Marshall University
Marshall Digital Scholar

Theses, Dissertations and Capstones

2019

A Machine Learning Recommender Model for Ride Sharing Based on Rider Characteristics and User Threshold Time

Govind Pramod Yatnalkar
yatnalkar@marshall.edu

Follow this and additional works at: https://mds.marshall.edu/etd

 Part of the Software Engineering Commons, and the Theory and Algorithms Commons

Recommended Citation

Yatnalkar, Govind Pramod, "A Machine Learning Recommender Model for Ride Sharing Based on Rider Characteristics and User Threshold Time" (2019). *Theses, Dissertations and Capstones*. 1259.
https://mds.marshall.edu/etd/1259

This Thesis is brought to you for free and open access by Marshall Digital Scholar. It has been accepted for inclusion in Theses, Dissertations and Capstones by an authorized administrator of Marshall Digital Scholar. For more information, please contact zhangj@marshall.edu, beachgr@marshall.edu.

A MACHINE LEARNING RECOMMENDER MODEL FOR RIDE SHARING BASED ON RIDER CHARACTERISTICS AND USER THRESHOLD TIME

A thesis submitted to
the Graduate College of
Marshall University
In partial fulfillment of
the requirements for the degree of
Master of Science
in
Computer Science
by
Govind Pramod Yatnalkar
Approved by
Dr. Wook-Sung Yoo, Committee Chairperson
Dr. Husnu S. Narman
Dr. Haroon Malik

Marshall University
December 2019

We, the faculty supervising the work of Govind Pramod Yatnalkar, affirm that the thesis, A MACHINE LEARNING RECOMMENDER MODEL FOR RIDE SHARING BASED ON RIDER CHARACTERISTICS AND USER THRESHOLD TIME, meets the high academic standards for original scholarship and creative work established by the Weisberg Division of Computer Science and the College of Information Technology and Engineering. This work also conforms to the editorial standards of our discipline and the Graduate College of Marshall University. With our signatures, we approve the manuscript for publication.

Dec. 9, '19

Dr. Wook-Sung Yoo, Weisberg Division of Computer Science
Committee Chairperson — Date

Dec. 9, 2019

Dr. Husnu S. Narman, Weisberg Division of Computer Science
Committee Member — Date

Dec-09-2019

Dr. Haroon Malik, Weisberg Division of Computer Science
Committee Member — Date

© 2019
Govind Pramod Yatnalkar
ALL RIGHTS RESERVED

ACKNOWLEDGEMENTS

I am thankful to my parents, Mr. Pramod Yatnalkar and Mrs. Nandini Yatnalkar, my brother, Mr. Prabodh Yatnalkar, his wife, Mrs. Manjiri Yatnalkar, and their lovely daughter, Madhura for providing me the support which was essential for accomplishing the crucial task of the thesis.

I extend my gratitude towards my thesis advisor, Dr. Husnu Narman, for his contributions to my thesis. Without his adept suggestions and expert guidance, I would not have reached this point of achievement in my research work. Along with Dr. Narman, I am highly obliged to my thesis committee members, Dr. Wook-Sung Yoo and Dr. Haroon Malik, for their helpful assistance and mentorship in the reviewing and fulfillment of the thesis document.

At last, I would like to thank the Weisberg Division of Computer Science, College of Information Technology and Engineering, Marshall University, for providing me an opportunity to present my research work. It was a huge honor to work with highly expertise professionals, along with the support of my friends, which led to the completion of my thesis work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In the present age, human life is prospering incredibly due to the 4^{th} Industrial Revolution or The Age of Digitization and Computing. The ubiquitous availability of the Internet and advanced computing systems have resulted in the rapid development of smart cities. From connected devices to live vehicle tracking, technology is taking the field of transportation to a new level. An essential part of the transportation domain in smart cities is Ride Sharing. It is an excellent solution to issues like pollution, traffic, and the rapid consumption of fuel. Even though Ride Sharing has several benefits, the current usage is significantly low due to limitations like social barriers and long rider waiting times. The thesis proposes a novel Ride Sharing model with two matching layers to eliminate most of the observed issues in the existing Ride Sharing applications like UberPool and LyftLine. The first matching layer matches riders based on specific human characteristics, and the second matching layer provides riders the option to restrict the waiting time by using personalized threshold time. At the end of trips, the system collects user feedback according to five characteristics. Then, at most, two main characteristics that are the most important to riders are determined based on the collected feedback. The registered characteristics and the two main determined characteristics are fed as the inputs to a Machine Learning classification module. For newly registering users, the module predicts the two main characteristics of riders, and that assists in matching with other riders having similar determined characteristics. The thesis includes subjecting the proposed model to an extensive simulation for measuring system efficiency. The model simulations have utilized the real-time New York City Cab traffic data with real-traffic conditions using Google Maps Application Programming Interface (API). Results indicate that the proposed Ride Sharing model is feasible, and efficient as the number of riders increases while maintaining the rider threshold time. The expected outcome of the thesis is to help service providers increase the usage of Ride Sharing, complete the pool for the maximum number of trips in minimal time and perform maximum rider matches based on similar characteristics, thus providing an energy-efficient and a social platform for Ride Sharing.

CHAPTER 1

INTRODUCTION

1.1 Impact of Automation and Ride Sharing

The world is progressing rapidly from a perspective of technology and innovation due to The 4^{th} Industrial Revolution. The ultimate motive of the revolution or the digitization is to convert time-consuming manual tasks to automation [1, 2]. Automation includes the interference of computing systems and sophisticated software to execute and speed up the manual tasks [3]. Also, automation not only acts as a catalyst for speeding up the processes but also significantly increases productivity. Several domains experience the usage of automation and technology like Finance and Banking, Manufacturing and Production, Information Technology (IT) Industry, Education, Health and Public Safety, Medicare, and many more [2, 4, 5]. The list also includes the domain of Transportation, on which the thesis mostly focuses [2, 3].

Transportation is one of the most vital domains for humankind [6]. The need for vehicles is comparable to the necessity of food and water to humans. Irrespective of weather conditions, vehicles provide the ability to swiftly traverse from a source to a destination [6, 7]. Engineers and researchers are continuously deploying advanced tools in vehicles, which offer smoother and faster transportation, making human life easier [6].

Incorporating the latest technologies avails a system to perform better and achieve more significant results [4]. The ability of a system to execute tasks while exploiting the features of the advanced technologies is a smart system [8]. Such systems include the coupling of an environment that is generally orchestrated by human beings with machinery plus computing power [2, 8]. A popular paradigm of smart systems is smart cities, and transportation forms a crucial component [4].

Smart transportation serves various benefits in terms of automobile communications and tracking. An example that revolutionized vehicle connectivity is manipulating cell controls via vehicle handles through Bluetooth and Network [9]. Also, driving is smarter and more manageable with features like Steering Assists, Cruise Control, and the latest innovation in the

market, Auto-Parking, and Auto-Pilot [10, 11]. Such examples constitute the smart transportation domain, and an essential part is Ride Sharing.

A simple definition of Ride Sharing is to share a ride among multiple users. The history of Ride Sharing dates back to the times of World War II [12, 13], during the oil and energy crisis. The concept of Ride Sharing emerged as a bright and potent idea of sharing a journey and was an effective way of saving plus sharing oil or fuel resources [13]. With time, world conditions improved, automobiles evolved, economies thrived, and as people started getting financially stable, a downfall in the utilization of Ride Sharing was observed, resulting in people owning self-purchased vehicles.

In the present age, under the topic of Green Computing, Ride Sharing is gaining much attention [14]. Ride Sharing is synonymous with names like Ride-Hailing, Car-Pooling, and Vehicle-Pooling. Hence, in the further chapters of the thesis, the term Ride Sharing is referenced with Ride-Hailing, Car-Pooling, and Vehicle-Pooling. Utilizing the basic idea of Ride Sharing, the thesis proposes an Enhanced Ride Sharing Model (ERSM). The enhanced model addresses several issues as surveyed in current Ride Sharing applications.

1.2 Motivation

A ride in a smart automobile is safer and better as compared to conventional vehicles. Even though automobiles provide many benefits, they also give rise to many problems. The problems arise due to the continuously rising requirements of people for fuel resources. A study of previous applications led to the discovery of a relation between the number of vehicles and the current rising population. The relation states that as the population increases, the number of vehicles increases [7, 15]. It is valid that the immense growth in the overall number of vehicles has risen exponentially in the past decade, which has directly impacted the present traffic conditions [16]. Solutions like High Occupancy Vehicle (HOV) lanes are proposed to address the traffic issue [17] in existing Ride Sharing systems, but there has not been a significant improvement in current traffic scenarios [18].

With traffic, vehicle fuel consumption has increased exponentially, and in the coming years, there is a possibility of outrunning the natural resources [19]. Governments from many

countries are investing in technologies for renewable energy generation, but the rate of fuel consumption is much higher than the rate of renewable energy consumption [20]. Other hurdles include the installation and production costs for renewable energy generation. The byproducts of burning fuel are the smoke and harmful gas emissions that have detrimental effects on the environment and human health [21].

One of the primary issues in the domain of transportation is the pollution resulting due to emissions from many vehicles [22]. As the population increases, the number of vehicles and emissions increases, resulting in Global Warming [7, 15, 23]. Additionally, the emissions not only affect human health but every living being on the planet Earth [21]. For example, the number of reported cases of respiratory issues has hiked up to notable levels in the past five years [24]. An increase in the number of vehicles also leads to car accidents, and a minor but critical issue of the decrease in the number of parking spaces [23, 25].

Ride Sharing is a possible candidate solution to the aforementioned problems of traffic, air pollution, and rapid fuel consumption. It is the process of sharing a ride among people who are traversing through a series of sources and destinations. In Ride Sharing, the journey is completed by following a specific trajectory that is formed using multiple locations [25, 26]. Moreover, Ride Sharing increases the number of HOV lanes, providing a smoother traffic flow.

Currently, there exist many Ride Sharing applications. The thesis includes a detailed study of several Ride Sharing applications and has listed several limitations in the previous works. The three primary persisting problems in most cases are that riders do not reach the seating capacity of the vehicle, the system suddenly adds or accepts passengers on an ongoing trip, and riders avoid Ride Sharing due to the social barriers, as riders do not know with whom they are going to travel on an upcoming trip [27, 28]. Such factors lead to consumer disappointment and frustration. The motivation of the thesis is to provide solutions to the three aforementioned issues, along with several others that are stated in further chapters.

1.3 The Enhanced Ride Sharing Model

Ride Sharing is a definite practical solution if applied effectively [29]. For example, consider a case of five users who have their distinct vehicles for commuting purposes. If the five

users decide to share a ride, they cut down the usage of four cars. Eliminating the usage of four cars leads to an overall reduction in traffic and a decrease in fuel consumption plus carbon emissions by almost 80% [23, 30]. Additional advantages include splitting the stress, fatigue, and fares among riders, increasing parking spots, and encouraging social interactions with others during the journey [25, 30, 31, 32]. The use case of the five users sharing a ride is portrayed in Figure 1.

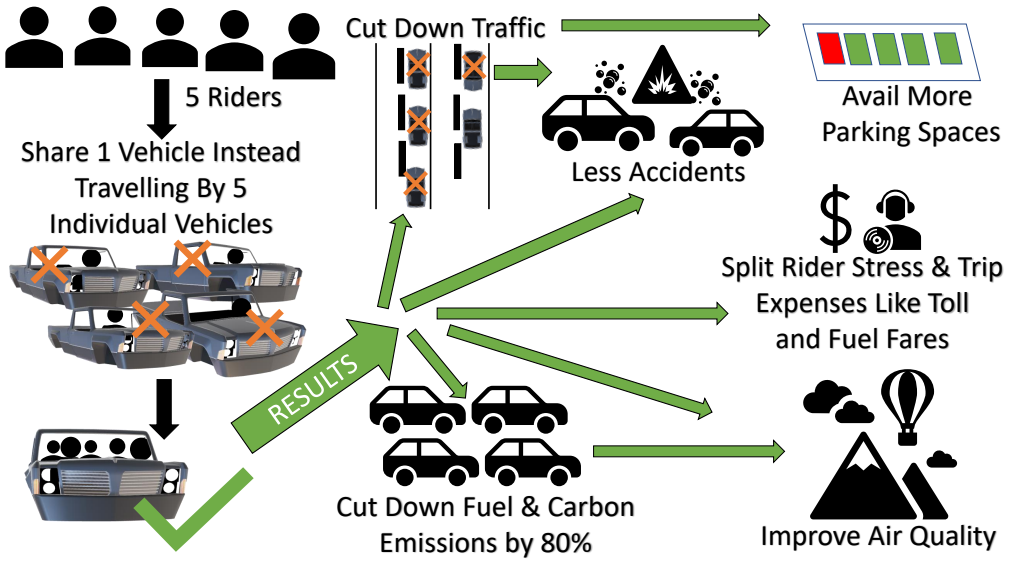

Figure 1: General Advantages of Ride Sharing.
Figure 1 represents the universal benefits of Ride Sharing by showcasing an example of five riders. All riders decide to share one vehicle instead of using five distinct vehicles. The result is reducing fuel consumption and gas emissions from the four vehicles. Other benefits include reduced traffic, fewer accidents, using one parking space instead of five parking spaces, and dividing the stress and expenses among users on the trip.

Currently, there exist problems in Ride Sharing applications like social barriers and the sudden rider addition without rider consent. Such factors cause people to avoid the usage of Ride Sharing [33, 34]. A fact obtained by research is that humans thrive on social relations and cannot stay isolated for long. Also, human beings tend to associate themselves with people having similar characteristics [35]. The thesis uses a similar kind of approach and utilizes human attributes in the rider matching layers. The *aim* of the thesis is to implement an Enhanced Ride Sharing Model that addresses the issues related to unknown characteristics of riders and the sudden elongation of the trip time. The Enhanced Ride Sharing Model, in a nutshell, is depicted in Figure 2.

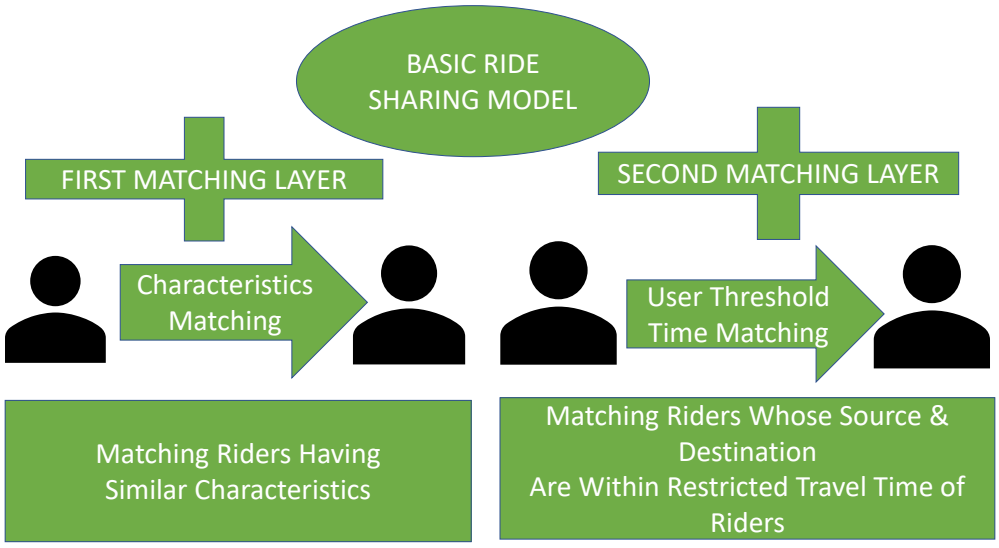

Figure 2: The Enhanced Ride Sharing Model (ERSM) in a Nutshell.
The Enhanced Ride Sharing Model includes the basic Ride Sharing approach with newly designed two matching layers. The first layer matches riders based on certain human characteristics. The second matching layer matches riders based on limited traveling time riders provide at the time of rider registration.

The designed model in the thesis includes Ride Sharing technology with two matching layers. The model begins with the rider registration, where users provide required profile data along with five specific characteristics. Characteristics are the requirements that define the search criteria for a match and are positive integers on a scale of 1 to 5. The selected human characteristics in the system are chatty, friendliness, safety, punctuality, and comfortability. Once a user registers to the system, the proposed model searches and matches riders having a similar set of characteristics. The matching of riders using the five characteristics constitutes the first matching layer.

The thesis proposes a novel concept of User Threshold Time (UTT). In registration, riders provide the User Threshold Time or User Tolerated Time. UTT is defined as the time in minutes that riders are willing to spend during the event of picking other riders. It is the maximum waiting time that both riders and drivers agree to accept a rider. UTT in the thesis is taken on a scale of 10 to 30 and in multiples of 5. Therefore, riders can select one of the following, 10, 15, 20, 25, and 30 as the UTT. Based on minimal UTT of a rider on a trip, drivers pick other riders to

respect the tolerated time of other riders. Hence, riders are only accepted if they are at a traveling time or waiting time, which is less than or equal to registered UTT. User Threshold Time assures travelers do not wait long, picking other riders during a journey.

The next stage in the proposed model is the execution of the matching layers, which begins with a broadcasting rider request. The ride request triggers a search for other riders having similar registered characteristics. The output list from the characteristics matching layer is the input to the UTT matching layer, where the traveling time between the locations of the broadcasting rider and other riders is computed using the Google Maps APIs and verified if the calculated time is less then trip UTT. If riders satisfy the matching layer conditions, the system adds them onto the final trip itinerary, marking the completion of trip formation.

After the trip formation, the execution of a novel designed feedback system begins where riders rate the driver as well as other riders on the trip. The feedback given by a user forms an essential data-set as the system uses the feedback data to compute the two main characteristics for every user. The determined characteristics are later employed by the Machine Learning algorithms to predict better rider recommendations. The generic control flow of the designed system in the thesis is reflected in Figure 3.

The two determined characteristics are the Feedback-Given-Characteristic and Feedback-Received-Characteristic. Feedback-Given-Characteristic is derived based on the feedback the rider gives to other riders, while Feedback-Received-Characteristic is computed based on the feedback the rider gets from other riders. The two main characteristics are used to determine the characteristics a rider most focuses on a trip while rating other riders. In the end, based on the feedback patterns in past trips, the system assigns the two most favored characteristics to every rider.

The computations for determining the main characteristics of a rider are quite complex and tediously high. Thus, after recording a sufficient number of trip and feedback records, the thesis made use of the Machine Learning classification algorithms or classifiers to predict the main characteristics of a rider, which eliminates the need for complex computations. Machine Learning (ML) is a technology where a system learns and trains based on an existing data-set and predicts outputs for new input data [36, 37]. In the case of the Ride Sharing model, the thesis employs the

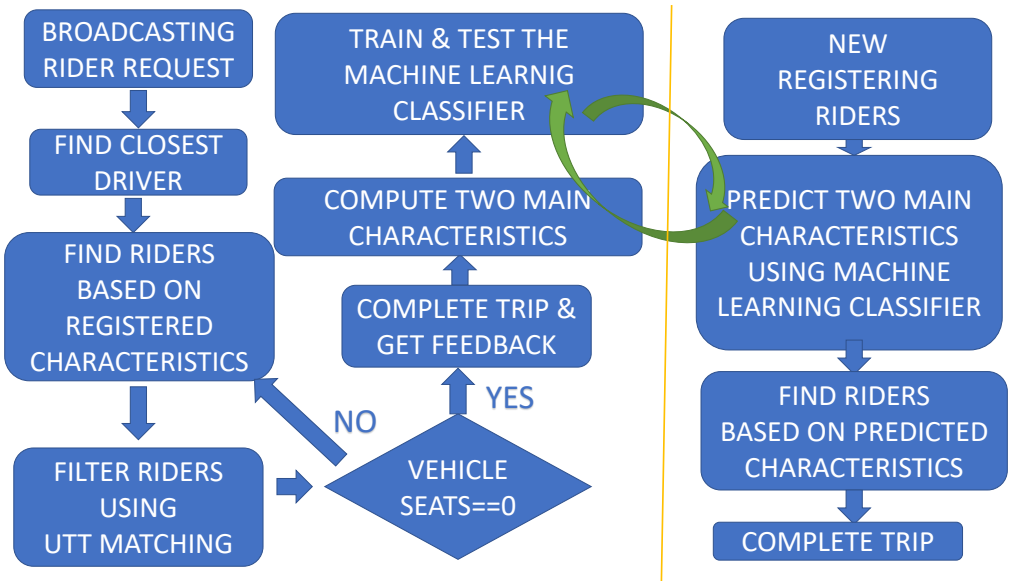

Figure 3: The Control Flow of Ride Sharing Model.
The control flow describes the consecutive execution steps of the system. The execution starts from the broadcasting rider request and follows by allocating a driver, executing matching layers, recording rider feedback, and computing the two main characteristics. The final step is to predict the two main characteristics based on the trained and tested Machine Learning classifier for the newly registering riders.

Support Vector Machine (SVM) classification algorithm. After appropriate training and testing, the SVM classifier predicts the two main characteristics of newly registering riders. Riders are recommended based on the predicted main characteristics.

In the chapter of results, the model's explorations and analysis showcase that it is possible to allocate the best-matched riders using characteristics and UTT. The proposed model in the thesis aims to increase the Ride Sharing while respecting rider considerations and decrease consumer frustration.

1.4 Contributions

The *key contributions* of the thesis are listed as follows:

i Performing the rider matching using the characteristics matching layer.

ii Filtering riders matched in the characteristics matching layer using the UTT matching layer.

iii Recording user feedback and computing the two main characteristics for every user, which are Feedback-Given-Characteristic and Feedback-Received-Characteristic.

iv Using a Machine Learning Algorithm to predict the two main characteristics and recommend riders for newly registering users.

v Evaluating the proposed model with an extensive simulation and real data to analyze the model efficiency.

Based on the user characteristics and UTT, the system allocates the riders based on similar characteristics on a trip to ensure they have a joyful and stress-free ride. The motive of the thesis is to reduce trip differences and promote an interactive journey. Through UTT, the model tries to minimize consumer frustrations in cases where the user unexpectedly waits for a long time on a short trip. The observations and results in the thesis show that The Enhanced Ride Sharing model is feasible, and can be deployed to increase the usage of Ride Sharing. Ultimately, the *objective* of the thesis is to enhance the usage of the present Ride Sharing services using the human characteristics, user feedback, and UTT, which will indirectly reduce the effects of Global Warming and increase the fuel reserves for future generations.

1.5 Organization of the Thesis

The organization of the rest of the thesis is as follows: Chapter 2 describes the related works for the present Ride Sharing applications. Chapter 3 includes the discussion of the system model, which possesses the problem statement and system architectures. Chapter 4 describes the methodologies followed for the proposed model, and Chapter 5 showcases the designs of the Enhanced Ride Sharing Model using Machine Learning algorithms and reports the simulations performed to test the system efficiency. Chapter 6 presents the model results plus observations, and Chapter 7 has the concluding remarks and plans to improve the proposed model.

CHAPTER 2

LITERATURE REVIEW

With the presence of advanced technologies and complex computing systems, there has been immense development in the field of Ride Sharing. Companies like Uber, Lyft, Via, Ola, and Juno are continuously developing ideas to improvise their applications and revenue models [38]. However, due to the lack of appropriate equipment and technology, Ride Sharing is discouraged in many states and countries. Even though governments are putting efforts and proposing plans to encourage Ride Sharing tactics like reducing taxes on vehicles affiliated with Ride Sharing applications and using public plus private transportation in conjunction with Ride Sharing services, the overall market for Ride Sharing remains low. [18, 39].

The chapter of the literature survey begins with the research of current popular Ride Sharing applications. The next section is of the study, which addresses different vehicle traversing approaches and the modern technologies integrated with the existing Ride Sharing applications. The section after the study of modern technologies with Ride Sharing applications presents the methods for determining the two main characteristics for a rider. In the final section of the chapter, the research provides the explorations of several Machine Learning classification algorithms. Also, the last section discusses the selected Machine Learning classifier, which is later utilized for predicting the two main characteristics.

2.1 Popular Ride Sharing Applications and Their Limitations

The literature survey began with an investigation of the most popular Ride Sharing applications like UberPool, LyftLine, Juno, Curb, Wingz, Via, Flywheel, Zimride, and Waze [40, 41, 42, 43, 44, 45, 46, 47, 48]. Uber, Lyft, Wingz, Via are Ride Sharing applications that allow any person to be a rider or driver [43, 46]. The study observed role restrictions in Juno, Gett, and Curb as they are taxi based Ride Sharing services [7]. The strong point of most of the applications is the usage of modern technologies like Internet of Things (IoT) and Cloud Computing. An application hosted with advanced technologies promotes quicker computations capabilities, easy availability of services, sophisticated notification abilities, and infinite data storage [16, 49].

Ride Sharing is employed throughout the United States of America, but states like New York, California, Florida, and Texas experience a higher usage of Ride Sharing services as compared to other states [50]. California is home to many Ride Sharing companies. Hence, Ride Sharing is highly popular in California. A separate Ride Sharing terminal at the San Francisco International Airport in California is a paradigm for the extensive usage of Ride Sharing [42, 44]. Gett, by Juno, is profoundly utilized in London, United Kingdom, as well as in the states of California, Texas, and New York in the US. New York City (NYC) Cab, which is a taxi-based service, is working with Uber, Lyft, Via plus Juno, and contributing notably to Ride Sharing services [51].

The findings from the research on the currently popular Ride Sharing applications listed several issues and some of the common limitations in all the applications are that drivers learn the count of passengers at the pickup location [34], and in most trips, the riders and driver do not reach the vehicle seating capacity [7, 26]. Additional issues include passengers do not possess the basic information of other passengers they are traveling with, unfair pricing for users [33], and the sudden addition of a rider whose destination is too far adds a significant time in trip completion [52]. Also, a critical issue observed is in the vehicle traversing approach or the route a car covers on a trip, which does not meet rider expectations of completing the journey in minimal time [53].

Acknowledging the listed issues in the existing Ride Sharing applications, the proposed model in the thesis is designed in a way that eliminates most of the problems. To accept most of the broadcasting riders in the system, the proposed model in the thesis presents the three types of rider matching. The first type of match is the Exact match, also referred to as the Same or Similar match. In the Exact match, the system finds riders with exactly matching characteristics. If the pool is incomplete or if the riders do not reach the seating capacity of the vehicle, the system triggers the search for riders with the second type of matching. The second type finds riders with Closer or Altered characteristics. Closer characteristics are the characteristics that are slightly different from the broadcasting rider's characteristics. If the pool is still incomplete, the system begins the third type of matching, which is comparable to the Uber and Lyft approach of matching riders [54]. The third type finds riders irrespective of characteristics i.e., matching based on the closest traveling time [54]. The third type of matching is called the Alternative type of

charactetistic matching as the system searches for passengers with alternative characteristics. The system serves most of the broadcasting rider requests by using the three types of characteristics matching and assures that it generates trips for a maximum number of riders. The three types of characteristics matching are portrayed in Figure 4.

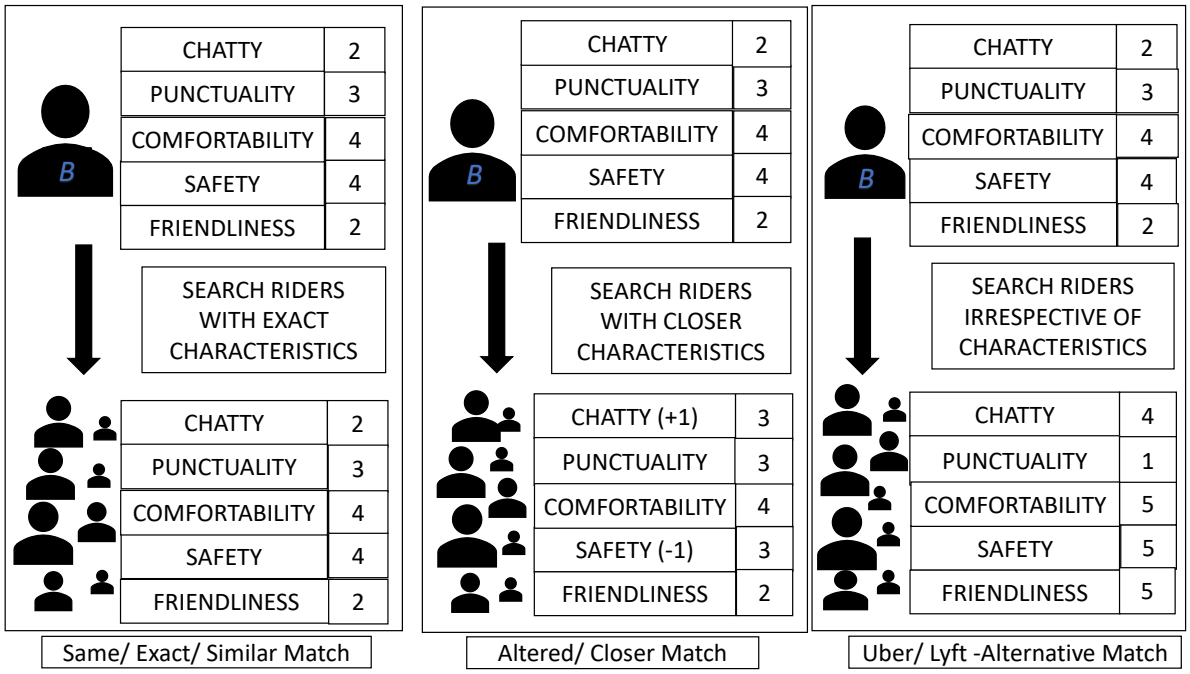

Figure 4: Three Types of Rider Matching.
The three types of rider matching are the Exact, Closer, and Alternative type of matching. The rider with the letter 'B' in the figure is the broadcasting rider. The Exact match searches riders with exactly matching characteristics. The Closer match searches riders with slightly different characteristics. In Figure 4, the riders to be searched in Closer matching have slightly different chatty and safety characteristics scores than the broadcasting rider scores. The Alternative matching searches riders who may have entirely different characteristics from the broadcasting rider characteristics. The three types of matching constitute the characteristics matching layer.

After completing the trip formation, the system sends the trip itinerary to every user, including the driver. The event of sending every user's basic information to other users reduces the social barriers among riders as riders get to know with whom they are going to travel on an upcoming trip. Also, with the User Threshold Time (UTT) matching layer and shortest path Multiple-Sources-Multiple-Destinations (MSMD), the ERSM ensures the accepted riders in a trip are not at a location that exceeds the trip's User Threshold Time. By studying the limitations in current Ride Sharing applications, it is concluded that for an application to be efficient and

popular, it is essential to implement the system features which reach the overall user expectations and improvises the user experience as much as possible [27, 28].

2.2 Modern Technologies with Ride Sharing

Significant technologies that are speeding up the building of smart cities are the Internet of Things (IoT), Artificial Intelligence, and Cloud Computing. Such technologies also contribute significantly to Ride Sharing services and applications.

IoT enables efficient device connectivity and communication while broadcasting the data. It is possible to push or send a notification to a million connected devices within a few seconds [55]. Integrated with Car-Pooling, every vehicle can connect and communicate to a data hub that logs every minor detail about the trip [16, 55]. Accordingly, the current status of a vehicle can be notified to broadcasting riders, facilitating faster decisions for road traversing, vehicle tracking, and location-based requests clustering. Such features result in continuous status updates, quicker rider-driver associations, and faster trip formation.

Cloud services bring numerous benefits to any computing system [56, 57]. Enabling Cloud services results in better system scalability, service availability, and efficient load balancing of requests [56]. Cloud services also decrease the overall costs of any system by offering resources like virtual machines, domain spaces for website hosting, and the databases. Also, the Cloud services facilitate efficient resource allocation plus management, and the resources are virtually made available within a few minutes.

If the system consumes too much time while responding to a client request, the system loses efficiency. In the Cloud environment, requests from a client device travels to the Cloud, interacts with the Cloud servers, and travels back to client devices to render server data introducing a latency. If the Cloud server and application reside at two different places, the traveling time of requests from the client to the server can cause a considerable time delay [49]. Further research on the Cloud Computing led to the finding of the topic, Fog Computing [58]. The Fog server constitutes a group of small servers that resides near the client location. Computations take place at the Fog server, which significantly reduces the request travel time as servers which are processing the client requests are placed nearer to the client machines than the

actual Cloud. [59].

The simulations in the thesis observed a large number of request and response transactions. For quicker computations of the client requests, the thesis utilized the technology of Fog Computing. Currently, the processing of requests takes place at a client machine that resides in a Cloud server. For storage purposes and exploiting the benefits of Cloud Computing in terms of databases [49], the system uses the Atlas MongoDB database, which is a Cloud-based database. To conclude, modern technologies play a crucial role in application design, data storage, and resource management. Also, factors like load balancing, timeliness of result, and quality of service are equally essential [26, 29, 49].

2.3 Multiple Sources Multiple Destinations (MSMD)

It is of utmost importance to meet the traversing requirements of the users. The traversing requirements are the possibilities of routing a vehicle to pick and drop users from their respective sources to destinations [47]. There are four traversing possibilities. The first traversing path is the Same-Source-Same-Destination (SSSD), where the trip starts from the same source and ends at the same destination for all riders. There are no stops included in the SSSD. The second traversing path is the Same-Source-Multiple-Destinations (SSMD), where users are picked up from the same source location and dropped at different destinations. In the third approach, which is the Multiple-Sources-Same-Destination (MSSD), riders are picked up from multiple locations, but they all end up at the same location. The research on the traversing modules observed the use of MSSD in many existing applications [40, 42, 44]. The last and most significant traversing approach is Multiple-Sources-Multiple-Destinations (MSMD). A notable feature of MSMD is that it includes all the traversing modules, which are SSSD, SSMD and MSSD [26]. MSMD reaches the primary user requirement that states users may start from multiple sources and end up on multiple destinations, or a trip may include multiple pickups and multiple drop-offs [28, 33]

The study on the vehicle traversing approaches included a search for algorithms that contributes to the formation of a MSMD path. The primary outcome of the MSMD approach is to consider the sources and destinations of all users and form an optimized itinerary. Some of the algorithms that promote the formation of a MSMD itinerary includes Mesh networks, Dijkstra's

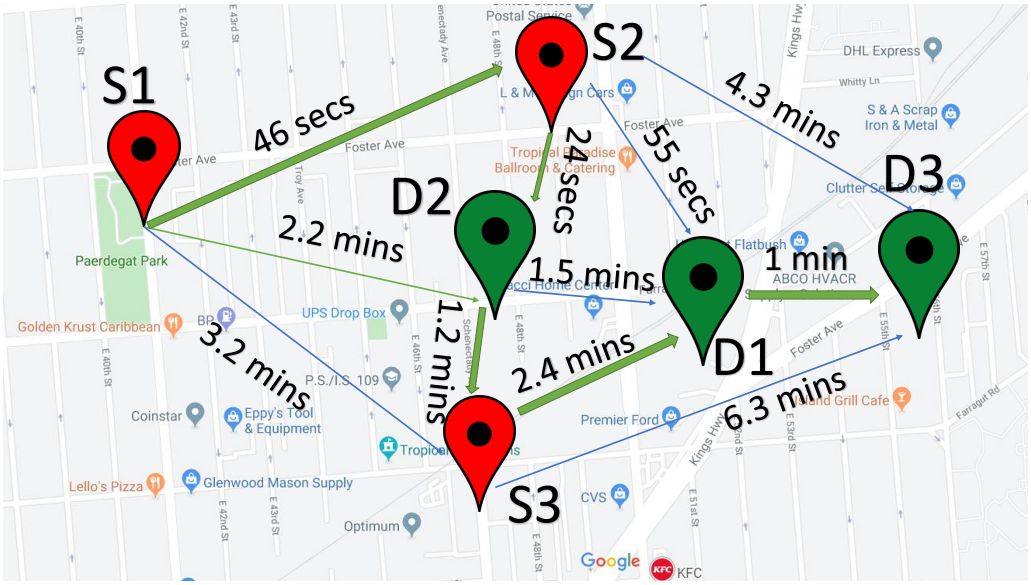

Figure 5: Multiple-Sources-Multiple-Destinations (MSMD) Traversing Approach. Figure 5 showcases the example of 3 riders having three sources, S1, S3, S3, and three destinations, D1, D2, D3. Initially, the system selects the broadcasting rider source and calculates the traveling time between all locations. The next station to be selected is the closest source or the destination to which the source is selected. The process continues until the system traverses through all the locations. Based on the computed traveling times, the green arrowed route shows the optimized travel path, which is S1-S2-D2-S3-D1-D3.

shortest path Many-Sources-Same-Destination approach, and Greedy algorithms [32, 39, 60].

The Mesh networks include the creation of a route based on the dynamic addition of locations [61]. The trip itinerary is regenerated if new locations are added to an ongoing trip [61, 39]. The drawback observed in the Mesh network is the computation time required for developing multiple optimized routes using different combinations of locations until finding the best one [39]. Another approach includes completing the journey through various public and private transportation systems like buses, cabs, and taxi [18, 22, 62]. In some cases, users had to walk a certain distance and meet other riders at a common location where passengers would be later picked for Ride Sharing. The limitation of completing the journey through different methods of transportation introduces latency due to the involvement and exchange of various means of transport during the entire trip.

The selected method in the thesis for creating a MSMD route is the Greedy algorithm [63]. In the Greedy algorithm, initially, any source from the available sources is selected. The

next step is the selection of the closest source or the destination of the rider whose source is initially selected. The process of location selection continues until all the locations are traversed. The journey created is an optimized one and formed by a Greedy approach. The modification performed in the thesis is starting the itinerary formation with the broadcasting rider's source and selecting further locations based on the traveling time instead of the traveling distance. Figure 5 demonstrates the formation of an optimised path using the MSMD vehicle traversing approach.

2.4 Tracking Rider Characteristics

One of the main motives of the thesis is to track or determine the main characteristics a rider most focuses while rating other riders. The method for tracking the main characteristic depends on feedback data. For example, a user may rate a score of 4 or a score of 0 to a specific characteristic for several trips, implying the user is less interested in a specific characteristic. The task is to find the characteristics the user is most interested in and recommend riders based on the computed main characteristics.

The research on the methods for tracking the main characteristics led to the finding of statistical methods like the range of a data-set [64], standard deviation [64], and variance [65, 66, 64]. The selected methodology for tracking the main rider characteristics is the variance. The concept of the variance is demonstrated with the help of stated three lists, L_1, L_2 and L_3.

$$L_1 = [1, 0, 5, 4, 0]$$

$$L_2 = [0, 0, 0, 0, 2]$$

$$L_3 = [4, 4, 4, 4, 4]$$

Let N be the total number of sample points in a list. The mean of the sample set is denoted by x_i. The distance of data-point x to the mean x_i or the spread of a specific sample point x around the mean x_i is computed by Equation 2.1 [66, 64].

$$x_{distance} = x - x_i \tag{2.1}$$

The variance of a data-set is computed using Equation 2.1. Variance indicates the level of spread of each sample point in a data-set [64]. Variance is also defined as the average of squared differences from the mean of the data-set [66, 64]. The differences are squared because the substraction of a sample point and the mean may result in a negative value [64]. The larger the variance of a data-set, the higher is the data-variety or the spred of data in the data-set. [65, 66, 64].

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x - x_i)^2}{N} \tag{2.2}$$

The variance for the list L_1 will be comparatively higher than the lists, L_2 and L_3. The spread of data around the mean in lists L_2 and L_3 is notable low [64]. If a similar methodology is applied for the feedback data-set of a rider in the proposed model, the characteristic feedback set with the highest variance is the main tracked characteristic of the rider. Thus, the thesis employed the variance to track the main rider characteristics based on feedback data-sets.

2.5 Machine Learning Module Selection

In the implementation of Phase 1, the Closer matching of riders consisted of manually altering the characteristics of broadcasting riders like adding or subtracting by 1 and then researching the riders. For automating the task of manual alterations, the research included the study of Machine Learning algorithms and led to the finding of the Machine Learning Content-Based recommendation system [67]. In the ML-based recommendation system, the features are converted to vectors and represented in a d-dimensional space, where d is the number of features [67, 68]. The angular distance or the Cosine of the angle θ, which is between the vectors is calculated using the equation of the Dot Product [68, 69]. The recommendation system plots the vectors with the highest Cosine values closer to each other [67, 68, 69]. The thesis uses a similar methodology where the selected features are the registered rider characteristics and the UTT. The ML-based recommendation system plots the rider vectors with higher Cosine values in proximity, and riders closest to each other are selected and added on a trip.

An additional need for Machine Learning algorithms in the thesis is the prediction of the two main characteristics of newly registering riders. There is room for a little error due to the presence of the imbalanced feedback data-sets in the proposed model [70]. In the case of an

imbalanced data-set, for similar inputs, different outputs may be recorded, creating uncertainties during predictions [71]. The research included a search for a Machine Learning classification algorithm or a classifier that could appropriately fit an imbalanced data-set and give quality predictions.

The search for a suitable Machine Learning classifier led to the training and testing of feedback data-sets with classifiers like Logistic Regression, K-Nearest Neighbours (KNN) classifier, Naive Bayes Multinomial classifier, Random Forests classifier, Neural Networks, and Support Vector Machine (SVM) [72, 73, 74, 75, 76]. Out of all tested classifiers, SVM turned out to be the most feasible because of the Radial Bias Function (RBF) Kernel [77].

For distinguishing classes, SVM uses the RBF kernel, which is a highly non-linear curve [76, 77, 78]. SVM works on the principle of placing the curve or the line to the closest data-point with maximum distance [76]. The regularization parameter, C, and the gamma parameter, γ dictates the shape and the placement of curve [77]. The process of governing the placement of the curve by manipulating the values of C and γ is called Kernelization [77, 78, 79]. Kernelization allows maximum fitting of data-points of a class, which may also include fitting fewer data-points from other classes. Hence, SVM considers a small error by the maximum fitting of data and works best for imbalanced data-sets. [70, 77, 79]. In the end, the classifier selected in the thesis is the SVM for predicting the main characteristics of riders.

CHAPTER 3

SYSTEM MODEL

The system model in the thesis reflects the proposed model framework and the purposes of the Enhanced Ride Sharing Model. The chapter begins by specifying the problem statement, which outlines the need for the designed model. The chapter concludes by describing the system architecture, which focuses on several components utilized in orchestrating the matching layers and the rider recommendation module.

3.1 Problem Statement

The increased number of vehicles has led to significant problems in the transportation domain like Global Warming, traffic congestion, and rapid consumption of fuel [23, 25, 26]. Along with humans, the stated issues also affect other living beings on our planet [80]. In such cases, the concept of Ride Sharing provides optimal solutions, and currently, many existing Ride Sharing applications tackle and solve the aforementioned problems. After an in-depth inspection of several applications and research papers based on the existing Ride Sharing models, the investigation concluded that the primary issue lies in the matching of riders, unexpected rider additions, vehicle traversing approach, and the overall time management in the completion of a trip [32, 39, 57]. Hence, even though there exist many Ride Sharing applications, Ride Sharing is not employed to its full potential.

The thesis presents a Ride Sharing platform that focuses on encouraging the services of Ride Sharing. The proposed solutions provide outcomes like higher rider matching rates and minimal time expenditure for trip formation plus trip completion. The system also provides the trip's metadata to all users to loosen the influence of social barriers among riders. Also, the model reaches user traversing expectations by creating an optimized path using the Multiple-Sources-Multiple-Destinations (MSMD) approach, which is an excellent choice for Car-Pooling to complete a trip with appropriate time management. The main idea of the designed model is to match riders based on human characteristics and the User Threshold Time (UTT). Riders having similar characteristics are grouped considering the minimal restricted

traveling time on a trip. The proposed model also uses Machine Learning algorithms to predict better rider recommendations plus to tune up the system efficiency. The thesis majorly focuses on the expansion of the Ride Sharing services, which will indirectly result in improving the weather conditions and preserve fuel resources.

3.2 Architecture

The architecture in the thesis resembles the blueprint of the designed Ride Sharing model. The thesis includes two design phases, and therefore, the chapter of the system model presents two distinct architectures for Phase 1 and Phase 2.

3.2.1 Architecture, Phase 1

Phase 1 provides the first design for the Enhanced Ride Sharing Model. The execution of Phase 1 commences with associating a driver on a trip. The driver allocation is followed by finding and filtering the riders based on rider characteristics and User Threshold Time. Figure 6 reflects the architecture of the implemented matching model in Phase 1.

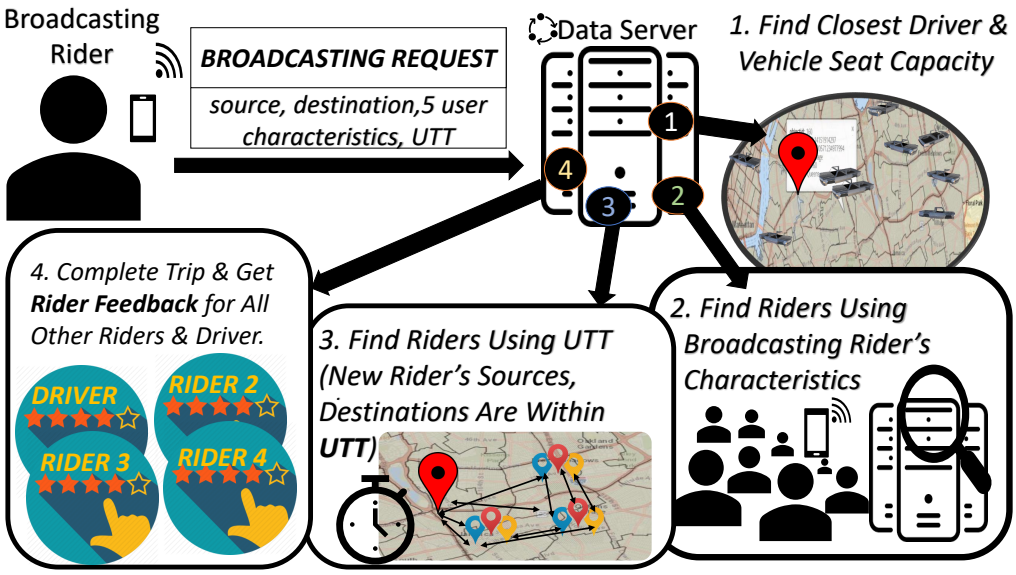

Figure 6: System Architecture, Phase 1.
The system architecture for Phase 1 illustrated the initial design for the model implementation. The entities in the architecture are the broadcasting rider, closest driver, rider matching layers, and the feedback system.

While researching the NYC Cab service, the study led to the finding of the NYC Cab

location data repository. The repository includes real-time NYC taxi zone locations and is publicly available [81]. The New York City Cab Department has divided New York City into small 265 areas, also referred to as the zones. Figure 7 gives an idea about the zones in New York City and showcases an example of the zone "Ridgewood." Each zone possesses almost 1000 locations in the form of latitudes and longitudes [81]. For accurate measurement of the system efficiency, the Ride Sharing model in the thesis made use of the NYC Cab location directory while performing model experimentations.

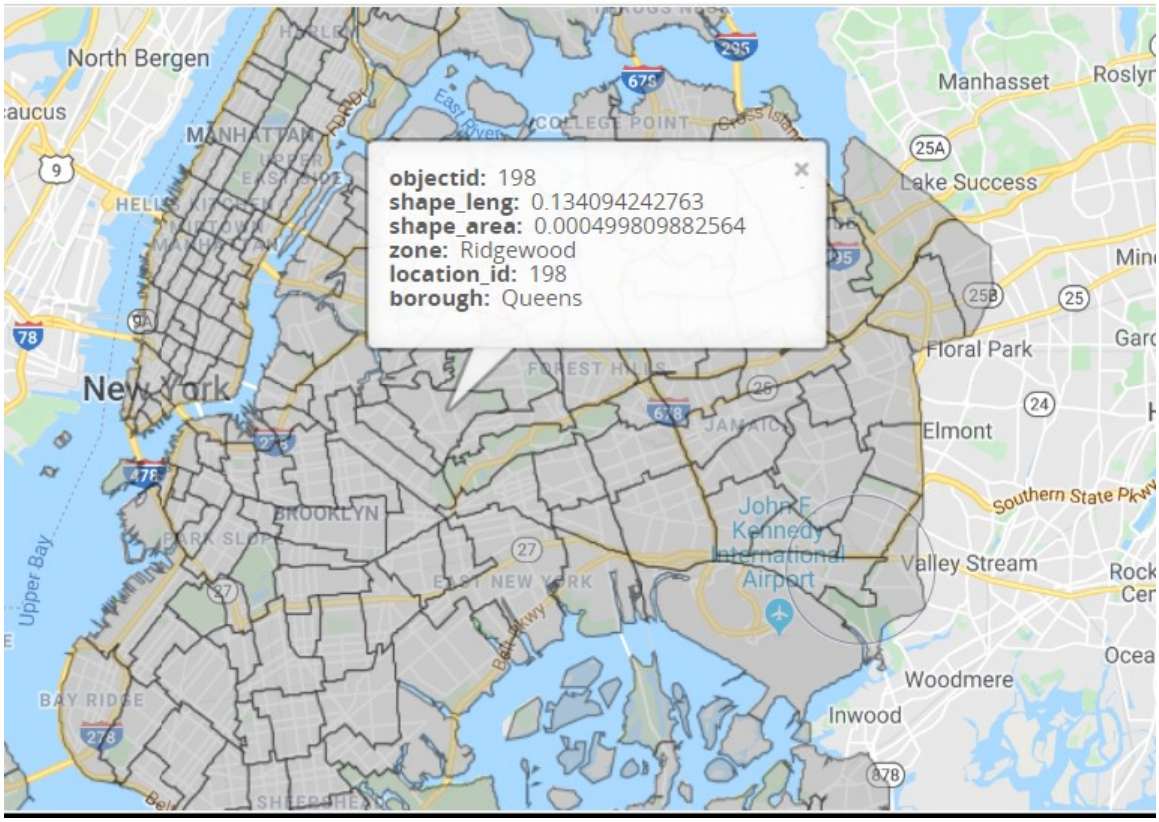

Figure 7: New York City Cab Zones.
The taxi drivers in the New York City Cab service traverse through smaller segregated areas in New York City called the zones. An object-id or a location-id uniquely identifies each zone. Also, each zone has a zone name and a borough name. The selected zone in Figure 7 has the object-id "198," the zone name "Ridgewood" and the borough name "Queens." The zone data by NYC Cab Department is an openly available location data repository for development purposes [81].

Throughout the implementation, the system maintains a client-server environment. In a client-server setting, a client device sends a request with the user data to a server. The server processes the client's request and sends the response data back to the client device. The system

then renders the received data from the server on the client device. In Phase 1, initially, a user broadcasts a rider request that holds the broadcasting rider's user-id, source, and destination. Based on the contents in the request, the data server retrieves the characteristics plus UTT and creates a data document, called the trip document, which includes the request data of the broadcasting rider.

The database on the server-side includes an active repository of all drivers. When drivers are active or awaiting broadcasting rider requests, the system keeps updating the location and status of the vehicles for quicker driver allotment to incoming requests. Additionally, the system notes the source zone from the trip document. The noted source zone indicates from which source zone the broadcasting request has originated. All the available drivers from the noted source zone are retrieved, and the closest available driver to the user's source location is selected. The model adds the selected driver to the trip document, and the activity of the driver association completes the first step of the trip.

Furthermore, the system sends the source zone as a parameter to the rider matching functions. The first function is the characteristics matching function, which includes the Exact, Closer, and Alternative types of characteristics matching. The function searches and retrieves all the active and broadcasting riders from the same source zone and gets a rider list based on the three characteristics matching types. The accepted passengers are further sent to the second function to perform a UTT check.

The second function is the UTT filtering function that computes the traveling time from every accepted rider's location to the broadcasting rider's location. If the traveling time is less than trip UTT, the rider is accepted. The characteristics and UTT functions continue the matching and filtering of riders until the number of accepted riders reaches the seating capacity of the vehicle or until there are no riders left in the characteristics matching accepted rider list. If the pool is incomplete or if the maximum number of seats in the car is not occupied, the model searches for active and broadcasting riders in other zones. Riders found in other zones undergo the same procedure of the characteristics and UTT matching.

After concluding the rider search, the system executes trip completion and records rider feedback. The simulation of trip completion consists of adding the time required to traverse

between all rider locations. The feedback in Phase 1 is the event where a user provides a single-digit rating to other users on a scale of 1 to 5. The single-digit feedback module forms the last step in the architecture of Phase 1.

3.2.2 Architecture, Phase 2

The characteristics and UTT functions provided successful results that reached the expectations of Phase 1. However, the simulations in Phase 1 incurred a significant time lag while running the characteristics matching function. After a minor inspection of the function, the investigation revealed that the time lag is due to the presence of numerous conditional statements in the Closer characteristics matching. For achieving optimized performance, it was necessary to eliminate the extra time consumed in the characteristics matching due to the several conditional loops. The model experienced significant programming changes that led to the creation of Phase 2 of the thesis. The changes in design did not imply changing the idea of characteristics matching but included updating the methodology for characteristics matching.

The second phase majorly focuses on the characteristics matching layer and the rider feedback systems. The improvisation in the architecture consists of elements like matching layers with recommendation systems, 1-minute threshold driver match, broadcasting rider requests with the feedback status, redesigned feedback system, computation of the main characteristics, and the Machine Learning classification model. Figure 8 reflects the system architecture for Phase 2 of the Ride Sharing model.

A new field added to the broadcasting request is the feedback status. The feedback status checks if there is the presence of historical data representing the user feedback pattern. The historical data of a rider consists of the rider feedback and the computed main characteristics assigned by the system based on past trips. If the user has performed trips before, the system gets the assigned main characteristics and prioritizes a search for other riders with similarly assigned main characteristics.

The next step is a similar step performed in the first phase, which is to find the closest available driver from the same source zone. In Phase 1, the system computed the traveling time for all available drivers and then selected the nearest driver. The driver search in Phase 1

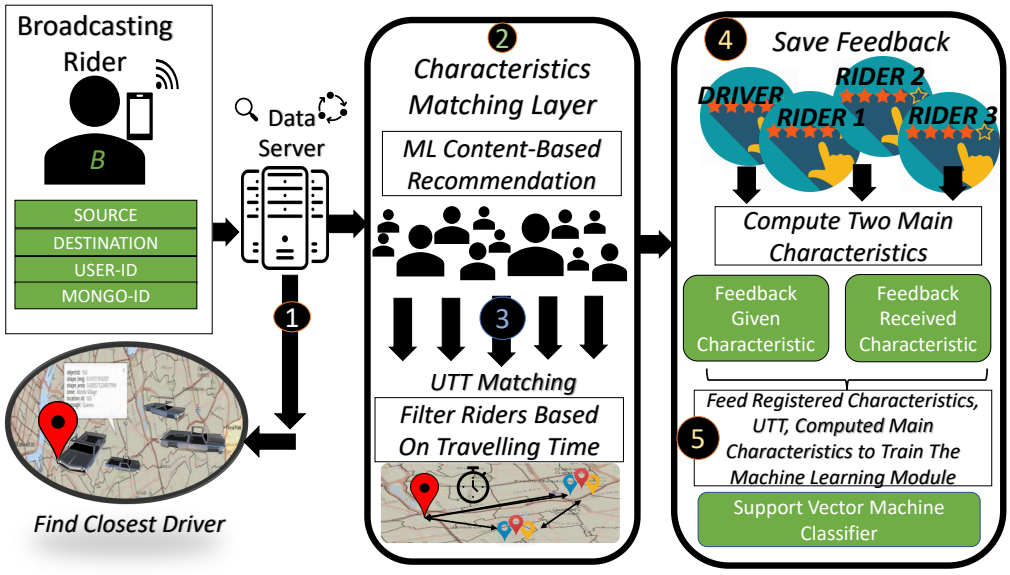

Figure 8: System Architecture, Phase 2.
Phase 2 architecture is the improvised version of the Phase 1 architecture. The new design incorporates the recommendation system in the characteristics matching layer. Also, the architecture includes the training of the Machine Learning module, which later predicts the two main characteristics for newly registering riders.

consumed much time as all drivers were initially searched and later compared based on computed traveling time. A solution implemented in such a case is the 1-minute driver threshold strategy, which is stopping the driver search if the system finds a driver at a location that is within a traveling time of 1 minute. Hence, there are limited iterations with the 1-minute threshold driver strategy. The allocation of the driver to a trip marks the completion of Step 1.

After the system associates a driver to a trip, the execution of the characteristics matching layers begins. In step 2, the system fetches all the broadcasting riders based on the Exact, Closer, and Alternative matching types. The enhancement in Step 2 is that the proposed model uses the Machine Learning recommendation system in all three types of characteristics matching. The recommendation system eliminates the process of manually updating the characteristics. As there is no interference for updating a characteristic, the time consumed for the trip formation in Phase 2 is notably less as compared to the time consumed for the trip formation in Phase 1.

After getting a rider list from the characteristic matching layer, the system computes traveling time between the broadcasting and selected rider locations. The step of computing plus

checking the traveling time is the UTT matching layer and is Step 3 of the architecture. Riders are added to the final trip itinerary if they satisfy the conditions in Step 2 and Step 3. The system continues the running of matching layers until the accepted riders fill up the seats of the selected driver's vehicle, or no more riders are left to traverse in the accepted rider list. The trip's pool completion status is labeled "Yes" if riders with driver occupy at least $n_{seats} - 1$ seats, where n_{seats} is the total number of seats in the vehicle. The pool completion status is labeled "No" if the riders and driver do not reach the vehicle seating capacity.

Step 4 is saving the feedback and computing the two main characteristics for every rider and driver. Also, the design in Phase 2 included a significant change in the feedback rating approach of Phase 1. In the new approach, a user rates the five characteristics of riders instead of providing a single-digit rating. Such an approach assists in tracking the user's most favored characteristics. Through the newly designed feedback module, it is possible to get the characteristics a rider expects in other riders while commuting on a trip. If the system groups riders having similar expectations based on the rider feedback patterns, the thesis achieves the ability to promote social journeys for maximum trips.

After recording the feedback by users, the system segregates the feedback records and enters the data into two distinct data-sets. The first data-set comprises the feedback data a user provides to other users, while the second data-set comprises the feedback data a user receives from other users. The Machine Learning classifier uses both data-sets to predict the main characteristics for every user in the system. Therefore, Step 5 is the training and testing of the Machine Learning classifier. The need for Machine Learning in the thesis is to predict the main characteristics of the newly registering riders. The step of predicting the characteristics by the Machine Learning classifier forms the final stage of Phase 2 architecture.

CHAPTER 4

METHODOLOGIES

The chapter of methodologies presents the approaches utilized to construct the proposed model. The current chapter specifically focuses on the implementations of the elements that constitute the system architecture. The chapter exhibits an in-depth visualization of the system components in the form of six prime sections: (i) The Broadcasting Rider Request (ii) The Search for the Closest Driver (iii) Searching Riders by Characteristics Matching (iv) Filtering Riders through UTT Matching (v) Saving User Feedback and (vi) The Final Trip Document.

4.1 The Broadcasting Rider Request

Figure 9: Structure of a Broadcasting Request.
A broadcasting request consists of a source, destination, source zone, destination zone, broadcasting rider's user-id, and mongo-id. The request gets updated on the server-side with the registered characteristics and UTT. A timestamp is added at the end to indicate the start time of the trip.

The program execution begins with a client device like a cell phone or a computer broadcasting a request to the data server. The most important part of the broadcasting request is the user-id. The server fetches the registered rider characteristics and UTT from the rider

registration records using the user-id. The next step is one of the vital steps in the Ride Sharing model, which is the creation of the trip document. Figure 9 showcases the structure and elements of a broadcasting request, which is a part of the trip document.

The trip document logs every essential trip detail or any minor trip updates. If the trip document is inspected deeply, to the presence of the user-id, there is also a mongo-id. The database, MongoDB, creates a new and unique mongo-id for every user, which is a 12-bit binary JSON string during the user registration. The reading of the mongo-id is complicated and needs conversion to a simple string format for later data handling purposes. Hence, the system generates a user-id for every rider in registration. User-id is a unique identification number that is easily readable and serves better for data handling tasks like data additions and alterations. The broadcasting rider's characteristics and UTT are referenced as the trip characteristics and trip UTT because, throughout the trip, the model refers to the broadcasting rider's characteristics and UTT present in the trip document while searching a rider or a driver.

4.2 The Search for the Closest Driver

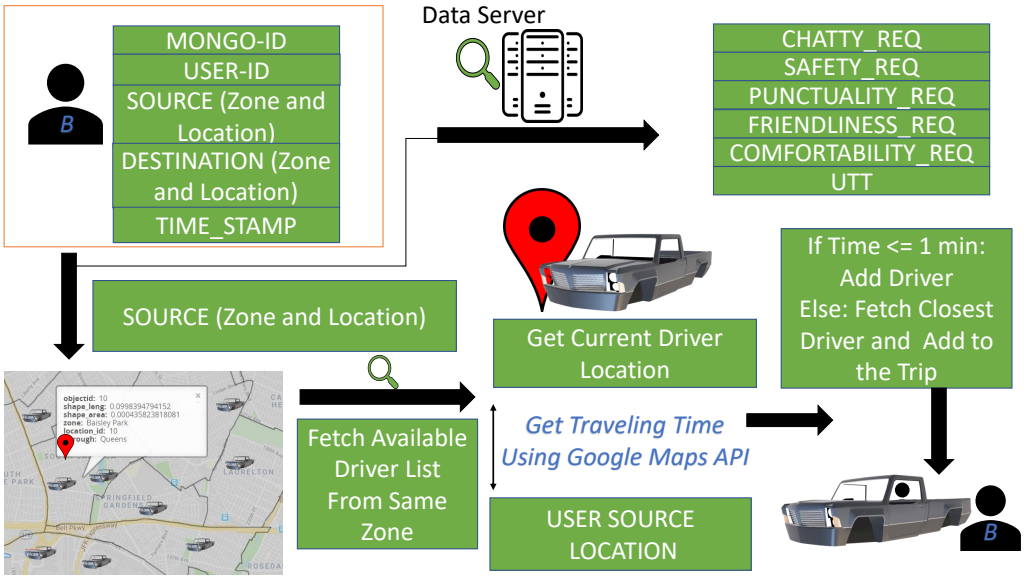

Figure 10: Adding Driver to a Trip.
Adding a driver starts by extracting the source zone from the trip document. Based on the source zone, all the available and active drivers are retrieved. The driver closest to the broadcasting rider in terms of traveling time is selected and added on the trip.

The system keeps a driver's status as available until the driver is active, and the riders have not reached the seating capacity of the vehicle. At first, the system gets the source zone from the recently created trip document and retrieves all the available drivers using the source zone as the parameter. After getting the driver list, the system records every driver's current location. The next crucial step is the computation of traveling time between the broadcasting rider's location and the logged driver's location.

The traveling time including real-time traffic is computed using the Google Maps Distance Matrix API. The calculated timings are compared to find the lowest one, and the driver with the shortest traveling time is selected and added on the trip. An essential step in the driver search is noting the vehicle seating capacity. Figure 10 represents the complete driver search module using the Google Maps API.

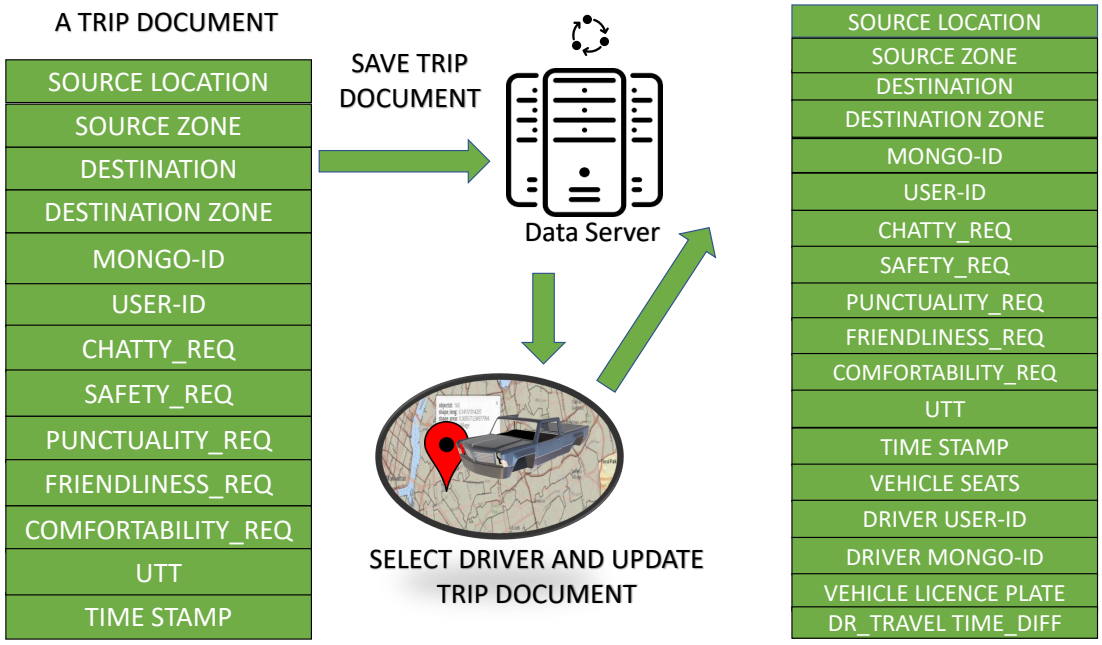

Figure 11: Adding Driver Details To Trip Document.
The system adds the driver's identification details to the trip document. Along with the driver details, the system updates the trip document with vehicle details, which includes the vehicle seating capacity, license plate number, and the traveling time difference, denoted by DR_TRAVEL TIME_DIFF. The traveling time difference is the computed traveling time between the broadcasting rider location and the selected driver location.

With the improvised driver search algorithm in Phase 2, driver allotment is more agile. The system stops the search for a driver if it finds a driver at a traveling distance of one minute.

Selecting and adding a driver to a trip implies updating the trip document with driver details like driver's mongo-id, user-id, the vehicle's license plate number, and the computed traveling time between the broadcasting rider and the selected driver. Figure 11 showcases the updated trip document after adding a driver to a trip.

4.3 Searching Riders with Characteristics Matching

A complete rider search module with the three types of characteristics matching is stated in Figure 12. The rider search commences with Exact characteristics matching. In Exact matching, the model searches for riders from the same zone with identical characteristics to that of the broadcasting rider's characteristics. The model adds the found riders in a list and sends the rider list for UTT matching.

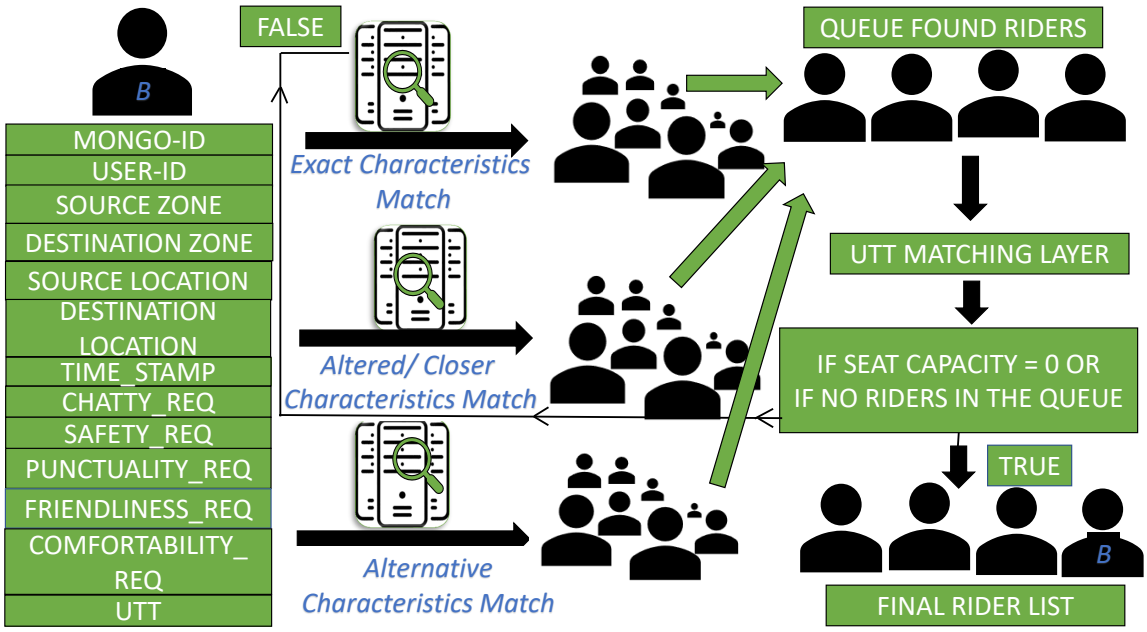

Figure 12: A Generic View of the Characteristics Matching Layer.
Figure 12 is one of the most important designs of the Enhanced Ride Sharing Model. The figure provides an in-depth visual of the three types of matching, Exact, Closer, and Alternative matching. After every search, the system sends the rider list to the UTT matching layer. In the end, the system checks if the pool is complete or incomplete. If the pool is incomplete, the search for riders begins in other zones until the trip completes the pool, or the accepted rider list is empty.

If the riders do not complete the pool, the system commences the Closer characteristics matching. In Closer matching, especially in Phase 1, the characteristics were manually altered. In

future stages, an iterative conditional program replaced the manual alterations of the characteristics. Even though the task was automated, the execution still consumed significant time. The root cause of the delay was the presence of numerous conditional executions. Figure 13 provides an idea of how the program altered the broadcasting rider's characteristics.

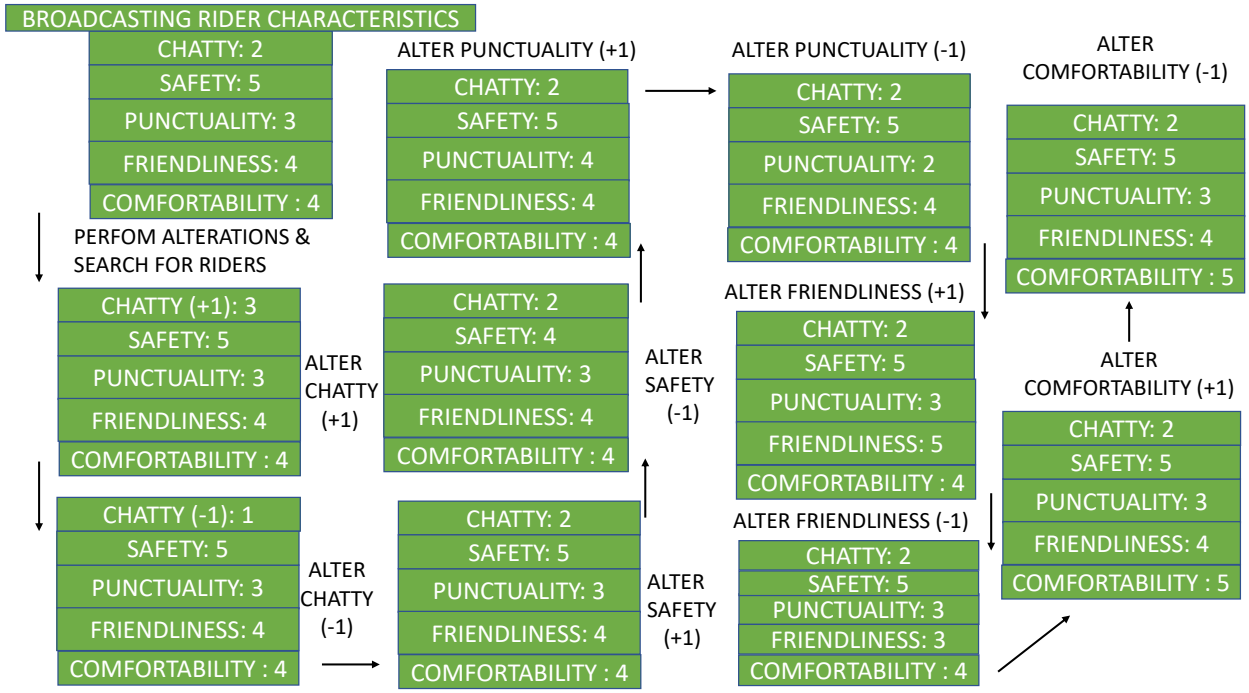

Figure 13: Closer Matching, Phase 1.
The Closer matching in the phase had a large number of loops. It served the purpose of completing the pool but incurred a drag on the system performance. For alterations, the system altered a specific characteristic by the value 1 and then researched the riders. In Figure 13, the system initially alters broadcasting rider's every characteristic by adding 1 and then by subtracting 1.

As stated in the chapter of the system model, a significant design change led to the replacement of the entire matching function with the Machine Learning recommendation system. The ML-based system facilitates computing a match between the broadcasting rider characteristics and countless possible combinations of other rider characteristics. The next chapter of matching layers with Machine Learning presents a detailed mathematical explanation of the Content-Based recommendation system. After the rider acceptance in Closer matching, the system adds all riders in a queue and sends them for the UTT matching.

Furthermore, if the seats of the vehicle remain unfilled, the model employs the last type of

matching. In the Alternative type of rider matching, the system searches for riders irrespective of the characteristics. The approach is a similar rider selection approach employed by companies like UberPool and LyftLine. Riders are then added in a queue and sent for the UTT matching.

4.4 Filtering Riders through UTT Matching

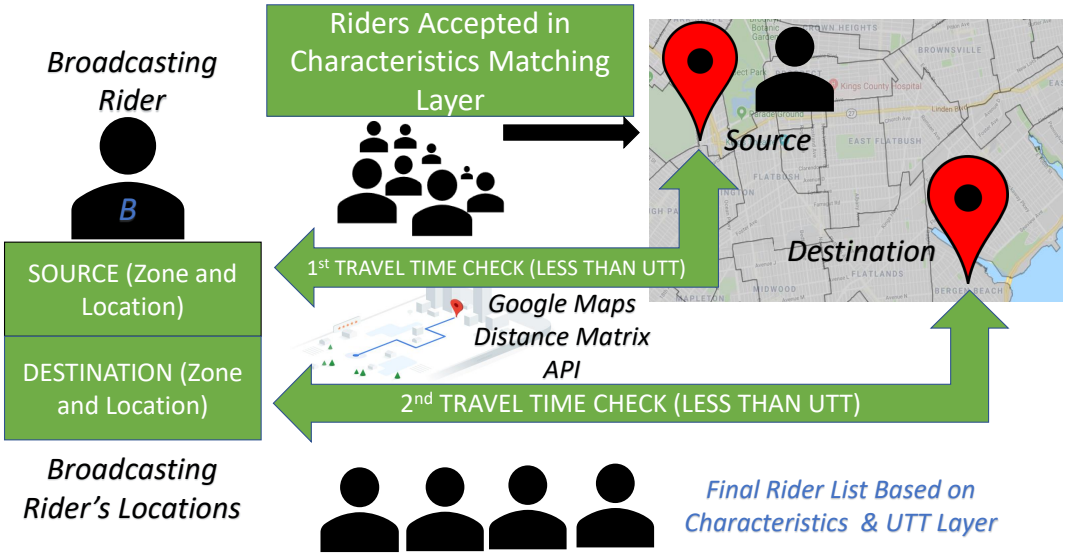

Figure 14: User Threshold Time (UTT) Matching Layer.
In UTT matching, the model initially records the rider locations and computes traveling time between broadcasting rider 'B' location and found riders' locations. If the traveling time is less than the trip UTT, the model adds riders into a final queue. Through UTT matching, the model pairs only the riders who do not exceed the registered restricted time or the UTT and completes the journey in minimal time.

The next vital action in the rider matching is sending the accepted riders from the characteristics matching layer to the UTT matching layer. At first, for all riders, the system calculates the traveling time between broadcasting and accepted rider's source locations using the Google Maps Distance Matrix API. The model then performs the first UTT check, which is to verify if the traveling time is equal or less than the trip UTT. If the rider satisfies the first UTT check, the model accepts the rider and sends the rider to a second UTT check. The model performs the second check, which is to verify if the traveling time between broadcasting and rider's destination locations is equal or less than the trip UTT. If the rider satisfies both UTT checks, the system adds the rider into a final itinerary. The process of UTT matching continues until the

riders reach the seating capacity of the vehicle, or the model does not find any more riders in the accepted rider queue. Figure 14 illustrates the two UTT checks in the UTT matching layer.

The thesis maintains a threshold of two minutes for the trip formation, which assures that the time for trip formations is not high. If the rider list is exhausted for the same zone, the system extends the rider search with characteristics and UTT matching to other zones. Also, if a rider gets rejected from one trip, the system redirects the rejected rider request to other ongoing trips. The rejected rider request may also be sent to an active and available driver to commence a new trip. Using the maximum rider check strategy from multiple zones allows the vehicle seats in a car to get occupied entirely.

4.5 Saving User Feedback

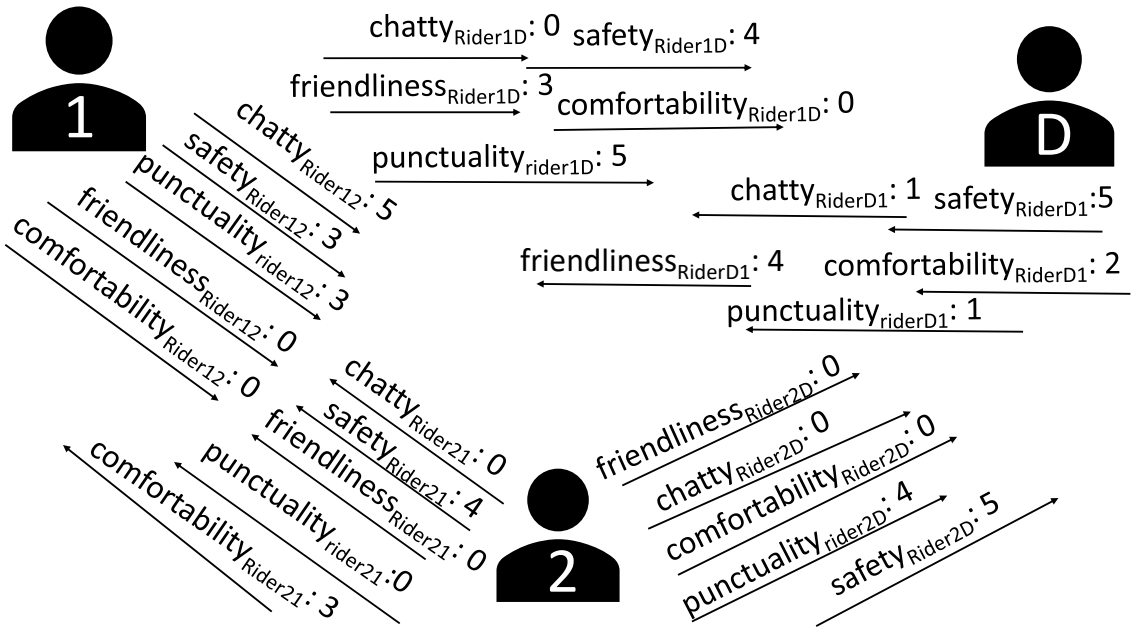

Figure 15: An Use Case of Phase 2 Feedback System.
Figure 15 illustrates the recrafted feedback system with riders $Rider_1$ and $Rider_2$ and the driver D. The arrows provide the direction of rating, and the label on the arrows specifies the characteristic the user is rating. In the given example, $Rider_1$ provides a rating of 3 to the punctuality characteristic of $Rider_2$. Another example is of the safety rating of 5, provided by the driver D to $Rider_1$.

The architecture in Phase 2 replaced the single-digit rating model with the five characteristics rating model to track the user characteristics. A rider provides ratings to other

riders in terms of five characteristics. The five characteristics are the same characteristics present at the time of rider registration. Figure 15 provides an example of ratings given by two riders and a driver on a trip. Equation 4.1 represents the rating user rates to other users for a specific characteristic.

$$characteristic_{ab} = rating\_value \tag{4.1}$$

In Equation 4.1, a represents the user rating other users, and b represents the user getting rated. If a user submits a feedback without rating a specific characteristic, the system assigns the value 0 to the charactetistic the user has not rated. The following is an example of the feedback given by $Rider_1$ to the Driver D.

$$chatty_{1D} = 0$$

$$safety_{1D} = 4$$

$$punctuality_{1D} = 5$$

$$friendliness_{1D} = 3$$

$$comfortability_{1D} = 0$$

The system adds the feedback into a feedback data-set after each user submits the ratings for other users. The feedback data-set is later used for the computation of the main characteristics.

4.6 Final Trip Document

From broadcasting of requests till the ending of the trip, the trip document keeps collecting data from several entities like the broadcasting rider requests, the allocated driver, and the accepted riders. At each stage, each part of the trip contributes to the creation of the final trip document. For example, from the broadcasting rider, the starting and ending location of the trip is saved, and based on the rider locations, an optimized path is created, which provides the total trip time. In the end, the model possesses a massive block of trip data, and the system saves

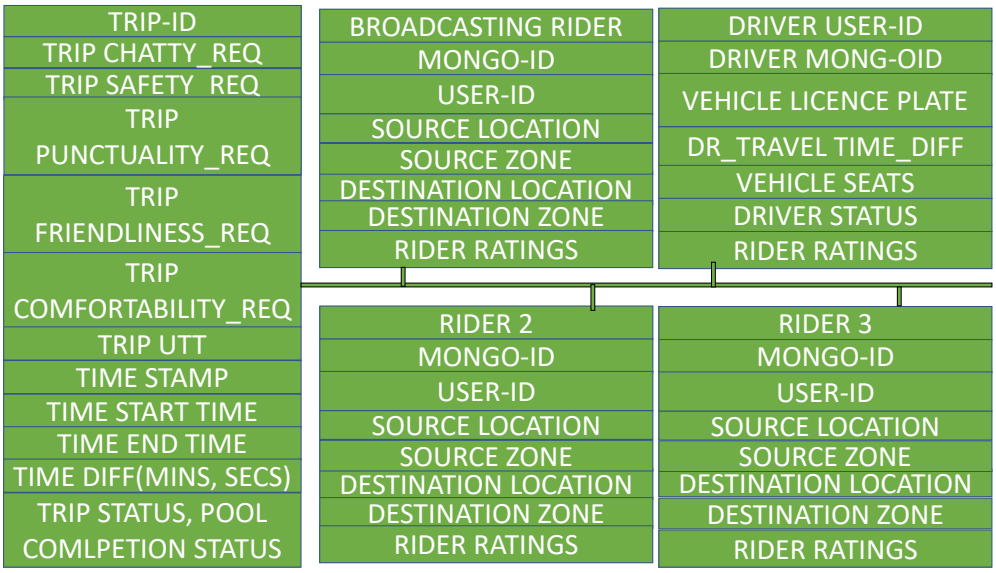

Figure 16: The Final Trip Document.
The final trip document is the terminal stage of the proposed solution. The trip document has basic trip details like the unique trip-id, the trip characteristics, trip UTT, and time consumed for trip completion in minutes and seconds. Besides noting the basic trip elements, the trip document also comprises the broadcasting rider request, selected driver details, and other distinct rider documents.

the trip document in a distinct trip data-set for future data management. As shown in Figure 16, every trip document is assigned a unique trip-id. The reason for the creation of the unique trip-ids is for future maintenance and support. If there are customer complaints on a trip, the complaints can be tracked using the trip-id.

The final section of the proposed solution concludes with a description of the final trip document. In the trip document, the trip characteristics and the trip UTT are the broadcasting rider's characteristics and UTT. Every rider data is itself a small document and contains information like mongo-id, user-id, source and destinations, and a copy of ratings given by the corresponding user to other users. Additionally, the trip document also contains an overall time taken for the completion of the journey. The time difference is the difference in time from the point the first rider broadcasts the request for a trip until the point where the driver acknowledges, "trip ended" as the trip status. The final trip document marks the completion of the entire trip and forms the final step of the proposed model.

CHAPTER 5

MATCHING LAYERS WITH MACHINE LEARNING MODULES

The chapter of matching layers with Machine Learning is a part of the proposed model, but the module of Machine Learning itself spans numerous details. Thus, it was essential to provide the research and contributions of the Machine Learning modules in a distinct chapter. The current chapter starts by discussing how the recommendation system improves the quality of matching. Later, the chapter describes the methodologies for computing the main characteristics of the rider and presents an in-depth discussion on the selected Machine Learning classifier for predicting the main characteristics. The chapter concludes with the simulations performed for testing the system efficiency.

5.1 Recommendation System With Characteristics Matching Layers

In the Exact matching type, the system searches for riders with exactly matching characteristics. The odds of finding an Exact match in the same zone are low because of the scenario where two or more broadcasting riders having exactly the same characteristics start around the same time and reach the same or nearby sources and destinations. Hence, the number of matches in Exact characteristics matching is notably low. Alternatively, the chances of finding a rider with little different characteristics and heading on the same trajectory are high. Hence, the number of riders accepted is largest by the Closer characteristics matching type. The alteration of characteristics in the Closer matching involved a large number of iterations in Phase 1. The thesis employed the concept of Machine Learning Content-Based recommendation system to reduce the higher number of loops in Phase 2. Initially, the system converts the characteristics of every $rider_a$ to a vector, $char\_v_a$ as shown in Equation 5.1.

$$char\_v_a = [chatty_a, safety_a, punctuality_a, friendliness_a, comfortability_a] \tag{5.1}$$

For example, let the registered characteristics of a broadcasting rider, $Rider_{br}$, be as follows:

$$chatty_{br} = 3$$

$$safety_{br} = 4$$

$$punctuality_{br} = 3$$

$$friendliness_{br} = 3$$

$$comfortability_{br} = 4$$

The vector representation $char\_v_{br}$ for the broadcasting rider is given by Equation 5.2.

$$char\_v_{br} = [3, 4, 3, 3, 4] \tag{5.2}$$

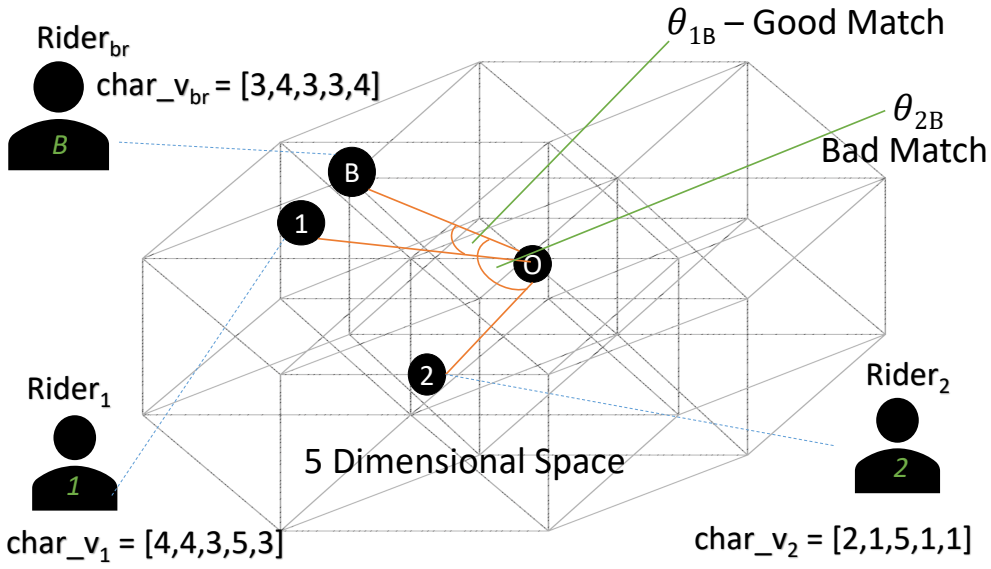

Figure 17: Rider Matching Using Content-Based Recommendation.
The methodology for the three types of matching is updated in Phase 2 using the Machine Learning Content-Based recommendation system. Each rider characteristic represents a vector in a d-dimensional space where d is the number of features. The measure for the match is computed based on the angular distance between two vectors. For a smaller angle, the match is higher, and riders are paired up for a higher matching value.

Consider a case of two riders, $Rider_x$ and $Rider_y$. The vector representing $Rider_x$ is given by $char\_v_x$ and the vector representing $Rider_y$ is given by $char\_v_y$. For measuring the level of

match between $Rider_x$ and $Rider_y$, the angular distance between two vectors or the Cosine of angle θ_{xy} is calculated using the Equation 5.3. The Cosine of θ_{xy} is equal to the Dot Product of vector divided by the product of the vector magnitude.

$$cos\theta_{xy} = \frac{(char\_v_x \cdot char\_v_y)}{\|char\_v_x\| \|char\_v_y\|} \tag{5.3}$$

If θ_{xy} is equal to 0, the value of $cos\theta_{xy}$ is 1. The angular distance is 0 when the system finds a rider with exactly matching characteristics. If there are other riders with identical characteristics as that of the broadcasting rider, it is a 100% match. Such a case is matching the riders through the Exact matching type. Hence, a greater Cosine value results in a higher match.

Figure 17 illustrates an example of the matching of riders using the ML-based recommendation system. The figure represents the vectorized characteristics in 5-dimensional space or a 5-dimensional hypercube. The reason for selecting a 5-dimensional space is because the number of dimensions is equal to the number of selected features. In the figure, 'O' represents the origin, and $Rider_{br}$ represents the broadcasting rider. Additionally, $Rider_1$ and $Rider_2$ represent the riders to be matched with the broadcasting rider $Rider_{br}$. Points B, 1, and 2 in the hypercube represent the points plotted by the vectors for $Rider_{br}$, $Rider_1$, and $Rider_2$.

By observing the visuals in Figure 17, $char\_v_1$ seems to be a better match than $char\_v_2$. The match is higher for $char\_v_1$ due to a smaller angle, θ_{1B} as compared to the angle θ_{2B}, which has a larger stretch than angle θ_{1B}. The higher match implies $Rider_1$ is a better match than $Rider_2$. Hence, $Rider_1$ is selected and added on the trip. The system redirects $Rider_2$ to other ongoing trips or any available and active drivers. In simulations, the system accepts riders only if a rider match results greater than 85% or only if the computed $cos\theta_{xy}$ value is above 0.85.

The benefit of the recommendation system is that it can be utilized to compute the angular match between any two characteristic vectors irrespective of the characteristic's matching type. Thus, using the ML-based recommendation model led to the elimination of altering the rider characteristics in Closer matching. The elimination resulted in cutting down a large number of conditional loops that profoundly affected the system performance in terms of time complexity.

It is now possible to get a match between broadcasting rider and a rider with any

combination of characteristic values. Therefore, the employment of the recommendation system is not only limited to Closer matching. The thesis also employs the recommendation system in the Exact and Alternative type of matching. A positive point to state is about the processing offered by Sklearn libraries. Sklearn libraries are the Machine Learning libraries and facilitate batch processing between multiple vectors. Batch processing is the process of computing the Cosine Similarity between multiple vectors at the same time. Using the feature of batch processing, the model computes the Cosine Similarity between any number of riders, which results in the elimination of massive computations and time-consuming processes.

5.2 Computation of the Main Characteristics

The computation of main characteristics initiates after saving the rider feedback and dividing the feedback data into two parts. The first main characteristic is the Feedback-Given-Characteristic and uses the first part of the feedback data, which includes the ratings given by each rider to other riders. The purpose of the first main characteristic is to track the characteristic the rider most focuses while giving feedback to other riders. The computation of the first main characteristic is discussed with an example of the feedback given by $Rider_1$ to $Rider_2$, $Rider_3$, and $Rider_4$, as shown in Table 1.

Riders	**Chatty**	**Safety**	**Punctuality**	**Friendliness**	**Comfortability**
$Rider_2$	0	2	1	4	0
$Rider_3$	0	3	0	4	0
$Rider_4$	1	5	0	4	0

Table 1: Feedback Given by $Rider_1$ to Other Riders.
Table 1 provides the feedback given by $Rider_1$ to $Rider_2$, $Rider_3$, and $Rider_4$. A rider rates the five characteristics while rating the other riders. A rating of 0 specifies that the rider did not rate the characteristic due to which the system assigned the value 0.

The next step is to segregate and append the feedback data based on every characteristic of the rider. The system creates the following lists based on the feedback given by $Rider_1$.

$$chatty_{Rider1} = [0, 0, 1]$$

$$safety_{Rider1} = [2, 3, 5]$$

$$punctuality_{Rider1} = [1, 0, 0]$$

$$friendliness_{Rider1} = [4, 4, 4]$$

$$comfortability_{Rider1} = [0, 0, 0]$$

The observation made from the five lists is that $Rider_1$ may continue to give a friendliness rating of 4 in future trips. Another observation is that the rider has submitted the feedback without rating the comfortability characteristic, and therefore, the system assigned the value 0 to the comfortability rating. The only data variety observed is in the safety rating. The characteristic with the highest data variety is the Feedback-Given-Characteristic. The first main characteristic of $Rider_1$ is the safety class. The system computes the first main characteristic using the equation of variance.

The created list from the given feedback data forms the sample sets for computing variance. The higher the spread of the data around the mean of a sample set, the higher is the characteristic variance. Total number of elements in a characteristic list is n_{char} or $data\_count_{char}$. x denotes a specific element from the characteristic sample set, and x_{char_i} denotes the mean of the characteristic sample set. The system calculates the squared differences using Equation 5.4.

$$x_{sqr\_diff} = (x - x_{char\_i})^2 \tag{5.4}$$

The selected characteristic variance is denoted by σ^2_{char} and is represented in Equation 5.5. The system selects the characteristic list with the highest variance, which implies the user is more diverse in rating the selected characteristic and therefore focuses on the selected characteristic. Besides noting the characteristic with the highest variance, the system also records the variance of other characteristics and saves in the feedback data-set for every user.

$$\sigma^2_{char} = \frac{\sum_{i=1}^{n_{char}} x_{sqr\_diff}}{data\_count_{char}} = \frac{\sum_{i=1}^{n_{char}} (x - x_{char\_i})^2}{data\_count_{char}} \tag{5.5}$$

After calculating the first main characteristic, the system proceeds to the computation of the second main characteristic or the Feedback-Received-Characteristic. The purpose of the

second main characteristic is to categorize users based on the feedback given by other users. For example, if 20 users have provided the highest rating to the chatty characteristic of $Rider_a$ on many trips, the best-observed characteristic in $Rider_a$ is chatty. If users are looking for a rider who enjoys conversations, the system will recommend the $Rider_a$ as the rider has the highest chatty rating. The methodology for the second main characteristic uses the second part of the feedback data-set, which is the feedback received by other users to a user.

Riders	**Chatty**	**Safety**	**Punctuality**	**Friendliness**	**Comfortability**
$Rider_2$	4*0.32	2*4.31	0*2.10	2*0.1	4*1.73
$Rider_3$	3*3.45	1*0.15	1*0.55	0*5.72	3*3.34
$Rider_4$	3*9.21	0*3.21	3*0.02	0*0.21	0*1.32
$\sum$ Total	39.26	8.77	0.61	0.2	16.92

Table 2: Feedback Given to $Rider_1$ by Other Riders.
While computing the second main characteristic, the system fetches every characteristic variance of all riders $Rider_i$, who are rating $Rider_1$. The resultant value is the product of given feedback and the respective $Rider_i$ characteristic variance.

Table 2 provides a use case of the feedback given to $Rider_1$ by $Rider_2$, $Rider_3$, $Rider_4$. Each element in the column has two values. The first value is the feedback given by $Rider_i$ for a specific characteristic, and the second value is the characteristic variance ($(\sigma_{i\_char})^2$) computed for the $Rider_i$ characteristics. Every time a rider provides feedback for a specific characteristic, the system multiplies the feedback value by their respective characteristic variance. To exemplify, $Rider_2$ variance for safety is 4.31, and the safety rating given by $Rider_2$ to $Rider_1$ is 2. The feedback to $Rider_1$ by $Rider_2$ for the safety characteristic is the product of variance and the value provided in the rating. The system computes the product of variance and rated value for every characteristic of every rider.

In the end, for every characteristic, all the multiplications are added and compared. The characteristic with the highest score is the Feedback-Received-Characteristic. In the same use case, the second main characteristic computed for $Rider_1$ is chatty, as the value of 39.26 is highest as compared to other characteristic values.

Based on the computed main characteristics, the system redefines the search criteria for every rider. Indeed, the scenario is a practical use-case where riders rate other riders based on their past experiences and provide a real-time idea of the characteristics a user possesses. The

main characteristics assist in promoting a better and real-time recommendation to the riders.

5.3 Machine Learning Model & Prediction

The Machine Learning module selected in the thesis is Support Vector Machine (SVM). The data-sets created for training the SVMs are the Feedback-Given-Characteristic Data-set and Feedback-Received-Characteristic Data-set. In both data-sets, the input fields are the registered user characteristics and the registered UTT. The outputs or the labels to be predicted are the computed main characteristics. Table 3 and Table 4 reflect the fields and sample rows of the created data-sets for the Machine Learning module.

Feedback-Given-Characteristic Data-set						
Class_Given	**Chatty**	**Safety**	**Punctuality**	**Friendliness**	**Comfortability**	**UTT**
Comfortability	3	5	4	1	4	20
Chatty	1	2	4	3	5	10

Table 3: Sample Rows in the Feedback-Given-Characteristic Data-Set.
Table 3 provides the sample rows in the Feedback-Given-Characteristic database. Each row comprises the first computed main characteristic, the rider's registered characteristics, and UTT. The first row in Table 3 signifies that for a rider with registered characteristics as chatty:3, safety:5, punctuality:4, friendliness:1, comfortability:4, and UTT:20 minutes, the computed Feedback-Given-Characteristic is the comfortability class.

Feedback-Received-Characteristic Data-Set						
Class_Received	**Chatty**	**Safety**	**Punctuality**	**Friendliness**	**Comfortability**	**UTT**
Punctuality	4	4	2	3	1	10
Safety	5	4	4	1	4	25

Table 4: Sample Rows in the Feedback-Received-Characteristic Data-Set.
Table 4 provides the sample rows in the Feedback-Received-Characteristic database. Every tuple contains the second main characteristic, the rider's registered characteristics, and UTT. The first row in Table 4 states that for a rider with registered characteristics chatty:4, safety:4, punctuality:2, friendliness:3, comfortability:1, and UTT:10 minutes, the computed Feedback-Received-Characteristic is the punctuality class.

The registered characteristics and UTT are the selected input fields or selected features for the SVM modules. The expected output is that the SVM should predict the main characteristics that match the computed main characteristics in the data-sets. For two data-sets, the thesis uses two distinct SVM modules. The function of the first SVM is to predict the Feedback-Given-Characteristic. Similarly, the function of the second SVM is to predict the

Feedback-Received-Characteristic. The working of the two SVM modules is reflected in Figure 18.

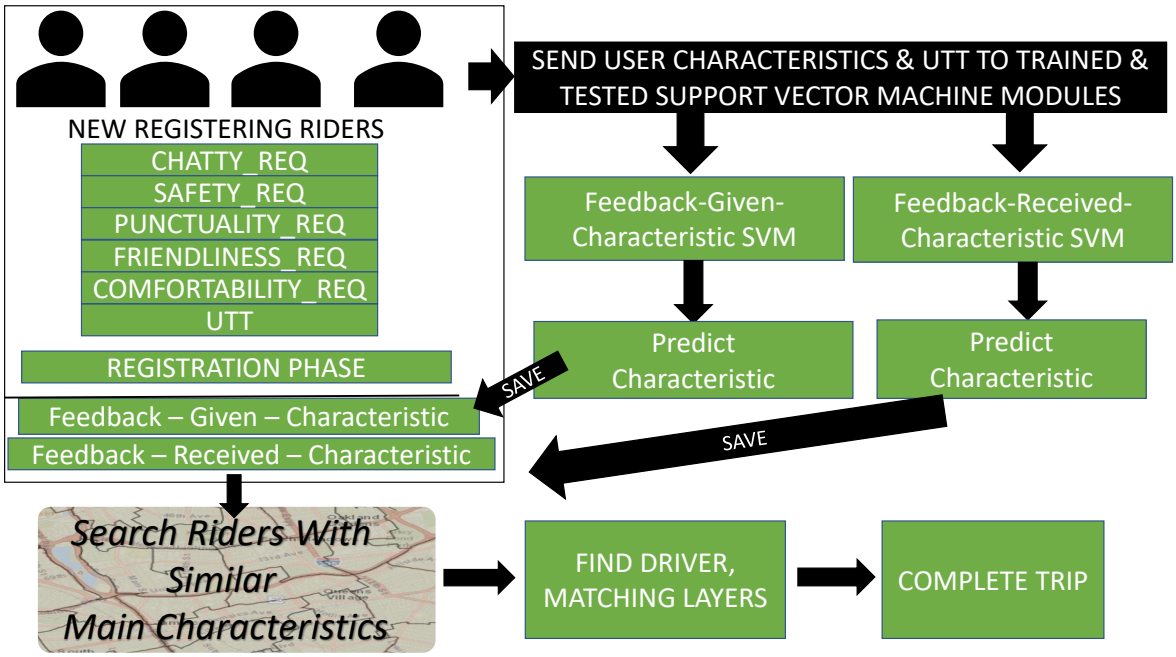

Figure 18: Working of Support Vector Machines in the Main Characteristics Prediction for Newly Registering Riders.
When new riders register to the system, the system notes the characteristics and UTT of the riders. The system then sends the characteristics and UTT to the SVM modules. Based on the trained and tested data-sets, the SVM modules predict the main characteristics. Based on the predicted main characteristics, the system recommends riders with similar main characteristics to the newly registered users.

The training and testing of SVMs are vital steps in the thesis. The module training has been completed using 27,000 records for both SVM classifiers. Through the data-sets, the SVM classifier learns that for a specific combination of the registered characteristics and UTT, the output is a specific main characteristic. The SVMs were tested using new 12,000 records. For the inputs in the testing data, initially, the system computed the respective first and second characteristics. For the same set of input data, the SVMs predicted main characteristics. The system then compared the predicted and computed values to check if the SVM is correctly predicting the main characteristics. The comparison is a part of evaluating the Machine Learning model accuracy, and the chapter of results provides a brief description of the SVM classifier evaluation.

After getting significant testing results, the thesis employed the trained and tested SVMs

to predict the main characteristics of newly registering riders. At first, the newly registered riders provide the characteristics and UTT in the registration phase. The system then sends the recorded characteristics and UTT to SVMs, which predict the main characteristics of the newly registered riders.

In the experimentations, the Machine Learning module was retrained using variance as an additional input feature. For an SVM, the higher the number of features, the higher is the accuracy of the module. The value of variance provides an extra edge for SVM to classify and plot the data points into labeled classes. With variance as an additional feature, the system achieved higher model accuracy.

5.4 Experimentations

A simulation is denoted by Equation 5.6 where U_i denotes the User Threshold Time, RC_i denotes the number of riders traversed, and S_i denotes a specific simulation event for the i^{th} simulation.

$$S_i = \{U_i, RC_i\} \tag{5.6}$$

At the beginning of every simulation, the trip starts by selecting a broadcasting rider from the rider records. The selected rider has a UTT equal to U_i. Consider the first simulation S_1. For the 1^{st} iteration, U_1 is taken as 10 minutes and RC_1 as 200. After selecting a broadcasting rider with a registered UTT of 10 minutes, the system begins the trip formation and the creation of the trip document. If the trip ends while traversing through the first 20 riders, the simulation continues by starting a new trip. The system again searches for a rider with a similar registered U_i and begins the trip. The process of rider traversing and trip completion continues until the RC_1 reaches 200. The following represents the first simulation.

$$S_1 = \{U_1, RC_1\} = \{10, 200\}$$

For every next simulation, the system keeps the similar value for U_i and increases the value of RC_i by 200 until it reaches 1000. Hence, the next simulation or S_2 is denoted by $S_2 = \{U_2, RC_2\} = \{10, 400\}$ and as the RC_i reaches 1000, the i^{th} simulation is

$S_5 = \{U_5, RC_5\} = \{10, 1000\}$. As the RC_i reaches 1000, the system resets RC_i to 200 and increases U_i by 5. The next simulation is denoted by $S_6 = \{U_6, RC_6\} = \{15, 200\}$ and is followed by simulations until $S_{10} = \{U_{10}, RC_{10}\} = \{15, 1000\}$. The system proceeds with the simulations till the U_i reaches 30. Indeed, the n^{th} or the last recorded simulation is given by S_n.

$$S_n = \{30, 1000\}$$

.

Variable	Description
U_i	Trip User Threshold Time for a simulation S_i
RC_i	Total number of riders traversed in a simulation S_i
RP_i	Total number of riders accepted in a simulation S_i
T_i	Total time consumed for completion of a simulation S_i
$trip\_count_i$	Total number of trips computed in a simulation S_i
MR_i	Matching rate of a simulation S_i
$closer_i$	Count of riders accepted through the Exact and Closer matching types in a simulation S_i
$alternative_i$	Count of riders accepted through the Alternative matching type in a simulation S_i
$match_{closer}$	Total count of riders accepted through the Exact and Closer matching types in all simulations
$match_{alternative}$	Total count of riders accepted through the Alternative matching type in all simulations

Table 5: Variables Responsible for Data Tracking in a Simulation.
Table 5 lists the variables that note the crucial changes or updates in a simulation. Every variable has a distinct significance and contributes crucially while evaluating the overall system efficiency.

For every simulation S_i, the system notes RP_i, the total number of riders accepted, T_i, the total time required for completing the simulation, and $trip\_count_i$, the total number of trips computed. Table 5 mentions the description of each variable which tracks the significant updates in every simulation. In some cases, the results stated that RP_i has a smaller value than RP_{i+1}. The expected result is that RP_i or the number of accepted riders should keep increasing with every progressing simulation event. The unexpected increase or decrease in RP_i was a randomness factor introduced due to uneven acceptance of the riders at the UTT matching layer. The system performed the same simulation without the Machine Learning model for ten times,

and with the Machine Learning module for five times to reduce the randomness factor. Performing the simulations several times reduced the randomness element, which led to the accurate measurements of the system efficiency.

An important measure in the system efficiency is the matching rate. Equation 5.7 defines the equation to compute the matching rate for a simulation. The matching rate is the division of accepted riders and the total number of traversed riders. According to the expectation of the thesis, the matching rate should keep increasing for consecutive simulations.

$$MR_i = \frac{RP_i}{RC_i} \tag{5.7}$$

Two variables, $closer_i$ and $alternative_i$ track the number of accepted riders based on the characteristics matching type which is the Exact, Closer and Alternative matching type in every simulation S_i. In the end, the system adds the values of $closer_i$ and $alternative_i$ from all simulations to compare the number of riders accepted by the type of matching. Equation 2 and Equation 3 represents the added rider count by the type of matching for all simulations. n marks the total number of simulations and $match_{closer}$ and $match_{alternative}$ are the variables that track the total number of accepted riders by the characteristics matching type.

$$match_{closer} = \sum_{S_i=1}^{n} closer_i \tag{5.8}$$

$$match_{alternative} = \sum_{S_i=1}^{n} alternative_i \tag{5.9}$$

Every tracking variable contributes to the performance measurement of the system. The next chapter of results provides an in-depth analysis of the entire Ride Sharing model and provides the contributions of the matching rate, the number of trips, and the time required for trip formation towards the overall system efficiency. Also, the chapter includes a comparison of results from Phase 1 and Phase 2, which states the improvements observed due to changes in Phase 2.

CHAPTER 6

ANALYSIS AND RESULTS

The chapter of analysis and results includes four sections that are most crucial while evaluating the system efficiency. The four sections are (i) Results from Phase 1 (ii) Machine Learning Accuracy Measurement and Evaluation (iii) Results from Phase 2 and (iv) Comparison of Results.

6.1 Results from Phase 1

The three significant components in the results of both phases of the thesis are the matching rate, the number of completed trips, and the total time taken for the completion of a simulation.

6.1.1 Matching Rate

The matching rate or the MR_i provides a fractional value of the accepted riders out of the total traversed riders RC_i in a simulation S_i. It is essential to consider the total traversed riders RC_i in a simulation to understand the impact of the matching rate. Consider an example of a simulation where the computed matching rate is 0.48. For better understanding, MR_i is multiplied by 100 to get the matching rate in percentage. If RC_i is 100, a matching rate of 0.48 implies that the system accepted 48% of riders while finding a match for the broadcasting riders in all the computed trips in a simulation. The expected outcome of the thesis is that the matching rate should improve as the number of traversed riders and the trip UTT increases. If the matching rate is constant or falls for an increasing number of riders, the system fails to be efficient. The X-axis in Figure 19 represents the RC_i or the total number of searched riders while the Y-axis specifies the scale of the matching rate. Each legend or the trend line indicates the trip UTT of the simulation. The matching rate for Phase 1 is drafted using a stacked-line graph in Figure 19.

The observations from the graph in Figure 19 state that with the rising number of riders, the system recorded a higher matching rate than the priorly recorded matching rates as shown in the simulations. The highest recorded matching rate in Phase 1 is 0.49 for 1000 riders and 30

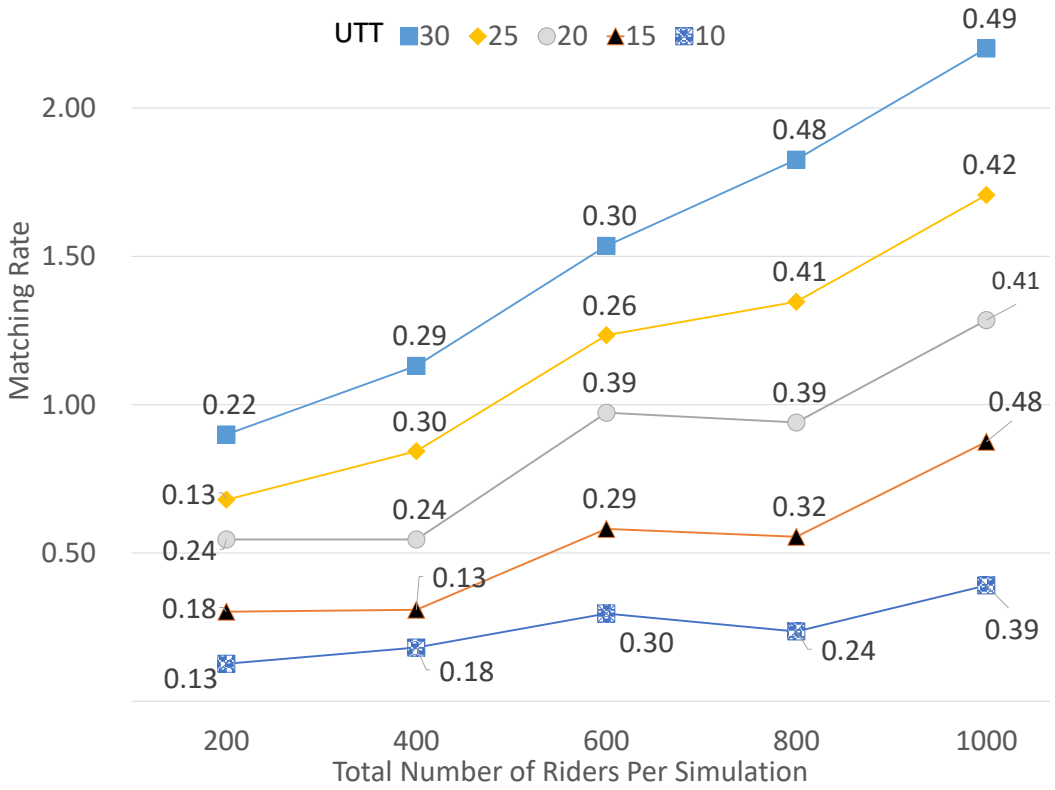

Figure 19: Rider Matching Rate, Phase 1.
Every data-point on the graph is a simulation event. The graph evaluation includes observing the plotted value for a specific UTT and a specific number of traversed riders. For example, the matching rate for UTT:25 minutes and 800 traversed riders is 0.41, which implies that out of 800 searched riders where the trip UTT was 25 minutes, the system accepted 41% or 328 riders using the designed Ride Sharing model.

minutes as the trip UTT. The matching rate of 0.49 implies that out of 1000 traversed riders, the model accepted 49% or 490 riders.

6.1.2 Total Number of Completed Trips

For every simulation S_i, the variable $trip\_count_i$ tracks the number of completed trips. The elements of the stacked-lined graph for the total number of completed trips are similar to the graph of the matching rate. The X-axis reflects the number of traversed riders from 200 to 1000, and the legends indicate the trip UTT with distinct markers. The only difference is with the value and scale on Y-axis. Y-axis represents the total number of computed trips. The expectations from the model in terms of computed trips are that the $trip\_count_i$ should increase with the increasing number of riders plus trip UTT, and the minimum number of computed trips should be at least 3

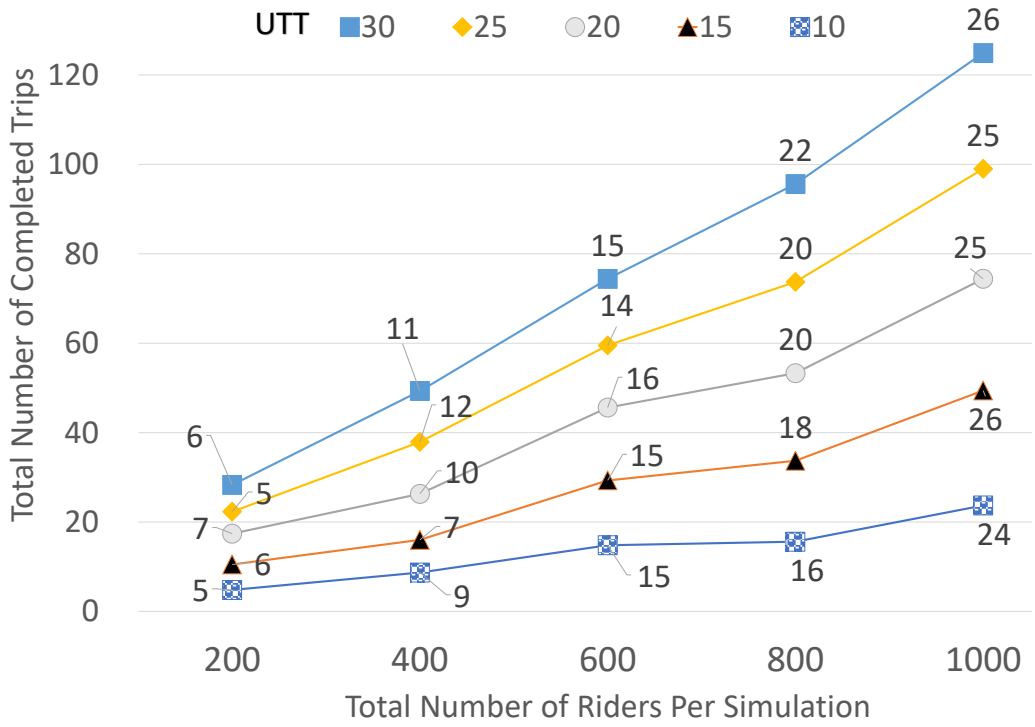

Figure 20: Total Number of Computed Trips, Phase 1.
The number of computed trips is the total count of completed trips in a simulation. The graph is evaluated by observing a simulation event and the value plotted for the simulation indicating the total number of completed trips. For example, for UTT:10 minutes and traversed rider count of 600, the total number of completed trips is 15.

for every 100 traversed riders. Like the matching rate, if the total number of computed trips drops down for progressing simulations, the Enhanced Ride Sharing Model proves to be inefficient. The graph of the computed trips against the count of traversing riders is reflected in Figure 20.

The result in Figure 20 reflects that the Ride Sharing model in Phase 1 of the thesis achieved a trip count of 5 for the first simulation where the UTT is 10 minutes, and the number of traversed riders is 200. The observations from the stacked-lined graph in Figure 20 states that the trip count increases with every simulation event or with the growing count of riders and trip UTT. The highest number of completed trips is 26 for UTT:15 minutes and UTT:30 minutes for a traversed rider count of 1000 riders.

6.1.3 Trip Simulation Time

The trip simulation time or the T_i specifies the total time consumed for completing a simulation S_i. It is an important measure as it states the total time required to complete a specific number of trips. The resultant graph for the trip simulation is portrayed in the form of a stacked-line graph in Figure 21.

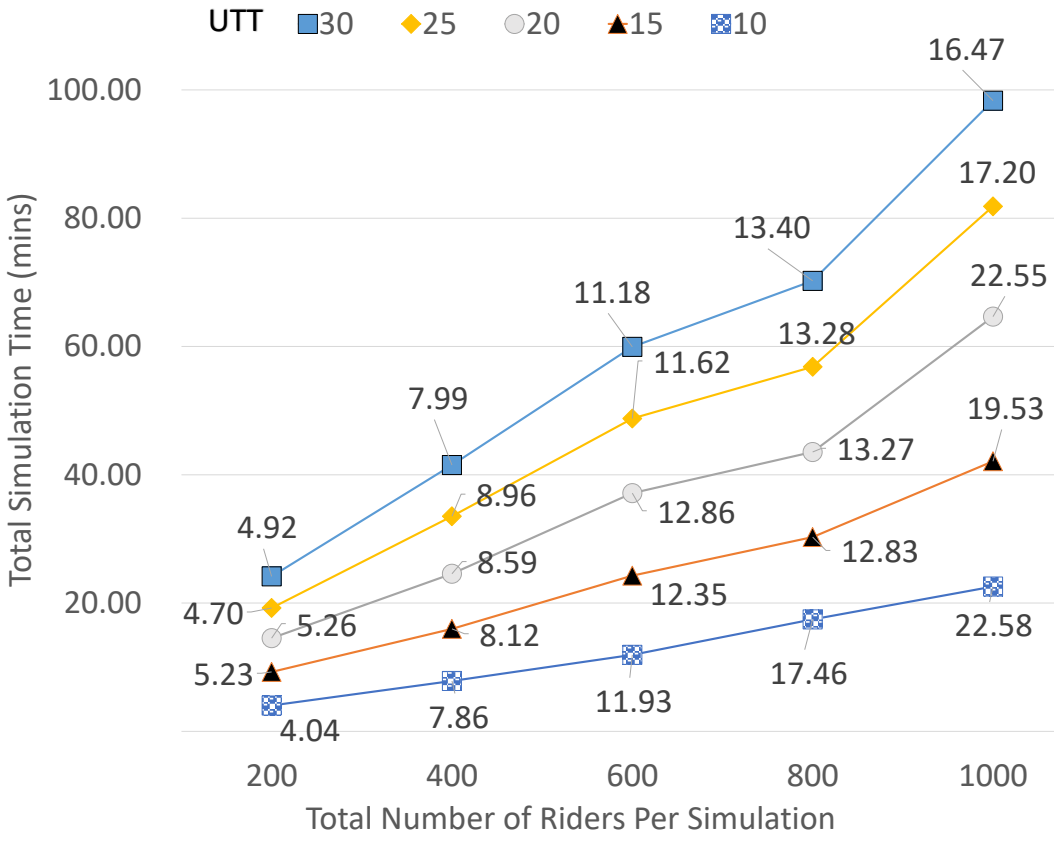

Figure 21: Trip Simulation Time, Phase 1.
The data-points in the stacked-line graph describe the total time recorded to complete one entire simulation. The graph is understood by providing an example of the simulation event where the trip UTT is 15 minutes, and the number of traversed riders is 800. The simulation time recorded for the selected example is 12.83 minutes.

The contribution of the simulation time towards the system efficiency depends on the matching rate and the total number of computed trips in a simulation. The increase or decrease in the simulation time does not affect the system. However, if the simulation time increases for every consecutive simulation, it is crucial to observe the values of the matching rate and the completed number of trips. The expected result in the thesis is that if the simulation time T_i keeps rising for

every simulation S_i, there should be a corresponding increase in the matching rate MR_i and the total number of computed trips $trip\_count_i$. From Figure 21, the noted observation is that the trip simulation time keeps increasing for every progressing simulation. If the figures, Figure 19, Figure 20 and Figure 21 are placed next to each other, the results specify that the matching rate MR_i and the number of completed trips $trip\_count_i$ increases with the rising trip simulation time T_i.

6.1.4 Number of Trips with Pool Completion

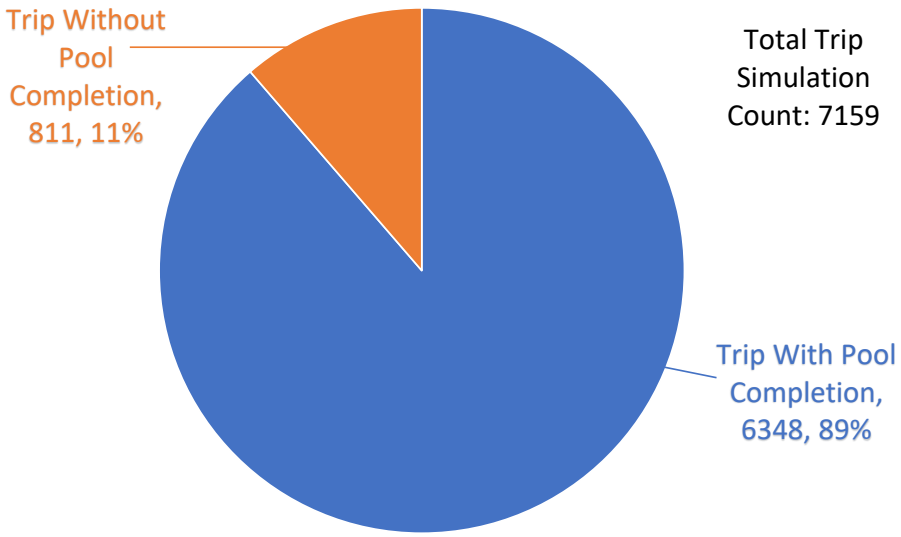

Figure 22: Number of Trips with Pool Completion, Phase 1.
The pie-chart in Figure 22 provides a classification of the trips based on the pool completion status. A higher number of trips with pool completion results in cutting down the fuel usage, reducing carbon footprint, and improving the air quality, which are the objectives of the Ride Sharing model. The total number of computed trips is 7159, out of which 6348 completed the pool, and 811 trips did not complete the pool.

The proposed model indirectly helps humanity preserve environment and fuel resources if the number of computed trips with pool completion is higher than the number of completed trips without pool completion. After the completion of Phase 1, the trip count from all simulations is added and classified based on the pool completion status. Trips that complete the pool entail that the accepted riders and driver reached the vehicle seating capacity. Figure 22 reflects the drafted classification of trips by the pool status in a pie-chart. The expected outcome in the thesis is that out of the total number of computed trips, at least 70% of trips should complete the pool.

The result achieved in the case of trips with pool completion is considerately acceptable.

The pie-chart in Figure 22 shows that the number of completed trips with pool completion is 89%, which is a significant measure in the case of the Ride Sharing model. For the total computed trips $trip\_count_n$ and n being the total number of simulations, it is confirmed that around 90% of the total trips completed the journey with pool completion.

6.1.5 Count of Matches By Characteristics Matching Type

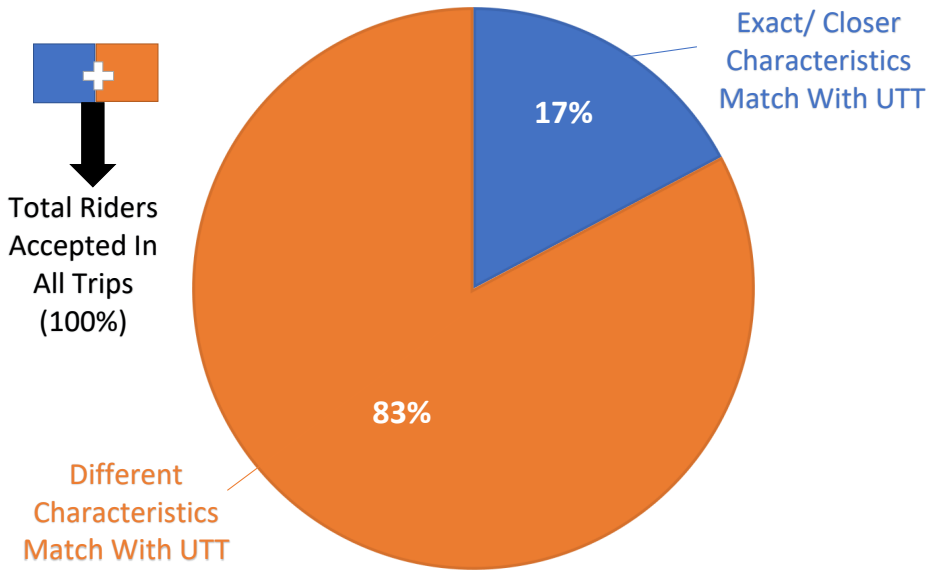

Figure 23: Rider Count Classified by Matching Type, Phase 1.
The pie-chart in Figure 23 provides a classification of the rider count based on the rider matching types. Out of all the accepted riders, the system accepted 17% of riders by the Exact or Closer matching type and the rest by the Alternative or Different matching type. The result achieved is an average quality result, but as every accepted rider goes through the UTT matching layer, the overall result is satisfactory.

The count of matches by characteristics matching type is one of the most critical measures of the system. The model design includes the Exact, Closer, and Alternative types of characteristics matching. As stated in the previous chapter, the variables which track the rider count by the matching type are the $match_{closer}$ and $match_{alternative}$. If the system accepts a rider by the Exact or Closer matching, the value of $match_{closer}$ is incremented by 1. Alternatively, if the system accepts a rider by the Alternative type of matching, the value of the $match_{alternative}$ is increased by 1.

The reason for generating results by the matching type is to check which rider characteristics matching type is the most utilized by the system while creating trips. The

expected outcome in the thesis is that the number of rider matches by the Exact or Closer characteristics matching must be higher than the number of rider matches by the Alternative matching type. Based on the values of $match_{closer}$ and $match_{alternative}$, the accepted riders are drafted on a pie-chart as showcased in Figure 23. The two classes in the pie-chart are the Exact or Closer characteristics match with UTT, and the Different or Alternative characteristics match with UTT.

The pie-chart in Figure 23 reveals that the result achieved in terms of the number of rider matches by characteristics matching types is an average quality result. The system recorded a lower number of matches by the Exact and Closer matching type than the Alternative matching type. But, as all riders undergo the UTT matching layer and as most of the trips complete the pool, the overall result is acceptable.

6.2 Machine Learning Accuracy Measurement and Evaluation

6.2.1 True Positive, True Negative, False Positive, False Negative

It is necessary to provide the definitions of true positive (tp), true negative (tn), false positive (fp), and false negative (fn) for evaluating the Machine Learning classifiers. The definations are given with the help of a confusion matrix which is illustrated in Figure 24.

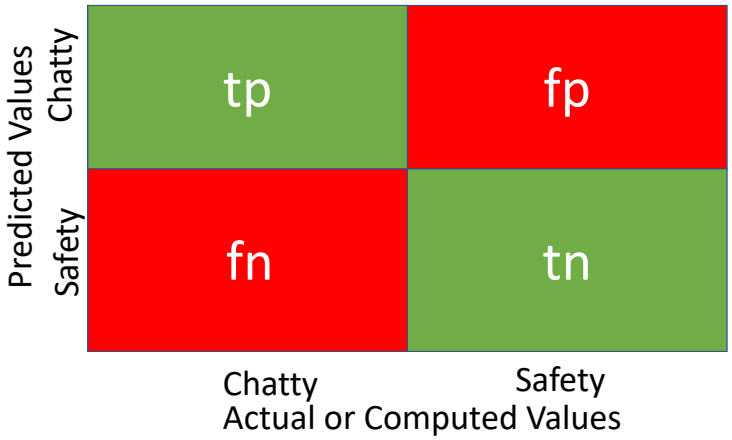

Figure 24: An Example of Confusion Matrix.
Figure 24 illustrates a simple example of the confusion matrix with two classes, safety and chatty. The values on the X-axis represent the actual or computed values and the values on the Y-axis represent the predicted values by a Machine Learning classifier. The matrix contains values which determine the quality of the prediction of a Machine Learning classifier in the form of true positive (tp), true negative (tn), false positive (fp), and false negative (fn).

The confusion matrix provides the level of correctness between a system's computed value for a class and a Machine Learning classifier's predicted value for the same class. For illustrating the confusion matrix, only two classes have been considered, the chatty and the safety class. Let the computed main characteristic for a rider be the chatty class. The system evaluates the Machine Learning accuracy by checking the main characteristic predicted by the SVM classifier.

If a Machine Learning classifier predicts a class that is similar to the computed class, the prediction is true positive (tp). For example, if the SVM predicts the class chatty as the main characteristic for the rider, the prediction is a true positive prediction.

Consider a case where an entity E does not belong to a class C_{false}. If the system subjects the classifier with the entity E and the class C_{false} and the classifier correctly predicts that E does not belong to class C_{false}, the prediction is true negative (tn). Directing to the same example, if the system subjects the SVM with the rider and the safety class, and if the SVM predicts that the rider is not associated with the safety class, the prediction is a true negative prediction.

The prediction is a false negative (fn) prediction when the Machine Learning classifier predicts the wrong class as the right class. In the example of the rider, if the SVM predicts the safety class, the prediction is a false negative prediction as the computed class is chatty.

Consider a case of the entity E, which belongs to a class $C_{correct}$. If the system subjects the Machine Learning classifier with class $C_{correct}$ and the entity E, and the classifier predicts that E does not belong to class $C_{correct}$, then the prediction is false positive (fp). In the same example, if the system subjects the SVM with the rider and the class chatty, and the SVM predicts that the rider does not belong to the chatty class, the prediction is a false positive prediction.

6.2.2 Performance Measures

With the definitions of tp, tn, fp, and fn, the important elements of a Machine Learning performance measure are the accuracy, F1 score, precision, and recall.

The first measure of performance for the Machine Learning classifier is the accuracy. Accuracy is the fractional value of the total number of correctly predicted samples to the total number of present samples in a data-set. Equation 6.1 represents the accuracy measure of a Machine Learning classifier.

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{6.1}$$

Consider a case of a data-set where the number of false positives and false negatives are the same. When the fp and fn are almost equal, the data-set is balanced. In the case of an imbalanced data-set, the accuracy is not enough to measure the quality of prediction of a Machine Learning classifier. The Enhanced Ride Sharing Model possesses imbalanced feedback data-sets, and therefore more measures are required to evaluate the efficiency of SVMs. The other performance measures for evaluating a Machine Learning classifier are precision, recall, and the F1 score.

The second performance measure is precision, which is the division of the true positive predictions to all the positively predicted predictions. From all the positive predicted values, precision states which are correctly predicted values that match precisely to the computed values. The system calculates the positive predictions by adding all true positive and false positive predictions. A classifier is expected to possess a higher precision value for quality prediction. The following Equation 6.2 represents the formula for computing precision of a Machine Learning classifier.

$$precision = \frac{tp}{tp + fp} \tag{6.2}$$

The third performance measure is the recall. Recall provides the fractional value of the correctly predicted samples to the total samples in a data-set. In the case of the recall, the system adds the true positive and false positive predictions to get the whole sample set. Equation 6.3 provides the formula for computing the recall of a Machine Learning classifier.

$$recall = \frac{tp}{tp + fn} \tag{6.3}$$

It may be the case that the classifier predicts accurately for a particular set of classes but predicts incorrectly for a few classes. In such a case, a possible solution is using the combination of precision and recall. The combination of precision and recall provides the F1 score. The major

focus of the F1 score is on false positives and false negatives. Equation 6.4 provides the formula for computing the F1 score.

$$F1\_Score = 2 * \frac{recall * precision}{recall + precision} \tag{6.4}$$

If a classifier has a higher computed F1 score, the precision and recall utilized for computing the F1 score are also high. Hence, a good or a higher F1 score indicates the classifier prediction is accurate. The last performance measure for a Machine Learning classifier is the Root Mean Square Error (RMSE). The concept of the RMSE is explained through Figure 25.

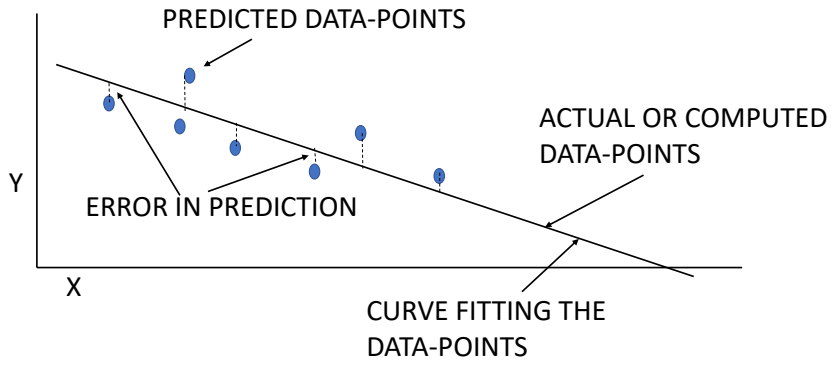

Figure 25: An Illustration of Root Mean Square Error (RMSE).
The Root Mean Square Error (RMSE) is a performance measure designed to get the difference between the predicted and computed values. The computation begins with the plotting of the predicted and computed data-points. The distance or the dotted lines between the fitted curve and the predicted points is the actual error. The system calculates the RMSE by taking the square root of the average of all the squared errors and provides an overall estimate of how correctly a classifier can predict.

RMSE computes a value based on the errors present in every predicted sample point. Initially, a curve or a line passes through all the computed data-points in the system. The first step for calculating RMSE is the computation of error. Equation 6.5 gives the error between the computed and predicted data-points. The error is the distance between the computed data-points and the predicted data-points.

$$error = y_{computed} - y_{predicted} \tag{6.5}$$

As the errors may result in negative values, all the errors are squared and added. The

added squares are divided by the total number of data-points, providing the mean of squared errors. The last step is computing the square root of the mean. A lower RMSE value represents less error and therefore signifies that the classifier prediction is accurate. Equation 6.6 is utilized for calculating the Root Mean Square Error (RMSE).

$$RMSE = \sqrt{\frac{(error)^2}{total\_sample\_points}} = \sqrt{\frac{(y_{computed} - y_{predicted})^2}{total\_sample\_points}} \tag{6.6}$$

The thesis uses the confusion matrix, F1 score, precision, recall, RMSE, and accuracy for measuring the quality of prediction and the performance of the Support Vector Machines.

6.2.3 Performance Measure of SVMs

Due to the presence of the imbalanced data-sets in the training and testing of Support Vector Machines, it is essential to measure F1 score, precision, and recall for the five classes and the overall RMSE and accuracy. The evaluation of the SVMs begins with the performance measure for the first SVM, which is the Feedback-Given-Characteristic SVM. Table 6 cites the calculated performance measures for the first SVM classifier. Also, to compare the system computed values with the SVM predicted values for every class, the section of performance measure includes the confusion matrix plotted for the first SVM, as shown in Figure 26.

Overall SVM Accuracy: 91.65%					
Root Mean Square Error: 0.64					
Accuracy Measure By Class					
Measurement(%)	**Chatty**	**Safety**	**Punctuality**	**Friendliness**	**Comfortability**
F1 Score	92.34	91.65	91.07	91.92	90.90
Precision	87.04	90.40	91.97	95.84	97.35
Recall	98.31	92.94	90.20	88.32	85.25

Table 6: Performance Measures for Feedback-Given-Characteristic SVM.
Table 6 provides the results of the first Support Vector Machine. The overall classifier accuracy is around 92%, and RMSE is 0.64. Also, the computed F1 score, precision, and recall for every class are greater than 85%, which defines that the first SVM classifier predicts accurately.

The expected result in the thesis from the perspective of Machine Learning is getting a higher score for accuracy, F1 score, precision, recall, and getting a minimal RMSE score for the Feedback-Given-Characteristic. From Table 6, it is confirmed that the Feedback-Given-Characteristic SVM achieves a higher score for every performance measure. The

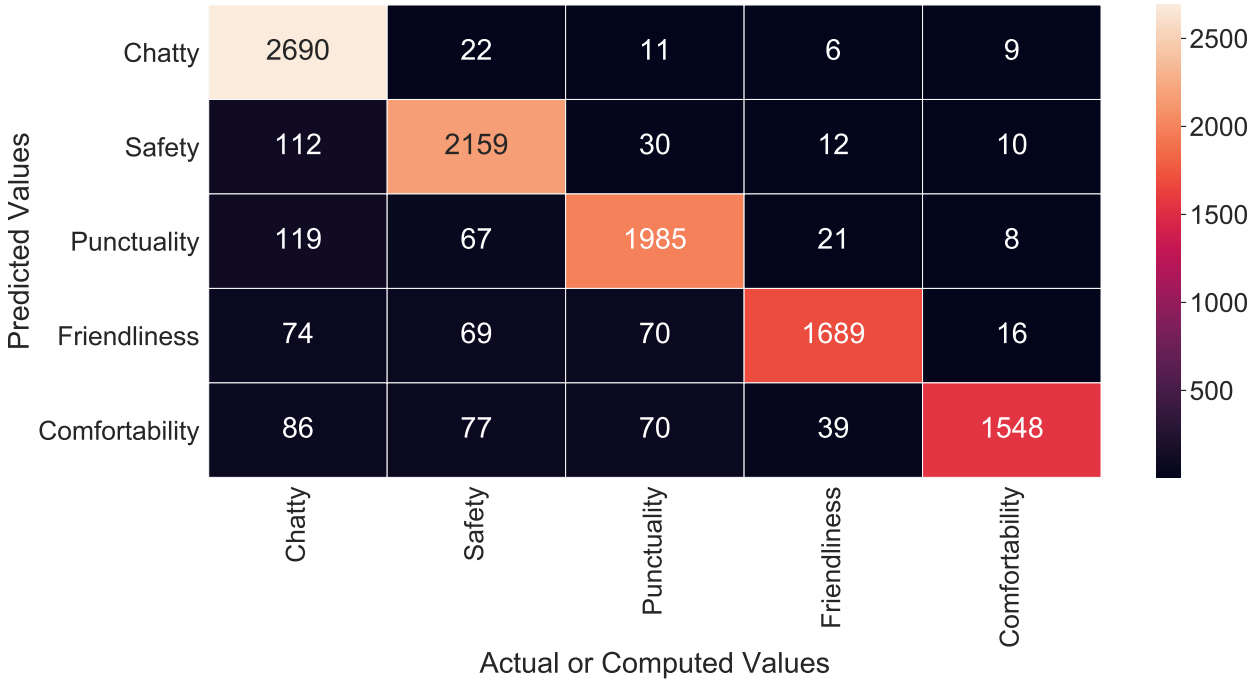

Figure 26: Confusion Matrix for Feedback-Given-Characteristic SVM.
The confusion matrix provides the relation between the predicted and computed values for the five classes. The numeric red scale on the right side of the heatmap defines the total number of tested records for the first SVM. If the computed values match with the predicted values, the system adds 1 to the true positives in the confusion matrix. The darkening of the red color based on the red scale signifies that the highest number of matches between the computed and predicted values occur in the true positives for all the classes.

computed F1 score, precision, and recall for all the classes are above 85%, proving the fact that the predicted main characteristics match the system computed main characteristics. Also, a lower RMSE score of 0.64 signifies that there is notably less error between the computed values by the system and predicted values by the SVM classifier.

A similar approach is to calculate the performance measures for the second SVM or the Feedback-Received-Characteristic SVM, which includes computing the overall classifier accuracy, RMSE, F1 score, precision, and recall for the five classes. Table 7 presents the evaluated performance measures for the second SVM classifier. To show the difference between the computed and predicted values, Figure 27 provides the confusion matrix for the second SVM classifier. The expected result from the second classifier is achieving a higher score for accuracy, F1 score, precision, and recall plus recording a minimal RMSE score.

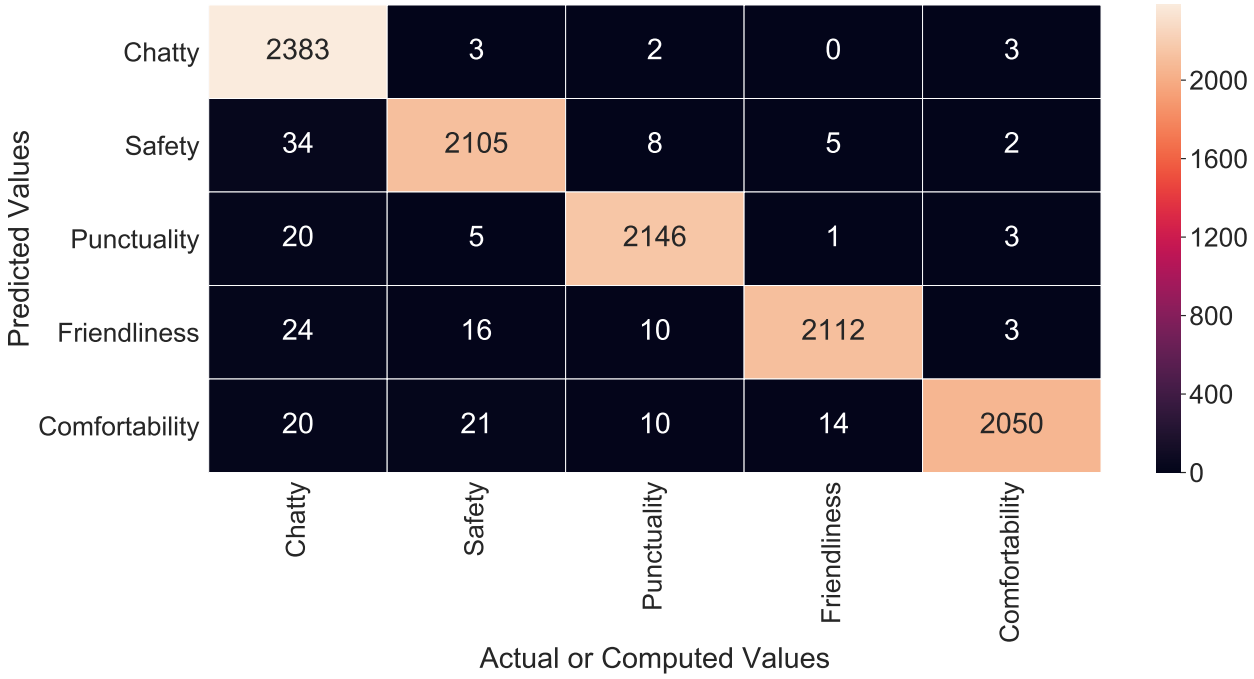

Figure 27: Confusion Matrix for Feedback-Received-Characteristic SVM.
The confusion matrix of the second SVM has a similar structure to that of the confusion matrix of the first SVM. The scale on the right side of the heat map represents the number of testing records from 0 to 2000 or more. The matrix states that the true positives have the maximum values in every column, which signifies that a higher number of predicted values match the computed values. With the observation of the minimal difference between the computed and predicted values, the Feedback-Received-Characteristic works accurately.

Overall SVM Accuracy: 91.33%					
Root Mean Square Error: 0.42					
Accuracy Measure By Class					
Measurement(%)	**Chatty**	**Safety**	**Punctuality**	**Friendliness**	**Comfortability**
F1 Score	87.85	89.02	90.63	93.22	93.21
Precision	86.13	87.52	92.58	91.97	95.48
Recall	89.21	88.82	89.67	94.49	96.96

Table 7: Performance Measures for Feedback-Received-Characteristic SVM.
Table 7 presents the performance measures for the Feedback-Received-Characteristic SVM. The accuracy of the second SVM rounds up to 91%, and RMSE is significantly low, which is 0.42. Also, the F1 score, precision, and recall are above 85%.

From the confusion matrix in Figure 27 and the observations in Table 7, it is inferred that the computed F1 score, precision, and recall for every class is close to 90%. A higher score for priorly stated performance measures contributes to the quality of prediction of the

Feedback-Received-Characteristic SVM. Also, the RMSE error is quite less, which is 0.42, which signifies that there the error is notably less between the computed and predicted values.

For both SVMs, the performance measurement of the Machine Learning classifier started with 2000 training records and 800 testing records. With fewer training and testing records, the accuracy computed to 20%. For gradually increasing the accuracy, the system increased the training records by 2000 and the testing records by 1000 for every accuracy-test event. The testing also included the meddling of the regularization and gamma parameter of the SVMs to attain maximum accuracy. The training and testing ceased after receiving a 90% accuracy for both SVMs. For getting such high accuracy, the system trained the SVMs with 27,000 records and tested the SVMs with 12,000 records. From Table 6 and Table 7, a positive result for both SVMs is that the accuracy, precision, recall, and F1 score is above 85% which is a good measure for a Machine Learning classifier. The quality of prediction is good, and the SVM classifiers assist in providing real-time recommendations to riders.

6.3 Results from Phase 2

The significant contributions of the thesis are in Phase 2. In Phase 2, the running of simulations continued until getting relevant and improved results than Phase 1. The results prove to be satisfactory after executing the entire simulation for 5^{th} time, which is half the number of the simulations executed in Phase 1. In Phase 2, the similar results are evaluated, which are the matching rate, the total number of completed trips, the total trip simulation time, the number of computed trips with pool completion, and the classification of riders based on the characteristics matching type. As the Phase 1 results specify the description of each evaluation measure for the Ride Sharing model, Phase 2 directly provides the thesis objectives, results achieved, and the supporting tables plus figures.

6.3.1 Matching Rate

The expected outcome in terms of matching rate for Phase 2 is that the computed matching rates in Phase 2 for every simulation should be higher than the recorded matching rates in Phase 1. Additionally, the matching rate should keep improving with the increasing number of riders and trip UTT.

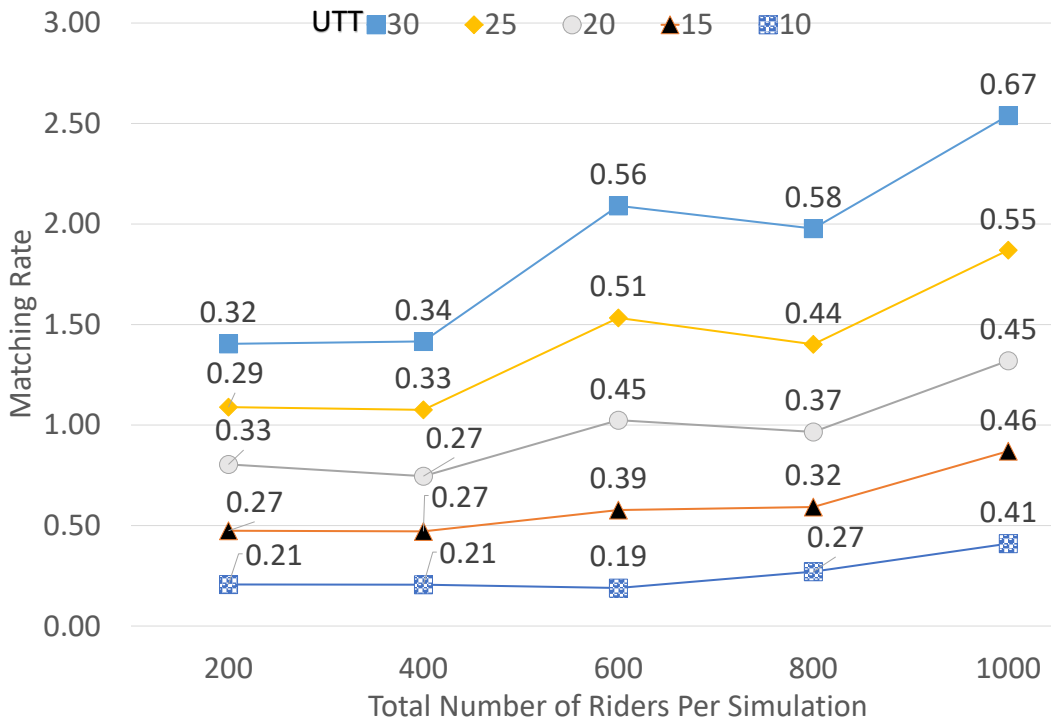

Figure 28: Rider Matching Rate, Phase 2.
Figure 28 is similar to the graph of the matching rate of the results in Phase 1. The evaluation of the graph includes noting the plotted point for a specific number of traversed riders and the respective trip UTT. For example, the system computed a matching rate of 0.58 for the trip UTT:30 minutes and traversed rider count of 800.

The observations from Figure 28 reflect that the model achieved a higher matching rate for every simulation in Phase 2 as compared to Phase 1. For example, for 200 traversed riders and trip UTT as 30 minutes, the matching rate in Phase 1 is 0.22, and the matching rate in Phase 2 is 0.32. Another example is of the UTT 30 minutes and 1000 traversed riders, where the matching rate has increased from 0.49 in Phase 1 to 0.67 in Phase 2. Hence, results from Phase 2 are better than Phase 1 in the aspect of the matching rate.

Moreover, from Figure 28, it is concluded that the matching rate in Phase 2 keeps improving for every consecutive simulation. The highest recorded matching rate in Phase 2 is 0.67 for 1000 traversed riders and UTT:30 minutes. The matching rate of 0.67 implies that out of 1000 searched riders, the system accepted 67% or 670 riders.

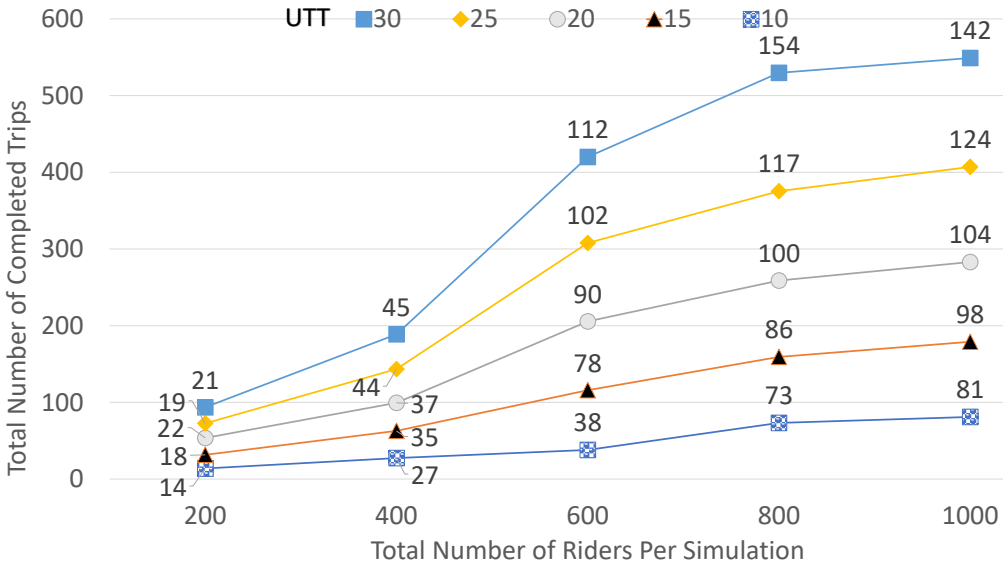

Figure 29: Average Number of Computed Trips, Phase 2.
Figure 29 provides the total number of computed trips for every simulation in Phase 2. The graph evaluation consists of reading the plotted data-points, which marks the individual simulation events. For example, the number of computed trips for UTT:25 minutes and 600 traversed riders is 102.

6.3.2 Total Number of Computed Trips

The expected result for the number of computed trips is that the number of computed trips in Phase 2 should exceed the completed number of trips in Phase 1 for every simulation. Also, the trip count in Phase 2 should not fall with the increasing number of riders and trip UTT.

Based on the results in Figure 29, the system achieved a higher number of trips as compared to Phase 1 for all simulations. For example, for RC_i or a traversed rider count of 800 and the trip UTT as 20 minutes, the computed $trip\_count_i$ in Phase 1 is 20, and the computed $trip\_count_i$ in Phase 2 is 100. Also, the observations from Figure 29 state that the $trip\_count_i$ or the number of completed trips keeps rising with the increasing number of riders and trip UTT. The highest number of recorded trips is 142 for the trip UTT:30 minutes and traversed count of 1000 riders.

6.3.3 Trip Simulation Time

One of the objectives in terms of trip simulation time for Phase 2 is that the matching rate and the count of computed trips should increase if the simulation time keeps increasing for

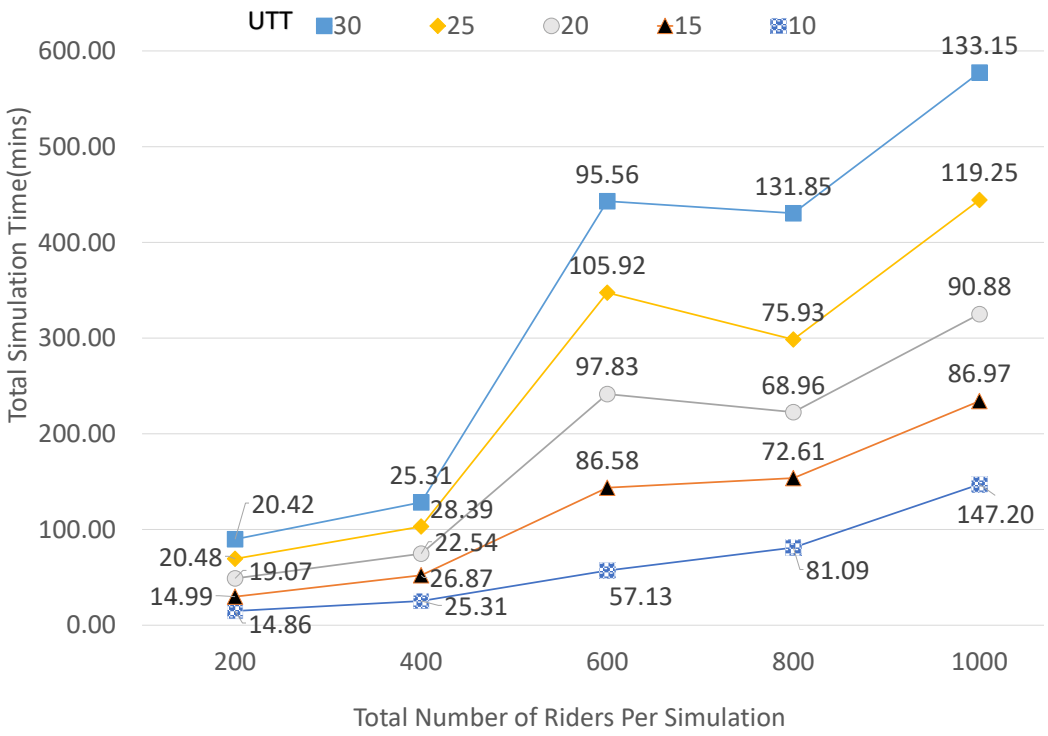

Figure 30: Trip Simulation Time, Phase 2.
The structure of the stacked-line graph for trip simulation time in Phase 2 is similar to the structure of the trip simulation time graph in Phase 1. The graph evaluation consists of noting the plotted data-point with traversed riders and trip UTT. For example, the computed trip simulation time for UTT 10 minutes and 800 traversed riders is 81.09 minutes. It is necessary to note the matching rate and computed trips in a simulation to analyze the impact of higher simulation times.

consecutive simulations. If the results are compared between Phase 1 and Phase 2, including the matching rate, the simulation time, and the number of completed trips, the system observes an increase in the matching rate MR_i and the trip count $trip\_count_i$ with the growing simulation time T_i for every simulation. The highest recorded simulation time is 133 minutes for the UTT:30 minutes and the traversing count of 1000 riders.

6.3.4 Number of Trips with Pool Completion

The expected outcome in terms of the trips with pool completion is that the number of trips that complete the pool should be higher than the number of trips that do not complete the pool. The second objective is to record a higher number of trips with pool completion than the trips recorded with pool completion in Phase 1.

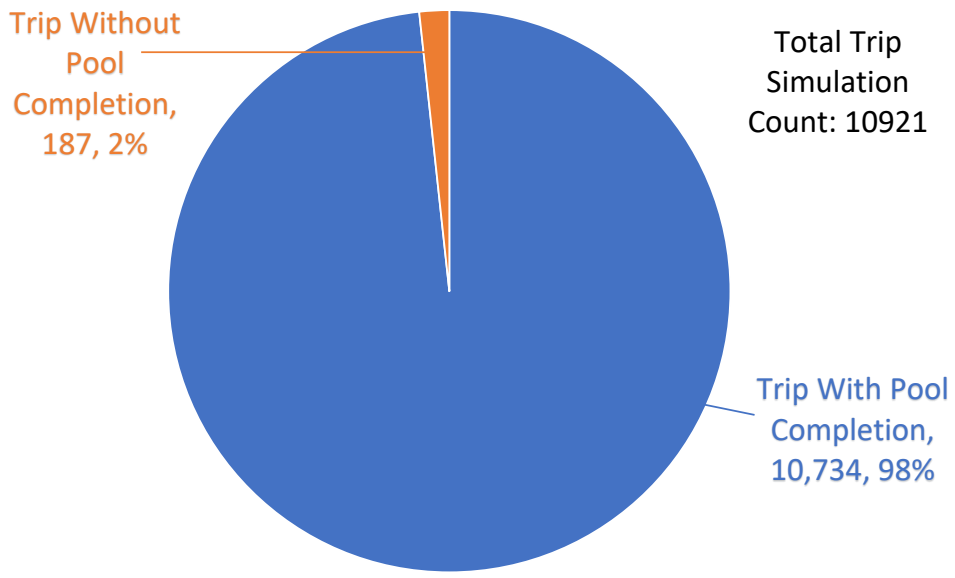

Figure 31: Number of Trips with Pool Completion, Phase 2.
The pie-chart in Figure 31 separates the trips based on a value, which indicates if the riders reached the seating capacity of the vehicle. The Ride Sharing model achieved acceptable results in both phases, which states that the number of trips that completed the pool is higher than the trips that failed to complete the pool.

One of the best results achieved in both phases is the result of the number of computed trips with pool completion. The result is notable in the case of Phase 2, where the number of trips that complete the pool is 98%. In the first phase, the total percentage of trips that completed the pool is 85%. Thus, Phase 2 results are better as compared to Phase 1 by 10%. The total computed trips in Phase 2 are 10,921 trips. Out of 10,921 trips, 10,734 or 98% of trips have the pool completion status as "Yes," while 187 or 2% of all the computed trips have the pool completion status as "No."

6.3.5 Count of Matches By Characteristics Matching Type

One of the major objectives in Phase 2 is to significantly increase the number of rider matches by the Exact and Closer types of characteristics matching. The objective also implies that the maximum number of matches should occur by the Exact or Closer characteristics types of matching and not by the Alternative type of characteristics matching.

The result of the rider matches by the characteristics matching type is of the utmost importance as it indicates the percentage of users matched by a specific characteristic matching

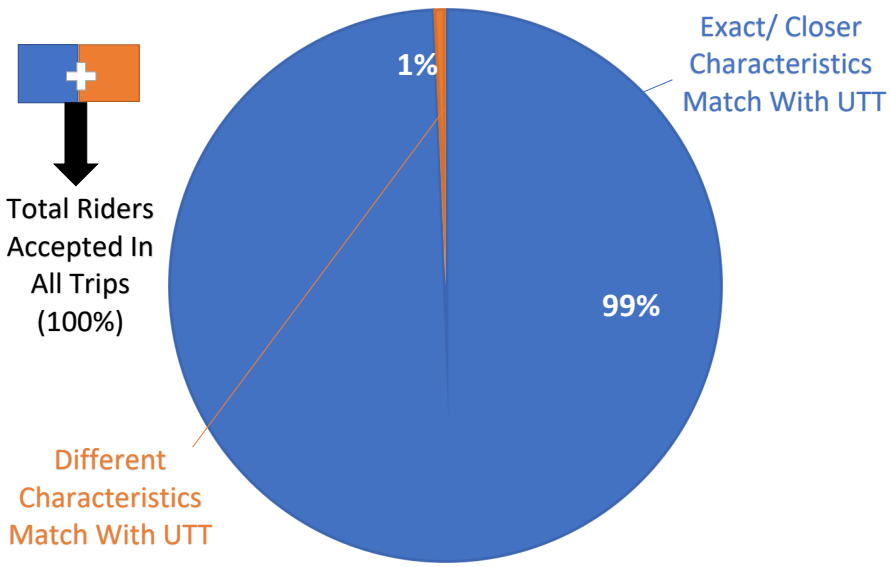

Figure 32: Rider Count Classified by Matching Type, Phase 2.
The pie-chart in Figure 32 provides the classification of the rider count by the characteristics matching type. In Phase 2, the model accepted a higher number of riders in the Exact and Closer matching type than the Alternative matching type. Of all accepted riders, the model matched and accepted 1% of riders having entirely different characteristics while the model paired and accepted 99% of riders with similar or a little bit different characteristics.

type. The observations derived from Figure 32 reflect that the Ride Sharing model experienced a more significant enhancement in the case of rider matches by Closer and Exact characteristics matching. The system performed the maximum number of rider matches by the Exact and Closer characteristics matching in Phase 2. Along with the result of maximum trips computation with pool completion, the result of the rider match count in Phase 2 is highly satisfactory.

6.4 Comparison of Results

The section of the result comparison focuses on the enhancements in Phase 2 and provides the reasons for improvements in the second phase. The section begins with Table 8, which showcases a comparison of the observations noted in Phase 1 and Phase 2.

The average trip formation time is the average time taken by the system to create an entire trip. The creation of the trip includes selecting a broadcasting rider and driver, adding riders based on the characteristics matching layer, filtering the riders through the UTT matching layer, and formation of the trip document. The observations provided satisfactory results for both phases in terms of the trip formation time. In both phases, the trip formation time rounds up to

Observations	Phase 1	Phase 2
Total number of computed trips	7159	10921
Total trips computed with pool completion	6348	10734
Total number of riders traversed	276400	90800
Average trip formation time (mins)	0.80	1.02

Table 8: Comparative Observations from Phase 1 and Phase 2.
To the addition of the number of computed trips with pool completion, the observations include the average trip formation time and the total number of riders traversed in all simulations. Observations provide the improvements and the drawbacks recorded in simulations which play a key role in evaluating the system performance.

a minute.

The total number of riders traversed is the count of all riders searched in all the simulations. In Phase 1, the system traversed through 276,400 riders for all simulations, while in the second phase, the system traversed through 90,800 riders. In Phase 2, the system not only computed a higher number of trips but also computed a higher number of trips with pool completion. The comparative observation of the trip count specifies that the system improved in Phase 2 of the thesis because of the Machine Learning algorithms. A higher number of computed trips with pool completion proves the system is efficient and assists in promoting the usage of Ride Sharing.

Figure 33 provides a comparison based on the number of computed trips with pool completion in Phase 1 and Phase 2. Comparatively, both the results are satisfactory as the percentage of trips that complete the pool is greater than 85% in Phase 1 and Phase 2.

Figure 34 comprises the vital results of accepted rider count based on Exact, Closer, and Alternative matching type. By comparing the results, it can be concluded that the results in Phase 2 are profoundly better than Phase 1. A higher count of matched riders possessing similar characteristics reaches the expectation of the thesis of computing maximum rider matches by Exact and Closer characteristics matching type.

The part of the performance measures comparison is one of the most critical parts of the results. The section of images begins by comparing the matching rates from Phase 1 and Phase 2 in Figure 35. Figure 36 compares the results in terms of the total number of computed trips for both phases, while Figure 37 presents a visual comparison of the total simulation time in both

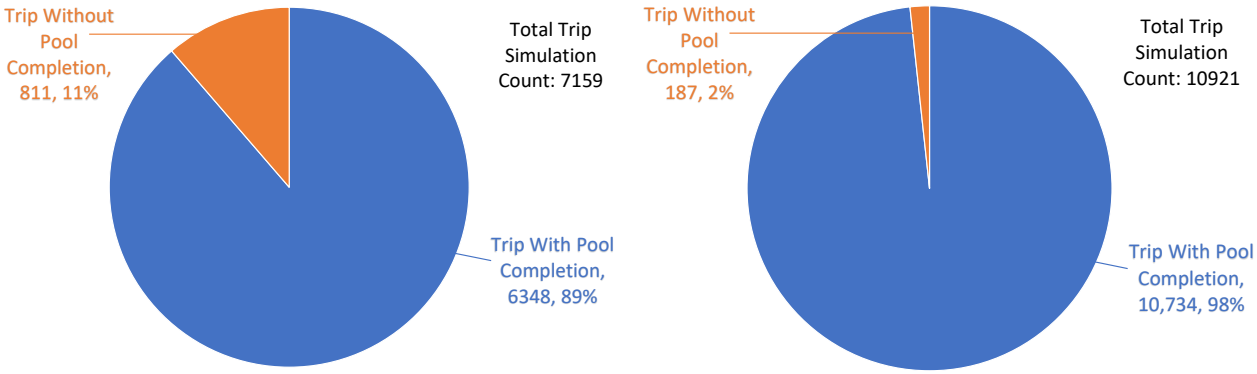

Figure 33: Result Comparison of the Classification of Trips Based on Pool Completion.
From a perspective of pool completion, both the phases provide the expected results of maximum trips computation with pool completion. The picture on the left is the result from Phase 1 while the picture on the right is the result from Phase 2.

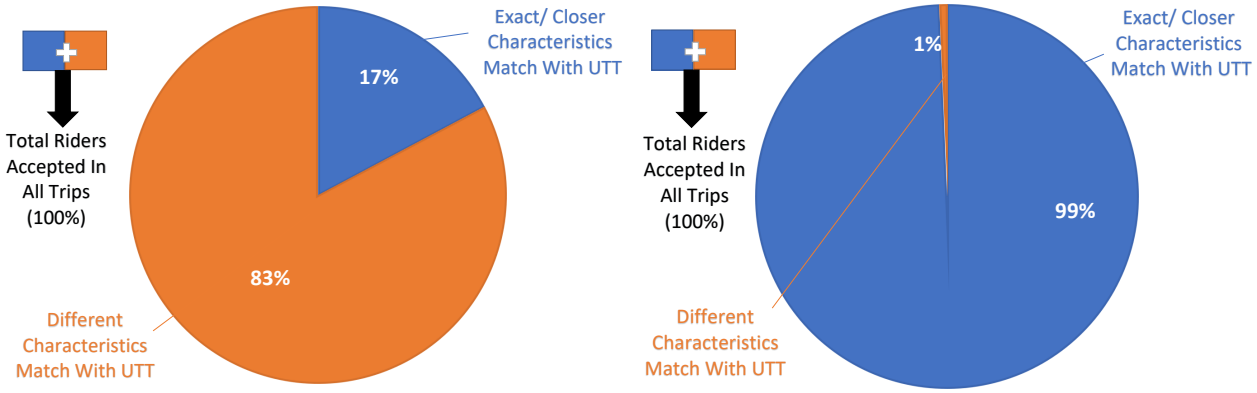

Figure 34: Result Comparison of the Classification of Match Count Based on Characteristics Matching Type.
From the comparison, it is recognizable that the system performance while matching the riders in Phase 1 is average. With the integration of the Machine Learning recommendation system, the system experienced a drastic improvement in the Closer and Exact Matching. The higher number of matches implies that most of the users are commuting with other users who have exactly similar or a little bit different characteristics.

phases. The images on the left are the results from Phase 1, and the images on the right are the results from Phase 2.

Machine Learning has a powerful impact on the Enhanced Ride Sharing Model. From the observations in Table 8, Phase 2 provided better outcomes. Even though the system traversed through fewer riders in Phase 2, with the help of a recommendation system, the system computed a higher number of trips with pool completion. The number of total traversed riders in Phase 1 is 276,400 traversed riders, and the number of computed trips with pool completion is 6,348 trips. Phase 2 observed a lower number of total traversed riders than Phase 1, which is 90,800 riders, but Phase 2 resulted in the computation of the higher number of trips with pool completion i.e., is 10,734 trips than Phase 1.

The further part of the section discusses the model efficiency based on the observations made in Figures 35, 36 and 37. The system's efficiency is measured using the total trip simulation time T_i, matching rate MR_i, and the number of computed trips $trip\_count_i$ for every simulation event S_i. In phase 2 there is a higher increase in T_i as compared to Phase 1. The system proves to be efficient if it follows the condition which is, for an increase in T_i, the values of MR_i and $trip\_count_i$ should also increase. If MR_i and $trip\_count_i$ decrease with the increasing T_i, the system fails to be efficient. As shown in Figure 35 and Figure 36, the MR_i and $trip\_count_i$ increases for every consecutive simulation. Also, from the result comparison of the three stacked-line graphs, in most of the cases, the plotted values of MR_i, $trip\_count_i$ and T_i in Phase 2 is greater than the plotted values in Phase 1.

In both phases, results reach the expectations of the thesis and achieve the best using Machine Learning SVM classifiers. The matching rate MR_i and the number of computed trips $trip\_count_i$ keep increasing due to the increasing number of traversed riders RC_i and increasing UTT or U_i. The greater the RC_i, there is more room for matching of riders. Also, with increased U_i, riders located at a distance that needs a little longer traveling time can be accepted. Hence, the matching rate and the number of trips depends on the number of riders and the UTT. It can be stated from the results, MR_i and $trip\_count_i$ is directly proportional to RC_i and U_i.

A significant change observed in the thesis is in the matching by characteristics type. In Phase 1, the system accepted fewer riders by the Exact or Closer characteristics matching type.

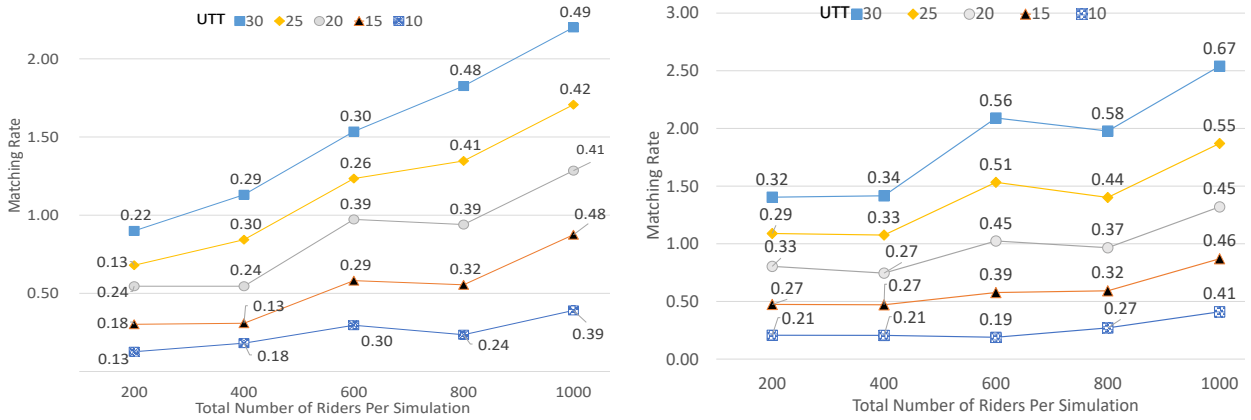

Figure 35: Result Comparison of the Matching Rates from Phase 1 (left) and Phase 2 (right).

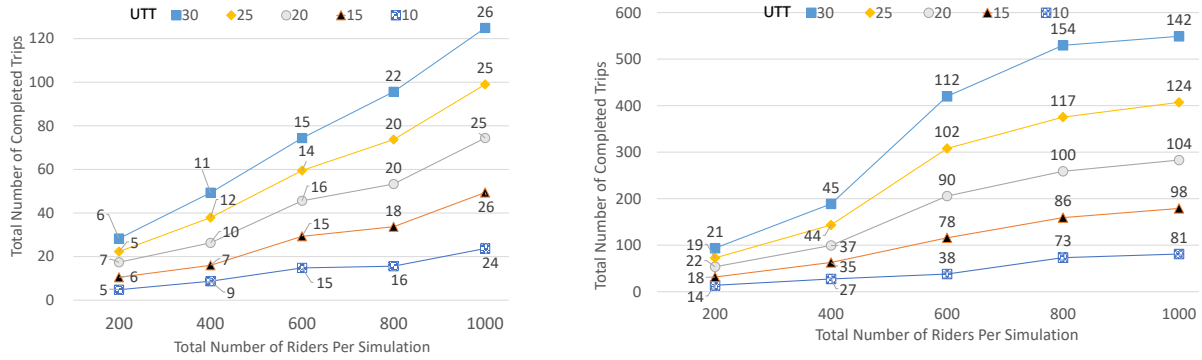

Figure 36: Result Comparison of the Number of Computed Trips in Phase 1 (left) and Phase 2 (right).

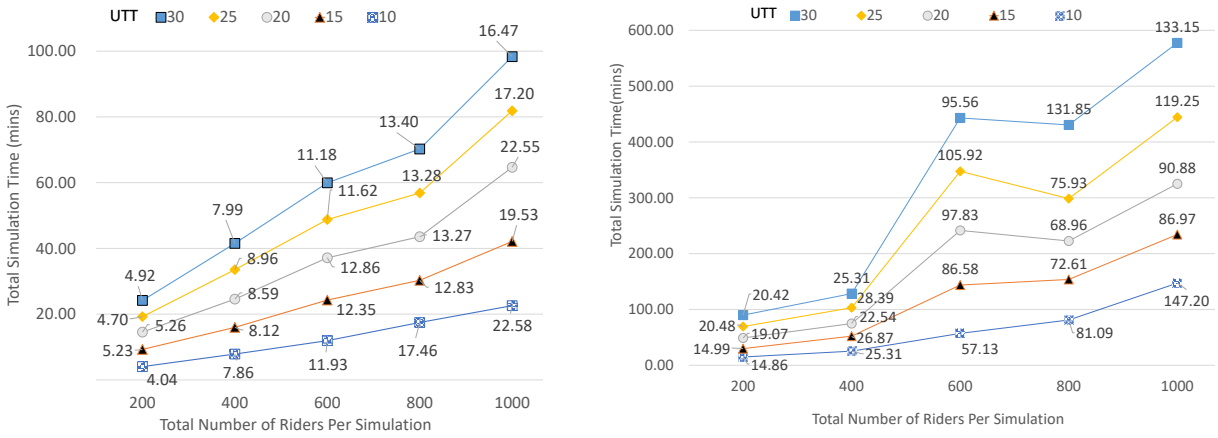

Figure 37: Result Comparison of the Total Simulation Time in Phase 1 (left) and Phase 2 (right).

In the second phase, $match_{closer}$ is much higher than the $match_{alternative}$ or the number of accepted riders by Closer and Exact characteristics type is much higher than the riders accepted irrespective of characteristics. The reason for a higher match is the replacement of manual altering of characteristics with the Machine Learning recommendation systems. In Phase 1, if the rider characteristics did not match other rider characteristics by 100%, the system rejected the rider in the three types of matching. In Phase 2, if the match is 85%, the system accepted the riders by any matching characteristics types, which provided more features for rider matching and resulted in a higher matching by the first two characteristics matching type i.e., by the Closer and Exact matching type. Getting a higher match in Closer and Exact matching is one of the profoundly expected outcomes of the system, where most of the users with similar likings are added on the same trip. The comparison of the rider count by matching type is stated in Figure 34.

Overall, the Machine Learning algorithms assisted significantly in tuning up the Enhanced Ride Sharing Model efficiency. The system achieved a higher matching rate with a higher number of trips with pool completion in Phase 2 as compared to Phase 1. Also, as viewed in Table 8 or the Table of Observations, in both phases, the average trip formation time rounds up to a minute, which is highly satisfactory. The chapter concludes by summarizing the simulation results that are maximum trips computed with pool completion, maximum rider matches observed by the Exact and Closer characteristics matching, and a continuous increase observed in matching rates plus the number of computed trips for progressing simulations.

CHAPTER 7

CONCLUSION

The chapter of the conclusion provides a final summary of all contributions of the thesis. The conclusion is the final chapter of the thesis and is categorized into two primary sections: (i) Conclusion and (ii) Future Work. The first section of the conclusion also includes the shortcomings of the implemented system. The proposed solutions for the listed shortcomings and the plans to extend the Ride Sharing model are included in future work.

7.1 Conclusion

Initially, the thesis began with an idea of Ride Sharing, which included the basic model of sharing a ride with the characteristics matching layer and the User Threshold Time. The system proved to be feasible after implementing the Ride Sharing model in Phase 1.

Phase 1 included the basic Ride Sharing model, while Phase 2 included the model implementation with Machine Learning algorithms. The system computed parameters like the matching rate, the average number of computed trips, total trip simulation time, the number of trips that completed the pool, and the count of rider matches by the characteristics matching type in both phases for evaluating the model efficiency.

A primary issue in the existing Ride Sharing models is the sudden addition of a rider on a trip. The time to pick up the rider may add significant time for trip completion if the rider is at a location that is too far from the vehicle's location. The solution proposed in the thesis to eliminate the extra rider waiting time is the User Threshold Time (UTT). UTT proved to be one of the most crucial aspects, as riders are only added to a trip if they satisfy minimal trip threshold time. Through UTT, the system makes sure that users do not wait for a long time due to the unexpected addition of a rider on a trip.

One of the priorly stated motives of the thesis is to improve the overall matching rate, which contributes to the expansion of the Car-Pooling services. The results computed in Phase 1 and Phase 2 are satisfactory and reach the expectations of the thesis in terms of the matching rates. With observations from Phase 1 and Phase 2 simulations, it is concluded positively that

the matching rate keeps increasing as the number of traversing riders and trip UTT keeps rising. The system observed a similar result in the case of the total number of computed trips. For both phases, the number of trips keeps increasing for every simulation event. The number of computed trips in Phase 2 is notably higher than the number of computed trips in Phase 1.

The efficiency of the Ride Sharing model is measured by coupling the results of simulation time, matching rate, and the average number of computed trips. The condition with the simulation time is that with the increasing simulation time, the matching rate, and the total number of computed trips should not decrease. From the simulations performed in both phases and based on the observations from the results, it is concluded that the matching rate and the number of computed trips keep increasing with the growing simulation time. With the combination of the results, it is true that the model efficiency increases as the number of traversed riders and the trip UTT increases.

The number of trips completed in Phase 2 is higher than the number of computed trips in Phase 1, even though the number of riders traversed in Phase 2 is less than the number of riders traversed in Phase 1. However, a satisfactory result in both phases is that the percentage of trips that complete the pool is above 85%. Such a measure encourages the usage of Ride Sharing services, indirectly leading to a reduction in the emissions from the vehicle, cutting down the usage of fuel resources, reducing traffic, and using the platform of Ride Sharing to tackle Global Warming.

The accuracy of the SVMs in both phases rounds up to 90%. The predicted characteristics by the SVMs assists in the rider matching and eliminates tedious computations for determining the main characteristics of a rider. Also, the Machine Learning recommendation system assists in tracking the rider's most favored characteristics, and the system tends to group riders having similar favored characteristics. Also, one of the results in Phase 2 states that most of the riders are matched by the Exact or Closer characteristics matching type. As most of the matching is performed using the Closer and Exact matching types, the proposed model pairs maximum riders with similar characteristics that promotes an environment for an interactive and social Car-Pooling journey.

The designed model in the thesis focuses more on users. The approach employed for rider

traversing is the Multiple-Sources-Multiple-Destinations (MSMD) approach, which meets the rider traversing expectation of starting and ending at any location in a journey. Also, using the optimized MSMD path with UTT ensures that riders complete the journey in minimally possible time and do not wait or travel for long on a trip. An additional result to conclude is about the average trip formation time. The average trip formation time in both phases of thesis rounds up to a minute. Also, details like the driver, the time for journey completion, necessary passenger details, and five characteristics based feedback helps understand and serve users better while commuting on a journey. Users get to know the basic profile of the riders through the trip document, which helps achieve a significant purpose of reducing stress, frustrations, disputes, and, most importantly, reducing the social barrier among riders.

Summarizing the section, the Enhanced Ride Sharing Model is feasible and can be employed to increase the usage of Ride Sharing. The thesis expects that the designed Ride Sharing model will help the current transportation companies to enhance their matching methodologies, which will indirectly help humanity to preserve the environment and save natural resources for future generations.

7.2 Shortcomings

Shortcomings define the limitations of a system and are the tasks that are out of the scope of the currently designed system. The solutions to current shortcomings form further implementation plans or future work.

One of the shortcomings in the thesis is the limitations of the zones. Currently, the model functions on the basis of the zones. If the system is set up for a new country or a state, the architecture of the system will require design changes. The system can use geofencing technology for an area without zones. For a new area or state, using the boundary of the area, the areas can be divided into several smaller sections using the Google Maps Geofencing API. The smaller sections may be declared and utilized as zones in the future.

Another critical limitation is the case when a user always wants to travel with other users having identical characteristics on every trip. If a rider wishes to travel only with the users having Exact characteristics, the system may find it challenging to search and accept riders with identical

characteristics on a trip. The system may even find users with exactly matching characteristics from the data-set, but associating riders on the same trip and at the same time may be difficult due to the UTT matching. The riders may want a user with the same characteristics, but the odds of finding users traveling at the same time and on the same trajectory are considerably low.

The limitation of the Google Maps Keys also forms one of the shortcomings of the system. The system experiences substantial transactions of the client requests and server responses in the Ride Sharing model. The application can only work if the system possesses an active Google Maps Key. The number of requests is limited to every Google Maps Key. After using all requests for a Key, no more client responses are received by Google Cloud data server. Hence, if the application functionality ceases after the complete usage of a Key, the only solution is to create a new key and restart the application. A proposed solution to the Google Maps Key request limitation is buying a premium plan of Google Maps, which may be expensive. Another solution is creating an array of Google Maps Keys. If one Key is entirely utilized and the system does not receive any further data, the next available Key in the array can be utilized, and the system's availability is not compromised.

7.3 Future Work

The future work of the Ride Sharing model includes building a full-fledged Android application. The application may provide functionalities like rider registration, broadcasting a request, completing a trip, and rating other users.

Additionally, in the further phases of implementation, the thesis may provide a sophisticated billing model for the riders in the future. The design will focus on solving the rider issues that states all riders are billed equally even though some riders travel for a considerately short distance. The pricing strategy will include billing the riders only for the miles they travel and not for the entire trip.

The future work in the thesis includes providing extra features for drivers. The system may provide a feature to switch a role between a rider and a driver instantly. In this way, any user can be a driver and a rider instantaneously. Additionally, a field named "luggage carrier" may be included in the driver's trip document details. Whenever a user is broadcasting and

specifically possesses luggage, the system may provide a feature that allows riders to specifically broadcast a request for a vehicle that is a "luggage carrier."

Moreover, the thesis may introduce the concept of "Favorites" in the future. A rider may tag certain users as Favorites based on past trips. Whenever the rider is broadcasting for a trip and the riders listed under the Favorites are active and broadcasting, the system may notify all riders and try to pair them up on a trip if they are traveling through the same trajectory. The solution of Favorites is proposed to overcome the shortcoming of the rider who only wants to travel with riders who have exactly matching characteristics.

The future work in the thesis may also extend to the idea of virtual "Badges." The system may provide Badges after specific rider and driver achievements. For example, if a rider performed ten trips, and the vehicle's seating capacity for the ten trips was full, the system may reward a badge that states "Environment Helper." The Badge may also possess points which may help to fill out the costs for the future trips. The Badges will signify how the user is indirectly contributing to the conservation of the environment and fuel resources.

REFERENCES

[1] J. Paek, A. Gaglione, O. Gnawali, M. A. Vieira, and S. Hao, "Advances in mobile networking for iot leading the 4th industrial revolution," *Hindawi Mobile Information Systems*, 2018.

[2] J. S. Y. Song, "Goliath in the age of digitization: Online platform network structure, content, and power dynamics," in *Academy of Management Proceedings*, no. 1. Academy of Management Briarcliff Manor, NY 10510, 2018, p. 17130.

[3] Q.-C. Pham, R. Madhavan, L. Righetti, W. Smart, and R. Chatila, "The impact of robotics and automation on working conditions and employment [ethical, legal, and societal issues]," *IEEE Robotics & Automation Magazine*, vol. 25, no. 2, pp. 126–128, 2018.

[4] G. H. de Almeida Correia, E. Looff, S. van Cranenburgh, M. Snelder, and B. van Arem, "On the impact of vehicle automation on the value of travel time while performing work and leisure activities in a car: Theoretical insights and results from a stated preference survey," *Elsevier Transportation Research Part A: Policy and Practice*, vol. 119, pp. 359–382, 2019.

[5] G. V. De Socio, F. Di Donato, R. Paggi, C. Gabrielli, A. Belati, G. Rizza, M. Savoia, A. Repetto, E. Cenci, and A. Mencacci, "Laboratory automation reduces time to report of positive blood cultures and improves management of patients with bloodstream infection," *Springer European Journal of Clinical Microbiology & Infectious Diseases*, vol. 37, no. 12, pp. 2313–2322, 2018.

[6] B. W. McDaniels, D. A. Harley, and D. T. Beach, "Transportation, accessibility, and accommodation in rural communities," in *Springer Disability and Vocational Rehabilitation in Rural Settings*, 2018, pp. 43–57.

[7] X. Wang, "Preparing the public transportation workforce for the new mobility world," in *Elsevier Empowering the New Mobility Workforce*, 2019, pp. 221–243.

[8] A. J. Neto, Z. Zhao, J. J. Rodrigues, H. B. Camboim, and T. Braun, "Fog-based crime-assistance in smart iot transportation system," *IEEE Access*, vol. 6, pp. 11 101–11 111, 2018.

[9] W. Yan, "Vehicle communication system based on controller-area network bus firewall," Patent, May 14, 2019, US Patent 10,291,583.

[10] Z. Ercan, A. Carvalho, H. E. Tseng, M. Gökaşan, and F. Borrelli, "A predictive control framework for torque-based steering assistance to improve safety in highway driving," *Vehicle System Dynamics*, vol. 56, no. 5, pp. 810–831, 2018.

[11] M. Ostrovsky and M. Schwarz, "Carpooling and the economics of self-driving cars," in *ACM Conference on Economics and Computation*, 2019, pp. 581–582.

[12] T. Z. Jaime Netzer. (2014) A brief (illustrated) history of carpooling. [Online]. Available: https://www.thezebra.com/insurance-news/406/history-of-carpooling/

[13] L. Cook, "The energy-crisis-going, going, but not quite gone," *IEEE Transactions on Industry Applications*, vol. 17, no. 6, pp. 548–551, 1981.

[14] M. Maksimovic, "Greening the future: Green internet of things (g-iot) as a key technological enabler of sustainable development," in *Springer Internet of Things and Big Data Analytics Toward Next-generation Intelligence*, 2018, pp. 283–313.

[15] "Highway statistics," Policy and Governmental Affairs Office of Highway Policy Information, 2017. [Online]. Available: https://www.fhwa.dot.gov/policyinformation/statistics/2017/

[16] X. Wang, Z. Ning, X. Hu, L. Wang, B. Hu, J. Cheng, and V. C. Leung, "Optimizing content dissemination for real-time traffic management in large-scale internet of vehicle systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1093–1105, 2018.

[17] X. Chen, Y. Zou, J. Tang, Y. Peng, L. Wu, and Y. Jiang, "Analysing the impact of traffic incidents on the travel time reliability of freeway high-occupancy vehicle lanes," *Hindawi Discrete Dynamics in Nature and Society*, 2018.

[18] H. Huang, D. Bucher, J. Kissling, R. Weibel, and M. Raubal, "Multimodal route planning with public transport and carpooling," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[19] N. A. of Engineering and N. A. of Engineering, *Energy: Production, Consumption, and Consequences*, J. L. Helm, Ed. Washington, DC: The National Academies Press, 1990.

[20] C. T. Tugcu, I. Ozturk, and A. Aslan, "Renewable and non-renewable energy consumption and economic growth relationship revisited: evidence from g7 countries," *Elsevier Energy economics*, vol. 34, no. 6, pp. 1942–1950, 2012.

[21] A. do Nascimento Rocha, L. S. Candido, J. G. Pereira, C. A. M. Silva, S. V. da Silva, and R. M. Mussury, "Evaluation of vehicular pollution using the trad-mcn mutagenic bioassay with tradescantia pallida (commelinaceae)," *Elsevier Environmental pollution*, vol. 240, pp. 440–447, 2018.

[22] V. M. de Lira, R. Perego, C. Renso, S. Rinzivillo, and V. C. Times, "Boosting ride sharing with alternative destinations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2290–2300, 2018.

[23] J. S. Apte, K. P. Messier, S. Gani, M. Brauer, T. W. Kirchstetter, M. M. Lunden, J. D. Marshall, C. J. Portier, R. C. Vermeulen, and S. P. Hamburg, "High-resolution air pollution mapping with google street view cars: exploiting big data," *ACS Environmental Science & Technology*, vol. 51, no. 12, pp. 6999–7008, 2017.

[24] M. Szyszkowicz, T. Kousha, J. Castner, and R. Dales, "Air pollution and emergency department visits for respiratory diseases: a multi-city case crossover study," *Elsevier Environmental Research*, vol. 163, pp. 263–269, 2018.

[25] S. Carrese, T. Giacchetti, S. M. Patella, and M. Petrelli, "Real time ridesharing: Understanding user behavior and policies impact: Carpooling service case study in Lazio region, Italy," in *5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2017, pp. 721–726.

[26] M. Mallus, G. Colistra, L. Atzori, M. Murroni, and V. Pilloni, "A persuasive real-time carpooling service in a smart city: A case-study to measure the advantages in urban area," in *20th IEEE Conference on Innovations in Clouds, Internet and Networks (ICIN)*, 2017, pp. 300–307.

[27] Y. Wang, J. Gu, S. Wang, and J. Wang, "Understanding consumers' willingness to use ride-sharing services: The roles of perceived value and perceived risk," *Elsevier Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 504–519, 2019.

[28] C. Boldrini, R. Bruno, and M. Conti, "Characterising demand and usage patterns in a large station-based car sharing system," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 572–577.

[29] S.-C. Huang, M.-K. Jiau, and C.-H. Lin, "A genetic-algorithm-based approach to solve carpool service problems in cloud computing," *IEEE Transactions on intelligent transportation systems*, vol. 16, no. 1, pp. 352–364, 2014.

[30] S. D. Contreras and A. Paz, "The effects of ride-hailing companies on the taxicab industry in Las Vegas, Nevada," *Elsevier Transportation Research Part A: Policy and Practice*, vol. 115, pp. 63–70, 2018.

[31] T. Teubner and C. M. Flath, "The economics of multi-hop ride sharing," *Springer Business & Information Systems Engineering*, vol. 57, no. 5, pp. 311–324, 2015.

[32] Y. Duan, T. Mosharraf, J. Wu, and H. Zheng, "Optimizing carpool scheduling algorithm through partition merging," in *IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[33] J. Cramer and A. B. Krueger, "Disruptive change in the taxi business: The case of Uber," *American Economic Review*, vol. 106, no. 5, pp. 177–82, 2016.

[34] X.-Y. Zhao and Q. Su, "Existing issues of ride sharing company operation and sharing economy in China: Uber case analysis," in *5th Annual International Conference on Management, Economics and Social Development (ICMESD 2019)*. Atlantis Press, 2019.

[35] K. Samuel, S. Alkire, D. Zavaleta, C. Mills, and J. Hammock, "Social isolation and its relationship to multidimensional poverty," *Oxford Development Studies*, vol. 46, no. 1, pp. 83–97, 2018.

[36] M. G. Campana, F. Delmastro, and R. Bruno, "A machine-learned ranking algorithm for dynamic and personalised car pooling services," in *19th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1856–1862.

[37] A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, "Lightweight machine learning-based approach for supervision of fitness workout," in *IEEE Sensors Applications Symposium (SAS)*, 2019, pp. 1–6.

[38] S. Li, F. Fei, D. Ruihan, S. Yu, and W. Dou, "A dynamic pricing method for carpooling service based on coalitional game analysis," in *IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2016, pp. 78–85.

[39] J. Jamal, A. Rizzoli, R. Montemanni, and D. Huber, "Tour planning and ride matching for an urban social carpooling service," in *MATEC Web of Conferences*, vol. 81. EDP Sciences, 2016, p. 04010.

[40] Gett, "Juno by Gett," 2019, [Online, accessed October 10, 2019]. [Online]. Available: https://gett.com/juno/

[41] Y. He, J. Ni, X. Wang, B. Niu, F. Li, and X. Shen, "Privacy-preserving partner selection for ride-sharing services," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5994–6005, 2018.

[42] U. T. Inc, "Uber," 2019, [Online, accessed October 10, 2019]. [Online]. Available: https://www.uber.com/

[43] Z. Li, Y. Hong, and Z. Zhang, "An empirical analysis of on-demand ride sharing and traffic congestion," in *AIS International Conference on Information Systems*, 2016.

[44] I. Lyft, "Lyft," 2019, [Online, accessed October 10, 2019]. [Online]. Available: https://www.lyft.com/

[45] W. Mobile, "Waze carpool," 2019, [Online, accessed October 10, 2019]. [Online]. Available: https://www.waze.com/

[46] G. Rodriguez, "Autonomous vehicles and unmanned aerial systems: Data collection and liability [leading edge]," *IEEE Technology and Society Magazine*, vol. 38, no. 3, pp. 14–16, 2019.

[47] S. Shaheen and A. Cohen, "Shared ride services in North America: definitions, impacts, and the future of pooling," *Transport Reviews*, vol. 39, no. 4, pp. 427–442, 2019.

[48] M. Tang, S. Ow, W. Chen, Y. Cao, K.-w. Lye, and Y. Pan, "The data and science behind grabshare carpooling," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2017, pp. 405–411.

[49] W. He, G. Yan, and L. Da Xu, "Developing vehicular data cloud services in the iot environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, 2014.

[50] WTOP. (2019) Top carpool percentages by city. [Online]. Available: https://wtop.com/council-of-governments-guaranteed-ride-home/2019/04/top-carpool-percentages-by-city/

[51] M. Samy and A. M. Elkorany, "Using semantic features for enhancing car pooling system," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018, pp. 702–708.

[52] J. Koebler, "Why everyone hates UberPool," May 2016. [Online]. Available: https://www.vice.com/en_us/article/4xaa5d/why-drivers-and-riders-hate-uberpool-and-lyft-line

[53] C. Huang, D. Zhang, Y.-W. Si, and S. C. Leung, "Tabu search for the real-world carpooling problem," *Springer Journal of Combinatorial Optimization*, vol. 32, no. 2, pp. 492–512, 2016.

[54] U. T. Inc., "How does Uber match riders with drivers?" 2019, [Accessed November 1, 2019]. [Online]. Available: https://marketplace.uber.com/matching

[55] I. Grgurevic, D. Perakovic, I. Forenbacher, and T. Milinović, *Application of the Internet of Things Concept in Carsharing System*, 10 2015, pp. 401–414.

[56] M. Zhu, X.-Y. Liu, and X. Wang, "An online ride-sharing path-planning strategy for public vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–12, 2018.

[57] C.-H. Lin, M.-K. Jiau, and S.-C. Huang, "A cloud computing framework for real-time carpooling services," in *6th IEEE International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM)*, 2012, pp. 266–271.

[58] Y. Lai, F. Yang, L. Zhang, and Z. Lin, "Distributed public vehicle system based on fog nodes and vehicular sensing," *IEEE Access*, vol. 6, pp. 22 011–22 024, 2018.

[59] M. Li, L. Zhu, and X. Lin, "Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing," *IEEE Internet of Things Journal*, 2018.

[60] R. Hasan, A. H. Bhatti, M. S. Hayat, H. M. Gebreyohannes, S. I. Ali, and A. J. Syed, "Smart peer car pooling system," in *3rd IEEE MEC International Conference on Big Data and Smart City (ICBDSC)*, 2016, pp. 1–6.

[61] Y. Huang, F. Bastani, R. Jin, and X. S. Wang, "Large scale real-time ridesharing with service guarantee on road networks," *Proceedings of the VLDB Endowment*, vol. 7, no. 14, pp. 2017–2028, 2014.

[62] K. Kalogirou, N. Dimokas, M. Tsami, and D. Kehagias, "Smart mobility combining public transport with carpooling: An ios application paradigm," in *IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2018, pp. 1271–1278.

[63] N. Agatz, A. L. Erera, M. W. Savelsbergh, and X. Wang, "Dynamic ride-sharing: A simulation study in metro Atlanta," *Elsevier Procedia-Social and Behavioral Sciences*, vol. 17, pp. 532–550, 2011.

[64] Math and Science, "What is variance in statistics? learn the variance formula and calculating statistical variance!" Youtube. [Online]. Available: https://www.youtube.com/watch?v=sOb9b_AtwDg

[65] X. Huang and H. Peng, "Efficient mobility-on-demand system with ride-sharing," in *21st IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3633–3638.

[66] X. Fang, B.-M. Hodge, L. Bai, H. Cui, and F. Li, "Mean-variance optimization-based energy storage scheduling considering day-ahead and real-time lmp uncertainties," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7292–7295, 2018.

[67] C. Luo, J. Zhan, X. Xue, L. Wang, R. Ren, and Q. Yang, "Cosine normalization: Using cosine similarity instead of dot product in neural networks," in *Springer International Conference on Artificial Neural Networks*, 2018, pp. 382–391.

[68] N. Dehak, R. Dehak, J. R. Glass, D. A. Reynolds, P. Kenny *et al.*, "Cosine similarity scoring without score normalization techniques." in *Odyssey*, 2010, p. 15.

[69] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Springer Asian conference on computer vision*, 2010, pp. 709–720.

[70] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "Svms modeling for highly imbalanced classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 281–288, 2008.

[71] J. A. Sáez, B. Krawczyk, and M. Woźniak, "Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets," *Elsevier Pattern Recognition*, vol. 57, pp. 164–178, 2016.

[72] L. Liang, H. Ye, and G. Y. Li, "Toward intelligent vehicular networks: A machine learning framework," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 124–135, 2018.

[73] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu, and M. Wu, "Machine learning for vehicular networks: Recent advances and application examples," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 94–101, 2018.

[74] L. Wang, X. Geng, X. Ma, D. Zhang, and Q. Yang, "Ridesharing car detection by transfer learning," *Elsevier Artificial Intelligence*, vol. 273, pp. 1–18, 2019.

[75] N. Shahane, M. Kaul, and Y. Zheng, "Exploratory analysis of Chicago taxi rides," in *20th ACM Annual SIG Conference on Information Technology Education*, 2019, pp. 158–158.

[76] S. Jiang, W. Chen, Z. Li, and H. Yu, "Short-term demand prediction method for online car-hailing services based on a least squares support vector machine," *IEEE Access*, vol. 7, pp. 11 882–11 891, 2019.

[77] S. Han, C. Qubo, and H. Meng, "Parameter selection in svm with rbf kernel function," in *IEEE World Automation Congress*, 2012, pp. 1–4.

[78] L. Ma, T. Fu, T. Blaschke, M. Li, D. Tiede, Z. Zhou, X. Ma, and D. Chen, "Evaluation of feature selection methods for object-based land cover mapping of unmanned aerial vehicle imagery using random forest and support vector machine classifiers," *ISPRS International Journal of Geo-Information*, vol. 6, no. 2, p. 51, 2017.

[79] S. Jiang, R. Hartley, and B. Fernando, "Kernel support vector machines and convolutional neural networks," in *IEEE Digital Image Computing: Techniques and Applications (DICTA)*, 2018, pp. 1–7.

[80] A. Swami, "Impact of automobile induced air pollution on road side vegetation: A review," *ESSENCE Int. J. Env. Rehab. Conserv. IX (1)*, pp. 101–116, 2018.

[81] NYC, "NYC open data," 2019. [Online]. Available: https://data.cityofnewyork.us/Transportation/NYC-Taxi-Zones/d3c5-ddgc

APPENDIX A

APPROVAL LETTER

MARSHALL UNIVERSITY
www.marshall.edu

Office of Research Integrity

November 22, 2019

Govind Yatnalkar
WAEC – Room 3107
College of Information, Technology, and Engineering
Marshall University

Dear Govind:

This letter is in response to the submitted thesis abstract entitled *"A Machine Learning Recommender Model for Ride Sharing Based on Rider Characteristics and User Threshold Time."* After assessing the abstract, it has been deemed not to be human subject research and therefore exempt from oversight of the Marshall University Institutional Review Board (IRB). The Code of Federal Regulations (45CFR46) has set forth the criteria utilized in making this determination. Since the information in this study does not involve human subjects as defined in the above referenced instruction, it is not considered human subject research. If there are any changes to the abstract you provided then you would need to resubmit that information to the Office of Research Integrity for review and a determination.

I appreciate your willingness to submit the abstract for determination. Please feel free to contact the Office of Research Integrity if you have any questions regarding future protocols that may require IRB review.

Sincerely,

Bruce F. Day, ThD, CIP
Director

APPENDIX B
ACRONYMS

API Application Programming Interface. 6

ERSM Enhanced Ride Sharing Model. ix, 2, 5, 11

fn false negative. 51, 52

fp false positive. 51, 52

HOV High Occupancy Vehicle. 2, 3

IoT Internet of Things. 9, 12

KNN K-Nearest Neighbours. 17

ML Machine Learning. 6

MSMD Multiple-Sources-Multiple-Destinations. 11, 13–15, 18, 71

MSSD Multiple-Sources-Same-Destination. 13

NYC New York City. 10, 19

RBF Radial Bias Function. 17

RMSE Root Mean Square Error. x, 54, 55

SSMD Same-Source-Multiple-Destinations. 13

SSSD Same-Source-Same-Destination. 13

SVM Support Vector Machine. 7, 17, 40

tn true negative. 51, 52

tp true positive. 51, 52

UTT User Threshold Time. 5, 6, 11, 18