note that the features listed in Figure 3.12 that end with a 'N' correspond to the features with normalized values while the features that end with a 'D' represent the deviation of that feature from the hourly mean values.

3.4.2.5 Analyzing Prediction Performance

In this phase, we apply different k-NN methods to both our datasets with 27 and 18 features respectively. For the experiment, we use 10-fold cross validation mechanism to test the performance of k-NN methods. Specifically, we compare the prediction performance by observing the error rates between traditional k-NN method and the k-NN (w/TDID) method. Note that the traditional k-NN methods considers each features with similar importance. Unlike traditional k-NN, the k-NN (w/TDID) incorporates the temporal driver-intelligence deviation (TDID) as feature importance scores during distance calculation for neighborhood selection. We do this comparison of the prediction error for varying k-values between 1 and 25. Figure 3.13 illustrates the error rates of different techniques. For clarification we denote the traditional k-NN methods as k-NN27 and k-NN18 when applied to our datasets of 27 and 18 features respectively. Similarly the k-NN27 (w/TDID) and k-NN18 (w/TDID) stand for the k-NN methods with temporal driver-intelligence deviation (TDID) based feature importance score.

We can see from Figure 3.13 that the the k-NN27 (w/TDID) and k-NN18 (w/TDID) produce less error rates compare to k-NN27 and k-NN18. This implies that the use of temporal

Table 3.7: Results Obtained From Paired t-Test

Paired t-test Between k-NN Methods {a,b}		p-value	95% CI
{k-NN18,	k-NN18 (w/TDID)}	3.31e-10	(0.028, 0.043)
{k-NN27,	k-NN18 (w/TDID)}	1.184e-10	(0.028, 0.041)
{k-NN18,	k-NN27 (w/TDID)}	2.333e-09	(0.020, 0.031)
{k-NN27,	k-NN27 (w/TDID)}	5.719e-11	(0.020, 0.029)
{k-NN27 (w/TDID),	k-NN18 (w/TDID)}	0.001466	(0.004, 0.016)

driver-intelligence deviation (TDID) based feature importance scores with the neighborhood-based methods can select good quality neighborhood thus lower the error rates. We also can see that the k-NN18 (w/TDID) produces the error rate which is smallest among these four for almost all k values. This implies that the k-NN method perform better in our scenario when applied to the dataset with reduced features.

Next, we examine the significance of this improvement of using the TDID-based feature importance scores with the neighborhood based methods. We conduct a paired t-test. The paired t-test examine and determine the statistical evidence that the mean difference between paired observations is significantly different. A mean difference of 0 implies no difference. We perform the paired t-test for all the error rates obtained for varying k-values. Let, a and b represent two matrices that contain the error rates for different k-values between 1 and 25 for two different methods say, k-NN18 and k-NN18 (w/TDID). The paired t-test (a,b) returns an interval of differences for a given confidence interval (CI). We can see from Table 3.7 that this difference in error rate reduction is significant with a 95% CI since the value 0 is outside these intervals. We also can see that the k-NN18 (w/TDID) shows the most significant improvement compare to others. The lower p-values ($p < 0.05$) also strengthen the claim.

3.5 Conclusion

This chapter focused on the problem of queue context and queue wait time prediction at the airport by using neighborhood based methods. Specifically, we investigated a large number of heterogeneous mobility-associated factors related to the mobility of taxi drivers and passengers of the JFK airport in New York City, including time of day, week, month, taxi trips to and

from the airport, flight arrival times and passenger numbers as well as features related to weather conditions. To conduct this research, we utilized two real-world mobility context datasets: taxi-passenger queue context dataset and taxi queue wait time dataset which were generated by fusing three real-world datasets: taxi trip data, airport passenger arrival data and weather condition data. We also conducted a comprehensive analysis on the associated factors. Then, we devised methods to select relevant factors and introduce a feature weighting scheme leveraging taxi driver intelligence to identify dense quality neighbors for k-NN based methods.

The experimental results show that our feature weighting scheme enhanced the performance of the state-of-the-art k-NN model for mobility context prediction. We also can see from the results of paired t-test with 95% confidence level that the improvement in obtained results is statistically significant compared to baselines. Furthermore, the results obtained from inter-neighbor distance analysis demonstrate that our method identified dense neighborhoods for varying neighborhood sizes, which was also the reason for this significant improvement in prediction accuracy. Our research suggests that such results obtained for the mobility context prediction can help taxi drivers' decision in terms of making an airport trip or not. Also, passengers can seek alternative transport if a long queue is forecasted. Overall, our approach has the potential for practical implementation as shown by the experiments with real-world datasets. This could help the taxi drivers decide to make an airport passenger pickup after a precedent airport passenger drop-off thus reducing their queue wait times at the airport taxi rank.

However, the results obtained from this research are restricted to prediction and cannot provide optimal decision-making solutions, which require other types of modelling. Future research can address the problem of optimal decision making for taxi drivers by analyzing their personalized objectives. As for any predictive analytics technique, the performance of our approach also depends on and requires appropriate domain adaptation. We inferred expert-like knowledge by combining historical mobility context data and the probability theory for the calculation of feature weights. We deployed our techniques during distance calculation of k-NN based methods only in our experiments. In future, another validation experiment with

our technique could be performed when the similar kind of mobility context datasets become available.

In summary, this chapter demonstrated the modelling of heterogeneous mobility-associated factors for expert-like knowledge acquisition to enhance the performance of mobility context prediction task by calculating and assigning appropriate feature weights.

Chapter 4

Inferring and Integrating Mobility Contexts in Trip Planning

In Chapter 1, we discussed context-aware trip planning and the different user perspectives of a context for trip plan recommendation. Inferring sparse mobility contexts from heterogeneous data sources and integrating multiple contexts into the trip plans can be challenging problems for effective trip planning. In relation to RQ-3, this chapter summarizes the existing literature to develop a conceptual framework for providing different steps and procedures of context-aware trip planning. The steps include contextual data collection, fusion, inference, representation and integration into trip planning. Moreover, we develop several algorithms to address the multi-context integration problem in context-aware trip planning where the main goal is to consider different user-perspectives of a mobility context and integrate them into trip planning. To demonstrate the reasonableness of our developed technique, we consider an active transport trip planning scenario with real-world deployment. We also present the experimental results to illustrate the effectiveness of our developed methodology.

4.1 Motivation and Contribution

Active transport trips refer to the collection of non-motorized forms of transport options such as manual wheelchair, pedal bike, push scooter and walking. These are important to user

mobility because the trips using active transport require active human effort and able to bring long term benefits. The urban planner are interested because increasing active transport usage can lower the traffic congestion and reduce greenhouse emissions. Policy makers all around the world are calling for rapid increases in active transport usage in daily travelling. To promote the active transport usage, many exclusive services are provided to active transport users such as dedicated bike lanes and walking trails. Also, the health professionals, recommend active transport to increase independent mobility for different age groups by making them flexible physically, mentally and socially [121, 122]. However, the adaptation of active transport modes requires many situational factors to be considered. For instance, a user's trip from one place to another can be influenced by various factors which may include quietness, steepness, and congestion along the route. Physical barriers such as stairs, ramps, closures can directly influence the preferences of some active transport users for planning and making their trips. Most importantly, these factors vary among travellers. Therefore, the active transport trip planning should consider these diverse situations known as trip contexts for computing traveller specific trip plans. One of the challenges faced by the active transport trip planners includes designing a unified framework to collect, fuse, infer and represent contextual information for the computation of trip plans according to user preferences.

Another motivation is that the world's population is ageing and people aged 65 and above is increasing at a high rate. According to World Banks report [123], Australia has 15% of their total population aged 65 and over in 2014 whereas in United States, Singapore and Portugal this rate is 14%, 11% and 19% respectively. Globally, within the older population group, persons aged 80 years or over account for 14% of the total population in 2013. This is projected to reach 19% in 2050 which is equivalent to 392 million persons aged 80 years or over by 2050 [124]. So, it is becoming more and more important to consider the special needs of this increasingly large number of people when developing public services for them. Among the various special needs of elderly people, mobility needs are becoming more important as more people retire from driving, and thus require improved trip planner options which may combine accessible public transport and walking routes to meet their mobility needs. For example, they will require mobility to access health care services, various social activities, shopping,

and simply maintain community connections. However, there are many perceived barriers which limit the usual mobility requirements of elderly people and those with special needs. Of these barriers, accessibility issues are considered the most important as highlighted in [125]. A review in [126] shows that public transport has a significant influence on access to various health services for elderly people and those who stop driving their own vehicles. Other research points out that special consideration must be given while constructing or upgrading road and footpath infrastructure [127], for instance, as this can impact on pedestrians who use a cane, guide dog or wheelchair. It is important to meet everyone's mobility needs so that they do not become isolated from society [128]. Although there has been improvement in aspects of public transport and civil engineering to improve accessibility, the following research question still remains:

"Which is the most accessible route to take between two points-of-interest (POIs) within walking distance?"

By point-of-interest, we mean a place where a trip starts or finishes (e.g. home, hospital, public transport station, or community place). The proliferation of mobile technologies and navigation services can help to provide solutions to this question. It has now become easier to go from one place to another by using various navigation devices. Route recommendation systems that are available compute choices of routes from a list of recommendations based on various criteria such as shortest route and fastest route. Although these systems are built to help people to be mobile, they cannot always satisfy every type of user. For example, a person with a manual wheel chair, who may be querying a route recommender system to travel between two locations, may not be satisfied with the outcome of their query. He may be directed to a path which is inaccessible or too steep and risky for him. This happens as the recommender only considers paths that are shortest and fastest. But, for this special user, route accessibility is the main key factor that needs to be considered. Moreover, recommendations for a route based on accessibility needs depend on the person's physical capability. For a daily commute, it may be considered less necessary to use a recommender system since the user would be well aware of the environment. However, the situation is different when the user wants to visit a new or unknown place. It is particularly necessary to design an accessible path recommendation

for the elderly people and people with special needs to fit their physical abilities.

Route accessibility is very difficult to model as there are many factors that can affect the accessibility of a route. Of them, the most influential factor is the gradient of the route. To be specific, people with a wheelchair can comfortably wheel themselves up a specific gradient but not beyond a slope of one-in-fourteen [129]. The United Nations has also provided a design manual for a barrier free environment for people with mobility problems[130]. How to give a route recommendation that is accessible for a wheelchair is a challenge, since there are so many possibilities that can happen along the path. For example, a very smooth route can be rendered inaccessible by a very sharp rise in gradient over a very small portion of the route. On the other hand, there may have several routes with a gentle rise in gradient in several portions but all of them could be accessible because this rise is below a certain margin. The challenge is to pick the best route from all the latter options. The existing path planning algorithms try to minimize the total travel distance or travel time. However, there is no measure defined for evaluating the accessibility of a path either. In the accessible path routing problem, there are the following challenges: *First*, the current network graph used for trip planning does not take into account the slope of the paths, and thus does not support the accessibility optimization. *Second*, there is no measure for evaluating the accessibility of the path. *Third*, there is no algorithm proposed for the accessible path routing.

To address the above challenges, this chapter presents a unified framework called context-aware active transport trip planning (CoAcT). We also propose technique to integrate multiple contexts (i.e. accessibility and distance) in trip plans considering different user-perspectives of a context. We demonstrate our developed techniques using a number of real-world deployments and experiments. Specifically, the contribution of this chapter is listed as follows:

- A unified framework to collect, fuse, infer and represent contextual information for providing context-aware trip planning.
- A Contour-based graph generation and query-based adaptation scheme is proposed to represent the slope of the paths in the graph with the aid of contour lines;

- Two metrics, total vertical distance and maximal slope, are defined for evaluating the accessibility of a path;
- A Multi-Objective A* search algorithm is designed for minimizing the total distance, total vertical distance and maximal slope;
- A diverse set of trade-off paths is provided, including the shortest path. The users can choose the most suitable path according to their own perspective of contexts (i.e. distance and accesibility).

4.2 Related Work

Some researchers focus on identifying different mobility aspects for the elderly and the people with special needs. Another direction of research focuses on different techniques for collecting information on physical accessibility barriers along the path whereas a number of researchers consider different parameters for calculating the score of a path. Also, there is another direction of research where the aim is to develop systems for mobility assistance.

4.2.1 Mobility Aspects for the Elderly and People with Special Needs

Several surveys have been conducted to identify the mobility aspects and accessibility barriers for the elderly and people with special needs [131, 132, 133, 38, 134]. A spatial analysis of accessibility of train stations and access to their surroundings for elderly passengers is presented in [131], where the authors leveraged data from State Government organizations and conducted a field survey of seven railway stations in Perth, Western Australia. The data collected from the survey identified the trip purposes and attitudes towards accessibility for the elderly travelers. This research found that accessibility at the train station and surrounding areas is affected by route directness, facility and service quality at station, mixed land use, and intermodal connectivity. The researchers calculated accessibility indices for train stations and surroundings by combining elderly patronage rates and identifying variables that affect accessibility. They classified the data into three types of elderly passengers: those who walk and ride, park and ride and those who take the bus and ride, since the main form of public transport in Perth is bus.

However, the research did not consider the route accessibility that can have an impact on the elderly and passengers with special needs and influence their attitudes towards their patronage of public transport in a major way. The research in [132] presented a way to determine the accessibility of public transport and evaluate the service quality by analyzing pervasive mobility data. The authors in [133] conducted a survey to learn about the opportunities and barriers associated with ridesharing from an elderly person's point of view. Some research also focuses on blind passengers' travel needs. An interview with a group of blind and deaf-blind public transport users revealed that they are primarily concerned with independence and safety [38]. In [134], the routing behaviour of pedestrians in an indoor environment is investigated by evaluating responses to active RFID and QR-code based route navigation systems for blind people. Such systems were also evaluated in [135], which recognized that all of these systems must work in an integrated manner to achieve desired accessibility outcomes for the individuals concerned.

4.2.2 Crowdsourcing as a Tool for Data Collection and Route Recommendation

Several studies have collected data on accessibility barriers along a path through crowdsourcing [136, 137, 138, 139, 140, 141, 142, 143, 144]. Crowdsourcing has been widely used for accessibility data collection in general as well as for pedestrian navigation. An accessibility information sharing platform for people with disabilities was explained in [136], which aimed to provide disabled people with a suitable path to their destination. The authors in [137] proposed and onlined crowdsourcing techniques with the Google Street View application to identify the bus-stop landmark locations and improve the accessibility of blind riders. Crowdsourcing was also used to collect information on stop identification landmarks in [38]. A platform for collaborative accessibility map generation was proposed in [138]. The system allowed users to add photos of the side walk accessibility barriers and comment on them. The authors in [139] designed a system which they call mPASS to collect indoor and outdoor accessibility data as well as analyse many outdoor accessibility requirements. The system crowdsourced notifications about a possible accessibility barrier (such as stairs for example) to alert other users of the

system to be aware but it may also need to be confirmed. By considering the user preferences and specific needs, the system aimed to provide personalized paths for users. It stores the user profiles based on their needs and preferences, which are then updated by allowing the users to select their choices (neutral, like, dislike and avoid) against a specific accessibility barrier. A route recommendation system based on crowdsourced data was presented in [140], where the authors quantified the human perceptions of quietness; happiness and beauty to recommend paths. Crowdsourcing was also used to select a small set of paths from a large set of recommendations. A crowd-driven turn-by-turn path selection technique was proposed in [141], where the authors collect live traffic information through crowdsourcing and then ruled out the less important paths. Crowd perceptions about routing directions were collected through a series of routing questions. The research also proposed a strategy to select the most important set of routing questions. A route recommender system based on crowd-voting data from social media was introduced in [142]. The aim was to suggest the most pleasurable route for urban walking rather than recommending a route based on time/distance. A crowd-aided mobile platform for user safety perception management was presented in [143]. The authors also extended their work by finding the safest route between two locations in [144], leveraging the data collected through their mobile crowdsourced platform. Though crowdsourcing is an effective tool for data acquisition, it can suffer from various issues such as trust, missing data, incorrect data, etc. Another model to estimate the probability of a crash on any road as a function of the traffic volume, road characteristics, and environmental conditions is presented in [145]. To compute the safest route between two locations, the authors employed Dijkstra routing algorithm.

4.2.3 Measuring Route Scores

Several authors have defined a walkability score for a pedestrian route or a specific location [146, 147, 148, 149, 150, 151, 152, 12]. A model for measuring walking accessibility towards public transport terminals was presented in [146] by introducing the concept of equivalent walking distance. The equivalent walking distance is the sum of the actual walking distance plus other factors along the route (crossing, ascending steps, conflict points), the values of

which are measured by calculating the trade-off of that factor with respect to the actual walking distance. "Walkscore" is a publicly available system which provides a score for the walk and a transit score for a specific address [147]. It uses the distance of local amenities and transit facilities from an address to assign the score. A map route ranking method that considers environmental factors is presented in [148]. The direction and elevation services are used to select and rank the routes recommended by the Google Maps application [149]. However, this approach did not consider a context aware route search and the routes generated from Google Maps are based on shortest distance or minimum time and no on accessibility issues. Another model for recommending a walking route was proposed in [150]. Routes were generated by combining the A* algorithm and genetic algorithms and were evaluated against safety, amenity and walkability criteria. In the system, the user was required to enter the weights for each of these criteria to define the objective functions for each route. However, the safety was a qualitative measure and users might find it difficult to assign weights for different parameters. RouteCheckr [151] is a Dijkstra-based client/server architecture which aims to provide personalized routing to mobility impaired users. The system is based on multimodal annotation of geo-data. Users can rank their choices and then, based on the multi-criteria cost associated with each route, the best route is presented. The problem with the weighted sum approach is that all the parameters are required to be converted to a common scale. A traffic aware real-time route recommendation system was proposed in [152]. A combination of Dijkstra and A* algorithms was used to recommend the best route based on shortest time. The technique employed the real-time and historical taxi data. A bi-criteria optimization algorithm for urban navigation was proposed by [12]. The aim was to provide a set of paths that shows trade-off between distance and safety.

4.2.4 Mobility Assistance

A significant amount of mobility assistance can be made available to aid different groups of users. Considering the concept that blind travelers navigate through a place based on some landmarks, a braille-based application was developed by [38] that provides information on bus and bus-stop landmarks. It can become a problem if the landmark is not available due to any

construction work. A train station navigation application for blind passengers was presented in [153] where descriptions of the station were stored at different levels of the tree structure: overview, floors, platforms and places of interest (POIs). The system starts with a basic overview of the station, i.e., how many floors the station has and how they are numbered with respect to the ground. The user can travel floor by floor and can have various descriptions about the POIs. M3I is an interactive platform for pedestrian navigation in both indoor and outdoor environments [154]. The platform incorporates speech and gesture recognition for navigation support. A rich overview of mobility assistance systems for elderly or mobility impaired persons was presented in [155], where the authors also explain the current status and usability of such systems.

Current literature does not provide a unified framework for context-aware trip planning. Also, these research works aim to integrate a single mobility context by achieving one objective such as minimising distance, optimising safety, or increasing accessibility of the path. Moreover, it is a complex issue to combine and integrate multiple mobility contexts concurrently to compute and provide trip plans. Also, the topographical information which is one of the most influential factors in accessibility based trip planning is not considered in current research works. In the next sections, we present a unified active transport trip planning framework that combines contour information from topographical map data with road network data to model path accessibility. Additionally, we integrate two mobility contexts (i.e. the distance and the accessibility) concurrently considering different user-perspectives of a context during trip plan computation.

4.3 The CoAcT Framework

In this section, we present a unified framework called context aware active transport (CoAcT) which is designed to compute and provide trip plans based on a user query. The framework has two main components: i) contextual data collection and ii) fusion and query processor. This framework summarizes the existing solutions of single context trip planning and presents the concept of multi-context trip planning. In this chapter, we present a multi-context integration technique for trip plan computation which can be incorporated in the ‘fusion and query pro-

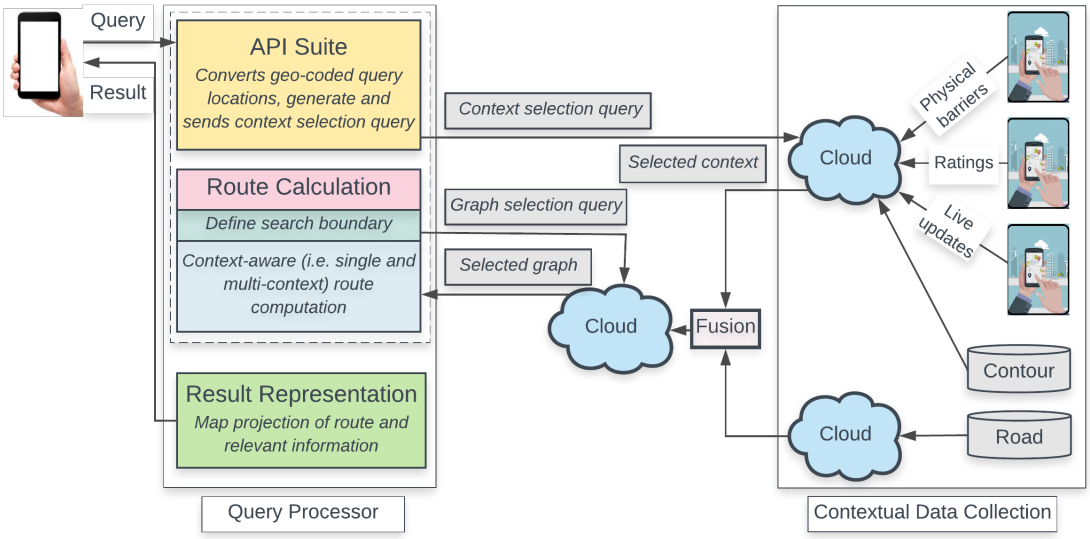


Figure 4.1: Overview of the CoAcT Framework for Context Aware Active Transport Trip Planning

cessor' module of the CoAcT framework. The core modules of CoAcT framework are shown in Figure 4.1.

4.3.1 Contextual Data Collection

The contextual data collection component collects and stores contextual data from heterogeneous data sources. As shown in Figure 4.1, the road network data is collected and stored in the cloud. The other mobility associated data from different heterogeneous sources are collected and stored into another cloud storage over time. The data associated with user mobility may include steepness, user ratings, live updates and information about physical barriers such as ramp, stairs corresponding to a specific geo location along the roads. In the next phase, these data are fused together to quantify the accessibility of a route.

4.3.2 Fusion and Query Processing

A user initiates a trip query by specifying from and to locations and trip context(s) to be considered for trip planning. These locations are usually geo-coded and hence converted into

geo-coordinates. Then based on the context(s) specified in the user query, a context selection query is generated and sent to the context cloud. The contextual data returned by the query is overlaid on to the road network data. This contextual information is assigned to corresponding units of roads called road segments. Then a search boundary is constructed by the route calculation module to limit the search space based on some parameter (i.e. user defined maximum walk-able distance). Based on the search boundary, a contextual graph is retrieved from the cloud which is used by the route calculation module. This module contains an algorithm suite consisting of different routing algorithms. The idea of using an algorithm suite is evident from the fact that there are different algorithms that serve different purposes taking various context information into account. Then a routing algorithm is employed from the algorithm suite to calculate the suitable paths based on the user specified trip contexts. The resultant trip plans are presented to the user in response to their queries. The response can include a map representation of additional information related to the trip plans such as contextual distribution of surroundings.

4.4 Context-aware Trip Planning

In this section, we present contour-based accessible path routing as an example of context-aware trip planning utilizing our CoAcT framework. We describe the process of data preprocessing and trip query handling in the following sections by presenting two scenarios of context integration in trip plan computation. In the first scenario, we compute trip plans based on a single context only. The second scenario integrates, computes and provides trip plans considering multiple contexts.

4.4.1 Data Preprocessing: Contour-based Graph Generation

Many of the papers discussed in Section 4.2 consider the road network as a representation of a graph where the road intersections are considered as the nodes of the graph. In an intersection, multiple roads cross each other. A *road segment* is referred to an edge between two nodes in the road network. It is different from a road or a street. For example, the Queen Street, an

Figure 4.2: Google Map of an area in Melbourne City, Australia.

area in Melbourne City, Australia is divided into several road segments as shown in Figure 4.2. A road segment of Queen Street can be seen between Lonsdale Street and Little Bourke Street. Another road segment is the portion of Queen Street between Little Bourke Street and Bourke Street.

Finding the shortest path between two locations in the road network is a commonly encountered problem in trip planning and tourist trip design. Here, a path is a sequence of nodes in the road network connecting with road segments. Dijkstra [156] and A* [157] search algorithms, and their variants [158, 159, 160, 161, 162] are mostly used to find the shortest path in terms of distance or travel time. There is no doubt about the effectiveness of such algorithms. However, this approach to representing the road network has a major drawback because the accessibility of one route segment might not always give a true reflection of the accessibility if the route segment is only the connection between two road crossings. It could happen that a route segment with a good accessibility rating may contain a very small portion which is wheelchair inaccessible due to a steep slope or steps. In this chapter, we consider this issue to be very important. That is, a road segment can have a number of different slopes in between corners and intersections. For example, the road segment AB has two different slopes, one upward from A (elevation of 60m) to C (elevation of 75m), and the other downward from

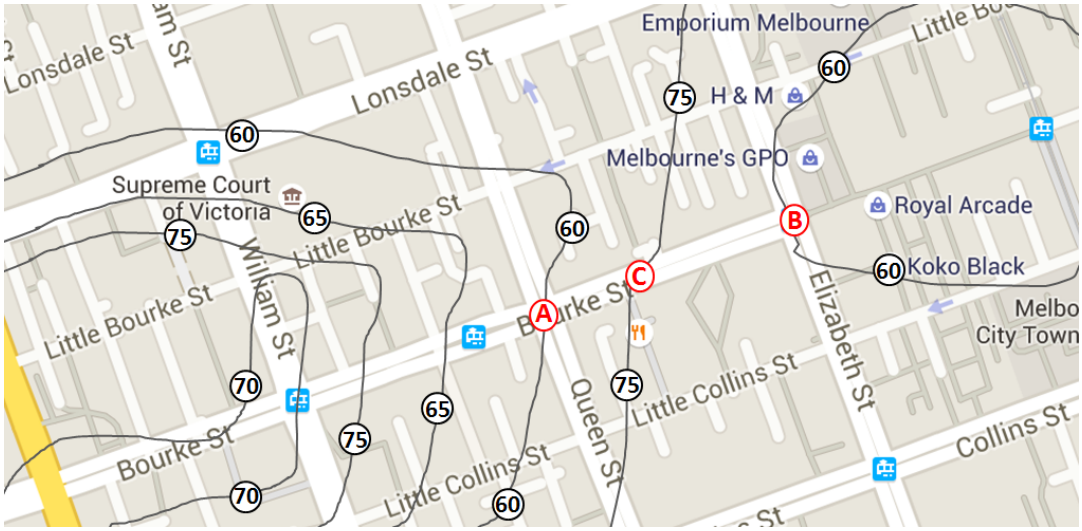


Figure 4.3: A contour map showing road segment AB with two different slopes.

C to B (elevation of 60m) as shown in Figure 4.3. The road network is a planar graph, only considering the latitude and longitude values. Therefore, the accessibility of AB given by the road network (the elevation difference between A and B) will be much different from its actual accessibility (the elevation difference between C and A, and C and B).

In practice, it is a challenging problem to identify the exact geographical locations of the turning points between the slopes in a road segment (the exact location of the point C in Figure 4.3 for example). Therefore, in this chapter, a contour-based graph generation technique is developed to approximate the locations of such turning points. Specifically, a new graph is generated by overlaying the contour lines on the road network, and adding new nodes at the cross-sections between the contour lines and the road segments. The contour lines are imaginary lines on the geographical surface connecting points with similar elevation score. The contour lines can be drawn for any elevation value on the earth's surface. Figure 4.3 can be considered as an illustration of a partial contour map of Melbourne, Australia, in which the grey curve lines on the map are contour lines. This contour map is an example of 5 meter contour interval which can be generated using the Open Street Map application, Srtm2Osm [163] for any location on the earth. The Srtm2Osm is a tool which can generate the contour lines from the digital elevation model provided by the Shuttle Radar Topography Mission (SRTM) [164].

The tool writes contours as OSM ways into an OSM file. It can be seen that there are many crossing points between the contour lines and the road segments on the map. We include these crossing points as nodes with our road network graph.

In Figure 4.3, we can see that there is only one contour line that intersects road segment AB at point C. Therefore, we add C to the road segment AB. After combining the road network and the contour lines, the nodes in the graph are defined as the union set of the intersections of the roads with the crossing points of other roads and other contour lines. As a result, there are more nodes and edges in the newly generated graph than the original one. For example, the original road segment AB is divided into two smaller segments AC and BC. The operations (i.e., intersection and union) which are required for the contour-based graph generation can be seen to be similar to the *ll_intersects* and *pp_plus* operators respectively as described by Güting et al. in [165]. The *pp_plus* operator outputs the union of two point objects. It scans and merges the point sequences from two point objects into a new points object. Given two line objects L_1 and L_2, the *ll_intersects* operator outputs whether they intersect or not. The output is true if both objects have no segments in common but at least one common point which is an intersection point but not a meeting point. Note that the elevation interval is an important parameter, since it determines both the accuracy of the turning point approximation and the number of new points added, and thus the size of the newly generated graph.

Each node in the contour-based graph, has a latitude and a longitude value given by the road network. In addition, the elevation value can be obtained by the Google Elevation API [166]. Therefore, the contour-based graph can be seen as a 3-D graph, where each node can be featured with the 3-dimensional vector (latitude, longitude, elevation). With the contour-based graph, one can calculate the elevation differences in different segments of a path much more accurately than by using only the pure road network.

In the proposed contour-based accessible path routing system, the contour-based graph is generated in the data preprocessing phase and stored in an XML file. First, the road network is extracted from Open Street Map (OSM), and contour data is extracted using Srtm2Osm. Then JOSM is used to merge the contour line data and OSM road network data. The JOSM is also used to identify the crossing points of the contour lines and the road segments. The

Algorithm 4: Data preprocessing: contour-based graph generation

1 Extract the contour lines using Srtm2Osm [163];
2 Extract the road network using Open Street Map [167];
3 Combine the road network and the contour lines using JOSM [168];
4 Identify the crossing points between the contour lines and streets;
5 Generate the contour-based graph by adding the new crossing points and edges;
6 Generate the XML file for the contour-based graph using JOSM;

JOSM is a cross-platform OSM editor. The details of the data preprocessing are described in Algorithm 4.

4.4.2 Query-Based Adaptation

The generated network graph only consists of the intersection points between the road segments and between the road segments and the contour lines. On the other hand, the query points (starting and ending points of the trip) can be anywhere on the map, and thus are highly likely to be outside the network graph. It is necessary to include the query points into the graph in real time. Intuitively, a trip must start and end somewhere in the middle of a street. Therefore, a scheme is proposed which is illustrated in Figure 4.4 and can be summarized as follows:

1. Identify the existing edge on the graph that is closest to the query point;
2. Remove the edge, and add an edge from the query point to each of the two end-nodes of the edge.

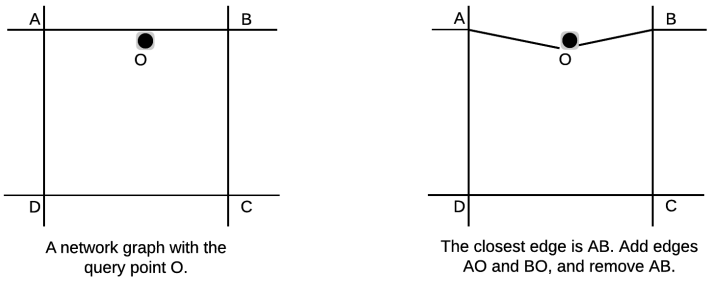


Figure 4.4: An Example of Including a Query Point into the Network Graph.

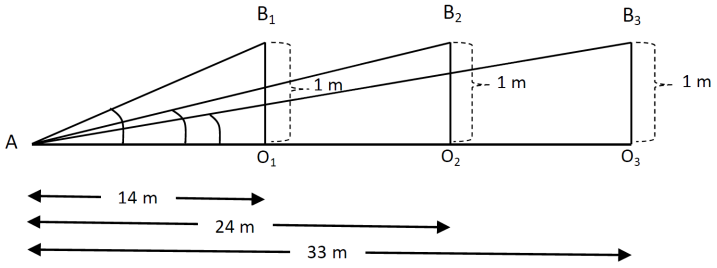


Figure 4.5: An Example of Gradient Calculation. Given a 1 meter rise, the gradients of three lines: AB_1, AB_2, and AB_3 can be calculated by dividing each rise: O_1B_1, O_2B_2, and O_3B_3 by the respective runs: AO_1, AO_2, and AO_3 of 14, 24 and 33 meters.

We can see from Figure 4.4 that given a query point O, the closest edge AB is first identified and removed. Then, the two edges AO and BO are added to the network graph. The above procedure is applied to both the starting and ending point of the trip.

4.5 Single Context Trip Planning

In single context trip planning, only one mobility context is considered during trip plan computation. This section describes two case studies of active transport trip planning using CoAcT framework and their real-world deployments considering steepness rating of the route as a mobility context for persons with limited mobility.

4.5.1 Steepness Rating of the Route

As discussed in Section 4.4.1, we use Algorithm 4 to collect, store and fuse road network data [167] with universal contour data [164] and represent it as a road network graph. Each edge of this graph represents a road segment with a respective steepness rating. The steepness ratings of different route segments are calculated according to the longitudinal grades for footpaths, walkways and bikeways [169]. These grades state that the construction should consider the maximum gradient a person with limited mobility can raise themselves comfortably is 1:14 (i.e. 1 meter in vertical rise to 14 meters of horizontal run) and a landing is required every 9 meters for them to rest or change direction. If the footpath is flatter than 1:33 (i.e. 1 meter in vertical rise to more than 33 meters of horizontal run), no landings are required. Figure 4.5 shows 1

meter vertical rises to three different horizontal runs of 14 meters, 24 meters and 33 meters from A to B_1, B_2 and B_3 respectively. We evaluate all the edges (road segments) of our road network graph based on these three ratios of rises to runs for assigning edge weights. If the gradient of a road segment is flatter than 1:33, it is considered to be an ideal road segment and assigns a steepness rating of 1. If the gradient is between 1:24 and 1:33, the road segment (i.e. the equivalent edge in the road network graph) is assigned with a steepness rating of 2. If the gradient is between 1:14 and 1:24, the steepness rating of the road segment is 3 and if the gradient is steeper than 1:14, the road segment is considered inaccessible and the equivalent accessibility rating is ∞. We use the Google elevation API [166] to find the steepness of a road segment. Later gradients of the edges are calculated and the road segment is assigned with equivalent steepness rating.

Note that we follow two different approaches to construct road segments. In *approach 1*, we consider the intersections of roads as nodes in the road network graph and the corresponding edges are the road segments. In *approach 2*, we consider intersections of roads and intersections between road and contour overlay as nodes in the road network graph and the corresponding edges are the road segments. Since the road network graph is complete with all the edges (road segment) with steepness ratings, the CoAcT framework is ready to handle a trip planning query. The query processing starts with the CoAcT query processor receiving a user query for trip planning between two locations. In our case studies, the trip queries are between two places where one is a public transport (train) stop and another POI is a restaurant or home location. The reason for choosing public transport is evident from the fact that the usage of public transport incurs some unavoidable portions of active transport i.e. walking to complete a journey. The start/destination locations specified by the trip query are geo-coded and need to be converted into geo-coordinates. For this purpose, the CoAcT framework uses the Google Geocoding API [170] that converts geo-coded locations to corresponding geo-coordinates.

4.5.2 Route Planning

At this stage, a route planning algorithm is employed to calculate the best route in terms of steepness from the road network graph. In this paper we use the A* algorithm [157] to

compute our routes. Specifically we apply A* algorithm on three occasions. First, we apply the traditional A* algorithm which computes the shortest path in terms of distance. We denote this deployment as A*(Distance). Next we apply the the A* algorithm considering road segments between two road cross-sections and denote it as A*(Steepness-road network) *approach 1*. Then we apply the A* algorithm considering road segments between one road cross-section and one road-contour cross section and denote it as A*(Steepness-contour) *approach 2*. Finally, we show all the routes on a Google map along with the Google route to illustrate the effectiveness of this real-world deployment.

The computed route is presented to the traveller with relevant information about the computed route. To reduce the search space of the routing algorithm, a search boundary is constructed. We initially consider a circular bounding area where the start/destination locations are considered as two ends of the diameter of the bounding circle. This approach may lead to a poor result since there may exist a node just outside the bounding area adjacent to origin and destination locations through which a less steep route can be suggested. At the same time it is not appropriate to search the entire road network. So, a parameter d is used that controls the diameter D' and hence the bounding area. The parameter d is tuned based on the maximum walk-able distance set by a user. The diameter of the circle is extended in both directions so that it remains within the maximum walking distance. If $D(s,t)$ is the distance between a point of interest (POI) and a preferred public transport stop, the diameter of the bounding circle is given by, $D' = D(s,t) + d$.

4.5.3 Experimental Studies

In the first case study, the suburb of Rosanna, a suburb in the north-east region of the city of Melbourne, Australia, is chosen. In another case study, we choose Heidelberg, another suburb in the north-east region of the city of Melbourne, Australia. The reason for choosing these two location is that these suburban areas are located in steep contours and hence provide good examples for active transport trip planning which considers the steepness of the route as a mobility context .

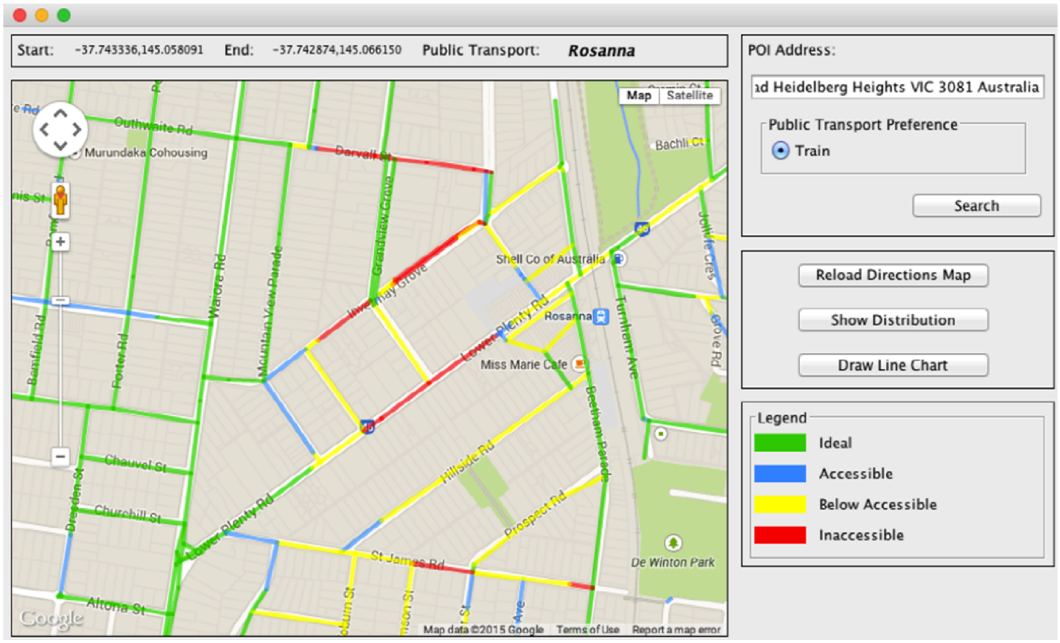


Figure 4.6: Accessibility Distribution of Surroundings in Rosanna, Melbourne, Australia

4.5.3.1 Case Study-1: Rosanna

For the first case study at Rosanna, we choose Rosanna train station and Waiora Road Medical Service as start and destination locations respectively. First, we compute the accessibility status of the surroundings connecting these two locations. For this, we refer to the steepness ratings defined in this article following the standard of building code of Australia [169]. Figure 4.6 shows the accessibility distribution of the surrounding bounding area between Rosanna train station and Waiora Road Medical Service considering the steepness ratings. We can see that some road segments are marked with red color which indicates that these segments of roads are too steep for a person with limited mobility and hence considered inaccessible. On the other hand the road segments marked with green color are completely accessible according to the steepness rating. We also can see that some segments are labelled with blue color. These road segments are acceptable with perhaps a small compromise in regards to comfort level, while road segments labelled with yellow color are more difficult compared to blue segments but somewhat walk-able. Then we apply the A*(Distance), A*(Steepness-road network) and A*(Steepness-contour) to provide a comparison of steepness of different routes

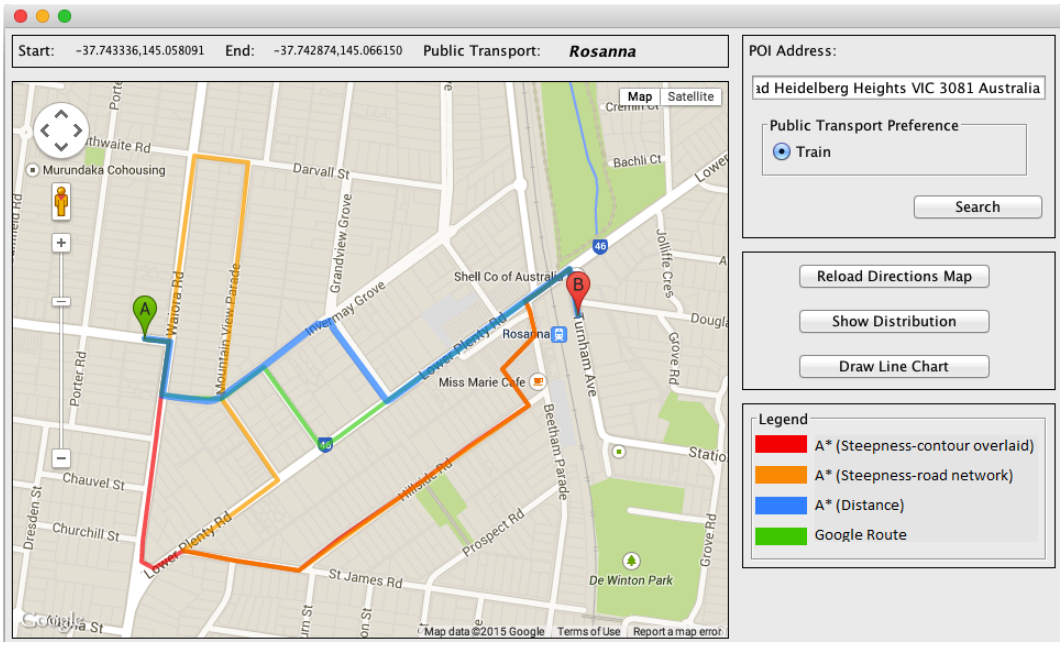


Figure 4.7: Steepness Context Aware Trip Planning in Rosanna, Melbourne, Australia

between the start/destination locations. Figure 4.7 shows the routes on the Google map given by A*(Distance), A*(Steepness-road network), A*(Steepness-contour) and Google.

4.5.3.2 Case Study-2: Heidelberg

For the case study at Heidelberg, we choose Heidelberg train station and Coconut Lagoon restaurant as start and destination locations respectively. First, we compute the accessibility status of the surroundings connecting these two locations. For this, we refer to the steepness ratings defined in this article following the standard of building code of Australia [169]. Figure 4.8 shows the accessibility distribution of the surrounding bounding area between Heidelberg railway station and Coconut Lagoon restaurant considering the steepness ratings. We can see that the road segments surrounding Heidelberg train station and within maximum walk-able distance are mostly inaccessible. Trip planning must consider this case so that a route with lowest inaccessible segment can be provided. Next we apply the A* search algorithm to find the least steep route between our start/destination locations. Figure 4.9 shows all four routes computed by A* (Distance), A* (Steepness-road network), A* (Steepness-contour) and Google.

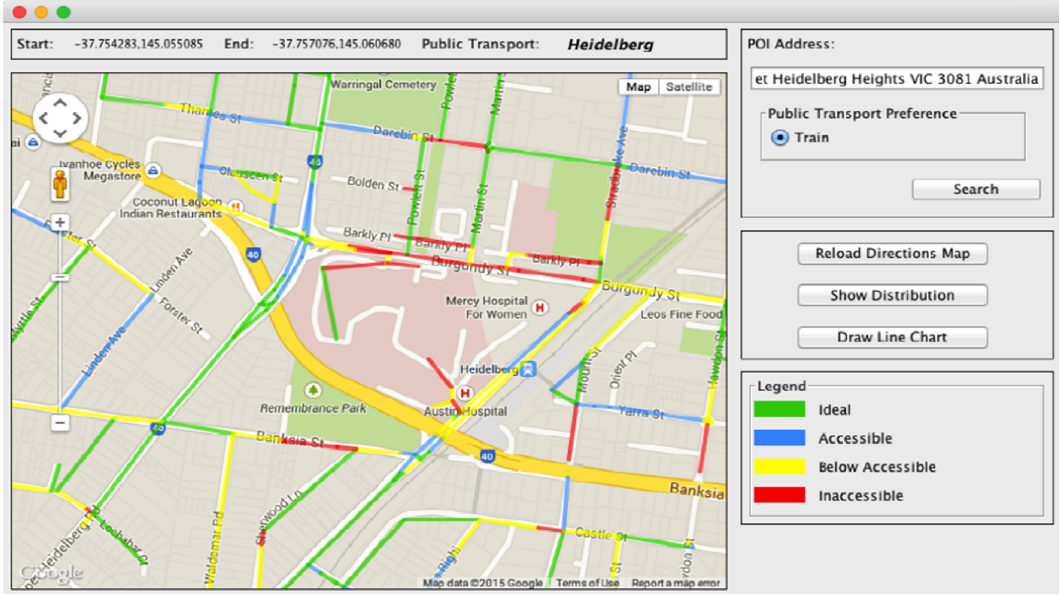


Figure 4.8: Accessibility Distribution of Surroundings in Heidelberg, Melbourne, Australia

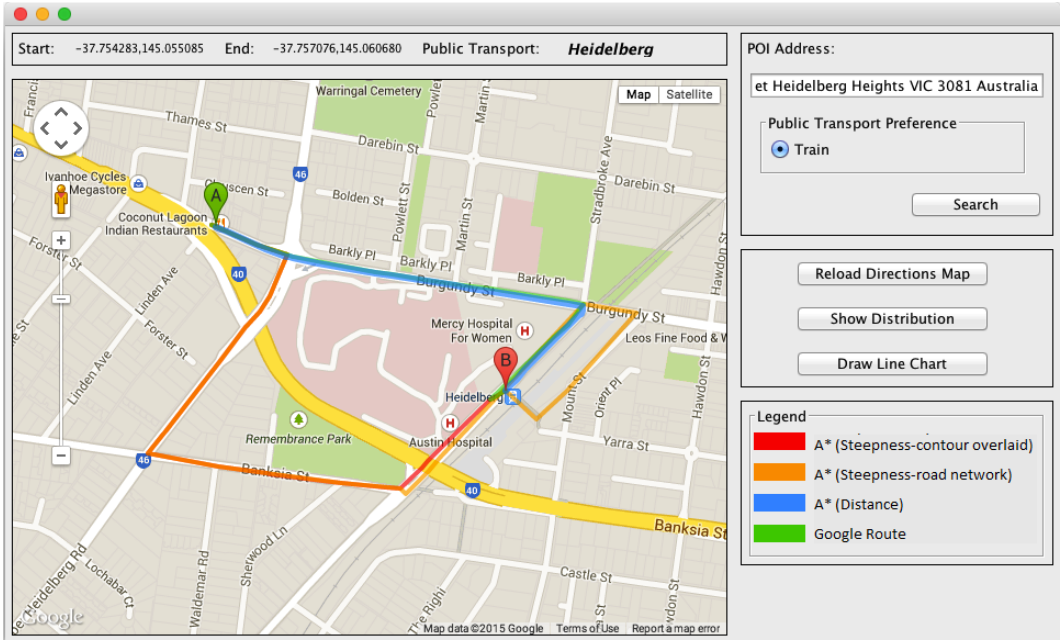


Figure 4.9: Steepness Context Aware Trip Planning in Heidelberg, Melbourne, Australia

4.5.4 Discussion of Results

In this section, we discuss different evaluation criteria of the routes produced by A* (Distance), A* (Steepness-road network), A* (Steepness-contour) and Google. We show a comparison of route length and the length of the total inaccessible road segments along a computed route. We also present a metric called the inaccessibility index which combines the length of the route and the length of inaccessible road segments along that route. Let, $D(s,t)$ be the length of shortest route between start location s and destination location t, $l(i)$ be the length of the i^{th} inaccessible segment, then the inaccessibility index of a route r, denoted as I_r is defined as follows:

$$I_r = \frac{\sum_{i=1}^{k} l(i)}{D(s,t)} \tag{4.1}$$

We also compute the average velocity along the routes. We use Tobler's hiking [171] function to do so. If V is the average velocity then, Toblers function is written as follows:

$$V = 6e^{-3.5|\frac{dh}{dx}+0.05|} \tag{4.2}$$

Here, $\frac{dh}{dx}$ is the slope of any route segment, dh is the change in elevation between start and end point and dx is the length of route segment. If $V = 5km/h$ for any route then the route is considered as an idle walking surface in practical.

In Rosanna, the routes suggested by Google and A* (Distance) go through Lower Plenty Road and Invermay Grove. It seems a promising suggestion considering shortest distance (Figure 4.7). But a closer look at Figure 4.6 shows that the a large portion of the included road segments are inaccessible which implies that these routes are no good considering the steepness context. On the other hand, routes computed by A* (Steepness-road network) and A* (Steepness-contour) avoid the inaccessible road segments while constructing the routes. Hence, the total lengths of inaccessible road segments produced by A* (Steepness-road network) and A* (Steepness-contour) are 0 in contrast with 281m and 260m produced by A*(Distance) and A*(Steepness-road network). Also, the inaccessibility index of the routes produced by A* (Distance) and Google is much higher compared to the routes produced by A* (Steepness-road

network) and A*(Steepness-contour). The routes computed from A* (Steepness-road network) and A* (Steepness-contour) have a length of 1519 m and 2248 m respectively compared to the route lengths of 1037 m and 1100 m produced by A* (Distance) and Google. However, this amount of distance is needed to compensate to find the most accessible trip route from Rosanna to Waiora Road Medical Service. Table 4.1 lists different evaluation metrics for the routes in Rosanna and Heidelberg. We can see from Table 4.1 that the A* (Steepness-contour) route requires less effort to travel compared to other alternatives since the highest average velocity (4.687 m/s) can be achieved by following this route.

Table 4.1: Summary of Evaluation Metrics for Four Different Routes

	Route Features	A*(Distance)	A*(road)	A*(contour)	Google
Case Study-1	Inaccessible segments (m)	281	0	0	260
	Route length (m)	1037	1519	2248	1100
	Inaccessibility index	0.271	0	0	0.251
	Avg. walking velocity (m/s)	4.227	4.512	4.687	4.327
Case Study-2	Inaccessible segments	465.77	128.35	128.35	465.77
	Route length (m)	888	1213	1800	888
	Inaccessibility index	0.525	0.145	0.145	0.525
	Avg. walking velocity (m/s)	4.549	4.684	4.731	4.549

Figures 4.8 and 4.9 show that all of the four roads suggested in Heidelberg go through some inaccessible roads segments. The reason is that there is no complete accessible route found within the maximum walk-able distance but we can see from Table 4.1 that the routes produced by the A* (Steepness-road network) and A* (Steepness-contour) have lower inaccessibility index and smaller total inaccessible road segments compared to A* (Distance) and Google. However, these the travellers need to travel longer distance if choose these roads compared to A* (Distance) and Google. Table 4.1 also shows that the computed average velocity of A* (Steepness-road network) and A* (Steepness-contour) are 4.684 m/s and 4.731 m/s respectively which are better than that of A* (Distance) and Google (both are 4.549 m/s).

4.6 Multiple Context Trip Planning and User Perspectives

In this section we present an algorithm to integrate multiple contexts (i.e. distance and accessibility) concurrently during trip plan computation. We model path accessibility in terms of path elevation and optimise two objectives of the paths: the distance and the accessibility. In summary our contributions are as follows:

- We optimise the accessibility along with the distance for the *first* time. No existing literature has addressed this issue before.
- For solving the problem, we develop a new multi-objective A* search algorithm known as Contour-based Accessible Path Routing Algorithm (CAPRA), more particularly the admissible heuristic functions for all the objectives, so that we can guarantee to obtain *all* the Pareto-optimal solutions in query time.
- We propose a new graph model that contains both the distance information and the elevation information for the A* search.

As discussed in the data preprocessing phase, the contour line is adopted to generate a new contour-based graph so that the elevation difference of each road segment can be evaluated more precisely. We have developed two new accessibility metrics: the vertical distance and maximal slope based on the contour graph to evaluate the accessibility of a path. Finally, we have designed a multi-objective A* search algorithm to find the best trade-off paths in terms of both distance and accessibility.

4.6.1 Accessibility Evaluation of Paths

The accessibility metrics of a path are derived from classical physics. In particular, assuming that the user keeps the same velocity while travelling along the path, the following two factors are closely relevant to the accessibility of a path: (1) the total energy consumed and (2) the maximal force needed to climb up the slopes along the path.

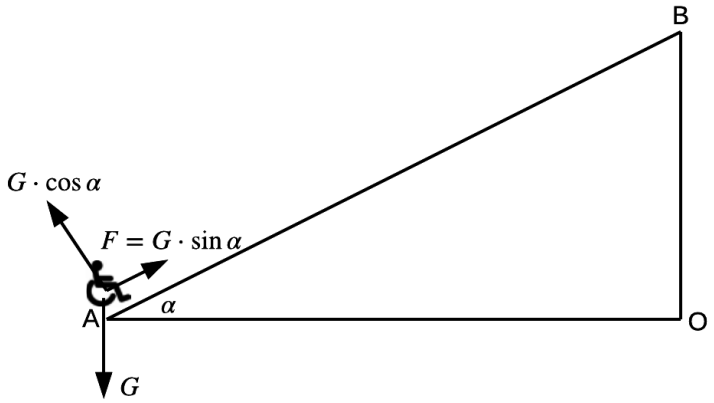


Figure 4.10: An Example of Moving Up a Slope of Incline α from A to B.

To facilitate this description, we take an example of a slope from point A to B in Figure 4.10, where a wheelchair user with gravity G is climbing up the slope whose steepness is α with a constant velocity v.

According to the relationship between work and mechanical energy, when moving up a slope from point A to B, we have

$$W_{AB} = TME_B - TME_A, \tag{4.3}$$

where W is the energy consumed (work done) by the user for climbing from A to B (the elevation of B is higher than that of A), and TME_A and TME_B are the total mechanical energy of the user at points A and B, respectively. It is known that the total mechanical energy is the sum of the kinetic energy KE and the potential energy PE. Then,

$$TME_A = KE_A + PE_A = \frac{1}{2}mv^2 + Gz_A, \tag{4.4}$$

$$TME_B = KE_B + PE_B = \frac{1}{2}mv^2 + Gz_B, \tag{4.5}$$

$$W_{AB} = TME_B - TME_A = G(z_B - z_A), \tag{4.6}$$

where m is the mass of the user, v is the velocity of the user, which stays the same during the climbing, G is the gravity of the user, and z_A and z_B are the respective elevation of points A and B so that $|z_B - z_A|$ is the vertical distance between point A and B.

On the other hand, if α is the steepness of the slope, the driving force needed for climbing up the slope from A to B while maintaining the velocity is as follows:

$$F_{AB} = G \cdot \sin\alpha = G \cdot \frac{|BO|}{|AB|}, \tag{4.7}$$

Similarly, when moving down from a higher point C to a lower point D, the two objectives are as follows:

$$W_{CD} = G(z_C - z_D), \tag{4.8}$$

$$F_{CD} = G \cdot \frac{|CO|}{|CD|}. \tag{4.9}$$

Note that $|AB|$ and $|CD|$ are not straightforward in practice. Therefore, they are replaced by $|AO|$ and $|OD|$, respectively, and $\sin\alpha$ is replaced by $\tan\alpha$ accordingly. Since α is always less than 90 degrees, minimizing $\sin\alpha$ is equivalent to minimizing $\tan\alpha$.

Then, given a path represented by a sequence of nodes $P = (v_0, v_1, \ldots, v_n)$, the total energy consumed $W(P)$ and the maximal force $F(P)$ needed to climb up and moving down all the slopes along the path are calculated as follows:

$$W(P) = \sum_{i=1}^{n} W_{v_{i-1}v_i}, \tag{4.10}$$

$$F(P) = \max_{i\in\{1,\ldots,n\}} \{F_{v_{i-1}v_i}\}, \tag{4.11}$$

where $W_{v_{i-1}v_i} = G \cdot |z_{v_i} - z_{v_{i-1}}|$, and $F_{v_{i-1}v_i} = G \cdot \frac{|z_{v_i} - z_{v_{i-1}}|}{d(v_{i-1}, v_i)}$ in which $d(v_{i-1}, v_i)$ is the horizontal distance of path segment between v_{i-1} and v_i and $|z_{v_i} - z_{v_{i-1}}|$ is the vertical distance between nodes v_i and v_{i-1}.

Given that the gravity G of the user is a constant, and $-\pi/2 \leq \alpha \leq \pi/2$, Eqs. (4.10) and (4.11) can be simplified to:

$$W(P) = \sum_{i=1}^{n} |z_{v_i} - z_{v_{i-1}}|, \tag{4.12}$$

$$F(P) = \max_{i\in\{1,\ldots,n\}} \left\{ \frac{|z_{v_i} - z_{v_{i-1}}|}{d(v_{i-1}, v_i)} \right\}. \tag{4.13}$$

Therefore, Eq. (4.12) illustrates the relationship between energy consumption and vertical distance travelled either up or down between successive points along the path. Note that

the total energy consumption for traveling along a path is related to the sum of the vertical distances along the path. So, minimizing the total vertical distance during accessible path planning will reduce the total energy consumption. Eq. (4.13) shows the greatest force required to move up or down the biggest elevation difference. Here, the required maximal driving force for moving up or down a slope is related to the vertical distance and the length of the slope. The elevation difference between start and end point of a slope is crucial. A longer path segment requires less travelling force compared to a shorter path segment with similar vertical distance. Also people may want to choose a path which is the shortest of all.

4.6.2 Path Routing Based on Distance and Accessibility

When an elderly user or person with special needs is planning to travel along a path from a source to a destination on the map, both the distance and accessibility are critical factors to consider. To be specific, we assume the user would prefer the path with shorter distance and higher accessibility. However, in practice, the distance and accessibility may be in conflict with each other. In this case, one should provide a set of trade-off paths, which are termed the *Pareto-optimal* paths, instead of one single global optimal path. The three objectives to be minimized in the accessible path routing can be described as follows:

$$\min_P f_1(P) = \sum_{i=1}^{n} d(v_{i-1}, v_i), \tag{4.14}$$

$$\min_P f_2(P) = W(P) = \sum_{i=1}^{n} |z_{v_i} - z_{v_{i-1}}|, \tag{4.15}$$

$$\min_P f_3(P) = F(P) = \max_{i \in \{1,\ldots,n\}} \left\{ \frac{|z_{v_i} - z_{v_{i-1}}|}{d(v_{i-1}, v_i)} \right\}, \tag{4.16}$$

where, $f_1(P)$ is the total horizontal distance of P, $f_2(P)$ is the total vertical distance of P, and $f_3(P)$ is the maximal slope of P, and $W(P)$ and $F(P)$ are defined in Eqs. (4.12) and (4.13) respectively. Note that $f_2(P)$ is consistent with the energy consumed for moving up and moving down all the slopes. $f_3(P)$ is standing for the maximal force needed.

Given two paths P_1 and P_2, P_1 is said to *dominate* P_2 if and only if all the objective values of P_1 are no worse than those of P_2, and there is at least one objective for which P_1 has a better value than P_2. We denote $P_1 \prec P_2$ for P_1 dominating P_2. A path P^* is said to

be *Pareto-optimal*, if and only if there is no other path that dominates P^*. The goal of this problem is to find all the possible Pareto-optimal paths.

In this chapter, the multi-objective A* search algorithm is employed to find the Pareto-optimal paths. Specifically, the framework of the multi-objective A* search algorithm proposed in [172] is adopted here. The framework is described in Algorithm 5. Two sets of labels $OPEN$ and $GOAL$ are defined where $OPEN$ is initialized with the source nodes and the algorithm steps through all nodes identifying non-dominated nodes which are stored in $GOAL$.

Once the target or destination node is reached, the elements in $GOAL$ and $OPEN$ are updated by removing the elements that are dominated by the new label. The search process stops when $OPEN$ becomes empty, and all the paths have been obtained by the backtracking procedure `Backtrack`($GOAL$). Further details of the multi-objective A* search algorithm can be found in [172]. The multi-objective A* algorithm requires the followings to be satisfied:

1. the costs $\vec{c}(u,v)$ of all the edges $(u,v) \in E$ must be non-negative;
2. the heuristic function is admissible, i.e., it never overestimates the actual minimal cost of reaching the goal.

Therefore, to design a multi-objective A* search algorithm for minimizing the objectives shown in Eqs. (4.14)–(4.16), we must design the cost functions $\vec{c}(u,v)$ and the heuristic functions `Heuristic`(v,t,G) that satisfy the above two requirements. From Eqs. (4.14), (4.15), and (4.16), we set $\vec{c}(u,v)$ and `Heuristic`(v,t,G) as follows:

$$c_1(u,v) = d(u,v), \quad h_1(v) = d(v,t),$$

$$c_2(u,v) = |z_v - z_u|, \quad h_2(v) = |z_t - z_v|,$$

$$c_3(u,v) = \max\left\{\frac{|z_v - z_u|}{d(u,v)} - g_3(u), 0\right\}, \quad h_3(v) = 0,$$

$$\vec{c}(u,v) \leftarrow (c_1(u,v), c_2(u,v), c_3(u,v)),$$

$$\texttt{Heuristic}(v,t,G) \leftarrow (h_1(v), h_2(v), h_3(v)).$$

Algorithm 5: The framework of multi-objective A* search algorithm.

Input: The graph G, source node s and target node t
Output: A set of trade-off paths $\mathbf{P} = \{P_1, \ldots, P_m\}$
// Initialization
foreach $v \in G$ **do** $\vec{g}_{cl}(v) \leftarrow \emptyset$, $\vec{g}_{op}(v) \leftarrow \emptyset$;
$GOAL \leftarrow \emptyset$, $OPEN \leftarrow \emptyset$;
$OPEN \leftarrow OPEN \cup (s, \emptyset, \vec{0}, \vec{h}(s))$, $\vec{g}_{op}(s) \leftarrow \vec{g}_{op}(s) \cup \vec{0}$;
// Search
while *$OPEN$ is not empty* **do**
  $L(u) := (u, \text{pred}(u), \vec{g}(u), \vec{h}(u)) \leftarrow$ `Extract`$(OPEN)$;
  $OPEN \leftarrow OPEN \setminus L(u)$;
  $\vec{g}_{op}(u) \leftarrow \vec{g}_{op}(u) \setminus \vec{g}(u)$, $\vec{g}_{cl}(u) \leftarrow \vec{g}_{cl}(u) \cup \vec{g}(u)$;
  if $u = t$ **then**
    Add $L(u)$ into $GOAL$, and remove from $GOAL$ the elements with dominated $\vec{g}(\cdot)$;
    Remove from $OPEN$ the elements whose $\vec{f}(\cdot) := \vec{g}(\cdot) + \vec{h}(\cdot)$ are dominated by $\vec{g}(u)$;
  else
    foreach $v \in \mathcal{N}(u)$ **do**
      if *Adding (u, v) forms a cycle* **then continue**;
      $\vec{g}(v) \leftarrow \vec{g}(u) + \vec{c}(u, v)$; // update $\vec{g}(v)$
      $\vec{h}(v) \leftarrow$ `Heuristic`(v, t, G); // calculate $\vec{h}(v)$
      $L(v) := (v, L(u), \vec{g}(v), \vec{h}(v))$;
      if *v is a new node* **then**
        $OPEN \leftarrow OPEN \cup L(v)$, $\vec{g}_{op}(v) \leftarrow \vec{g}_{op}(v) \cup \vec{g}(v)$;
      else
        if *$\vec{g}(v)$ is non-dominated by any $\vec{g} \in \vec{g}_{op}(v) \cup \vec{g}_{cl}(v)$* **then**
          Remove from $\vec{g}_{cl}(v)$ and $\vec{g}_{op}(v)$ the elements whose $\vec{g}(\cdot)$ are dominated by $\vec{g}(v)$;
          $OPEN \leftarrow OPEN \cup L(v)$, $\vec{g}_{op}(v) \leftarrow \vec{g}_{op}(v) \cup \vec{g}(v)$;
        end
      end
    end
  end
  return $\mathbf{P} \leftarrow$ `Backtrack`$(GOAL)$;
end

First, we note that $\forall (u, v) \in E$, $c_i(u, v) \geq 0$, $i = 1, 2, 3$. Then, for the total distance f_1, the heuristic $h_1(v)$ is admissible under the assumption of triangular inequality. For the total

vertical distance f_2, for any other point $v' \neq v$ and $v' \neq t$, we have

$$|z_t - z_v| \leq |z_{v'} - z_v| + |z_t - z_{v'}|.$$

That is, $h_2(v) \leq c_2(v, v') + h_2(v')$. Therefore, $h_2(v)$ is admissible.

Finally, since $c_3(u, v) \geq 0$, $h_3(v) = 0$ is clearly admissible. In fact, since it is difficult to predict the maximal slope from any point to the target, we set $h_3(v) = 0$ to reduce the A* search in terms of f_3 to the Dijkstra algorithm. The function $g_3(\cdot)$ is naturally defined by A*. That is, $g_3(s) = 0$, where s is the source node, and for any edge (u, v), $g_3(v) = g_3(u) + c_3(u, v)$

In addition, since the function `Extract`$(OPEN)$ can return any elements with non-dominated $\vec{f}(\cdot)$, we choose the one with the shortest estimated distance $f_1(\cdot)$ so as to reach the target node as soon as possible and reduce the search space.

4.6.3 Experimental Studies

For the experimental studies, case studies are conducted for various hilly cities in the world, including San Francisco (USA), Lisbon (Portugal) and Singapore. These cities are good examples of the experimental studies as they are built on slopes which means that moving up and down hills usually occurs in these cities. In addition, different city layouts are taken into account and we selected four random trips for our experiment. We selected San Francisco because the streets are normally laid out as a grid system. For historical reasons, such rectangular city blocks are not common in many European and Asian cities. Therefore, we selected Lisbon and Singapore as the representatives of hilly cities with more complex city layouts.

There is no existing algorithm which takes the elevation into account for path computation. Since we have designed CAPRA to employ the multi-objective A* search to find the paths, it is guaranteed to find the shortest path and there is no need to compare with other shortest path finding algorithms. Here, we only compare CAPRA with the path produced by Google Directions API [173] to show its reasonableness in reality.

In the preprocessing phase, the contour interval is set to 5m. The reason behind the selection of such a small contour interval is that it allows us to obtain even small changes in elevation. Once the contour interval is selected, the corresponding contour-based road network

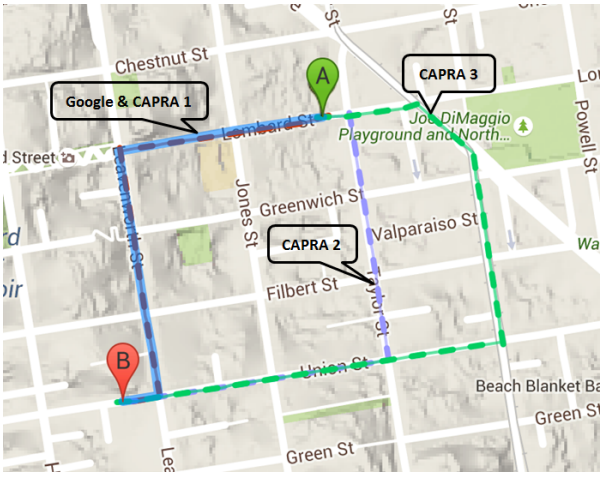


Figure 4.11: San Francisco, USA: The paths from 817 Lombard St to 1132 Union St. The solid path is obtained by Google Directions, and the dashed paths are obtained by CAPRA.

graph is generated and stored in the memory. For each test scenario, both CAPRA and Google Directions API are applied and the paths obtained by them are compared in terms of the three accessibility measures, i.e., horizontal distance, vertical distance and maximal slope defined in Eqs. (4.14), (4.15), and (4.16) respectively. To evaluate the efficacy of our contour based graph generation technique, the accessibility measure values for both CAPRA and Google Directions API paths are also calculated without considering the contours and compared with the obtained accessibility measure values of CAPRA from contour-based graph built with the contour interval of 5m. Specifically, to calculate the accessibility measure values of a path in the later case, the path is first divided into 10m-long small segments. Then, for each segment, the vertical distance and slope are calculated. Finally, the total vertical distance of the path is obtained by summing up all the vertical distances and the maximal slope of the path is obtained by selecting the maximal segmental slope. Although these are still not true values, they are good approximations by choosing a sufficiently small segment length.

4.6.3.1 Case Study-1 in San Francisco, USA

Figure 4.11 gives an example from 817 Lombard St (point A) to 1132 Union St (point B), San Francisco, USA. We selected this path because the path mainly consists of upward slopes and the elevation of point B is higher than point A.

Table 4.2: The accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario are shown in Fig. 4.11. "Distance", "Vertical" and "Slope" stand for the total horizontal distance, total vertical distance $W(P)$, and maximal slope $F(P)$, respectively. There is no accessibility measure value for Google from the 5m contour interval network graph, since the path is obtained by the Google API.

Path	Considering 5m Contour interval			Considering 10m-long road segment		
	Distance(m)	Vertical(m)	Slope	Distance(m)	Vertical(m)	Slope
Google	-	-	-	623	72.7	0.23
CAPRA1	623	73.0	0.23	623	72.7	0.23
CAPRA2	688	78.6	0.21	688	78.3	0.22
CAPRA3	943	82.2	0.14	943	82.2	0.15

There are four paths from A to B shown in the figure. The solid path is the shortest path found by the Google Directions. The three dashed paths are the trade-off paths obtained by our new algorithm CAPRA. One can see that the first path CAPRA1 (brown dashed) obtained by CAPRA is the same as the one obtained by Google Directions. In addition, CAPRA has provided two other paths CAPRA2 and CAPRA3 (purple and green dashed respectively).

A comparison summary of *cost-benefit* between distance and accessibility measure values of the paths obtained from 5m-interval contour based network graph and 10m-long segment based network graph are given in Table 4.2. We can see that the CAPRA1 does not provide the best accessibility score in terms of slope. On the other hand, the CAPRA2 path has better slope score, but longer distance and vertical distance compared to CAPRA1. The CAPRA2 also provides shorter distance compared to the CAPRA3 but pays more in terms of slope. Therefore, the paths are non-dominated to each other. Users can choose the best path based on their distance and accessibility requirements.

We also can see that the accessibility measure values of the paths obtained by CAPRA from the contour based network graph is very close to the corresponding values from 10m-long segment based network graph. This implies that a contour interval of 5m is sufficient to build an accurate contour-based network graph. In addition, while increasing the length of the path, the maximal slope decreases from 0.23 to 0.15.

Next, we examine the elevation changes along the paths given in Fig. 4.11. The CAPRA3 path has many more segments than the other paths due to the much larger horizontal distance

(a) Google (b) CAPRA1

(c) CAPRA2 (d) CAPRA3

Figure 4.12: The Elevation (in meters) Changes Along the Paths Given in Fig. 4.11.

as can be seen from Figure 4.12. However, it achieved a much smoother slope (evidenced by the maximal slope of 0.15) by choosing the longer distance to travel.

4.6.3.2 Case Study-2 in San Francisco, USA

Fig. 4.13 shows another scenario from 1260 Green St (point A) to 1398 Lombard St (point B), San Francisco, USA, but with mainly downward slopes and the elevation of point B is much lower than point A. In this scenario, CAPRA obtained four different paths. CAPRA3 (green dashed) is the same as that obtained by Google Directions. It should be noted that CAPRA managed to obtain two shorter paths CAPRA 1 and CAPRA 2 (red and purple dashed) than Google Directions, but with larger vertical distance and maximal slope.

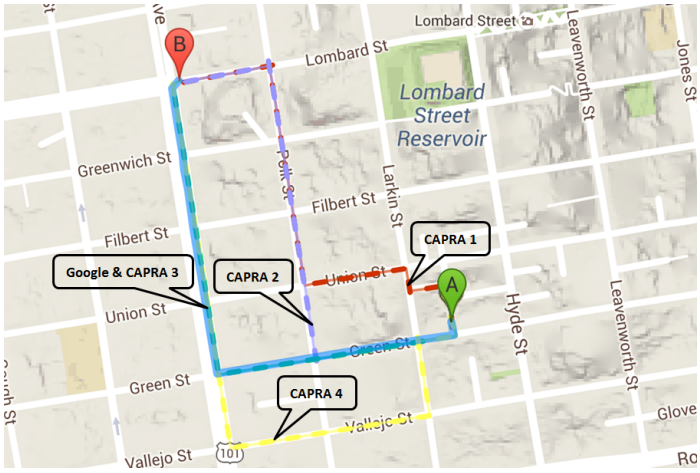


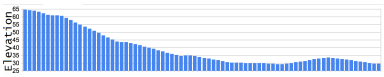
Figure 4.13: San Francisco, USA: The paths from 1260 Green St to 1398 Lombard St. The solid path is obtained by Google Directions, and the dashed paths are obtained by CAPRA.

Table 4.3: The accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 4.13. "Distance", "Vertical" and "Slope" stand for the total horizontal distance, total vertical distance $W(P)$, and maximal slope $F(P)$, respectively. There is no accessibility measure value for Google from 5m contour interval network graph, since the path is obtained by Google API.

Path	Considering 5m Contour interval			Considering 10m-long road segment		
	Distance(m)	Vertical(m)	Slope	Distance(m)	Vertical(m)	Slope
Google	-	-	-	787	43.9	0.15
CAPRA1	730	55.6	0.19	730	55.6	0.21
CAPRA2	772	47.4	0.14	772	47.6	0.15
CAPRA3	787	43.6	0.14	787	43.9	0.15
CAPRA4	997	43.6	0.09	997	43.9	0.09

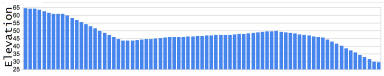
Table 4.3 shows a comparison summary of *cost-benefit* between distance and accessibility measure values of the paths obtained by Google Directions and CAPRA in the second scenario shown in Fig. 4.13. It can be seen that when the length of the path increases, the vertical distance and maximal slope tend to decrease. This way, the users can choose the most suitable path based on their own preferences in terms of distance and accessibility.

Fig. 4.14 gives the elevation changes for the paths shown in Fig. 4.13. It can be seen that for the path obtained by Google Directions and the first three paths obtained by CAPRA, the downward slopes are concentrated in the first half of the path (and the end of the path for CAPRA2). In contrast, the slopes are more uniformly distributed throughout the path for CAPRA4, which leads to a much smoother path overall.

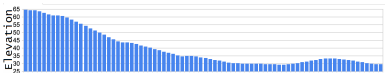


(a) Google

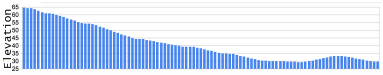
(b) CAPRA1



(c) CAPRA2



(d) CAPRA3



(e) CAPRA4

Figure 4.14: The Downhill Elevation (in meters) Changes Along the Paths Given in Fig. 4.13.

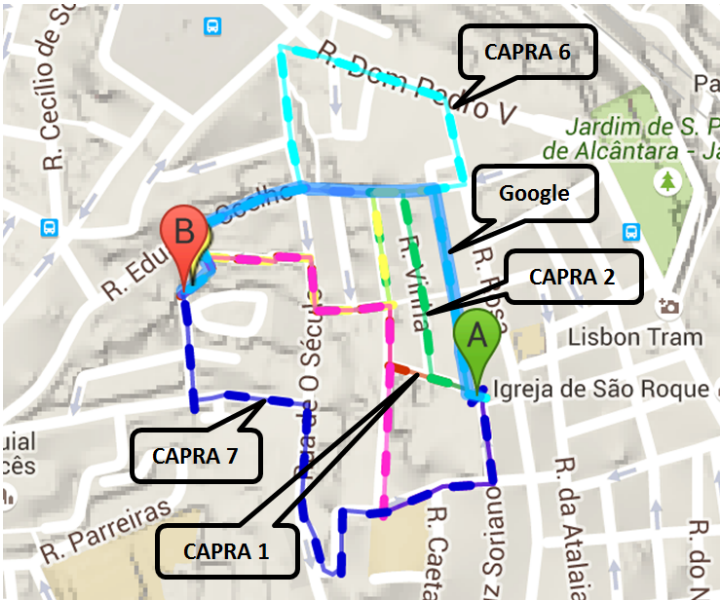


Figure 4.15: Lisbon, Portugal: The paths from Rua São Boaventura 182 to Travessa Horta 21. The solid path is obtained by Google Directions, and the dashed paths are obtained by CAPRA.

4.6.3.3 Case Study in Lisbon, Portugal

Fig. 4.15 shows a scenario from Rua São Boaventura 182 (point A) to Travessa Horta 21 (point B), Lisbon, Portugal. We selected this area of Lisbon because it is no longer a simple grid-like road network and is therefore more complex than that in San Francisco. The road network partly consists of some parallel streets (e.g., R. Vinha) which increases multiple routing possibilities. It can be seen that CAPRA obtained seven different paths in this scenario, none of which was the same as the Google path. A shortcut path CAPRA1 (brown dashed) was found, and the second path CAPRA2 (green dashed) was very similar to the Google path (turn right at a parallel street). In order to reduce the slope, two longer paths CAPRA6 and CAPRA7 (light and deep blue dashed) were also obtained, which have much reduced maximal slope. In this case, the irregular roads were employed as well.

Next, we summarize the *cost-benefit* between distance and accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 4.15.

Table 4.4: The accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 4.15. "Distance", "Vertical" and "Slope" stand for the total horizontal distance, total vertical distance $W(P)$, and maximal slope $F(P)$, respectively. There is no accessibility measure value for Google from 5m contour interval network graph, since the path is obtained by Google API.

Path	Considering 5m Contour interval			Considering 10m-long road segment		
	Distance(m)	Vertical(m)	Slope	Distance(m)	Vertical(m)	Slope
Google	-	-	-	464	31.6	0.19
CAPRA1	376	36.8	0.18	376	35.4	0.23
CAPRA2	463	32.1	0.18	463	31.5	0.19
CAPRA3	568	51.7	0.17	568	50.4	0.20
CAPRA4	575	45.7	0.17	575	45.3	0.20
CAPRA5	601	37.7	0.17	601	36.6	0.19
CAPRA6	613	44.7	0.13	613	43.1	0.14
CAPRA7	720	36.8	0.13	720	36.2	0.14

The CAPRA2 path has very similar distance and accessibility measure values to the Google path, due to the similar structure as shown in Table 4.4. For the paths obtained by CAPRA, although the value of the maximal slope for the paths from 5m contour interval is slightly higher than the 10m-long road segment one, the partial order is still consistent (i.e., a larger estimated value still leads to a larger real value). Therefore, one can still find the correct relative position of the paths on the Pareto front which is the set of Pareto optimal outcomes. It means that a CAPRA user is still able to choose a Pareto-optimal path which suits him/her best.

Fig. 4.16 gives the elevation changes over the paths given in Fig. 4.15. It can be seen that the vertical motions of the paths can be quite different from each other. For example, the first half of the Google path is relatively flat (slightly upward), while the CAPRA6 path keeps falling down until the last 15% of the path, and then goes up to reach the destination. They are trade-off paths and thus it is hard to tell which elevation change is better unless we look at elevation changes of each segment separately.

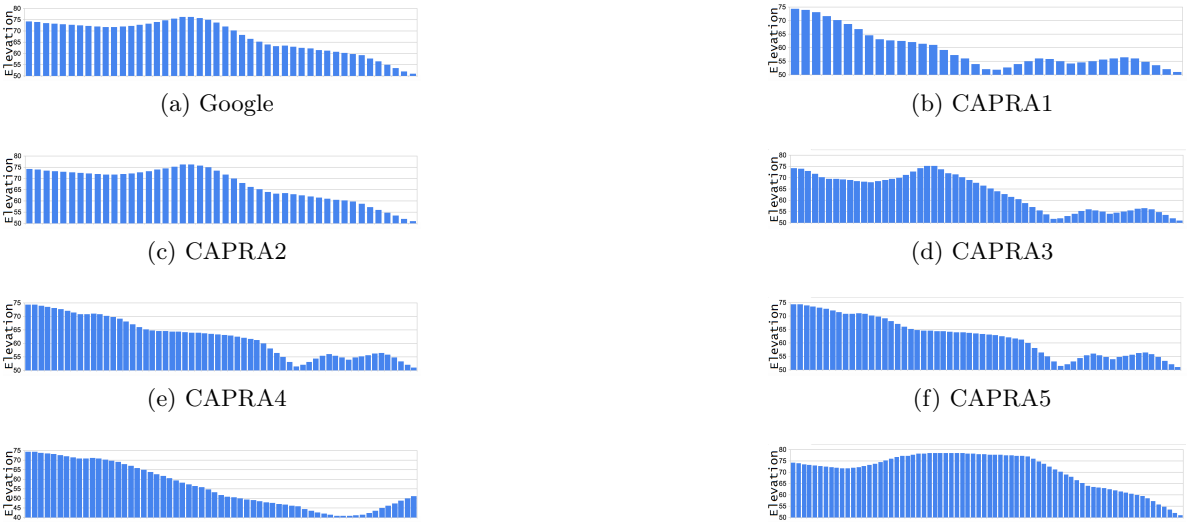


Figure 4.16: The Elevation (in meters) Change Along the Paths Given in Fig. 4.15.

4.6.3.4 Bukit Timah, Singapore

Fig. 4.17 shows a scenario from 23 Victoria Park Rd (point A) to 21 Duke's Rd (point B), Singapore. We selected this place because the roads in Singapore are very hilly and do not follow a grid. In this case, only two paths were obtained by CAPRA. The first path CAPRA 1 (brown dashed) is same as the Google path.

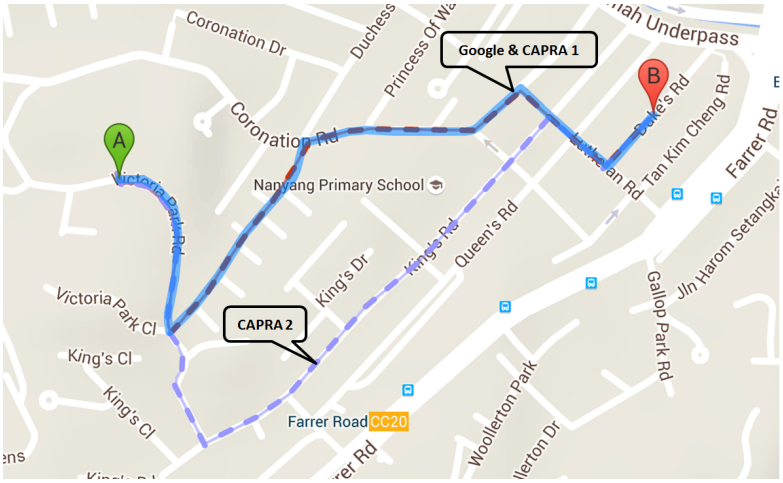


Figure 4.17: Singapore: The paths from 23 Victoria Park Rd to 21 Duke's Rd. The solid path is obtained by Google Directions, and the dashed paths are obtained by CAPRA.

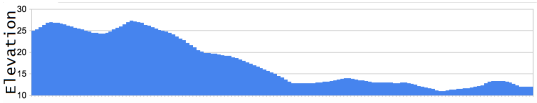
Table 4.5: The accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 4.17. "Distance", "Vertical" and "Slope" stand for the total horizontal distance, total vertical distance $W(P)$, and maximal slope $F(P)$, respectively. There is no accessibility measure value for Google from 5m contour interval network graph, since the path is obtained by Google API.

Path	Considering 5m Contour interval			Considering 10m-long road segment		
	Distance(m)	Vertical(m)	Slope	Distance(m)	Vertical(m)	Slope
Google	-	-	-	1444	30.7	0.06
CAPRA1	1444	28.8	0.06	1444	30.7	0.06
CAPRA2	1595	29.4	0.05	1595	31.7	0.05

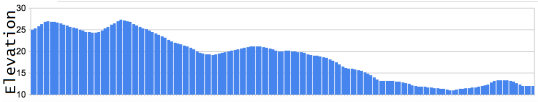
Table 4.5 shows the *cost-benefit* between distance and accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 4.17. As in the other scenarios, CAPRA managed to reach a smoother slope at the cost of a longer distance.

Fig. 4.18 gives the elevation change through the paths given in Fig. 4.17. In this case, the elevation change of the three paths are similar to each other. This is because their directions are roughly the same, and a major portion of the paths are parallel to each other.

(a) Google



(b) CAPRA1



(c) CAPRA2

Figure 4.18: The Elevation (in meters) Change Along the Paths Given in Fig. 4.17.

4.6.4 Discussion

Overall, the results for all the above case studies show that CAPRA is able to provide a wide range of reasonably good paths in terms of both vertical distance $W(P)$ and slope $F(P)$, including the optimal path in terms of distance. In most of the cases, CAPRA can obtain the Google path, or the paths with the same measure values as the Google path. In addition, the

Table 4.6: Summary of the Four Scenarios.

Case Study	Total Nodes	Total Edges	Trade-off Paths
San Francisco-1	33,122	2,963	3
San Francisco-2	33,122	2,963	4
Lisbon	10,411	2,515	7
Bukit Timah	16,177	1,870	2

trade-off paths with larger horizontal distances but smoother slopes are obtained as well. The estimated values of the CAPRA paths obtained from 5m contour intervals are close to their equivalent values obtained from 10m-long small road segments, which verifies the accuracy of the contour-based graph generation.

From the summary of *cost-benefit* analysis in Table 4.2-4.5, it can be seen that CAPRA can achieve a good trade-off between path length and accessibility (i.e., vertical distance and maximal slope). This way, the CAPRA users can choose the most suitable path based on their own preferences and accessibility requirements.

We note that there may be other physical accessibility barriers (i.e., stairs, ramps, traffic and road conditions) that can influence the accessibility of a walking path. For example, a path with an accessible elevation score may have a segment with stairs that cannot be traversed by people with wheelchairs. In this chapter, we assume that those physical accessibility barriers are handled with care.

Table 4.6 summarizes the four test scenarios used in this research to illustrate the corresponding number of nodes, edges and trade-off paths. We further note the *computation complexity* for calculating the Pareto-optimal trade-off paths. The worst-case time complexity of the adopted MOA* framework is $O(d^b)$, where d is the length of the longest non-dominated path, and b is the branching factor, i.e. the number of neighbours of each node in the graph. This computation complexity is no more than the traditional MOA* presented in [172]. The search space is an issue for the multi-objective framework since the function `Extract`$(OPEN)$ can return any elements with non-dominated $\vec{f}(\cdot)$. Therefore, we choose the element with the shortest estimated distance $f_1(\cdot)$ so as to reach the target node as soon as possible. In this way our adopted multi-objective framework is able to reduce the search space. The experiments

showed that our system can provide results in *query time* (< 1 seconds) on normal machine configuration (4GB RAM, Windows 7 OS, Intel Core-i7 CPU with 3.40 GHz clock speed) for all the test scenarios.

4.7 Conclusion

This chapter presented a framework called CoAcT for context aware trip planning using active transport. Our framework was able to provide unified solutions for providing trip plans based on a user query by integrating single or multiple trip contexts. The conceptual framework illustrated the procedures of collecting, integrating and managing contextual data from different data sources to plan context aware active transport trips. We also presented several real-world deployments of of trip planning to demonstrate the reasonableness of the CoAcT framework. The deployment showed that the framework can compute routes which aid the persons with limited mobility.

In order to serve the elderly and disabled people and those with special needs, we defined walking path accessibility considering the elevation of the path. A new contour-based path planning system called CAPRA was developed in this chapter. This new algorithm considered the accessibility of the path as well as the horizontal distance. The paths constructed may well serve healthy commuters while travelling, bike-riding, roller-skating as alternative routes with more gentle slopes.

We have demonstrated our CAPRA in four different hilly environments where the path elevation could be very steep and problematic for a person in a wheelchair. The experimental studies on several representative hilly cities in the world showed that CAPRA can provide not only the standard shortest path which is the same as that provided by Google Directions or an A* algorithm, but also other alternatives which may be longer but have smoother slopes. Our new algorithm can give the users a wider range of options to choose from. The users may not necessarily be elderly people or disabled but could instead, be bike riders or rollerbladers or people pushing prams. In fact, anyone who might prefer to know about alternate routes to their required destination for a variety of preferences. We have explained our three preferences, but other preferences based on user needs could also be implemented in the future.

The main contribution of this chapter is a generalized solution to the problem of multi-context integration in trip planning. We have shown how to consider distance and accessibility together to compute trip plans. We have discussed the process of route accessibility quantification using slope and total vertical distance along the route. Note that these quantification techniques are not fit for all solutions. Therefore, other types of quantification techniques are required if other contexts such as weather, safety and congestion are needed to be integrated into the trip plans. We believe that there might be several ways to speed up the route computation algorithm. For example, to reduce the search space and speed up the algorithm, we could return the shortest distance element until a complete path to the destination is found. From then on, we could return the element with the smallest vertical distance f_2 until a path to the destination with smaller f_2 is found. Then we could switch to returning the element with smallest f_3 until a path with smaller f_3 is found, then switch back to f_1 and so on. This way, we would make sure to successively decrease the limits for f_1, f_2, and f_3. Moreover, to know whether a proposed route is "good" or "acceptable" to the user is a complex and challenging task. Also, having more options may not necessarily be good for users. A personalized user-path mapping technique can be adapted to solve this problem. If we could know the specific requirement of a user, we could sort out the best path among the set of recommended paths. The future research also may include experiments with real users to study the user experience regarding our technique.

In summary, this chapter introduced an active transport trip planning framework called CoAcT. The contributions of the chapter include a new contour-based graph generation for path planning. We have developed new accessibility measures for routes and have designed a multi-objective A* path routing algorithm to satisfy different user perspectives of mobility contexts.

Chapter 5

Conclusion

In this section, we summarise the implications for practice, draw overall conclusions, and offer recommendations for further research. Mobility analytics and trip planning are two vital components of user mobility. We considered context-awareness in mobility analytics and trip planning, and investigated various mobility context scenarios to understand different situational factors that influence user mobility decisions. We leveraged the ubiquity of urban sensing technologies to collect and integrate data various situational factors. We performed context-aware mobility analytics and trip planning to facilitate a wide range of end users to make effective mobility decisions.

The core chapters of this thesis addressed key research challenges related to context-aware mobility analytics and trip planning which includes intelligent analysis of mobility contexts, mobility context prediction, representation and integration considering different user perspectives. Three research questions were constructed and we researched, developed and analysed specific solutions to these research issues. Specifically, we devised frameworks and efficient algorithms to provide intelligent contextual analysis and predict mobility contexts. In this thesis, we also presented techniques to incorporate expert-like knowledge for mobility context prediction and provide methods to consider different user perspectives of a context during context-aware trip planning. We highlighted different case-specific solutions and real-world deployment scenarios to illustrate the reasonableness of our solutions for context-aware mobility analytics and trip planning.

The *first research question* (RQ-1) was answered in Chapter 2 where we presented a new framework to provide step by step procedures for developing mobility context prediction given a future time stamp. We presented a complex taxi-passenger queue context prediction scenario at the airport as an example of mobility context prediction tasks. A real-world queue context dataset was generated for our experiments by fusing heterogeneous datasets including taxi trip logs, passenger arrivals and processing times, and weather conditions at a major international airport (JFK international airport, New York City). We investigated different mobility-associated factors including time, trip frequency of taxis, frequency of passenger arrivals and weather conditions since these factors have influence on the occurrence of different queue contexts. Using our framework, we predicted different queue contexts related to taxis and passengers which are of imbalanced distribution. We also provided analysis on the queue context prediction from different user perspectives. To predict the imbalanced taxi and passenger queue contexts, our framework used a suite of existing sampling and machine learning techniques. We observed that the Support Vector Machine and Random Forest delivered the top prediction performances when oversampling was employed for our queue context dataset. We also noted that the Support Vector Machine outperformed Random Forest when the taxi drivers point of view was considered. Random Forest exhibited better results compared to the Support Vector Machine from the airport passengers' points of view during our queue context prediction experiments.

To address the *second research question* (RQ-2), we developed a new technique to incorporate expert-like knowledge in mobility context prediction by modelling mobility associated factors from historical data. Specifically, a feature weighting scheme based on conditional mutual information was introduced in Chapter 3 to combine the expert-like knowledge and the probability theory for feature weight estimation (i.e. importance score computation). We considered two mobility context prediction scenarios. The both of the mobility context scenarios was influenced by many diverse factors associated with user mobility. The mobility context was considered as a categorical label of taxi-passenger queue contexts at the airport. In the second scenario, the mobility context was a numeric target score of queue wait times for the taxi drivers waiting at the airport taxi rank. We employed neighborhood based algorithms for

mobility context prediction in our experiments and to test our methodology. The experimental results showed that our developed feature weighting scheme could identify a good quality neighborhood and thus can improve the prediction outcomes both for categorical as well as numerical mobility contexts. We showed the statistical significance of this improvement in terms of confidence interval of paired t-test.

In Chapter 4, we addressed the third *research question* (RQ-3) and studied the problem of mobility context inference and integration based of user perspectives in an active transport trip planning scenario. We introduced a conceptual framework called CoAcT that summarized the existing approaches related to context-aware trip planning. This framework provided a list of key procedures to be followed for context-aware trip planning including contextual data collection, fusion, context inference, representation and integration considering diverse user perspectives of a context. We designed an algorithm to infer a sparse mobility context provided by accessibility conditions. We overlaid contour information into road network data. We also presented a graph representation of context that allowed us to access the context information for trip plan computation in near real time. As part of the trip planning module of our framework, we utilized the A* algorithm to provide trip planning considering one single mobility context at a time. Aiming to incorporate multiple mobility contexts and to consider diverse user perspectives of contexts in trip planning, we developed a multi-objective A* algorithm. Our framework and algorithms were tested with real-world scenarios in several cities around the world. The experimental results showed the practicality and effectiveness of our developed approaches.

The contribution of this thesis on context-aware mobility analytics and trip planning can be summarized as follows:

- Fusion and representation of mobility-associated factors collected from large heterogeneous data sources.
- Intelligent analysis and prediction of mobility contexts and associated factors.
- Incorporation of expert knowledge into neighborhood based methods for enhancing the mobility context prediction.

- Inference of sparse mobility contexts from heterogeneous data sources for facilitating the task of trip planning.
- A new graph representation technique for inferred mobility contexts to accelerate the computation of efficient trip plans.
- Effective approach to answer user's trip planning queries by considering multiple mobility contexts simultaneously, given the variety of different user perspectives.

5.1 Limitations and Future Directions of Research

The frameworks and algorithms presented in this thesis can be used for problem-specific context-aware mobility analytics and trip planning. This research is built on top of the existing machine learning tools and techniques to solve problems regarding mobility context inference, prediction and integration in trip planning considering different user perspectives. The proposed techniques outperform related baselines in terms of performance or reasonableness, however, there remains scope for improvement in these approaches. Here we briefly discuss the limitations of our study and recommend some directions for future research.

In chapter 2, we developed and emphasized context-aware mobility analytics using spatio-tempotal mobility associated factors. In this the era of big data, the spatio-temporal contextual information is mainly collected through autonomous loggers from heterogeneous data sources. These autonomous loggers may be prone to machine error which was ignored in this research. We conducted mobility analytics with an airport taxi-passenger queue context dataset which was generated by fusing several real-world heterogeneous datasets of mobility associated factors. We did not perform experiments with other datasets as these kinds of mobility context datasets were not publicly available. Our generated dataset served our purpose since the occurrences of the mobility associated factors fused in our dataset were dynamic in nature (i.e. changed over time and various external factors) which made it complex for experiments. Also, the data collected had different sampling rates and representations which posed challenges during data fusion and representation. We fused and represented various mobility factors associated with hourly time windows to achieve our goal for mobility analytics. We believe

this can be adapted for future datasets as they become available. However, investigation for advanced techniques for heterogeneous data fusion and representation remains a direction of future research in effective user mobility analytics using heterogeneous data. The queue context prediction framework presented in this thesis does not provide the relative queue lengths. Future research would enable the measurement of queues and their lengths in real time. The values used for two thresholds to infer the queue contexts were arbitrary. Future research may address the optimal thresholding. For the deployment, the taxi regulations at the JFK were taken into consideration. The taxi regulations of other places need to be considered carefully for future deployments.

In chapter 3, we inferred expert-like knowledge from historical mobility context data. We also presented a technique to combine the inferred expert-like knowledge and the probability theory for the calculation of feature weight scores to be used by the prediction algorithms. The results obtained from this research are restricted to prediction only and do not provide optimum decision-making solutions. For providing optimum decision-making, other types of modelling may be required. Future research could address this problem of providing optimal decision-making for taxi drivers by considering their personalized objectives. The performance of predictive analytics techniques is usually domain specific. Therefore, appropriate domain adaptation is required for the analytics techniques adopted from other domains into the mobility analytics with big spatio-temporal data. Given the fact that the neighborhood-based prediction algorithms are suitable for many real-world problems, we only employed k-NN based methods in our experiments to test our developed techniques. The value of k varied between 1 and 15. We did not conduct experiments with any other prediction algorithms. The related literature has shown that incorporating expert like knowledge can enhance prediction performance. Hence, we believe that our technique will also be adaptable to other prediction methods. However, further research is recommended to study how our technique can be integrated into other prediction algorithms by identifying the required adaptation parameters and adjustments. Also, further validation of our technique would be found by performing experiments with other datasets when the similar kinds of mobility context datasets become available.

In Chapter 4, we presented context-aware trip planning which considers topographical information to infer route accessibility. However, there may be other physical accessibility barriers present that may need to be taken into account. For example, the walk accessibility of a path may be affected by stairs, high curbs and busy intersections. We assumed that the physical barriers along the routes were handled accordingly. In future, a matrix such as the SAW criteria [150] and walkability score [147] could be incorporated with our approach to help disabled and elderly people check whether their route is affected by any physical accessibility barriers. The integration of such data can also be achieved through crowdsourcing, as some of these hazards are not permanent, but temporarily constructed for road maintenance or building construction. For the purpose of real-time data collection, the crowdsourcing platform described in [143] could be used. The R-Q based method proposed in [141] is able to provide an answer to the routing queries related to traffic conditions. Also, it can be adapted with our model to provide live updates about the busyness of a road. In this regard, the urban data from pedestrian sensors could be utilized with the crowdsourcing platform. Also, user profiles could be incorporated to satisfy individual requirements. The data collected through crowdsourcing can be used for providing context-aware mobility decisions since the information contains actual user perceptions. However, it is challenging to ensure the data quality of such crowd-generated context information. Our proposed technique can handle multiple mobility contexts concurrently during trip planning; however, we tested with only two mobility contexts (i.e. distance and accessibility). We have not investigated any further implications for concurrent consideration of more mobility contexts. Future research could address these issues. Note that the quantification technique used for calculating route accessibility is not fit for all solutions. Therefore, different types of quantification techniques may be required to integrate other contexts such as weather and congestion. To know whether a proposed route may be "good" or "acceptable" to the users is still a challenging issue. The user-path mapping and personalized path sorting could be addressed in future by considering user-specific preference levels. The future research may include real user study to examine the user experience regarding our technique. Future research could be conducted to speed up the route computation algorithm.

In conclusion, the contributions of this thesis include the development of several frameworks and algorithms for context-aware mobility analytics and trip planning. The significance of this research is to provide comprehensive support in mobility decision making for the users of urban spaces. The techniques developed can be deployed in real-world scenarios and can aid a wide range of users. The potential impact of our research include reduction of disruption that occurs due to the inefficiency in manual mobility context estimation at the airports. This thesis also presents new techniques for integrating multiple perspectives of mobility contexts in the trip planning process. A new context-aware trip planner is introduced which will also benefit elderly commuters and those with limited mobility to travel more conveniently around the city and beyond.

Bibliography

[1] Muhammed Fatih Bulut, Murat Demirbas, and Hakan Ferhatosmanoglu. LineKing: Coffee Shop Wait-Time Monitoring Using Smartphones. *IEEE Transaction on Mobile Computing*, 14(10):2045–2058, 2015. **[Cited on pages xii, 17, 41, 43, 44, 54, 56, 57, 58, 59, 61, and 62]**

[2] M.S. Rahaman, M. Hamilton, and F.D. Salim. *Using Big Spatial Data for Planning User Mobility*, pages 1–6. Springer International Publishing, 2018. **[Cited on pages 3 and 5]**

[3] Yunhe Pan, Yun Tian, Xiaolong Liu, Dedao Gu, and Gang Hua. Urban big data and the development of city intelligence. *Engineering*, 2(2):171 – 178, 2016. **[Cited on page 3]**

[4] Ainhoa Serna, Jon Kepa Gerrikagoitia, Unai Bernab, and Toms Ruiz. Sustainability analysis on urban mobility based on social media content. *Transportation Research Procedia*, 24:1 – 8, 2017. 3rd Conference on Sustainable Urban Mobility, 3rd CSUM 2016, 26 27 May 2016, Volos, Greece. **[Cited on page 3]**

[5] Laura Gebhardt, Daniel Krajzewicz, Rebekka Oostendorp, Mirko Goletz, Konstantin Greger, Matthias Kltzke, Peter Wagner, and Dirk Heinrichs. Intermodal urban mobility: Users, uses, and use cases. *Transportation Research Procedia*, 14:1183 – 1192, 2016. Transport Research Arena TRA2016. **[Cited on page 3]**

[6] Sergio Alvarez-Napagao, Arturo Tejeda-Gómez, Luis Oliva-Felipe, Dario Garcia-Gasulla, Victor Codina, Ignasi Gómez-Sebàstia, and Javier Vázquez-Salceda. Urban context detection and context-aware recommendation via networks of humans as sensors. In Fer-

nando Koch, Felipe Meneguzzi, and Kiran Lakkaraju, editors, *Agent Technology for Intelligent Mobile Services and Smart Societies*, pages 68–85, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. **[Cited on page 3]**

[7] Minjie Wang, Su Yang, Yi Sun, and Jun Gao. Human mobility prediction from region functions with taxi trajectories. *PLOS ONE*, 12(11):1–23, 11 2017. **[Cited on page 3]**

[8] Shenggong Ji, Yu Zheng, and Tianrui Li. Urban sensing based on human mobility. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 1040–1051, New York, NY, USA, 2016. ACM. **[Cited on page 3]**

[9] Yannis Tyrinopoulos and Constantinos Antoniou. Factors affecting modal choice in urban mobility. *European Transport Research Review*, 5(1):27–39, Mar 2013. **[Cited on page 3]**

[10] Dandan Chen, Yong Zhang, Liangpeng Gao, Nana Geng, and Xuefeng Li. The impact of rainfall on the temporal and spatial distribution of taxi passengers. *PLOS ONE*, 12(9):1–16, 09 2017. **[Cited on page 3]**

[11] Evangelia Anagnostopoulou, Efthimios Bothos, Babis Magoutas, Johann Schrammel, and Gregoris Mentzas. Persuasive technologies for sustainable urban mobility. *CoRR*, abs/1604.05957, 2016. **[Cited on page 3]**

[12] E. Galbrun, K. Pelechrinis, and E. Terzi. Safe Navigation in Urban Environments. In *The 3rd International Workshop on Urban Computing (UrbComp 2014)*, 2014. **[Cited on pages 3, 77, and 78]**

[13] H. Liu, Y. Gao, L. Lu, S. Liu, H. Qu, and L. M. Ni. Visual analysis of route diversity. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 171–180, Oct 2011. **[Cited on pages 3, 6, and 7]**

[14] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.*, 5(3):38:1–38:55, September 2014. **[Cited on page 3]**

[15] Simon Elias Bibri and John Krogstie. The core enabling technologies of big data analytics and context-aware computing for smart sustainable cities: a review and synthesis. *Journal of Big Data*, 4(1):38, Nov 2017. **[Cited on page 3]**

[16] Mohammad Saiedur Rahaman, Yi Mei, Margaret Hamilton, and Flora D. Salim. Capra: A contour-based accessible path routing algorithm. *Information Sciences*, 385386:157 – 173, 2017. **[Cited on pages 3, 5, 6, 7, and 8]**

[17] Ivana Semanjski, Sidharta Gautama, Rein Ahas, and Frank Witlox. Spatial context mining approach for transport mode recognition from mobile sensed big data. *Computers, Environment and Urban Systems*, 66:38 – 52, 2017. **[Cited on page 3]**

[18] Nivan Ferreira, Jorge Poco, Huy T. Vo, Juliana Freire, and Cláudio T. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149–2158, December 2013. **[Cited on page 3]**

[19] Apostolos Pyrgelis, Emiliano De Cristofaro, and Gordon J. Ross. Privacy-friendly mobility analytics using aggregate location data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '16, pages 34:1–34:10, New York, NY, USA, 2016. ACM. **[Cited on page 3]**

[20] A. Garz, A. A. Benczr, C. I. Sidl, D. Tahara, and E. F. Wyatt. Real-time streaming mobility analytics. In *2013 IEEE International Conference on Big Data*, pages 697–702, Oct 2013. **[Cited on pages 3 and 4]**

[21] Song Gao. Spatio-temporal analytics for exploring human mobility patterns and urban dynamics in the mobile age. *Spatial Cognition & Computation*, 15(2):86–114, 2015. **[Cited on page 3]**

[22] Nianbo Liu, Yong Feng, Feng Wang, Bang Liu, and Jinchuan Tang. Mobility crowdsourcing: Toward zero-effort carpooling on individual smartphone. *International Journal of Distributed Sensor Networks*, 9(2):615282, 2013. **[Cited on pages 3 and 5]**

[23] N. Samaan and A. Karmouch. A mobility prediction architecture based on contextual knowledge and spatial conceptual maps. *IEEE Transactions on Mobile Computing*, 4(6):537–551, Nov 2005. **[Cited on page 3]**

[24] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In *2012 IEEE 12th International Conference on Data Mining*, pages 1038–1043, Dec 2012. **[Cited on page 3]**

[25] Pablo Samuel Castro, Daqing Zhang, and Shijian Li. Urban traffic modelling and prediction using large scale taxi gps traces. In Judy Kay, Paul Lukowicz, Hideyuki Tokuda, Patrick Olivier, and Antonio Krüger, editors, *Pervasive Computing*, pages 57–72, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. **[Cited on page 4]**

[26] Mohammad Saiedur Rahaman, Margaret Hamilton, and Flora D. Salim. Predicting Imbalanced Taxi and Passenger Queue Contexts in Airport. In *Proc. of the Pacific Asia Conf. on Info. Systems (PACIS)*, 2017. **[Cited on pages 4, 5, 6, 8, 41, 45, and 65]**

[27] Felix Caicedo, Carola Blazquez, and Pablo Miranda. Prediction of parking space availability in real time. *Expert Systems with Applications*, 39(8):7281 – 7290, 2012. **[Cited on page 4]**

[28] Yu Lu, Shili Xiang, and Wei Wu. Taxi Queue, Passenger Queue or No Queue? A Queue Detection and Analysis System using Taxi State Transition. In *Proc. of the 18th International Conference on Extending Database Technology (EDBT)*, pages 593–604, Brussels, Belgium, 2015. **[Cited on pages 4, 17, 19, 20, and 27]**

[29] Anil Yazici, Camille Kamga, and Abhishek Singhal. A big data driven model for taxi drivers' airport pick-up decisions in new york city. In *Proc. of the IEEE Int. Conf. on Big Data*, pages 37–44, 2013. **[Cited on pages 5, 19, 20, and 43]**

[30] Jing-Quan Li, Kun Zhou, Liping Zhang, and Wei-Bin Zhang. A multimodal trip planning system with real-time traffic and transit information. *Journal of Intelligent Transportation Systems*, 16(2):60–69, 2012. **[Cited on page 5]**

[31] K. Rehrl, S. Bruntsch, and H. J. Mentz. Assisting multimodal travelers: Design and prototypical implementation of a personal travel companion. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):31–42, March 2007. **[Cited on page 5]**

[32] Jau ming Su and Chih hung Chang. The multimodal trip planning system of intercity transportation in taiwan. *Expert Systems with Applications*, 37(10):6850 – 6861, 2010. **[Cited on page 5]**

[33] Nilesh Borole, Dillip Rout, Nidhi Goel, P. Vedagiri, and Tom V. Mathew. Multimodal public transit trip planner with real-time transit data. *Procedia - Social and Behavioral Sciences*, 104:775 – 784, 2013. 2nd Conference of Transportation Research Group of India (2nd CTRG). **[Cited on page 5]**

[34] K. G. Zografos, K. N. Androutsopoulos, and V. Spitadakis. Design and assessment of an online passenger information system for integrated multimodal trip planning. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):311–323, June 2009. **[Cited on pages 5 and 6]**

[35] Bin Yang, Chenjuan Guo, Yu Ma, and Christian S. Jensen. Toward personalized, context-aware routing. *The VLDB Journal*, 24(2):297–318, Apr 2015. **[Cited on page 5]**

[36] Zhong-Ren Peng and Ruihong Huang. Design and development of interactive trip planning for web-based transit information systems. *Transportation Research Part C: Emerging Technologies*, 8(1):409 – 425, 2000. **[Cited on page 6]**

[37] Rahim A. Abbaspour and Farhad Samadzadegan. Time-dependent personal tour planning and scheduling in metropolises. *Expert Systems with Applications*, 38(10):12439 – 12452, 2011. **[Cited on page 6]**

[38] S. Azenkot, S. Prasain, A. Borning, E. Fortuna, R.E. Ladner, and J.O. Wobbrock. Enhancing Independence and Safety for Blind and Deaf-blind Public Transit Riders. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3247–3256. ACM, 2011. **[Cited on pages 6, 7, 75, 76, and 78]**

[39] Camille Kamga and M. Anl Yazc. Temporal and weather related variation patterns of urban travel time: Considerations and caveats for value of travel time, value of variability, and mode choice studies. *Transportation Research Part C: Emerging Technologies*, 45:4 – 16, 2014. Advances in Computing and Communications and their Impact on Transportation Science and Technologies. **[Cited on page 7]**

[40] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science*, 6(1):111–121, Feb 2012. **[Cited on page 7]**

[41] Jing Yuan, Yu Zheng, Liuhang Zhang, XIng Xie, and Guangzhong Sun. Where to find my next passenger. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, pages 109–118, New York, NY, USA, 2011. ACM. **[Cited on page 7]**

[42] Mohammad Sharif and Ali Asghar Alesheikh. Context-aware movement analytics: implications, taxonomy, and design framework. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(1):n/a–n/a, 2018. **[Cited on page 7]**

[43] Cynthia Chen, Jingtao Ma, Yusak Susilo, Yu Liu, and Menglin Wang. The promises of big data and small data for travel behavior (aka human mobility) analysis. *Transportation Research Part C: Emerging Technologies*, 68:285 – 299, 2016. **[Cited on page 7]**

[44] R. Meier, A. Harrington, and V. Cahill. Towards delivering context-aware transportation user services. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 369–376, Sept 2006. **[Cited on page 7]**

[45] Abdul Majid, Ling Chen, Gencai Chen, Hamid Turab Mirza, Ibrar Hussain, and John Woodward. A context-aware personalized travel recommendation system based on geotagged social media data mining. *International Journal of Geographical Information Science*, 27(4):662–684, 2013. **[Cited on page 7]**

[46] Victor Codina, Jose Mena, and Luis Oliva. Context-aware user modeling strategies for journey plan recommendation. In Francesco Ricci, Kalina Bontcheva, Owen Conlan, and Séamus Lawless, editors, *User Modeling, Adaptation and Personalization*, pages 68–79, Cham, 2015. Springer International Publishing. **[Cited on page 7]**

[47] Lijun Sun and Kay W. Axhausen. Understanding urban mobility patterns with a probabilistic tensor factorization framework. *Transportation Research Part B: Methodological*, 91:511 – 524, 2016. **[Cited on page 8]**

[48] A.P. Sinha and H. Zhao. Incorporating Domain Knowledge into Data Mining Classifiers: An App. in Indirect Lending. *Decision Support Systems*, 46(1):287–299, 2008. **[Cited on pages 8, 9, and 44]**

[49] M.R.A. Iqbal, M.S. Rahaman, S.I. Nabil, and I. A. Chowdhury. Knowledge Based Decision Tree Construction with Feature Importance Domain Knowledge. In *Proc. of the 7th International Conference on Electrical and Computer Engineering (ICECE)*, 2012. **[Cited on pages 8, 9, and 44]**

[50] Mohammad Saiedur Rahaman, Margaret Hamilton, and Flora D. Salim. Queue context prediction using taxi driver knowledge. In *Proceedings of the Knowledge Capture Conference*, K-CAP 2017, pages 35:1–35:4, New York, NY, USA, 2017. ACM. **[Cited on page 8]**

[51] Alison Conway, Camille Kamga, Anil Yazici, and Abhishek Singhal. Challenges in managing centralized taxi dispatching at high-volume airports: Case study of john f. kennedy international airport. *Transportation Research Records*, 2300:83–90, 2012. **[Cited on pages 8 and 18]**

[52] M. S. Rahaman, M. Hamilton, and F. D. Salim. Coact: A framework for context-aware trip planning using active transport. In *Proc. of the Percom Workshops*, 2018. [**Cited on page 8**]

[53] Y. Zheng, W. Wu, Y. Chen, H. Qu, and L. M. Ni. Visual analytics in urban computing: An overview. *IEEE Transactions on Big Data*, 2(3):276–296, Sept 2016. [**Cited on page 8**]

[54] Y. Zheng. Methodologies for cross-domain data fusion: An overview. *IEEE Transactions on Big Data*, 1(1):16–34, March 2015. [**Cited on page 8**]

[55] Haibo He and Edwardo A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 21(9):1263–1284, 2009. [**Cited on pages 9 and 21**]

[56] Ye Zhang, Le T. Nguyen, and Joy Zhang. Wait Time Prediction: How to Avoid Waiting in Lines? In *Ubicomp Adjunct*, pages 481–490, Zurich, Switzerland, 2013. [**Cited on pages 17, 41, and 43**]

[57] Ron Davis, Tamara Rogers, and Yingping Huang. A Survey of Recent Developments in Queue Wait Time Forecasting Methods. In *Proc. of the International Conference on Foundations of Computer Science (FCS)*, pages 84–90, Athens, Greece, 2016. [**Cited on page 17**]

[58] Afian Anwar, Mikhail Volkov, and Daniela Rus. ChangiNOW : A Mobile Application for Efficient Taxi Allocation at Airports. In *Proc. of the IEEE Conf. on Intelligent Transportation Systems*, pages 694–701, Hague, Netherlands, 2013. [**Cited on pages 19, 21, and 43**]

[59] Camille Kamga, Alison Conway, Abhishek Singhal, and Anil Yazici. Using Advanced Technologies to Manage Airport Taxicab Operations. *Journal of Urban Tech.*, 19:23–43, 2012. [**Cited on page 19**]

[60] J. Hilkevitch. O'hare taxi passengers, drivers often in holding pattern. Chicago tribune, 12-Jul-2015, http://www.chicagotribune.com/business/ chi-taxicabs-ohare-getting-around-met-20150316-column.html(Online). **[Cited on page 19]**

[61] Jane Lin, Sandeep Sasidharan, Shuo Ma, and Ouri Wolfson. A Model of Multimodal Ridesharing and its Analysis. In *Proc. of the 17th IEEE International Conference on Mobile Data Management (MDM)*, pages 164–173, Porto, Portugal, 2016. **[Cited on page 19]**

[62] Hao Dong, Xuedan Zhang, Yuhan Dong, Chuang Chen, and Fan Rao. Recommend a Profitable Cruising Route for Taxi Drivers. In *Proc. of the* 17^{th} *International Conference on Intelligent Transportation Systems (ITSC)*, pages 2003–2008, Qingdao, China, 2014. **[Cited on page 19]**

[63] Yong Ge, Hui Xiong, Alexander Tuzhilin, Keli Xiao, Marco Gruteser, and Michael Pazzani. An Energy-Efficient Mobile Recommender System. In *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 899–907, Washington DC, USA, 2010. **[Cited on page 19]**

[64] R. Wang, C. Y. Chow, Y. Lyu, V. C. S. Lee, S. Kwong, Y. Li, and J. Zeng. Taxirec: Recommending road clusters to taxi drivers using ranking-based extreme learning machines. *IEEE Transactions on Knowledge and Data Engineering*, 30(3):585–598, March 2018. **[Cited on page 19]**

[65] Jing Yuan, Yu Zheng, Liuhang Zhang, Xing Xie, and Guangzhong Sun. Where to Find My Next Passenger? In *Proc. of the 13th international conference on Ubiquitous computing (UbiComp'11)*, pages 109–118, Beijing, China, 2011. **[Cited on pages 20, 25, and 42]**

[66] Javed Aslam, Sejoon Lim, Xinghao Pan, and Daniela Rus. City-scale traffic estimation from a roving sensor network. In *Proc. of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pages 141–154, Toronto, ON, Canada, 2012. **[Cited on page 20]**

[67] Pablo Samuel Castro, Daqing Zhang, Chao Chen, Shijian Li, and Gang Pan. From Taxi GPS Traces to Social and Community Dynamics: A Survey. *ACM Computing Surveys*, 46:17:1–34, 2013. **[Cited on page 20]**

[68] Luis Moreira-Matias, J. Gama, Michel Ferreira, J. Mendes-Moreira, and Luis Damas. Predicting Taxi-Passenger Demand Using Streaming Data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402, 2013. **[Cited on page 20]**

[69] Siyuan Liu, Yunhuai Liu, Lionel M. Ni, Jianping Fan, and Minglu Li. Towards Mobility-based Clustering. In *Proc. of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 919–927, Washington, DC, USA, 2010. **[Cited on page 20]**

[70] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining Interesting Locations and Travel Sequences from GPS Trajectories. In *Proc. of the 18th international conference on World wide web (WWW)*, pages 791–800, Madrid, Spain, 2009. **[Cited on page 20]**

[71] C. Qiao, M. Lu, Y. Zhang, and K. N. Brown. An efficient dispatch and decision-making model for taxi-booking service. In *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pages 392–398, Aug 2015. **[Cited on page 20]**

[72] Hongjian Wang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. A simple baseline for travel time estimation using large-scale trip data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '16, pages 61:1–61:4, New York, NY, USA, 2016. ACM. **[Cited on page 20]**

[73] Austin W. Smith, Andrew L. Kun, and John Krumm. Predicting taxi pickups in cities: Which data sources should we use? In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM*

International Symposium on Wearable Computers, UbiComp '17, pages 380–387, New York, NY, USA, 2017. ACM. **[Cited on page 20]**

[74] Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, and Xing Xie. T-Finder: A Recommender System for Finding Passengers and Vacant Taxis. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 25(10):2390–2403, 2013. **[Cited on pages 20, 44, and 52]**

[75] Ren-Hung Hwang, Yu-Ling Hsueh, and Yu-Ting Chen. An effective taxi recommender system based on a spatio-temporal factor analysis model. *Information Sciences*, 314:28–40, 2015. **[Cited on pages 20 and 42]**

[76] Xudong Zheng, Xiao Liang, and Ke Xu. Where to Wait for a Taxi? In *Proc. of the ACM SIGKDD International Workshop on Urban Computing (KDD'12)*, pages 149–156, Beijing, China, 2012. **[Cited on page 20]**

[77] Rajesh Krishna Balan, Nguyen Xuan Khoa, , and Lingxiao Jiang. Real-Time Trip Information Service for a Large Taxi Fleet. In *Proc. of the International Conference on Mobile Systems, Applications, and Services (MobiSys'11)*, Bethesda, Maryland, USA, 2011. **[Cited on page 20]**

[78] Leyi Song, Chengyu Wang, Xiaoyi Duan, Bing Xiao, Xiao Liu, Rong Zhang, Xiaofeng He, and Xueqing Gong. TaxiHailer : A Situation-Specific Taxi Pick-Up Points Recommendation System. In *Proc. of the 19th International Conference of Database Systems for Advanced Applications, LNCS*, pages 523–526, Bali, Indonesia, 2014. **[Cited on pages 20 and 42]**

[79] Guande Qi, Gang Panand Shijian Li, Zhaohui Wu, Daqing Zhang, Lin Sun, and Laurence Tianruo Yang. How Long a Passenger Waits for a Vacant Taxi – Large-Scale Taxi Trace Mining for Smart Cities. In *Proc. of the IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 1029–1036, Beijing, 2013. **[Cited on pages 20 and 42]**

[80] Jie Xu, Dingxiong Deng, Ugur Demiryurek, Cyrus Shahabi, and Mihaela van der Schaar. Context-aware online spatiotemporal traffic prediction. In *Proc. of the IEEE Interna-*

tional Conference on Data Mining Workshops (ICDMW), pages 43–46, 2015. **[Cited on page 20]**

[81] Wei Wu, Wee Siong Ng, Shonali Krishnaswamy, and Abhijat Sinha. To Taxi or Not To Taxi? - Enabling Personalised and Real-Time Transportation Decisions for Mobile Users. In *Proc. of the IEEE Int. Conf. on Mobile Data Management (MDM)*, pages 320–323, 2012. **[Cited on pages 20 and 43]**

[82] Ke Zhang, Ke Zhang, Supeng Leng, and Shuo Xu. Adaptive Airport Taxi Dispatch Algorithm Based on PCA-WNN. *Proc. of the IEEE 11th International Conference on Dependable, Autonomic and Secure Computing*, pages 340–343, December 2013. **[Cited on page 21]**

[83] NYC-TLC. Tlc trip record data. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. Last Accessed: 10-Feb-2015. **[Cited on page 23]**

[84] Chris Whong. Should i stay or should i go? nyc taxis at the airport. http://chriswhong.com/open-data/should-i-stay-or-should-i-go-nyc-taxis-at-the-airport/. Last Accessed: 20-Feb-2015. **[Cited on page 25]**

[85] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009. **[Cited on page 32]**

[86] Ron Davis, Tamara Rogers, and Yingping Huang. A Survey of Recent Dev. in Queue Wait Time Forecasting Methods. In *Proc. of the Int. Conf. on Foundations of Comp. Science*, pages 84–90, 2016. **[Cited on page 41]**

[87] N. S. Altman. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *ACM Transactions on Sensor Networks*, 46(3:175), 1992. **[Cited on page 41]**

[88] Yongli Ren, Gang Li, Jun Zhang, and Wanlei Zhou. The efficient imputation method for neighborhood-based collaborative filtering. In *Proc. of the ACM International Conference*

on Information and Knowledge Management (CIKM), pages 684–693, 2012. **[Cited on page 41]**

[89] Yuki Endo, Hiroyuki Toda, Kyosuke Nishida, and Jotaro Ikedo. Classifying spatial trajectories using representation learning. *International Journal of Data Science and Analytics*, 2(3):107–117, Dec 2016. **[Cited on page 42]**

[90] Johannes Paefgen, Florian Michahelles, and Thorsten Staake. Gps trajectory feature extraction for driver risk profiling. In *Proceedings of the 2011 International Workshop on Trajectory Data Mining and Analysis*, TDMA '11, pages 53–56, New York, NY, USA, 2011. ACM. **[Cited on page 42]**

[91] Yasuko Matsubara, Lei Li, Evangelos Papalexakis, David Lo, Yasushi Sakurai, and Christos Faloutsos. F-trail: Finding patterns in taxi trajectories. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 86–98, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. **[Cited on page 42]**

[92] Andreas Keler, Jukka M. Krisp, and Linfang Ding. Detecting vehicle traffic patterns in urban environments using taxi trajectory intersection points. *Geo-spatial Information Science*, 20(4):333–344, 2017. **[Cited on page 42]**

[93] Shane B Eisenman, Emiliano Miluzzo, Nicholas D Lane, Ronald A Peterson, Gahng-Seop Ahn, and Andrew T Campbell. BikeNet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks*, 6(1):6, 2009. **[Cited on page 43]**

[94] Jon Froehlich, Joachim Neumann, and Nuria Oliver. Sensing and Pred. the Pulse of the City through Shared Bicycling. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1420–1426, 2009. **[Cited on page 43]**

[95] Yun-Wei Lin and Yi-Bing Lin. Mobile Ticket Dispenser System With Waiting Time Prediction. *IEEE Transaction on Vehicular Technology*, 64(8):3689–3696, 2015. **[Cited on page 43]**

[96] Oleg S. Pianykh and Daniel I. Rosenthal. Can We Predict Patient Wait Time? *Journal of the American College of Radiology*, 12(10):1058–1066, 2015. **[Cited on page 43]**

[97] Jorge Goncalves, Hannu Kukka, Iván Sánchez, and Vassilis Kostakos. Crowdsourcing Queue Estimations in Situ. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW)*, pages 1040–1051, 2016. **[Cited on page 43]**

[98] Arik Senderovich, Matthias Weidlich, Avigdor Gal, and Avishai Mandelbaum. Queue mining for delay prediction in multi-class service processes. *Information Systems*, 53:278–295, 2015. **[Cited on page 43]**

[99] Avigdor Gal, Avishai Mandelbaum, François Schnitzler, Arik Senderovich, and Matthias Weidlich. Traveling time prediction in scheduled transportation with journey segments. *Information Systems*, 64:266–280, 2017. **[Cited on page 43]**

[100] Safak Kr, Nihat Yzgll, Nurdan Ergn, and A. Alper evik. A knowledge-based scheduling system for Emergency Departments. *Knowledge-Based Systems*, 23(8):890–900, 2010. **[Cited on page 43]**

[101] Shiliang Pu, Tao Song, Yuan Zhang, and Di Xie. Estimation of crowd density in surveillance scenes based on deep convolutional neural network. *Procedia Computer Science*, 111:154 – 159, 2017. The 8th International Conference on Advances in Information Technology. **[Cited on page 43]**

[102] J. Weppner and P. Lukowicz. Bluetooth based collaborative crowd density estimation with mobile phones. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 193–200, March 2013. **[Cited on page 43]**

[103] Toshio Ito and Ryohei Kaneyasu. Predicting traffic congestion using driver behavior. *Procedia Computer Science*, 112:1288 – 1297, 2017. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France. **[Cited on page 43]**

[104] Simon Kwoczek, Sergio Di Martino, and Wolfgang Nejdl. Predicting and visualizing traffic congestion in the presence of planned special events. *Journal of Visual Languages & Computing*, 25(6):973 – 980, 2014. Distributed Multimedia Systems DMS2014 Part I. **[Cited on page 43]**

[105] Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T Campbell. NextPlace: a spatio-temporal prediction framework for pervasive systems. In *Proc. of the 9th international. conf. on Pervasive computing*, pages 152–169, 2011. **[Cited on page 43]**

[106] Guande Qi, Gang Pan, Shijian Li, Zhaohui Wu, Daqing Zhang, Lin Sun, and Laurence Tianruo Yang. How Long a Passenger Waits for a Vacant Taxi? Large-scale Taxi Trace Mining for Smart Cities. In *Proc. of the 2013 IEEE Int. Conf. on Green Comp. and Comm. and IEEE IoT and IEEE Cyber, Physical and Soc. Comp.*, pages 1029–1036, 2013. **[Cited on page 43]**

[107] Vsevolod Salnikov, Renaud Lambiotte, Anastasios Noulas, and Cecilia Mascolo. OpenStreetCab: Exploiting Taxi Mobility Patterns in New York City to Reduce Commuter Costs. In *CoRR, abs/1503.03021*, 2015. **[Cited on page 43]**

[108] Krisztian Buza, Alexandros Nanopoulos, and Gbor Nagy. Nearest neighbor regression in the presence of bad hubs. *Knowledge-Based Systems*, 86:250–260, 2015. **[Cited on page 44]**

[109] J. Gou, L. Du, Y. Zhang, and T. Xiong. A New Distance-weighted k-nearest Neighbor Classifier. *Journal of Info. & Comp. Sc.*, 9(6):1429–1436, 2012. **[Cited on page 44]**

[110] S. Dudani. The Distance Weighted k-Nearest Neighbor Rule. *EEE Transactions on Systems, Man, and Cybernetics*, 4(1):325–327, 1975. **[Cited on page 44]**

[111] A. T. Lora, J. M. R. Santos, A. G. Expsito, J. L. M. Ramos, and J. C. R. Santos. Electricity Market Price Forecasting Based on Weighted Nearest Neighbors Techniques. *IEEE Transaction on Power Systems*, 22(3):1294–1301, 2007. **[Cited on page 44]**

[112] Maryam Dialameh and Mansoor Zolghadri Jahromi. A general feature-weighting function for classification problems. *Expert Systems with Applications*, 72:177–188, 2017. **[Cited on page 44]**

[113] Aiguo Wang, Ning An, Guilin Chen, Lian Li, and Gil Alterovitz. Accelerating wrapper-based feature selection with K-nearest-neighbor. *Knowledge-Based Systems*, 83:81–91, 2015. **[Cited on page 44]**

[114] M R A Iqbal, Mohammad Saiedur Rahaman, and S I Nabil. Construction of Decision Trees by Using Feature Importance Value for Improved Learning Performance. In *Proc. of the International Conference on Neural Information Processing*, pages 242–249, 2012. **[Cited on page 44]**

[115] Diego P. Vivencio, Estevam R. Hruschka Jr, M. do Carmo Nicoletti, Edimilson B. dos Santos, and Sebastian D. C. 0. Galvio. Feature-weighted k-nearest Neighbor Classifier. In *Proc. of the IEEE Sym. on Found. of Comp. Intelligence*, pages 481–486, 2007. **[Cited on page 44]**

[116] N. Jankowski and K. Usowicz. Analysis of Feature Weighting Methods Based on Feature Ranking Methods for Classification. In *Proc. of the International Conference on Neural Information Processing*, pages 238–247, 2011. **[Cited on page 44]**

[117] P. J. Garcia-Laencina, J. Sancho-Gomez, A.R. Figueiras-Vidal, and M. Verleysen. K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputing*, 72:1483–1493, 2009. **[Cited on page 44]**

[118] Zhibin Pan, Yidi Wang, and Weiping Ku. A new general nearest neighbor classification based on the mutual neighborhood information. *Knowledge-Based Systems*, 121:142–152, 2017. **[Cited on page 44]**

[119] D. W. Aha. Feature Weighting for Lazy Learning Algorithms. In *Feature Extraction, Cons. and Selection: A Data Mining Perspective*, pages 13–32, 1998. **[Cited on page 44]**

[120] Max Kuhn. *Variable Selection Using The caret Package*, 2010. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.168.1655, Last Accessed: 10-May-2017. **[Cited on page 65]**

[121] Marie H. Murphy, Alan M. Nevill, Elaine M. Murtagh, and Roger L. Holder. The effect of walking on fitness, fatness and resting blood pressure: A meta-analysis of randomised, controlled trials. *Preventive Medicine*, 44(5):377 – 385, 2007. **[Cited on page 72]**

[122] Guy E.J. Faulkner, Ron N. Buliung, Parminder K. Flora, and Caroline Fusco. Active school transport, physical activity levels and body weight of children and youth: A systematic review. *Preventive Medicine*, 48(1):3 – 8, 2009. **[Cited on page 72]**

[123] The World Bank. World Bank Group Annual Report 2014,Population Ages 65 And Above (% of Total). http://data.worldbank.org/indicator/SP.POP.65UP.TO.ZS. Last Accessed: 27-Jul-2015. **[Cited on page 72]**

[124] United Nations. United Nations, World Population Ageing 2013. http://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2013.pdf. Last Accessed: 28-Jul-2015. **[Cited on page 72]**

[125] H. Wennberg, A. Ståhl, and C. Hydén. Older Pedestrians' Perceptions of The Outdoor Environment in A Year-round Perspective. *European Journal of Ageing*, 6(4):277–290, 2009. **[Cited on page 73]**

[126] K. Corcoran, J. McNab, S. Girgis, and R. Colagiuri. Is Transport a Barrier To Healthcare for Older People With Chronic Diseases? *Asia Pacific Journal of Health Management*, 7(1):49–56, 2012. **[Cited on page 73]**

[127] M. Whelan, J. Langford, J. Oxley, S. Koppel, and J. Charlton. *The Elderly and Mobility: A Review of The Literature*. Monash University Accident Research Centre Australia, 2006. **[Cited on page 73]**

[128] J.A. Davey. Older People and Transport: Coping Without A Car. *Ageing and Society*, 27(1):49–65, 2007. **[Cited on page 73]**

[129] Disability Services Commission (Govt. of WA, Australia). Buildings and facilities checklist:Access and Inclusion Resource Kit-January 2014. http://www.disability.wa.gov.au/Global/Publications/For%20business%20and%20government/DAIPs/Buildings-and-facilities-checklist-for-Outcome-2-and-Outcome-7.pdf. Last Accessed: 10-Feb-2016. **[Cited on page 74]**

[130] United Nations. Accessibility for the Disabled - A Design Manual for a Barrier Free Environment. http://www.un.org/esa/socdev/enable/designm/AD2-01.htm. Last Accessed: 20-Feb-2016. **[Cited on page 74]**

[131] T.G Lin, J.C. Xia, T.P. Robinson, K.G. Goulias, R.L. Church, D. Olaru, J. Tapin, and R. Han. Spatial Analysis of Access to And Accessibility Surrounding Train Stations: A Case Study of Accessibility for The Elderly in Perth, Western Australia. *Journal of Transport Geography*, 39:111–120, 2014. **[Cited on page 75]**

[132] L. Ferrari, M. Berlingerio, F. Calabrese, and B. Curtis-Davidson. Measuring public transport accessibility using pervasive mobility data. *IEEE Pervasive Computing*, pages 26–33, 2013. **[Cited on pages 75 and 76]**

[133] J. Meurer, M. Stein, D. Randall, M. Rohde, and V. Wulf. Social Dependency and Mobile Autonomy: Supporting Older Adults' Mobility with Ridesharing ICT. In *Proceedings of The 32nd Annual ACM Conference on Human Factors in Computing Systems*, pages 1923–1932. ACM, 2014. **[Cited on pages 75 and 76]**

[134] M. Nasir, C.P. Lim, S. Nahavandi, and D. Creighton. Prediction of Pedestrians Routes Within a Built Environment in Normal Conditions. *Expert Systems with Applications*, 41(10):4975–4988, 2014. **[Cited on pages 75 and 76]**

[135] S. Alghamdi, R. van Schyndel, and M. Hamilton. Blind User Response to a Navigational System to Assist Blind People Using Active RFID and QR-Code. In *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, pages 313–316. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014. **[Cited on page 76]**

[136] T. Miura, K. Yabu, M. Sakajiri, M. Ueda, J. Suzuki, A. Hiyama, M. Hirose, and T. Ifukube. Social Platform for Sharing Accessibility Information Among People with Disabilities: Evaluation of a Field Assessment. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 2013. **[Cited on page 76]**

[137] K. Hara, S. Azenkot, M. Campbell, C.L. Bennett, V. Le, S. Pannella, R. Moore, K. Minckler, R.H. Ng, and J.E. Froehlich. Improving Public Transit Accessibility for Blind Riders by Crowdsourcing Bus Stop Landmark Locations with Google Street View. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 2013. **[Cited on page 76]**

[138] C. Cardonha, D. Gallo, P. Avegliano, R. Herrmann, F. Koch, and S. Borger. A Crowdsourcing Platform for The Construction of Accessibility Maps. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. ACM, 2013. **[Cited on page 76]**

[139] C. Prandi, P. Salomoni, and S. Mirri. mPASS: Integrating People Sensing and Crowdsourcing to Map Urban Accessibility. In *Proceedings of the IEEE International Conference on Consumer Communications and Networking Conference*, pages 10–13, 2014. **[Cited on page 76]**

[140] D. Quercia, R. Schifanella, and L.M. Aiello. The Shortest Path to Happiness: Recommending Beautiful, Quiet, and Happy Routes in The City. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, pages 116–125. ACM, 2014. **[Cited on pages 76 and 77]**

[141] C. Jason, Z. Yongxin, and T. Lei. Where to: Crowd-aided path selection. In *40th International Conference on Very Large Data Bases (VLDB'2014)*, pages 2005–2016. ACM, 2014. **[Cited on pages 76, 77, and 117]**

[142] M. Traunmueller and A. Fatah gen Schieck. Introducing the Space Recommender System: How Crowd-sourced Voting Data Can Enrich Urban Exploration in The Digital Era. In

Proceedings of the 6th International Conference on Communities and Technologies, pages 149–156. ACM, 2013. **[Cited on pages 76 and 77]**

[143] M. Hamilton, F. Salim, E. Cheng, and S. Choy. Transafe: A Crowdsourced Mobile Platform for Crime and Safety Perception Management. In *IEEE International Symposium on Technology and Society 2011*, pages 32–37. IEEE, 2011. **[Cited on pages 76, 77, and 117]**

[144] S. Elsmore, I.F. Subastian, F. Salim, and M. Hamilton. VDIM: Vector-based Diffusion and Interpolation Matrix for Computing Region-based Crowdsourced Ratings: Towards Safe Route Selection for Human Navigation. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, pages 212–215. ACM, 2014. **[Cited on pages 76 and 77]**

[145] John Krumm and Eric Horvitz. Risk-aware planning: Methods and case study for safer driving routes. In *Proc. of the 29th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI 2017)*, 2017. **[Cited on page 77]**

[146] S.S. Wibowo and P. Olszewski. Modeling Walking Accessibility to Public Transport Terminals: Case Study of Singapore Mass Rapid Transit. *Journal of the Eastern Asia Society for Transportation Studies*, 6:147–156, 2005. **[Cited on page 77]**

[147] Walkscore. Walkscore.com. http://www.walkscore.com. Last Accessed: 10-Jan-2014. **[Cited on pages 77, 78, and 117]**

[148] J. Li. Map Route Ranking with Weighted Distance using Environmental Factors. *arXiv preprint arXiv:1404.0934*, 2014. **[Cited on pages 77 and 78]**

[149] Google. Google Maps. http://www.maps.google.com. Last Accessed: 10-Jan-2014. **[Cited on pages 77 and 78]**

[150] W. Sasaki and Y. Takama. Walking Route Recommender System Considering SAW Criteria. In *Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on*, pages 246–251. IEEE, 2013. **[Cited on pages 77, 78, and 117]**

[151] T. Völkel and G. Weber. RouteCheckr: Personalized Multicriteria Routing for Mobility Impaired Pedestrians. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 185–192. ACM, 2008. **[Cited on pages 77 and 78]**

[152] H. Wang, G. Li, H. Hu, S. Chen, B. Shen, H. Wu, W.S. Li, and K.L. Tan. R3: A Real-Time Route Recommendation System. *Proceedings of the VLDB Endowment*, 7(13):1549–1552, 2014. **[Cited on pages 77 and 78]**

[153] M. Guentert. Improving Public Transit Accessibility for Blind Riders: A Train Station Navigation Assistant. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 317–318. ACM, 2011. **[Cited on page 79]**

[154] R. Wasinger, C. Stahl, and A. Krüger. M3I in a Pedestrian Navigation & Exploration System. In *Human-Computer Interaction with Mobile Devices and Services*, pages 481–485. Springer, 2003. **[Cited on page 79]**

[155] B. Krieg-Brückner, C. Mandel, C. Budelmann, B. Gersdorf, and A.B. Martínez. Indoor and Outdoor Mobility Assistance. In *Ambient Assisted Living*, pages 33–52. Springer, 2015. **[Cited on page 79]**

[156] E.W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische mathematik*, 1(1):269–271, 1959. **[Cited on page 82]**

[157] P.E. Hart, N.J. Nilsson, and B. Raphael. A Formal Basis For The Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. **[Cited on pages 82 and 87]**

[158] M. Valtorta. A Result on The Computational Complexity of Heuristic Estimates for the A* Algorithm. *Information Sciences*, 34(1):47–59, 1984. **[Cited on page 82]**

[159] R. Geisberger, P. Sanders, D. Schultes, and D. Delling. Contraction hierarchies: Faster and Simpler Hierarchical Routing in Road Networks. In *Experimental Algorithms*, pages 319–333. Springer, 2008. **[Cited on page 82]**

[160] Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. Engineering Route Planning Algorithms. In *Algorithmics of Large and Complex Networks*, pages 117–139. Springer, 2009. **[Cited on page 82]**

[161] Y. Disser, M. Müller-Hannemann, and M. Schnee. Multi-criteria Shortest Paths in Time-dependent Train Networks. In *Experimental Algorithms*, pages 347–361. Springer, 2008. **[Cited on page 82]**

[162] L. Mandow and J.L.P. De La Cruz. Multiobjective A* Search with Consistent Heuristics. *Journal of the ACM (JACM)*, 57(5), 2010. **[Cited on page 82]**

[163] OpenStreetMap. OpenStreetMap, Srtm2Osm. http://wiki.openstreetmap.org/wiki/Srtm2Osm. Last Accessed: 26-Feb-2015. **[Cited on pages 83 and 85]**

[164] NASA. Shuttle Radar Topography Mission (SRTM). http://srtm.usgs.gov/. Last Accessed: 26-Feb-2015. **[Cited on pages 83 and 86]**

[165] R.H. Güting, Th. de Ridder, and M. Schneider. Implementation of the ROSE Algebra: Efficient Algorithms for Realm-Based Spatial Data Types. In *Proc. of the 4th Intl. Symposium on Large Spatial Databases*, pages 216–239, 1995. **[Cited on page 84]**

[166] Google. Google Elevation API. https://developers.google.com/maps/documentation/elevation/. Last Accessed: 26-Feb-2015. **[Cited on pages 84 and 87]**

[167] OpenStreetMap. OpenStreetMap (OSM). http://www.openstreetmap.org/. Last Accessed: 26-Feb-2015. **[Cited on pages 85 and 86]**

[168] OpenStreetMap. JOSM: OpenStreetMap (OSM) Editor in JAVA. https://josm.openstreetmap.de/. Last Accessed: 26-Feb-2015. **[Cited on page 85]**

[169] Department of Transport and Australia Main Roads, State of Queensland. Technical note 38: Longitudinal grades for footpaths, walkways and bikeways, 2010. http://www.tmr.qld.gov.au/. Last Accessed: 20-Dec-2016. **[Cited on pages 86, 89, and 90]**

[170] Google. Google geocoding api. https://developers.google.com/maps/documentation/geocoding/. Last Accessed: 23-Dec-2016. **[Cited on page 87]**

[171] W. Tobler. Three presentations on geographical analysis and modeling: Non- isotropic geographic modeling; speculations on the geometry of geography; and global spatial analysis, 1993. **[Cited on page 92]**

[172] L. Mandow and J.L.P. De La Cruz. A New Approach to Multiobjective A* Search. In *Proceedings of the 19th international joint conference on Artificial intelligence (IJCAI '05)*, pages 218–223. Citeseer, 2005. **[Cited on pages 98 and 109]**

[173] Google. Google Directions API. https://developers.google.com/maps/documentation/directions/. Last Accessed: 3-July-2015. **[Cited on page 100]**