RideShare using Degrees of Separation: A Social Network-Based Approach

Gokul Garikipati
San Jose State University
RideShare using Degrees of Separation: A Social Network-Based Approach

A Project
Presented to
The Faculty of the Department of Computer Science
San José State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Gokul Garikipati
May 2023
RideShare using Degrees of Separation: A Social Network-Based Approach

by

Gokul Garikipati

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

May 2023

Dr. Navrati Saxena Department of Computer Science
Dr. Katerina Potika Department of Computer Science
Abhishek Roy Director, Mediatek Inc, USA
ABSTRACT

Conventional ride-sharing services, such as Lyft and Uber, routinely match drivers with riders based on their proximity to each other, using GPS coordinates and mapping technology. The application then calculates the cost of the ride based on factors such as distance traveled and time spent in the car. The concept of six degrees of separation suggests that a maximum of 6 steps or relationships can connect any two individuals in the world. This idea could be applied to a ride-share service to provide a more personalized and efficient experience for users. Instead of just matching riders with drivers based on proximity, the app could take into account the social connections between users. The motive of this paper is to develop an algorithm that is capable of allocating rides based on the strength of ties.

Index Terms: Ride-share, mapping technology, degrees of separation, social connections, strength of ties
ACKNOWLEDGMENTS

I want to express my deepest appreciation to my project advisor, Dr. Navrati Saxena, for her expertise, continuous guidance, and valuable support throughout this project and the rest of my academic journey. I’m also immensely grateful to Dr. Abhishek Roy for his meticulous and detailed review of my work, using his extensive field expertise and time. His insightful feedback and suggestions have greatly improved the quality of my work. I would also like to extend my gratitude to Dr. Katerina Potika for her invaluable input and suggestions that have played a critical role in shaping my work. In addition, I would like to take this opportunity to thank my family and friends for their unwavering support and encouragement throughout my academic journey. Their love and support have been a constant source of motivation, and I’m grateful for their belief in me.
TABLE OF CONTENTS

CHAPTER

1 Introduction

1.1 Ridesharing Apps and the need

1.2 History - Uber, Lyft

1.3 Challenges based on Trust

1.4 Project Statement

2 Background

2.1 Graph matching algorithms

2.2 Six Degrees of Separation

2.3 Social Media Clustering and Linking

3 Related Works

4 Dataset

5 Proposed Solution

5.1 Design

5.2 Neo4j and Graph Search

6 Implementation

6.1 User Web Application

   6.1.1 Create Account

   6.1.2 Login - Ride Search

   6.1.3 Matching Algorithm

6.2 Testing

   6.2.1 Unit Testing

   6.2.2 Integration testing
6.2.3 Functional Testing .......................................................... 23
6.2.4 Performance Testing ...................................................... 24
6.2.5 Security Testing ............................................................ 24
6.3 Deployment ........................................................................ 25

7 Experiments and Results ...................................................... 27
7.1 Peak vs normal hours ...................................................... 28
7.2 Trade-off: Degrees of Separation vs. Acceptance Percentage ... 30
7.3 Relationship between Ride Match Time and Time of Day .......... 32

8 Analysis ............................................................................ 33

9 Ethics in AI ......................................................................... 35

10 Conclusion ......................................................................... 37

LIST OF REFERENCES ............................................................. 39

APPENDIX
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BFS vs DFS</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Degrees of Separation</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Flowchart of the Application</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>BFS vs Bidirectional</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Login Page</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>Ride Search page</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>Ride matching based on DoS</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>User graph and Connections</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>Drivers graph</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>Social connections in User Graph</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>FreeDNS Domain Registration</td>
<td>26</td>
</tr>
<tr>
<td>12</td>
<td>Microsoft Azure Virtual Machine Configuration</td>
<td>26</td>
</tr>
<tr>
<td>13</td>
<td>Rides vs Time Frames</td>
<td>29</td>
</tr>
<tr>
<td>14</td>
<td>Ride Acceptance vs DoS</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>Users connected at various Dos</td>
<td>31</td>
</tr>
<tr>
<td>16</td>
<td>Match time vs Time Frames</td>
<td>32</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th></th>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Time Frames</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>Cost Savings vs Ride Type</td>
<td>33</td>
</tr>
</tbody>
</table>
CHAPTER 1

Introduction

1.1. Ridesharing Apps and the need

Ride-sharing apps have become essential in our fast-paced urban world for several reasons. First of all, they provide users with convenient transportation in areas with limited public transit options or when traditional means may not be readily available or suitable. Users simply tap their smartphones to request rides - the driver arrives right at their location within minutes!

Second, ride-sharing apps provide flexibility for both drivers and riders; drivers can earn income on their own schedule while riders have various options such as carpooling, shared rides or luxury vehicles depending on their preferences and budgets. Ride-sharing applications provide convenient transportation solutions for commuters, travelers and those without private cars - whether commuting daily, making travel plans or without access to private transport themselves. Ride-sharing apps also often provide transparent pricing, real-time tracking and safety features like driver and vehicle information, GPS tracking and in-app feedback to provide a safer transportation experience compared to traditional taxi services. Ride-sharing has quickly become an indispensable component of modern life; providing convenience, flexibility and safety benefits in today's fast paced society.

1.2. History - Uber, Lyft

Uber and Lyft, two prominent ride-sharing services, have both made headlines for their innovative approaches and disruptive impact in the transportation industry. Uber first emerged as an executive car service in 2009 before rapidly growing traction through its user-friendly app that allowed people to request rides electronically while paying electronically via credit or debit card. Meanwhile, Lyft emerged in 2012 to address community driven ride-sharing and offered rides from regular drivers using their cars; both companies quickly expanded quickly in multiple markets around the globe disrupting
traditional taxi cab operations with their innovative offerings while challenging existing regulations with regulatory authorities worldwide.

Over the years, Uber and Lyft have rapidly developed their services, adding carpooling, food delivery, and electric scooters/bikes as options and legal battles over safety/labor rights issues to market dominance disputes; yet still remain highly popular choices due to user-friendly apps with transparent pricing structures, user convenience features and global reach.

As part of our project, Uber and Lyft serve as a basis for creating an algorithm that takes account of social ties when allocating rides. Uber and Lyft currently rely on proximity-based matching while our proposed algorithm uses 6 degrees of separation as its foundation to deliver more personalized and efficient ride-sharing experiences.

By taking into account social ties between users, our algorithm could enhance safety, foster interaction, and offer more tailored user experiences. Led by Uber and Lyft's success and lessons learned over their history, our project hopes to push the envelope of ride-sharing innovation while contributing to its rapid development as an industry.

1.3. Challenges based on Trust

Trust is at the core of ridesharing app adoption and success, and users need to have faith that drivers and vehicles provided through an app are safe. Incidents of unsafe or fraudulent driver behavior have caused users to express concern, prompting greater regulatory action within the industry. Conducting thorough background checks, vehicle inspections, and safety protocols is paramount when building trust among users.

Issues arise with regards to app reliability and accuracy in terms of information provided to users by an application. Users rely on such applications for precise data on driver location, estimated arrival times, trip fares etc that is up-to-date and accurate at all times; any discrepancies or inaccuracies in such information could potentially erode trust resulting in poor user experiences and result in negative user ratings for your service provider. Maintaining high levels of accuracy and reliability within such apps require robust technology infrastructures such as data management solutions with stringent
quality control measures in order to guarantee user experience.

Privacy can also be an issue in ride-sharing apps, with users needing to trust that their personal information like location, payment details and interactions with drivers is safe from breach or misuse by drivers or third-parties. Data breaches or misuse can lead to privacy violations and consequently lead to lost trust from other users - therefore having strong data encryption, authentication and privacy policies implemented is paramount in protecting user data while building rapport among riders and drivers alike.

Trust among users themselves is of critical importance in our proposed algorithm that considers social connections for ride allocation. They need to trust that their social ties will be respected and not used improperly by anyone; protecting user privacy and consent while using connections solely for ride allocation can go a long way toward building that trust among users.

These concerns should be addressed by employing robust safety measures such as assuring accuracy of information and protecting user privacy while respecting social connections that they can build and uphold among their user community resulting in greater adoption rates and customer satisfaction.

1.4. Project Statement

Existing ridesharing apps face challenges in matching rides between passengers efficiently, leading to underutilized rides and increased travel time. Ridesharing apps often rely on pick-up/drop-off locations as the basis for pairing rides, leading them to overlook potential matches who share similar routes but who may live further away. This could result in missed opportunities when potential matches may exist but none were immediately close at hand. This could result in increased costs, longer travel times, inefficient resource usage, and ultimately compromise the sustainability and effectiveness of ridesharing services. The lack of trust among users, privacy concerns, and data protection issues, and the need for an effective algorithm that matches rides based on social connections is also a serious barrier to ridesharing apps being effective.
Building a graph database using social network profiles requires collecting and processing personal data about users such as their location, profession, and interests. To meet user privacy expectations and comply with relevant data protection laws while giving them the option of opting out at any time is of vital importance in building user trust and maintaining the integrity of carpooling services.

Furthermore, the algorithm used to match rides based on social interactions presents another difficulty. Formulating an algorithm that makes use of 6 degrees of separation to successfully match riders for ridesharing involves carefully considering factors like social connections, frequency and duration of communication between users as well as the level of separation between them. Machine learning techniques like supervised, unsupervised, or reinforcement learning may be used to optimize an algorithm, yet difficulties may still arise in accurately predicting user preferences and behavior. Overcoming these algorithmic challenges is essential to creating a smarter carpooling system that maximizes social connections for ridesharing apps.
CHAPTER 2

Background

2.1 Graph matching algorithms

1. Dijkstra's Algorithm: It is a shortest-path algorithm that finds the path between two nodes in a weighted graph. It measures closeness by calculating the minimum distance from one node to all other nodes in the graph.

2. Breadth-First Search (BFS): It is a traversing algorithm that visits all the nodes of a graph at a given depth before moving on to the next level. It measures closeness by counting the number of levels or edges between two nodes.

3. Depth-First Search (DFS): It is also a traversing algorithm that visits all the nodes of a graph by exploring as far as possible along each branch before backtracking. It measures closeness by calculating the number of edges in the shortest path between two nodes.

![Figure 1. BFS vs DFS](image-url)
4. **A* Search Algorithm**: It is an informed search algorithm that uses a heuristic function to estimate the distance between the current node and the goal node. It measures closeness by calculating the estimated distance from one node to the goal node.

5. **Betweenness Centrality**: It is a measure of the importance of a node in a network. It calculates the number of shortest paths between pairs of nodes that pass through a given node, and measures closeness based on the number of times a node appears on the shortest paths between other nodes.

6. **Eigenvector Centrality**: It is a measure of the influence of a node in a network. It measures closeness by calculating the sum of the centrality measures of its neighbors, giving more weight to nodes that are connected to other important nodes in the graph.

### 2.2 Six Degrees of Separation

The concept of Six Degrees of Separation first originated from Hungarian author Frigyes Karinthy's 1929 hypothesis that any two individuals in the world are connected through no more than six mutual acquaintances or "links" in a social network [13]. Later popularized by social psychologist Stanley Milgram's 1960 "Small World" experiment where participants successfully passed messages to target people they didn't know through approximately six intermediaries on average, thus giving rise to its namesake phrase: Six Degrees of Separation.

The Six Degrees of Separation concept has gained wide recognition and has since become a topic of research in fields as diverse as social network analysis, sociology, psychology and communication studies. It has been applied to examine areas including social
influence, diffusion of information, disease transmission and formation of social ties - among many others. Furthermore, algorithms designed specifically for social networking platforms, recommendation systems or matchmaking services use this concept extensively as well. Eventually this has led to greater insight into how social networks function as well as understanding how information, behaviors and influences spread within societies - leading to better insight than before into how individuals connect within society - providing greater understanding.

Figure 2. Degrees of Separation

2.3 Social Media Clustering and Linking

Social media platforms have quickly become ubiquitous, connecting billions of people around the globe. Individuals use them to create profiles, share information and interact with one another, producing vast amounts of data that can be utilized for various purposes. One key application of social media data analysis involves clustering and linking people - gathering together individuals with similar attributes or behaviors into clusters before identifying relationships between them and uncovering any connections or links that exist
between them. In his research Lawrence [12] applies the Ant colony optimization technique on Facebook to prove that the concept of six degrees of separation applies to the real world too.

Clustering techniques are used to organize individuals who share similar characteristics, behaviors, or interests into groups. Clustering may employ machine learning algorithms, statistical methods, graph-based approaches or any combination thereof in order to recognize patterns in social media data. For instance, individuals engaging in similar discussions, using similar hashtags or following similar profiles may be clustered together if they share common interests or affiliations; clustering can also be employed for targeted marketing, content recommendation and community detection on various social media platforms.

Linking individuals on social media involves identifying connections or relationships among them, such as friendships, collaborations or interactions. Linking techniques such as graph-based algorithms, social network analysis or link prediction methods to detect patterns in social media data that reveal relationships. For instance, individuals who frequently interact with each other, share common friends or mention each other may be linked together as evidence of social ties or relationships; such linking techniques provide insights into the structure, influence and dynamics of social networks on these platforms - it even has applications in areas like recommendation systems, fraud detection and influencer marketing.
Social media clustering and linking techniques have gained widespread attention for their ability to analyze the complex and ever-evolving nature of social media networks. They're utilized across various fields - marketing, advertising, political campaigns, public health research and social sciences alike - for analysis and understanding user behavior, preferences, interactions and interactions on these platforms. But challenges such as data privacy concerns, algorithm bias issues and scalability must also be overcome in order to ensure ethical use of such data when clustering and linking individuals together.
CHAPTER 3

Related Works

The concept of six degrees of separation has captivated researchers and the general public alike, as it highlights the interconnectedness and small-world nature of social networks. Conducting studies on the concept of six degrees of separation in the context of ridesharing and other domains allows researchers to explore novel and innovative approaches to optimize transportation systems, improve user experience, and promote sustainable transportation behavior. It also adds to the body of knowledge in the field of complex networks, transportation planning, and human behavior, contributing to the advancement of research and understanding in these areas.

One notable study in this field was conducted by Xia et al. (2019), who proposed a carpool matching model that integrated both social and route networks. The model utilized a graph-based representation to match drivers and riders based on their preferences and travel patterns. The study found that incorporating social network information into the matching algorithm significantly improved the matching efficiency and reduced travel time, demonstrating the potential of utilizing the concept of six degrees of separation in ridesharing.

Another relevant study was conducted by Kuwahara et al. (2022), who proposed an evaluation framework for an efficient commuting carpool program. The framework considered user preferences, matching algorithms, and social and spatial context to optimize the carpooling process. The study suggested that incorporating multiple factors, including social network information, into the matching algorithm could lead to an improved user experience and promote the adoption of carpooling programs.
Furthermore, Wang et al. (2022) presented an algorithm for integrating multi-proximity for trust-based group recommendation in ridesharing. The algorithm utilized both social and spatial proximity to recommend groups of riders and drivers with similar travel patterns and preferences. The study found that incorporating trust into the recommendation algorithm could improve matching efficiency and user satisfaction, highlighting the potential of utilizing social network information in the context of six degrees of separation.

In addition to ridesharing, the concept of six degrees of separation has also been explored in other contexts. For instance, Kardes et al. (2012) conducted a study to explore the concept of six degrees of separation among US researchers. The study found that most researchers could be connected in six steps or less, indicating that the concept could potentially be applied to ridesharing among individuals who are not directly connected on social media platforms. Lunze (2016) presented a study on the six degrees of separation in multi-agent systems, which aimed to improve coordination and communication among agents by utilizing the concept of six degrees of separation.

Lawrence and Latha (2015) utilized the six degrees of separation theory to analyze Facebook networks using ant colony optimization, with the aim of identifying the most influential individuals and their role in shaping the network structure. Ke (2010) proposed a social networking services system based on the six degrees of separation theory and damping factors, which aimed to improve connectivity among individuals and facilitate information exchange.
Overall, the prior work in the field of six degrees of separation and ridesharing has provided valuable insights into the potential benefits of incorporating social network information into the matching algorithms. These studies have demonstrated the effectiveness of utilizing the concept of six degrees of separation in improving the efficiency and user experience of ridesharing programs. However, further research and development are needed to explore the full potential of this concept in real-world ridesharing scenarios.
CHAPTER 4

Dataset

The dataset used for our ridesharing application came from The Social Computing Data Repository at Arizona State University and comprises two CSV files, "nodes.csv" and "edges.csv". "Nodes.csv" contains Twitter IDs of users, serving as unique identifiers for nodes in our Neo4j graph database; while "Edges.csv" represents connections between them based on whether one follows another on Twitter.

But in order to implement login and ride request functionality in our application, we require additional user details - such as email addresses - than are provided in the dataset, including their Twitter IDs and no personal data such as addresses or contact information. In order to do this effectively we will utilize Faker, a Python library which provides realistic fake data generation capabilities for various uses.

The Faker library provides various methods for generating fake data such as email addresses, names, phone numbers and addresses. We can use these methods to generate fake information for each of the users in "nodes.csv", assigning each generated record with their Twitter ID as they generate fake info for all nodes in this file.

As it's important to keep in mind that generated data does not represent actual user information, it should only be used as part of our ridesharing application and should not be taken as actual personal data.
CHAPTER 5

Proposed Solution

5.1 Design

Neo4j provides us with an effective and scalable method to store our ridesharing application's data, with its nodes representing users and edges representing connections. This allows us to model user relationships in an efficient and scalable fashion - essential when matching and linking users based on social or spatial proximity.

Node.js is an extremely versatile JavaScript runtime that we are employing on the backend to construct the server-side logic of our app. Node provides an efficient and scalable environment for creating server-side logic applications, with its event-driven I/O model making it well suited to handle real-time processing of Neo4j graph database queries as well as RESTful API implementation for various functions such as user registration, ride creation/marriage matching/communication between users.

On the frontend, we are using React - a popular JavaScript library for building user interfaces - to develop its UI components for our application. React's component-based architecture enables us to build reusable UI components, making the development process more modular and efficient. React components help create responsive and interactive user experiences, providing riders and drivers alike a seamless user experience when using our ridesharing application.

As for hosting, we will co-locate both Neo4j graph database and application server on one server to maximize communication between frontend and backend components of the application, reducing latency and ensuring smooth data flow. Furthermore, we will follow Agile methodology in
our development process - emphasizing iterative and incremental development, frequent feedback loops, collaboration among team members, as well as agile-like changes as needed based on user feedback and changing needs of the app.

![Flowchart of the Application](image)

**Figure 3. Flowchart of the Application**

### 5.2 Neo4j AND GRAPH SEARCH

Once our fake data has been generated and combined with the original dataset, we can use it to populate a Neo4j graph database with user information that includes generated email addresses for login and rideshare purposes in our application. Utilizing the shortest path algorithm, our application can query this graph database in search of two users who share similar follow relationships on Twitter; using this technique helps identify potential rideshare partners more likely than not being in close proximity to one another.
Cypher queries used by Neo4j to find the shortest path are an iteration of Breadth-First Search (BFS). Neo4j uses bi-directional BFS algorithm in their shortestPath function to locate paths between nodes in their graph database and find its shortest paths.

Bi-directional BFS begins searching from both source ('p1') and target nodes simultaneously, traversing their neighbors in both forward and backward directions simultaneously. It keeps track of visited nodes by maintaining two queues dedicated to forward and backward searches respectively, until both forward and backward searches reach a common node, signifying that a shorter path has been found.

Neo4j's bi-directional BFS algorithm used by its shortestPath function provides a fast and effective means for finding shortest paths in large graph databases, by exploring both directions simultaneously and rapidly converging towards it compared with traditional BFS or Depth-First Search (DFS) algorithms. This makes Neo4j an attractive solution for applications such as ridesharing or social network analysis that rely heavily on graph analysis.

Figure 4. BFS vs Bidirectional Search
CHAPTER 6

Implementation

6.1 User web application

The web application development for this project entails a robust technical process. The application features a create account page where users can register and provide their details, including an email ID. This email ID is used to create a node in the Neo4j graph database, which serves as a central repository for storing all the user details. The application also offers the option for users to log in with their Gmail account, which is utilized for authentication and retrieving user details from Google.

Initially, the plan was to utilize the retrieved email ID or the one entered during registration to fetch friends' details from social networking platforms like Facebook, Instagram, and Twitter, and establish connections between users in the graph database based on these friendships. However, due to limitations imposed by social networking platforms in granting developer account requests, this approach may not be feasible. Consequently, the application may need to solely rely on the email ID entered during registration or retrieved from Google for user authentication, skipping the step of retrieving friends' details from social networking platforms.
The login page of the application enables users to login and search for a ride. The search page provides a seamless integration with Google Maps API, allowing users to conveniently select the pickup and drop-off locations on the map. Based on the search criteria, the application scans for parallel ride requests from other nearby users whose requests may align with the initial user's request, facilitating potential ride-sharing matches.
Figure 6. Ride Search page

Subsequently, the application queries the Neo4j graph database to determine if a shortest path exists between the two users who have matching ride requests. If a path is found, the application calculates the number of nodes (i.e., degrees of separation) between the two users, leveraging the concept of "six degrees of separation," which suggests that theoretically, there should always exist a path between any two users in less than six steps.
Figure 7. Ride matching based on DoS

Figure 8. User graph and Connections
In cases where no path is found between the two users, it indicates that there is insufficient data to determine the path. In such scenarios, the application provides the option for the user to travel solo or opt to travel with unknown users. When a ride is completed with a stranger, a connection is formed between the two users in the graph database for future references, thereby strengthening the algorithm and improving the efficiency of the matching process.

Furthermore, upon completion of a ride, the application establishes a connection between the driver and the user in the graph database. This helps to enhance the efficiency of the matching algorithm for future references, as it allows for better tracking of past ride-sharing matches and potential future matches based on the history of completed rides and connections formed between users and drivers.

Figure 9. Driver graph
In summary, the web application development entails various technical components, including user registration and authentication, seamless integration with Google Maps API for location selection, scanning and matching of ride requests, querying the Neo4j graph database for shortest path determination, handling cases with no path, forming connections between users, and improving the matching algorithm for future references.

6.2 Testing

Testing the web application involves multiple stages and various potential test cases to ensure its functionality and reliability. The technical process involved in testing the web application can be summarized as follows:

Unit Testing: This involves testing individual components or modules of the application in isolation to ensure they function as expected.
• Testing user registration and authentication process, including email verification.

• Testing location selection feature using Google Maps API and verifying the accuracy of selected locations.

• Testing the algorithm for scanning and matching ride requests to ensure it accurately identifies potential matches.

Integration Testing: This involves testing the integration of different components or modules of the application to ensure they work together seamlessly.

• Testing the integration of user registration and authentication with Gmail API for authentication using Gmail accounts.

• Testing the integration of Google Maps API with the application for accurate location selection and display on the map.

• Testing the integration of the application with the Neo4j graph database for storing and retrieving user data and ride information.

Functional Testing: This involves testing the overall functionality of the web application to ensure it meets the intended requirements.

• Testing user registration and login process, including email verification and social media authentication.

• Testing the ride search feature to verify if it accurately matches ride requests and calculates the shortest path between users.

• Testing the formation of connections between users based on completed rides.

• Testing the efficiency and accuracy of the matching algorithm in various scenarios, including cases with no path between users or incomplete user data.
Performance Testing: This involves testing the performance and scalability of the web application under different load conditions.

- Testing the response time of the application for different operations, such as user registration, login, ride search, and database queries.
- Testing the application's ability to handle multiple concurrent requests from different users.
- Testing the application's performance with a large number of users and ride data in the Neo4j graph database.

Security Testing: This involves testing the security features of the web application to ensure it protects user data and prevents unauthorized access.

- Testing the application for vulnerabilities, such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
- Testing the authentication and authorization process to ensure it is secure and prevents unauthorized access.
- Testing the application's data handling processes, including encryption and protection of sensitive information.

Overall, a comprehensive testing approach involving unit testing, integration testing, functional testing, performance testing, security testing, usability testing, and regression testing can help ensure the reliability, functionality, performance, security, and usability of the web application. It is important to thoroughly test the application at each stage of development and address any identified issues before deploying it to production.
6.3 Deployment

Launching a web application and Neo4j on Microsoft Azure involves several detailed steps. First, an Azure account needs to be created before accessing the virtual machine (VM) service to launch an instance that meets specific needs and budget. When the instance has been successfully deployed, security groups are configured in order to control inbound/outbound traffic rules for the instance, providing protection by only permitting required traffic while restricting any unauthorized attempts at entry or departure from it.

After setting up the instance, it is necessary to install the necessary software for your web application and Neo4j database on a VM instance. This includes installing a web server such as Internet Information Services (IIS); Neo4j database server; dependencies or libraries needed by your website application; as well as setting up virtual hosts, configuring database connections, specifying permissions/access controls etc.

Once the software has been installed and configured, we deploy the web app code onto a VM instance using IIS. This involves uploading it, setting file permissions appropriately and configuring a web server to serve it. In some cases, additional information is required by the application - like user data or any relevant details related to its functionality - might need to be imported into Neo4j databases prior to deployment.

It is also important to implement backup and monitoring solutions, such as automated backups, monitoring tools, and alerts, to ensure data protection, system stability, and timely detection of any issues. Regular updates and scaling may be required to keep the web application and Neo4j up-to-date with security patches and updates, and to handle increased traffic and demand.
In this deployment, IP address conversion was done using freedns software. Neo4j is configured to run on a private IP along with port 7474, and bolt runs on 7687. Proper planning, configuration, testing, and maintenance are essential for a successful deployment of the web application and Neo4j on the Azure VM, ensuring reliable and efficient operation.

Figure 11. FreeDNS Domain Registration

Figure 12. Microsoft Azure Virtual Machine Configuration
CHAPTER 7

Experiments and Results

The results section of the report presents the findings obtained through the analysis of data collected from the web application. Based on the research conducted by Yau [15], who performed analysis on the various activities that a regular American would do in their daily lives, the window of travel times was calculated. Additionally, the observations of Ashkrof [9] in measuring the ride acceptance emphasized the parameter of driver’s shift times affecting ride acceptance times. Using all these factors, we mapped the data related to the percentage of travelers at every major window of the day and noticed a larger variation in the percentage value at certain hours. The data was classified based on different time frames that categorized the different windows of the day where the comparisons were run. These time frames were early morning, morning rush hour, afternoon, evening rush hour, night hours, and midnight.

Table 1: Time Frames

<table>
<thead>
<tr>
<th>Window of the day</th>
<th>Time range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early morning</td>
<td>4AM - 7AM</td>
</tr>
<tr>
<td>Morning rush hour</td>
<td>7AM - 11AM</td>
</tr>
<tr>
<td>Afternoon</td>
<td>12PM - 4PM</td>
</tr>
<tr>
<td>Evening rush hour</td>
<td>4PM - 7PM</td>
</tr>
<tr>
<td>Night hours</td>
<td>7PM - 12AM</td>
</tr>
<tr>
<td>Midnight</td>
<td>12AM - 4AM</td>
</tr>
</tbody>
</table>
The above table describes the time frames that we will be using to widely categorize the different windows of the day where we will be running our comparisons.

We attempt to measure the possibility of a rider accepting a ride with 1 more or 2 more riders, to test this scenario of the number of users choosing rides with or without sharing, we followed the below steps:

1) Calculate the entire cost of each projected ride as a combination of the petrol, toll, and any other charges shared by the riders.

2) Add the expected duration of travel for each leg of the trip (from pick-up to drop-off for each rider) to get the total travel time for each prospective ride.

3) The degrees of separation are fetched from the neo4j database.

4) Based on user preferences, give each of the three factors (degrees of separation, cost, and travel time) a weight. As we are simulating a trust based ride sharing application, we have chosen the weight for degrees of separation to be higher than cost or time of ride.

5) Each parameter should be multiplied by its weight to create a score that is then normalized to a scale of 0 to 1 for each potential ride. For example, if the weights for degrees of separation, cost, and travel time are 0.5, 0.2, and 0.3, respectively, the score for a potential ride may be calculated as follows. \((0.5 \times \text{Degrees of Separation}) + (0.2 \times \text{Cost}) + (0.3 \times \text{Travel Time})\) is the score. These numbers have been standardized to range from 0 to 100.

6) We have normalized these values to range from 0 to 100. If a user’s score ranges from 0 - 30, he will pick a single ride option. If it ranges from 30-70, he will pick the 2 rider option, and he picks the three sharing option if the score is higher than 70.
The results obtained from the data analysis shown in Figure 13, revealed that during peak hours, there is a higher number of riders who opt for the three-share option. This can be attributed to the efficiency of the matching algorithm, which is designed to link people based on the six degrees of separation concept. This indicates that the algorithm is effective in finding suitable matches among riders during peak hours, resulting in a higher adoption of the three-share option.

On the other hand, the results also highlighted a lack of trust among riders after midnight. During this time period, there is a preference for booking solo rides rather than opting for ride-sharing. This could be due to a variety of issues, including safety concerns or discomfort with sharing transportation with strangers late at night. This study shows that late-night ride-sharing usage may be declining, highlighting a possible need for further measures to address trust and safety concerns among riders during these times.
Figure 14 presents a graph demonstrating the relationship between user acceptance numbers and degrees of separation in ride-sharing networks. The x-axis represents degree of separation ranging from 1 to 6, while the y-axis displays user acceptance count. This chart provides a visual depiction of how users connect within ride-sharing networks based on degree of separation concepts.

The graph displays that, as degrees of separation increase, the number of users connected to each other decreases - an indication that users in ride-sharing networks tend to experience more connectivity among members closer in terms of degrees of separation (i.e. those directly connected or having few degrees between them). When degrees of separation increase further still, users become less connected, suggesting less connectivity among distant participants in their network.
Based on the research conducted by Ashkrof[9] to simulate the acceptance rates of users based on the degrees of separation, we have used an inverse distribution for the trust values, where the values decrease as the degrees of separation increase. This choice is based on our assumption that a rider will accept a ride that is being shared when they are more likely to know each other. We make use of an alpha value that is a constant known as the inverse distribution parameter. The trust values are then calculated based on the formula $T = 1 / (d ** \alpha)$ for $d$ in degrees. These trust values are further normalized to generate the graph for the acceptance rates.

This graph can provide insights into the overall structure and connectivity of ride-sharing networks, showing how users are linked together based on degrees of separation between them. This data can be useful for understanding dynamics within ride-sharing networks such as identifying hubs or clusters of users as well as optimizing matching algorithms to increase ride sharing options' efficiency.

Figure 15. Users connected at various DoS
Figure 15 provides a visual representation of how users connected by various degrees of separation are distributed throughout a ride-sharing network. The x-axis represents degrees of separation ranging from 1 to 6, while on the y-axis are user connections at each degree of separation. This graph gives a visual depiction of their distribution in this ride-sharing environment.

To perform this experiment we ran a search in the neo4j database and populated values for each degree of separation based on the number of edges that the search passed through for each user to reach the other 999 users. The graph shows that most users are connected through higher degrees of separation, as there is a higher possibility of users being connected via various intermediaries.

Figure 16. Match time vs Time Frames
Figure 16 displays the time needed to match users during different times of day, as measured in seconds. Each marker on this line plot represents how long it took for that particular time of day to match users successfully. The time taken to match the rides was calculated based on the graph search algorithm by neo4j. For example, the time to search for a user who is a single edge away in the graph will be less than a user who is 4 edges away.

Here we can see the trend where the match time of users is faster during rush hours as the number of users who are using the application will be higher. This was sampled over a simulation of 1000 simultaneous users and benchmarked against each window of the day.

From the graph, it can be observed that the time taken to match users varies throughout the day. During the early morning and night periods, the time taken is relatively high, averaging around 15-12 seconds. However, during the morning rush hour and evening rush hour, the time taken decreases, with an average of 7-6 seconds. This indicates that the matching algorithm may take longer to find suitable matches during the early mornings and night times, which could be due to factors such as reduced availability of users or increased demand during those times.
CHAPTER 8

Analysis

In our analysis, we compared the costs associated with solo rides and ride-share rides over an entire month, considering a user traveling 500 miles total. Solo rides were found to cost $2.50 per mile for this distance while ride-share rides offered significant cost savings resulting in fare fees totaling only $750; thus offering significant monthly discounts of $500 when compared to solo rides.

Table 2: Cost Savings vs Ride Type

<table>
<thead>
<tr>
<th>Ride Type</th>
<th>Total Distance (mi)</th>
<th>Fare per mile ($)</th>
<th>Total Fare ($)</th>
<th>Total Time (mins)</th>
<th>Fare Savings ($)</th>
<th>Time Savings (mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solo</td>
<td>500</td>
<td>2.50</td>
<td>1250</td>
<td>600</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Rideshare</td>
<td>500</td>
<td>1.50</td>
<td>750</td>
<td>800</td>
<td>500</td>
<td>200</td>
</tr>
</tbody>
</table>

Ride-share services often offer discounted per-mile rates as multiple passengers share one ride together, enabling the cost of transportation to be shared more evenly among them and therefore providing cost savings compared to solo trips. Users who travel similar routes or destinations will find ride-sharing more cost-effective as they can split fares and enjoy lower per mile costs than with solo journeys.
Reducing costs with ride-share rides by choosing ride-share rides over solo rides can lead to significant long-term savings for users who rely on transportation services for daily commuting or regular travel needs. These savings could prove especially valuable for budget-minded travelers as this offers them a way to reduce transportation expenses without compromising on service quality or convenience.

Ride-sharing provides more than financial advantages; it can also reduce traffic congestion, environmental sustainability, and promote social interactions between passengers. Ride-sharing's efficiency in matching users based on degrees of separation can decrease matching times significantly; by booking solo rides at certain hours of the day during particular time zones users can speed up matching processes, leading to time savings for them. Thus using ride-sharing services not only saves costs but contributes towards an increasingly sustainable and efficient transportation system.

Our analysis indicates that ride-sharing provides significant cost advantages over solo rides, with a monthly discount of $500 over 500 miles of travel. This highlights the financial advantages of ride-sharing for users with regular or frequent transportation needs; time savings, reduced traffic congestion, environmental sustainability and social interactions all highlight why ride-sharing should be seen as a viable transportation option; hence our findings show the efficiency and effectiveness of this solution as an efficient yet sustainable transportation alternative for users.
CHAPTER 9

Ethics in AI

The way people travel in the modern age of technology is completely different due to ride sharing platforms. These platforms make it extremely accessible to anyone with a smartphone to book a ride for themselves or even for someone who doesn’t even have a device on them. With the growing number of users registering on various platforms, the issues pertaining to the ethics of collecting and storing user data is now under spotlight. Data is the currency that most companies deal with, since there is a direct correlation between the insights you can get with a larger dataset. Rideshare platforms are at a unique position in the market where they have their hands on a large amount of user data. How they enforce rules such as GDPR and CCPA is of particular interest as they will have different privacy settings based on the location they operate in.

General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA) are two of the most popular privacy protection laws that are mandatory for most large scale applications operating in various continents. The GDPR is applicable to all European Union citizens. CCPA is a data privacy mechanism where the control is shifted to the user for determining the settings related to the privacy of their data. The objective of these regulations is to police the organizations from misusing their user data and maintaining a transparent process in handling sensitive information.
Having access to data that is extremely precise related to a single profile such as location information, banking information, history of trips, and contact details make the abuse of user data a serious ethical concern. The information is lucrative for the advertisement industry as they need all the data they can get to profile the users for targeted recommendations. Third party suppliers will be processing this data and adequate measures need to be taken to mask the personally identifiable information (PII) and strict legal agreements set in place that punishes misuse of data.

Consumer consent is an important issue with the growing number of application users. It is more often than not where a user is not aware of the extent to which user data is collected. The only way to ensure the user knows how his data will be used is by agreeing to policies that are set in place by the government and enforced by the application. The platform should also implement workflows where if a user decides to be forgotten/opt out of the data collection policy, the organization can delete the data. This brings the power back to the consumers hand where he can choose to be anonymous and not have his personal data monetized.
CHAPTER 10

Conclusion

In this paper, we present an innovative ride-sharing algorithm that integrates the six degrees of separation concept into ride matching. Leveraging social network analysis and graph theory techniques, it uses users and their social connections to prioritize those with stronger ties when matching riders with drivers. We believe this approach could revolutionize ride sharing by offering users more tailored, efficient experiences.

One of the key advantages of our proposed approach is its potential to promote social interaction and trust between strangers. Ride-sharing services have often been criticized for failing to create any sense of social bonding, leading to individualism that runs counter to the spirit of community and sharing. Our proposed solution addresses this concern by emphasizing social ties as being vitally important and by connecting those with shared interests, friends, or communities.

However, we recognize that our approach has numerous challenges that must be met in order to be effective. One major consideration is data privacy - social network data can be extremely sensitive, making users reluctant to share it with a ride-sharing service. Securing users' data securely is vital in earning their trust and developing successful services.

Integration of Social Network APIs With Ride-Sharing Platform is another challenge faced when operating ride-sharing services. Accuracy and completeness of social network data
depend heavily on API quality and accessibility from providers; close working relations must therefore exist between these providers to ensure successful service operations.

Complex social networks present another difficulty that must be managed effectively. As networks expand in size and scope, matching arrangements may become increasingly challenging to manage effectively. Optimization strategies must be created to handle the increasing complexity of networks and guarantee their scalability, thus keeping algorithms from becoming computationally expensive or inefficient.

Though not free from challenges, our proof-of-concept study yielded encouraging results in terms of matching efficiency and travel time reduction. Further research and development are necessary to address identified issues as well as expand this approach beyond ridesharing to other contexts such as carpooling or social network-driven applications such as dating apps or job matching platforms.

In conclusion, our proposed approach provides an exciting avenue for research and development in ride-sharing and social network analysis. By harnessing social connections to foster more efficient ride-sharing experiences that foster community, trust, and sustainability; our approach could contribute to creating more equitable transportation systems where people feel connected, empowered, and engaged with one another.
LIST OF REFERENCES


