ITÄ-SUOMEN YLIOPISTO

University of Eastern Finland

School of Computing

Master's Thesis

Predicting Routes of Mobile Users

Muhammad Raheel Mansoor

2020-07-10

ABSTRACT

The use of GPS enabled devices is increasing day by day. This increase is associated with the growing use of GPS-enabled smart phones, smart watches and GPS-enabled vehicles. People have started tracking their sports activities, daily runs and even saving their navigation data. All this tracking gives us a rich geospatial data but occasionally the data is incomplete due to a GPS error or hardware malfunction.

Route prediction and navigation applications such as *Google maps*, *Apple maps* and *Waze* collect GPS data of the application user. The data provided by these applications have missing records due to multiple reasons such as application crashing, signal interruption or disabled connection. We have taken into account similar data gathered through MOPSI service for our research.

This research uses the navigation data collected using GPS-enabled smart phones with which users can track their movements throughout the day, and tries to fill in the gaps of the incomplete geospatial data. It is accomplished by using previous data of the users and predicting their routes based on their historical route trends. In this research, we have compared the performance of different prediction models by analysing various properties of the user routes.

ACKNOWLEDGEMENTS

LIST OF ABBREVIATIONS

GPS	Global Positioning System
CAGR	Compound Annual Growth Rate
USD	United States Dollar
UEF	University of Eastern Finland
NATO	North Atlantic Treaty Organization

TABLE OF CONTENTS

1 Introduction

Our lifestyle and communication systems have been revolutionized by the usage of mobile devices and internet these days. This usage has become a source of information outburst in this era of technological advancements. One important information source in the modern communicational devices is *location-awareness.* Location awareness is a feature in devices such as mobile phones and tablets, which tells us about the geographical location of that device. Three common methods to identify the location of a device are [24]:

- *Global Positioning Systems* (GPS) satellite tracking
- Tower triangulation
- *Media Access Control* (MAC) address accessed by a device in a *Wireless Fidelity* (Wi-Fi) network

In this research, we focus on GPS satellite tracking data collected through mobile devices. Location based technology has proved its importance in various applications for example, in *Geographic Information Systems* (GIS), surveillance, navigation, military equipment tracking, wireless location services, GPS systems used in vehicles, memory cards of cameras (with automated location tagging of images) and healthcare device management [25].

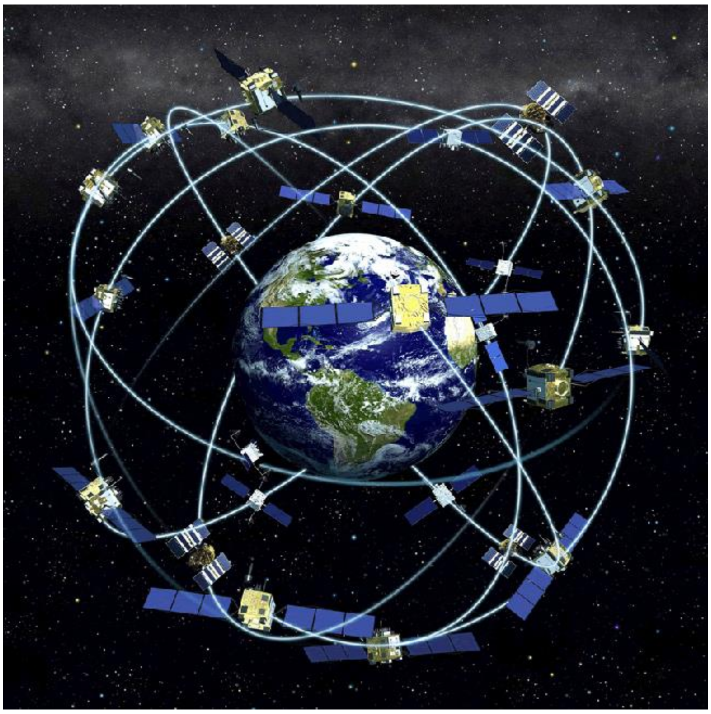


Figure 1: A GPS satellite network [30]

GPS is a satellites based system that provides constant navigation through mobile devices worldwide (shown in Figure 1). The GPS system has 24 satellites (at least) and algorithms to synchronize location, velocity and time with the recieving devices on Earth; and facilitate travelling through air, water and land. The satellite system is comprised of six orbitting planes (four satellites per orbital plane) around the Earth. Each satellite emits a unique signal so that the GPS devices on Earth catch and interpret those signals. Figure 2 shows how communication and control is handled in a GPS. It helps in five main domains: locating a position, navigating a *route*, tracking and monitoring movements, creating world maps, measuring time. [26]

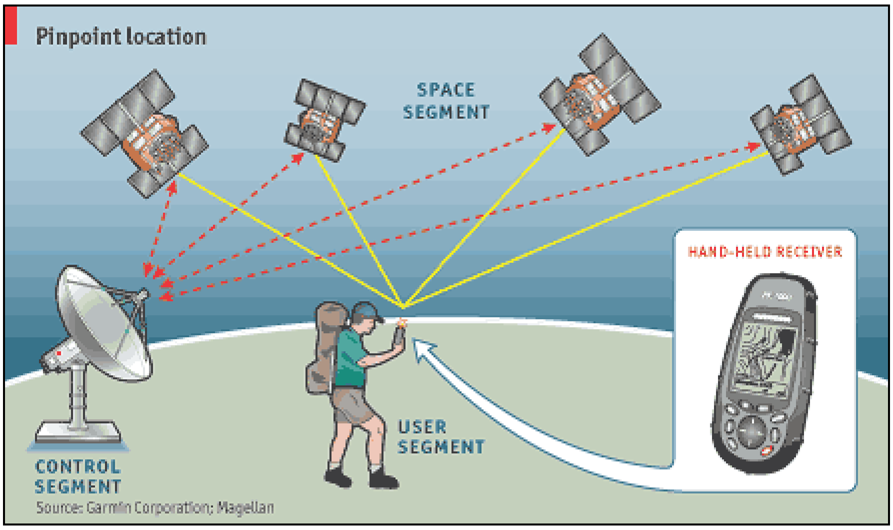


Figure 2: Three main segments of a GPS. [31]

The increased use of GPS-enabled smart phones (while traveling) and in-vehicle location-based systems have provided us with a large amount of geographical trajectory data which we will refer to as *routes*. This rich data has lead to the creation of *location-based services* (LBS). These services cater to the needs of navigation, location based recomendations and games, such as Google Maps, Uber, Foodora, PokemonGo.

Tracking the movement of a continously moving user gives us a route. A *route* is an ordered sequence of *latitude* and *longitude* points which shows how a moving object or person moves from one point to another (Figure 3).

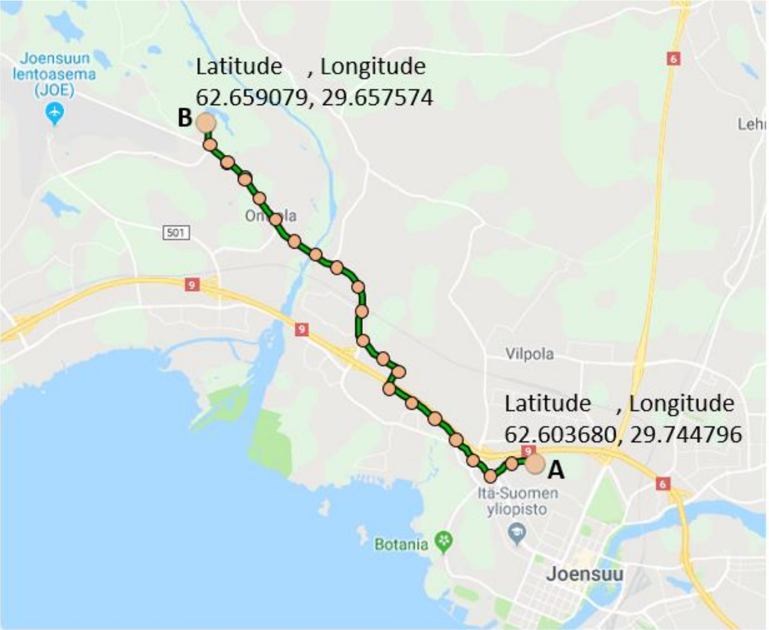
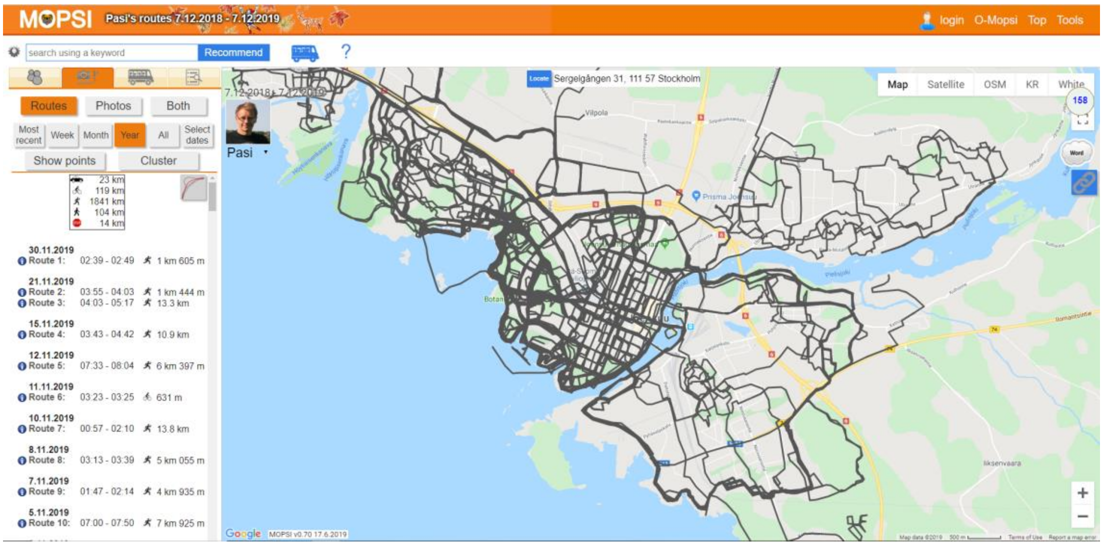


Figure 3: An example of a route from point A to point B.

The ease of movement tracking gave rise to different mobile tracking applications where a user can track and store their routes data. This type of tracking data can be stored using different services, one such social network service is MOPSI [1].



23 km
119 km
1841 km
104 km
14 km

30.11.2019
Route 1: 02:39 - 02:49 1 km 605 m

21.11.2019
Route 2: 03:55 - 04:03 1 km 444 m
Route 3: 04:03 - 05:17 13.3 km

15.11.2019
Route 4: 03:43 - 04:42 10.9 km

12.11.2019
Route 5: 07:33 - 08:04 6 km 397 m

11.11.2019
Route 6: 03:23 - 03:25 631 m

10.11.2019
Route 7: 00:57 - 02:10 13.8 km

8.11.2019
Route 8: 03:13 - 03:39 5 km 055 m

7.11.2019
Route 9: 01:47 - 02:14 4 km 935 m

5.11.2019
Route 10: 07:00 - 07:50 7 km 925 m

Figure 4: User Pasi’s 2019 travel data in Joensuu region taken from MOPSI.

The movement data can be stored in different formats. In MOPSI, the data is stored as separate route files for each user (Figure 4). The route tracking through such services is not always correct as it is prone to GPS error, signal interference and device malfunction. This would result in missing segments in the tracked route and the resultant route would not be accurate representation of user’s movement. One such example from the MOPSI data is shown in Figure 5.

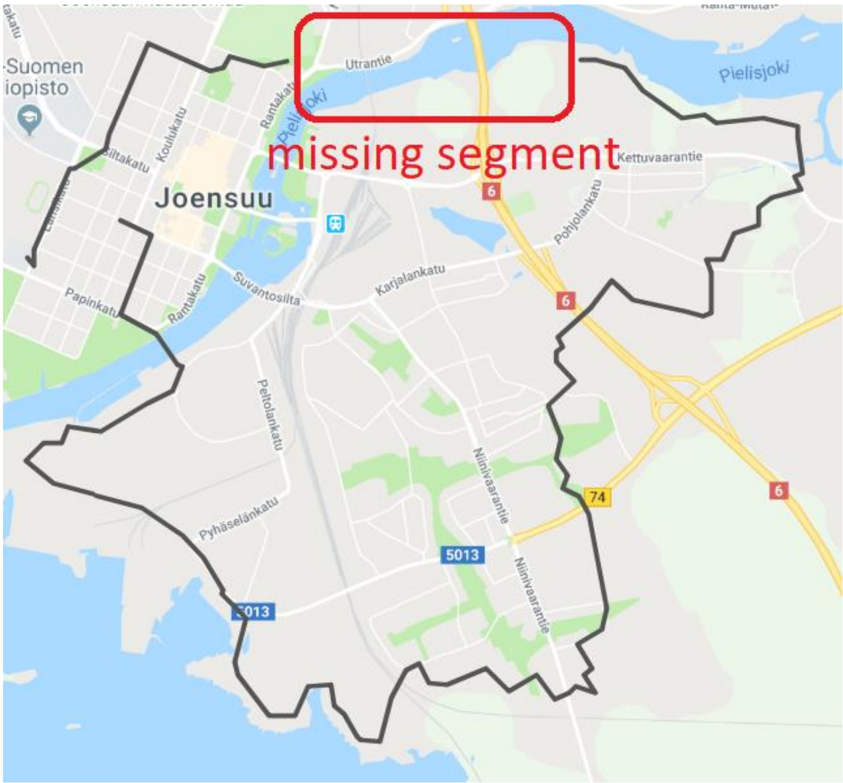


Figure 5: Missing segment in a tracked route (by MOPSI).

Predicting routes is also important when we wish to warn mobile users of upcoming incidents [7]. For example, when an accident or traffic jam has happened towards the user’s trajectory as shown in Figure 6. Another useful application of route prediction is a tool to store route collections that contain missing segments, which occurred due to battery loss of the mobile device or application failure.

A *prediction* is a statement about a future event, often based upon experience or knowledge. Predictive analytics use historical and current statistics to estimate, or ‘predict’, future outcomes of an event. The possible future occurrences are predicted using statistics based methods such as data mining and machine learning. [23]

In order to deal with the missing segments in routes, we aim to implement a route prediction mechanism which predicts the route of a user on road networks learning through the previous route history of that user. Given the scenerio; when users lose the

GPS signal on their devices while walking, cycling or driving and some part of the route is not tracked or recorded; our route prediction algorithm will predict that missing part of a route.

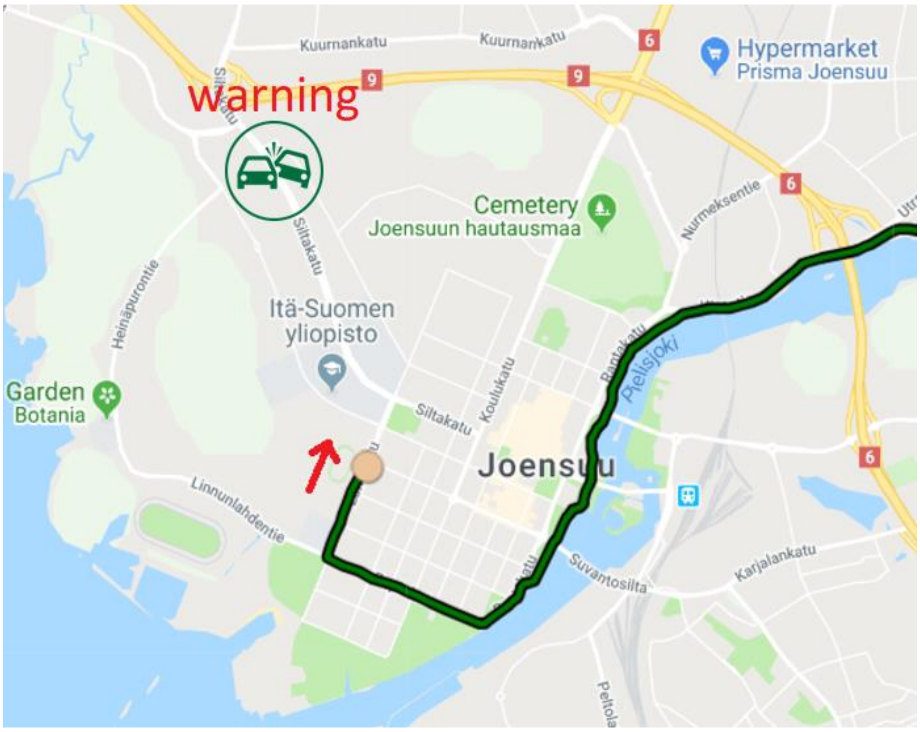


Figure 6: A scenario when route prediction is helpful: An accident warning on route.

The future of next generation mobility schemes depends on the ability to predict a correct route for users [3]. Tracking the past behaviour of user can predict the correct or most likely route with 93% accuracy [4]. In our work, we have considered the fact presented in [5], that analyzing the route history of a user reveals that their mobility is not random but rather direction or destination oriented. So, utilizing the movement history of a user along with the context information like time or day, it is possible to predict the correct route of that user. [6].

According to Georgiou et al. [7] a *trajectory* is a sequence of pairs of location and time (recorded for that location) such as <(p_0, t_0), (p_1, t_1),…, (p_i, t_i),…>, here p_i is the point of location of an individual and t_i represents the time this location was recorded at. In simple words, trajectory is a route taken by a moving individual from one point to another in a certain space and time. The simulation of such moving objects between two points is done using different curve fitting methods like *Linear interpolation* (LERP) or *B-splines*. There are two main divisions of trajectories – complete and incomplete, a trajectory is called incomplete (or open) if there are unknown points in the movement between starting and ending locations; otherwise, the trajectory is called complete (or closed). Based on these concepts, the two majorly discussed prediction-related problems are, *future location prediction* (FLP) and *trajectory prediction* (TP) as depicted in Figure 7 in [7].

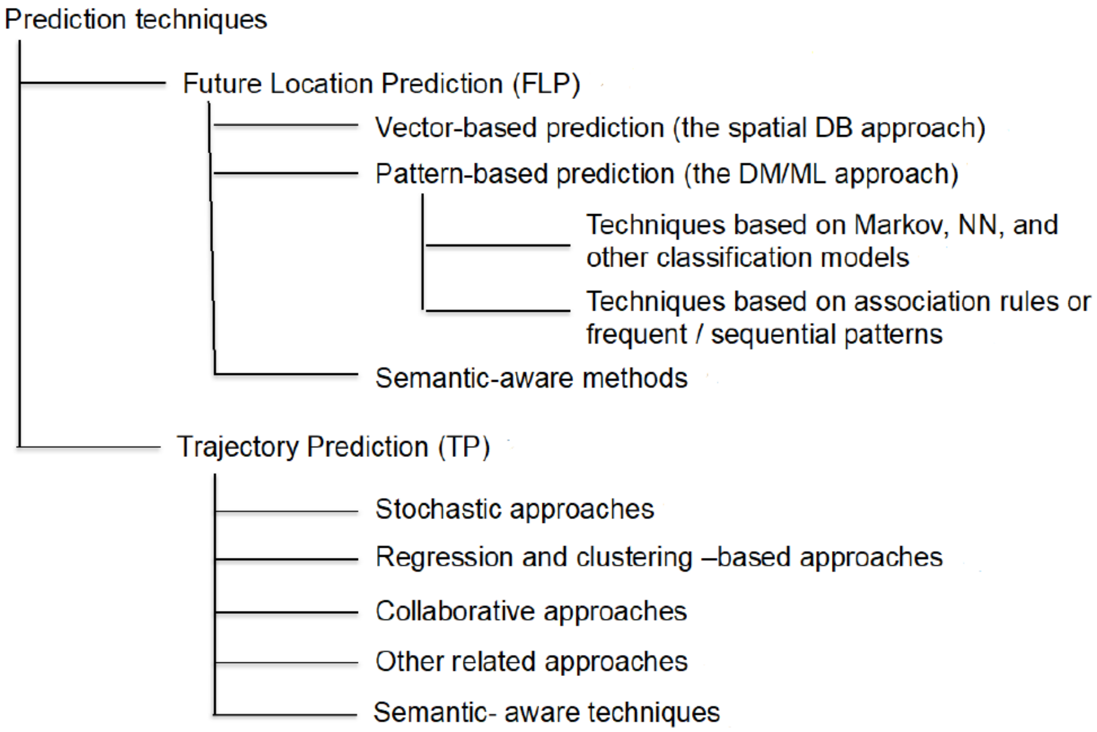

Prediction techniques
Future Location Prediction (FLP)
Vector-based prediction (the spatial DB approach)
Pattern-based prediction (the DM/ML approach)
Techniques based on Markov, NN, and other classification models
Techniques based on association rules or frequent / sequential patterns
Semantic-aware methods
Trajectory Prediction (TP)
Stochastic approaches
Regression and clustering –based approaches
Collaborative approaches
Other related approaches
Semantic- aware techniques

Figure 7: Taxonomy of Future Location Prediction and Trajectory Prediction methods surveyed in [7].

In FLP, the future locations of the user are predicted by learning their movement patterns from the past visited locations [8] while in TP, such a route is predicted that the user will most likely take to reach from one point to another based on their previous route history [7]. In context of Figure 7, our method lies in the trajectory prediction category to estimate the route with the help of data-driven prediction approaches.

As our thesis is linked to the MOPSI dataset, the following section better explains what MOPSI is and what data is available inside MOPSI service.

1.1 MOPSI Application

MOPSI is a social networking web application developed by Machine Learning lab at University of Eastern Finland (UEF). MOPSI provides location-based services such as location search, event recommendation, data collection, bus transportation and user tracking. MOPSI can be accessed on the web at http://cs.uef.fi/mopsi. Users can share their photos and routes, track their routes and chat with each other in MOPSI.

For data collection in MOPSI, the user data is stored as geo-tagged *photos* and routes in MOPSI. Photos are stored with location information as latitude and longitude points. Users can also input relevant discription while uploading the photos. The system distinguishes four different transportation modes as walk, run, bicycle and car.

2 Prediction Algorithm

We have developed a data driven algorithm to predict the route of a user between two points. These days there is an increased trend of tracking an individual's location information and movement activity in various mobile phone applications. This has resulted in rich location aware movement data. In our proposed prediction algorithm, we have used the historical movement data of a user to train a model. After the training process the model is ready to do the prediction. We must input a starting and ending location of the user and the model will output a route between the starting and ending location.

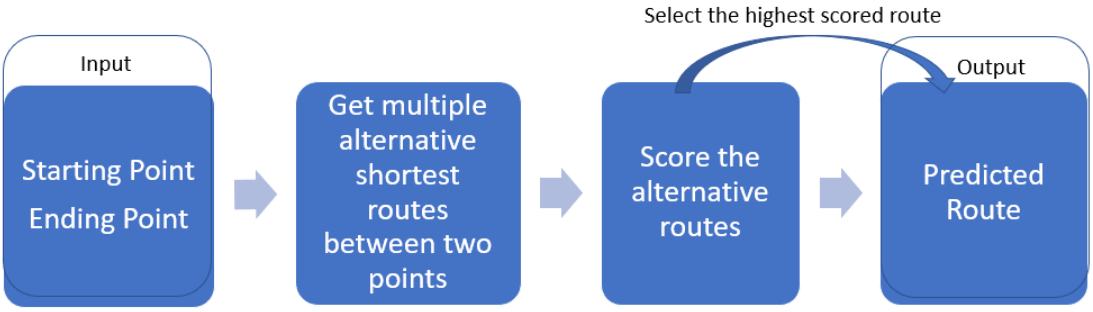

Select the highest scored route
Input
Starting Point
Ending Point
Get multiple alternative shortest routes between two points
Score the alternative routes
Output
Predicted Route

Figure 8: Prediction Algorithm workflow.

The workflow of our prediction algorithm is shown in Figure 8. In the first step, our algorithm takes the starting and ending points of the routes as input. As a second step, we predict multiple shortest candidate routes between the start and end points. Then, we assign score (as score = 1, 2, 3,...) to each candidate route. At the end, we select that route which achieves the highest score of all the candidate routes.

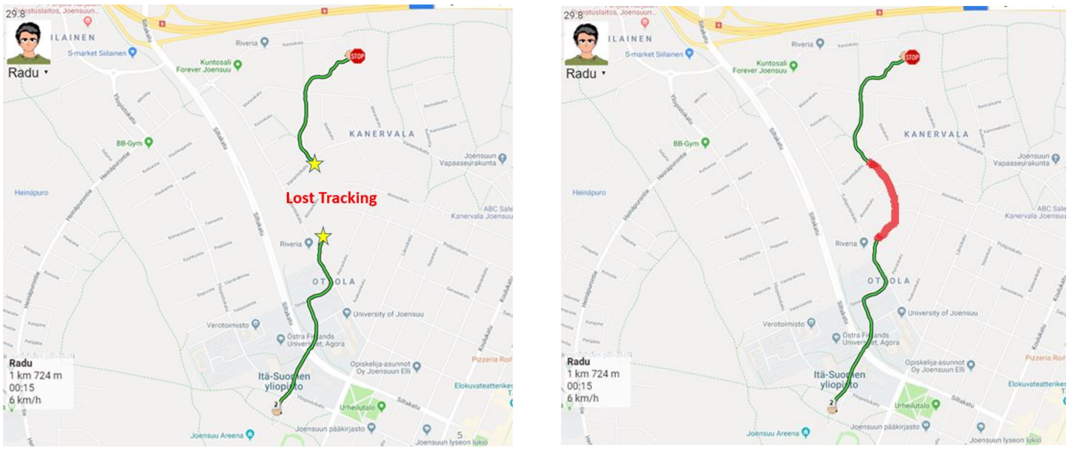


Figure 9: An example of a lost tracking problem (Left) and its solution (Right).

The use case of the prediction algorithm is shown in the example shown in Figure 9, where a user has lost the route tracking for a small duration during a morning run that resulted in a broken route. The missing route is shown between the two yellow stars in Figure 9 (Left). The start and end points (locations) of a lost track (represented by stars) would be used as inputs into the prediction algorithm. The algorithm will then return the predicted missing route as shown in Figure 9 (Right). The problem stated above and its solution are discussed step by step in Section 2.1 below.

2.1 Prediction of Missing Route

The prediction of broken or incomplete route between two points of a user has been described here in few simple steps. First, all the historical routes data available for a user is divided into *latitude, longitude* (lat, long) points through which the user has traversed; the lat, long are the coordinate points measured in degrees; used to represent the locations on the surface of earth. Then, we generate multiple short/fast paths between starting and ending points of the missing route. In the second step, we use the list of all those lat, long points as an input to our grid model. Our grid model divides the whole world surface into many grids each having a unique Id and then uses the lat, long list to assign scores to the grids of each short candidate route generated in the first step. The path is then selected on the basis of highest score and the missing part of the route can be predicted.. All of the above stated steps are illustrated in Figure 10.

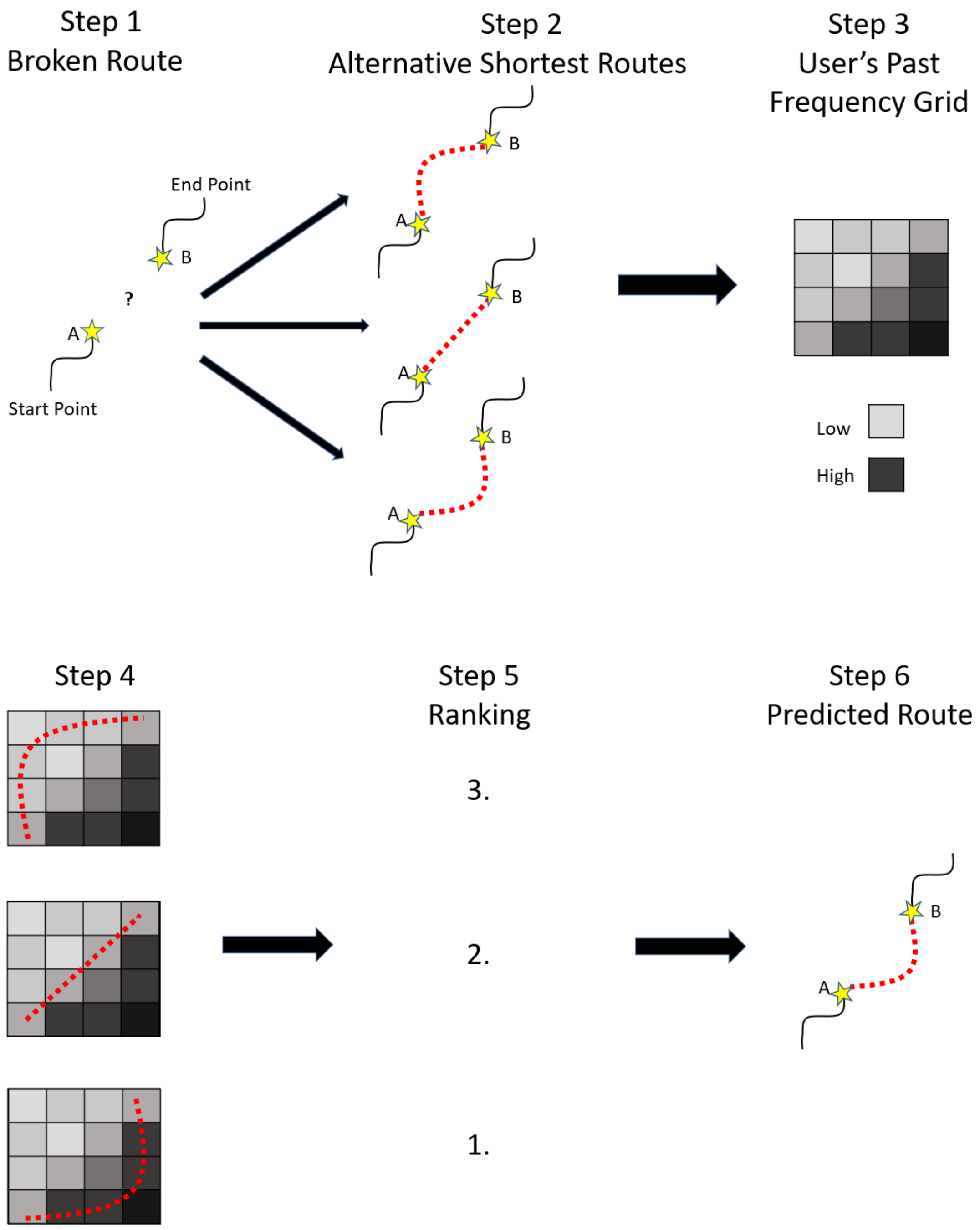


Figure 10: An illustrative example showing the steps to solve missing route problem.

Our prediction algorithm has six main steps, each of them is explained in sections below along with a real-world example.

2.2 Cell Representation

The Military Grid Reference System (MGRS) [1] is the geocoordinate standard used by NATO militaries for locating points on Earth. The MGRS is derived from the Universal Transverse Mercator (UTM) grid system and the Universal Polar Stereographic (UPS) grid system but uses a different labeling convention. The MGRS is used as geocode for the entire Earth.

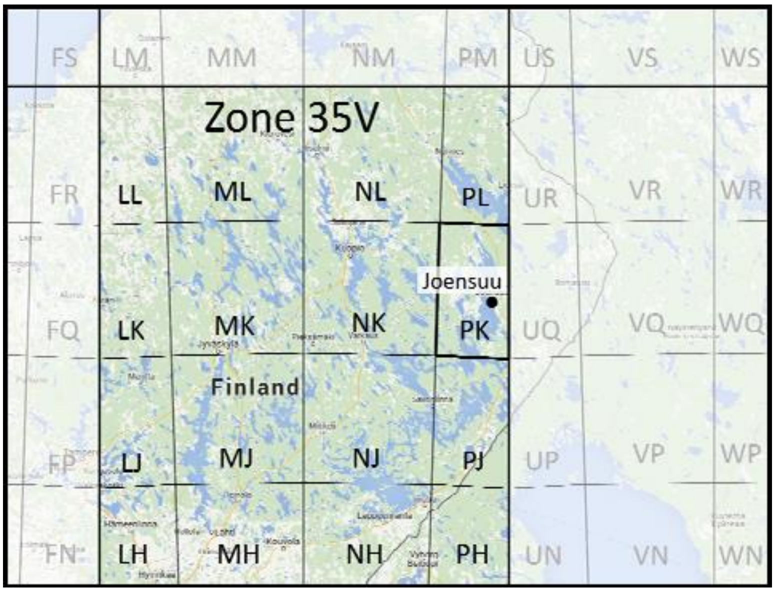


Figure 11: Map depicting a 100-kilometer grid using MGRS to locate Joensuu city in square PK of Zone 35V. [1]

We have used a modified version of MGRS to divide the earth’s surface into 25 × 25-meter square (sqm) grids as shown in Figure 12.

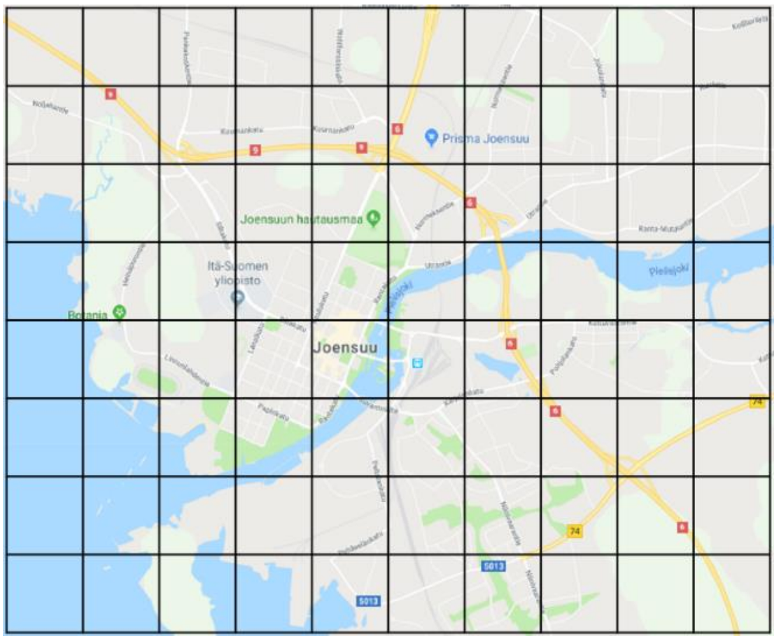
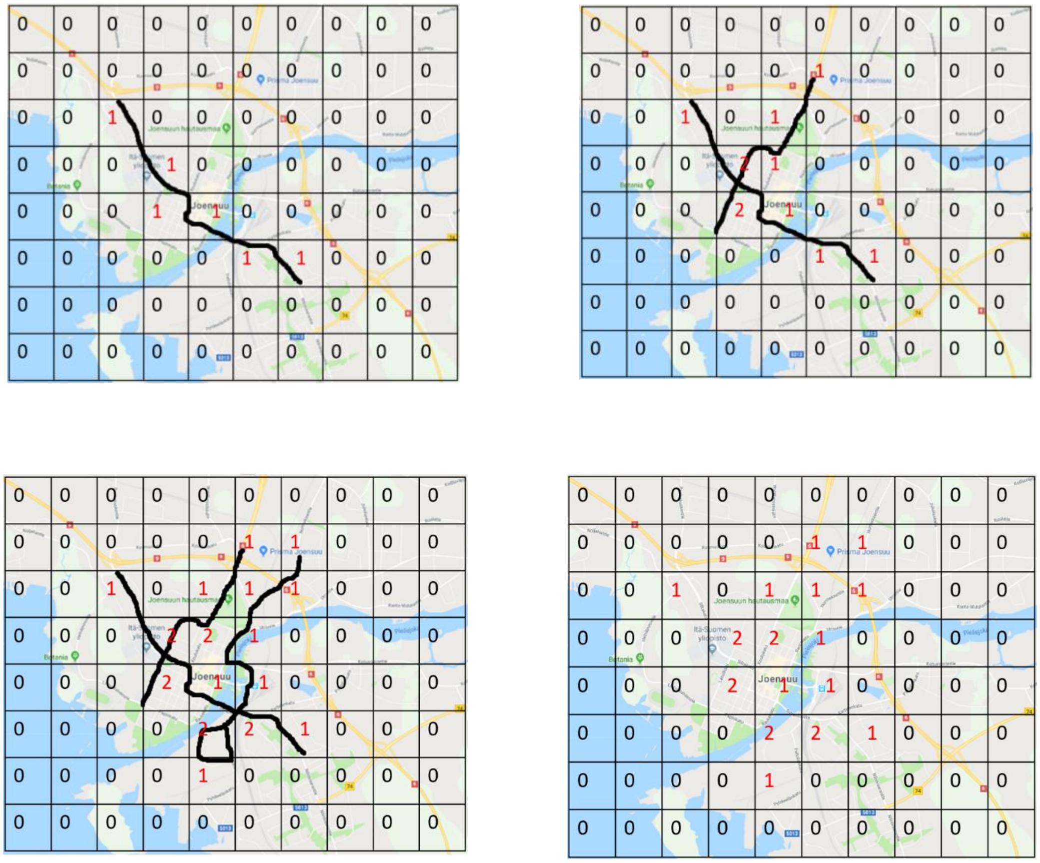


Figure 12: Google Map showing Earth surface divided into grids of size 25 x 25 sqm.

The grid in Figure 12 is shown using Google Maps API [32], this is discussed in detail in Section 2.5 [1]. Each grid is of approximate size of 25 × 25 meters. Every grid has a key string that contains Easting and Northing values. This grid key holds the value or score of the grid.

2.3 Frequency Grid Calculation:

The second step of our algorithm is to calculate the frequency grid. For this purpose, we use the historical route data of user and calculate each grid score that the route has been passed by. We append the value of the grid scores by the number of times the route goes through a specific point/location in a grid. This frequency grid calculation has been illustrated in Figure 13.



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	0	0	2	1	0	0	0	0	0
0	0	0	2	1	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	1	0	1	1	1	0	0	0
0	0	0	2	2	1	0	0	0	0
0	0	0	2	1	1	0	0	0	0
0	0	0	0	2	2	1	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

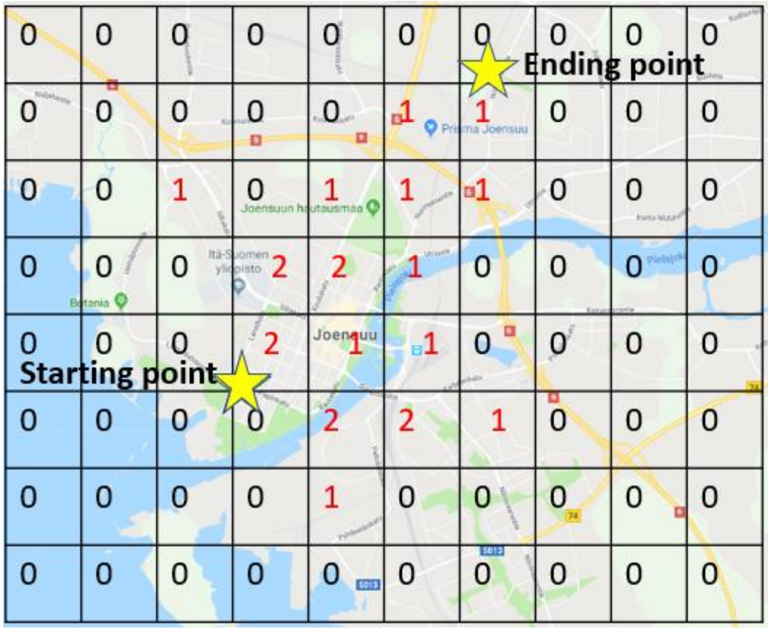
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	1	0	1	1	1	0	0	0
0	0	0	2	2	1	0	0	0	0
0	0	0	2	1	1	0	0	0	0
0	0	0	0	2	2	1	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Figure 13: Frequency grid calculation of single and multiple routes.

The images in Figure 13 are showing how each route of a user is mapped on the grid and the grid key is appended depending on how many times any route passes through a grid. When all the routes are scored, we get a frequency grid like that shown in the bottom right image of Figure 13. This frequency grid is used as a base for our prediction in the steps presented in sections below.

2.4 Input to the Algorithm

We need two input locations for the algorithm that is the starting and the ending point of the route. This can be extracted from the Broken Route problem by taking the starting and the ending location of the broken route (Figure 14).



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	1	0	1	1	1	0	0	0
0	0	0	2	2	1	0	0	0	0
0	0	0	2	1	1	0	0	0	0
0	0	0	0	2	2	1	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Figure 14: Start and end point locations of a broken route.

2.5 Getting Candidate Routes

Candidate shortest routes can be fetched using free direction services like OSRM and Google directions.

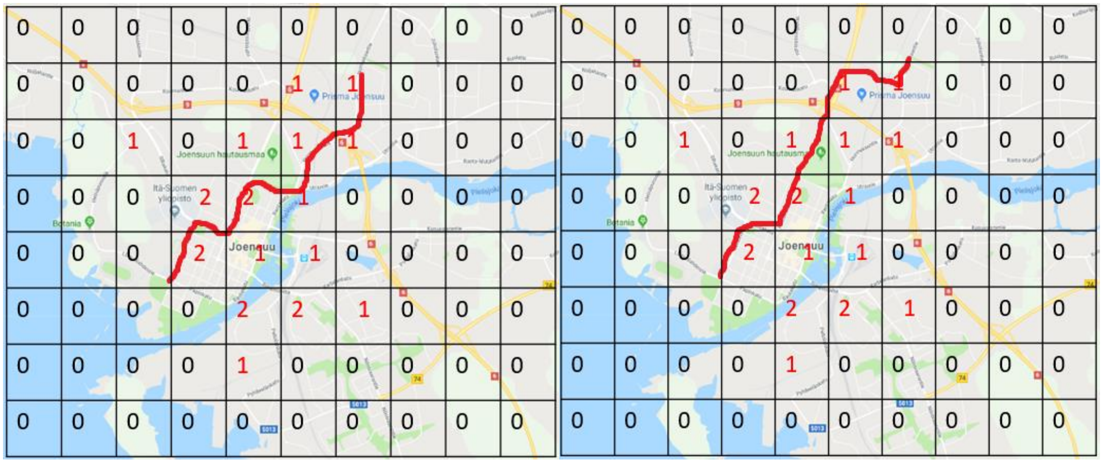
A Directions *Application Program Interface* (API) is a web service that is designed to provide information about a route. We have used *Google Directions API*[1] [32] in our experiments to estimate directions between locations. The Directions API helps in searching for directions in various transportation modes such as transit, driving,

[1] https://developers.google.com/maps/documentation/directions/start

walking, or cycling. We have accessed the Directions API through a *Hypertext Transfer Protocol* (HTTP) interface and used latitude/longitude coordinates to identify the locations according to the standard API request method by Google Maps Platform.

Additionally, we have used *Open Street Routing Machine* (OSRM) directions API. OSRM[2] is another dynamic and coherent routing service provider to calculate shortest paths in road maps. OSRM is an open source routing service and can be used license free. It can also be implemented on on-premise servers for better connection speed and availability. We have used OSRM to fetch candidate shortest routes for prediction.

After taking the input for the starting and ending point of the prediction. We get multiple shortest candidate routes from routing service. Figure 15 shows examples of two such candidate routes from the input shown in the last step. Each point on the route is considered as a pair of latitude (x) and longitude (y) on the Earth surface.



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	1	0	1	1	1	0	0	0
0	0	0	2	2	1	0	0	0	0
0	0	0	2	1	1	0	0	0	0
0	0	0	0	2	2	1	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	1	0	1	1	1	0	0	0
0	0	0	2	2	1	0	0	0	0
0	0	0	2	1	1	0	0	0	0
0	0	0	0	2	2	1	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Figure 15: Examples of shortest candidate routes.

[2] http://project-osrm.org/

2.6 Calculating Candidate Route Score

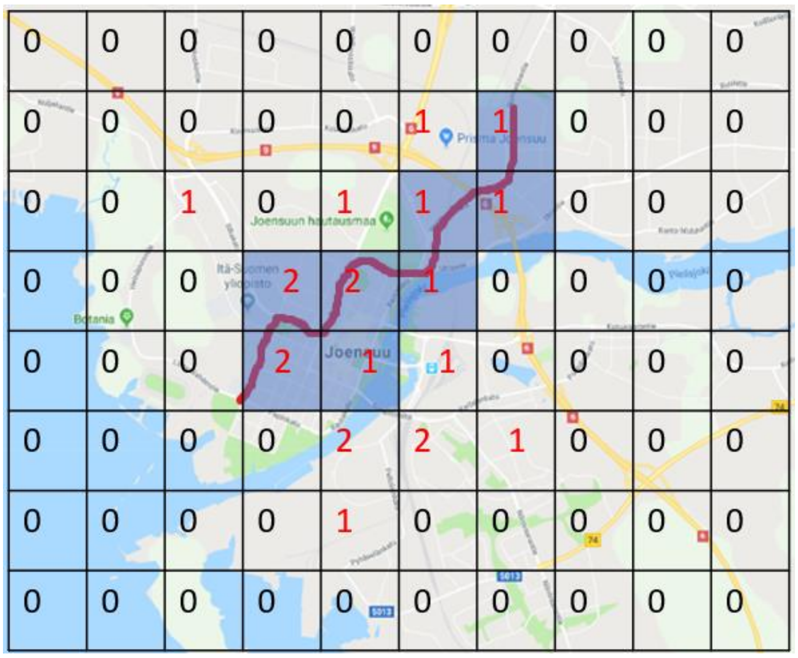
The candidate route score is calculated by summing up all the frequency score of the grids through which the candidate routes have passed through. Below is a visual explanation of calculation of the route score of the candidate routes fetched in the previous step.

$$CandidateScore = \sum Vxy$$

Where V is the Frequency value of Grid at location xy.

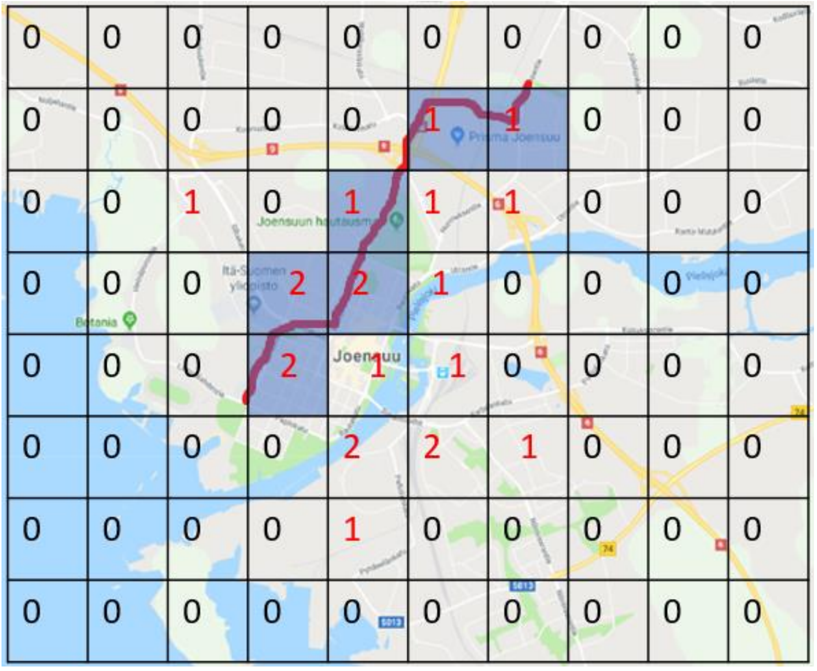
The candidate route 1 in Figure 16 (a) has a score of 11 **(1+1+1+1+2+2+1+2 = 11)** and candidate route 2 in Figure 16 (b) has a score of 9 **(1+1+1+2+2+2 = 9)**.

In this case the highest scoring route would be route 1 in Figure 16 (a) with score 11.



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	1	0	1	1	1	0	0	0
0	0	0	2	2	1	0	0	0	0
0	0	0	2	1	1	0	0	0	0
0	0	0	0	2	2	1	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Figure 16 (a): Candidate route 1 scored as 11.



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	1	0	1	1	1	0	0	0
0	0	0	2	2	1	0	0	0	0
0	0	0	2	1	1	0	0	0	0
0	0	0	0	2	2	1	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Figure 16 (b): Candidate route 2 scored as 9.

2.7 Ranking Candidate Routes

The next step in the prediction is to select the most suitable candidate route based on different strategies. We have used two different approaches here for ranking.

- Simple Ranking
- Normalized Ranking

2.7.1 Simple Ranking

Simple Ranking uses the candidate route score to rank all the candidate routes and the highest scoring route is ranked as *Top Rank*. For example, among the candidate routes shown in Figure 15 and Figure 16, the route with higher rank is candidate route 1 in Figure 15. So, according to simple ranking approach, the candidate route 1 is chosen as top rank.

$$Top\ Rank = MAX(\ Candidate\ Score)$$

2.7.2 Normalized Ranking

In *Normalized Ranking* approach, we consider the length of candidate route in addition to its score. The candidate route score we calculated in Section 2.6 is then divided by the length of the candidate routes.

$$Top\ Rank = MAX\left(\frac{Candidate\ Score}{Distance\ of\ Candidate\ Route}\right)$$

2.8 Route Prediction

Candidate route with the top rank is selected as the predicted route between two points for the user.

Each point on the route is considered as a pair of latitude (x) and longitude (y) on the Earth surface, given **p** = (x, y) and the route is a sequence of all these points arranged in order as **R** = (p_1, …, p_N).

Furthermore, a hashing method is used to store the cells that have already been tracked in the route representation. A grid measuring as 4,000 × 4,000 cells is created by 100 × 100 kilometers square containing each cell with dimensions of 25 × 25 meters (Figure 17). Each route passing through these cells contain Easting and Northing values.

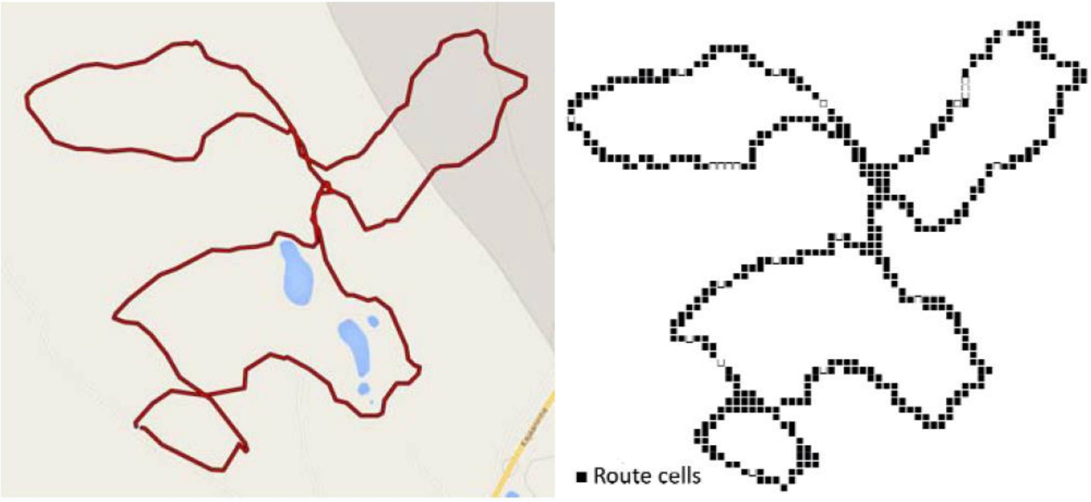


Figure 17: Example of a route (left) and its cell representation with 25 × 25 meters cell size (right) [1].

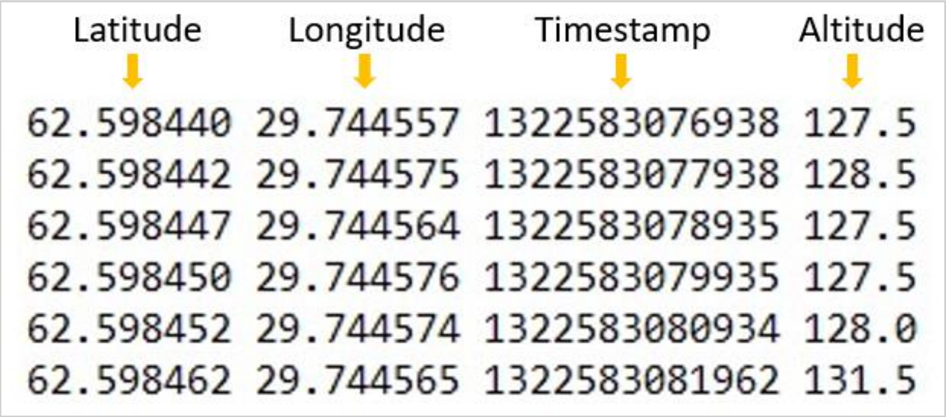
3 Experimentation

This section describes the details of our data selection and enhancement, feature engineering of the routes, experimental setup and the evaluation method used for our model.

3.1 MOPSI Routes 2014 Dataset

Mopsi Routes 2014 is a subset of the Mopsi routes collection dataset. In MOPSI Routes 2014 dataset, there are 6,779 routes (7,850,387 points) recorded by 51 users. Routes are recorded from every continent. Majority of the routes in this dataset has been collected from Joensuu region, Finland. Routes show a variety of activities and transportation modes for example, hiking, jogging, orienteering, skiing, walking, and traveling by cycle, car, bus, train or boat. We can analyze the life routines through these routes recorded by users such as going to work or shopping.

Mopsi dataset can be used for different location-based experimental evaluation purposes. We have used this dataset for evaluating our route prediction algorithm. Each route is stored in a separate text file containing one point per line ordered by time. The structure of the data is shown in Figure 20. We have considered timestamp in milliseconds and altitude in meters above sea level (may or may not be available always).



Latitude	Longitude	Timestamp	Altitude
62.598440	29.744557	1322583076938	127.5
62.598442	29.744575	1322583077938	128.5
62.598447	29.744564	1322583078935	127.5
62.598450	29.744576	1322583079935	127.5
62.598452	29.744574	1322583080934	128.0
62.598462	29.744565	1322583081962	131.5

Figure 20: An example file of a user's route history.

We have considered three sample users with richest travel history from Mopsi Routes 2014 data for our experiments. We store the route files separately for each user's travel history. These route files contain latitude, longitude, timestamp, and altitude in every row as shown in Figure 20. The travel modes considered for the experiments are walking, cycling, and driving.

The three sample users selected from Mopsi Routes 2014 dataset have routes distributions in three travelling modes as shown in Figure 21.

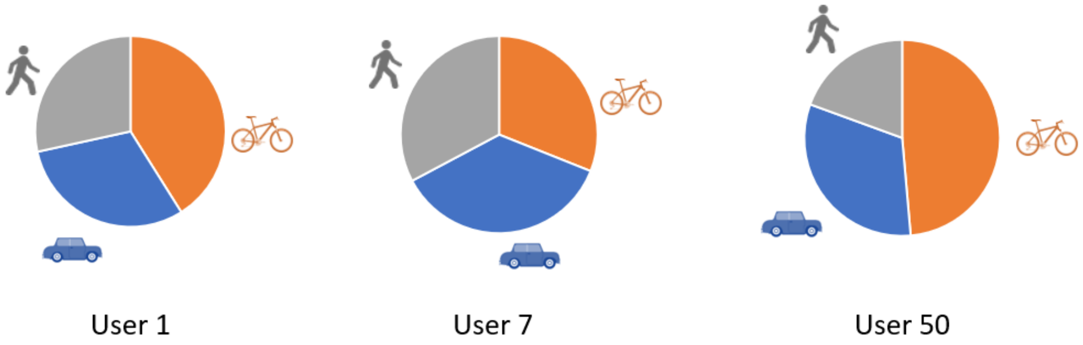


Figure 21: The distribution of walking, cycling, and driving routes in Mopsi Routes 2014 dataset for three sample users.

3.2 Feature Engineering

To derive additional values from the data, different transformations of route properties are applied. Following methods have been used for our route features enhancement:

To calculate the length of a route from the starting point till the ending point, the distance between each data point has been measured and the cumulative distance is calculated as the total length of the route.

$$\text{Length} = \sum_{x=0}^{n-1} \text{haversine\_distance}(P_x, P_{x+1})$$

The total duration of a route is calculated by measuring difference between timestamps of the starting and ending points of the route.

$$\text{Duration} = \text{End Time} - \text{Start Time}$$

The speed at which the user traversed through a route is calculated by dividing the length of the route by the total duration of the route.

Speed = Length/Duration

The location of a route is calculated by a binary dimension which determines if the route is in Joensuu region or outside Joensuu region. This is calculated by going through each data point of route to check whether it lies between specific latitude and longitude points as:

62.50118 > Latitude > 62.70118 and 29.56316 > Longitude > 29.96316

If a route fulfills the above bounded condition of between the specific latitude, longitude points, then the route is inside Joensuu region else considered outside Joensuu region.

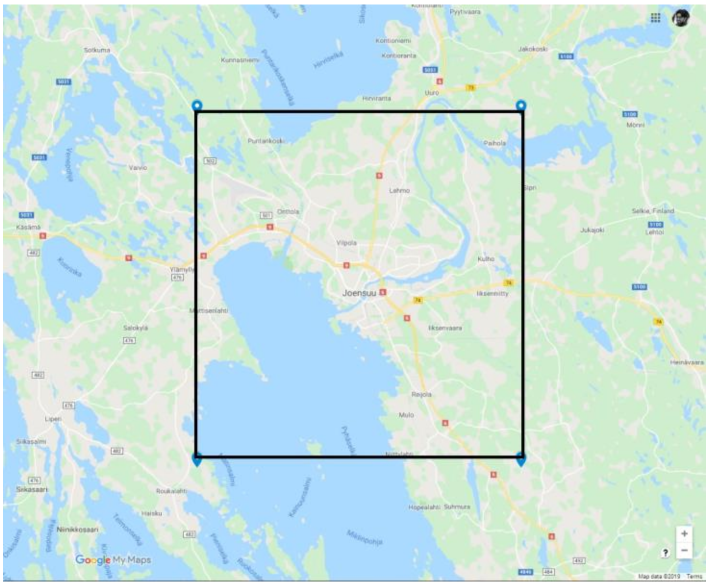


Figure 22: Location feature selected for Joensuu region inside the rectangular box.

3.3 Experimental Setup

We have split the route files of the three sample users into 50% *training* and *testing* data. Then these route files are sorted by date. In the training set, we have included every alternate route from a sorted route file of the user to avoid any seasonality effects. This training set is then used to calculate the underlying frequency grid. The remaining data is termed as testing data, which is then used to evaluate the performance of our model.

The starting and ending point of the testing routes are used as the input for the model. The routes within the Joensuu region are selected as shown in Figure 22 for data preparation by considering the latitude and longitude points of the area

The reason for restricting the location feature is to refine our experimental data. In this way, we have eliminated those routes which make sparse frequency grids. We have also removed those routes from our experimental data for which OSRM does not return any candidate route suggestions. Furthermore, those routes for which OSRM return exactly similar candidates for every travel mode (walking, cycling, and driving), are also eliminated to avoid data redundancy. The routes that return a single candidate route are excluded from test data because it does not take use of our model where we require at least two shortest candidate routes for prediction of an incomplete route.

3.4 Evaluation Method

We have used the similarity measure (C-Sim) proposed by Istodor and Fränti (2017) in [1], to measure the similarity between routes. According to [1], two routes will be considered *similar* if they overlap with each other. The more overlap means the more similarity between the routes. We try to find out the similarity between the candidate routes predicted by our model. We also look for the similarity of our resulting routes with the original route. There are several applications of measuring route similarity. For example, Ying et al. (2010) have used route similarity to identify similar behavior (in travel history) between users in a social network and recommend friends based on the measured similarity [9]. In [10] Shang et al. (2012) have used route similarity to give trip recommendations to the users based on their previous preferences. In [11] Evans et al. (2013) have used route similarity to build new bicycle paths based on the need and similarity measure. In [12] Ying et al. (2009) have used the similarity measure to optimize traffic and identify the overcrowding areas based on density clustering in areas.

According to our route similarity approach, two-dimensional grid cells are used for route representation and then set operations are applied on the routes. The upper limit stated in this method has cost of O(K_1+K_2) where K_1 and K_2 are the actual lengths of route 1 and route 2 (respectively) calculated in meters. But the actual cost depends on the chosen size of cells. Moreover, the *Route Similarity Ranking* (RSR) algorithm is used in this approach to find out all the similar routes in the database for a given route as an input. The algorithm has been implemented on Mopsi Routes 2014 dataset containing 6,700 routes. [1]

Other such similarity approaches represented in [14], [15] and [16] have used *dynamic programming* with O(N_1N_2) time complexity. Here, N_1 and N_2 are consecutive points in route 1 and route 2 respectively. For example, the *general time series analysis* techniques discussed by Hamilton (1994) in [13] and Agrawal et al. (1993) in [14] used the same methodology. *Longest common subsequence* methods presented by Vlachos et al. (2002) in [15] and *real sequence* method for edit distance proposed by Chen et al. (2005) in [16] also fall in the same method category. For the similarity computations, the research n [14], [15] and [16] compare the given input route with every other route in the database and result into O(MN^2) cost. Here, M represents the number of routes stored in the database and N represents the mean of consecutive points in a route.

4 Evaluation

We analyze in our experiments that the routes predicted for cycling mode have achieved the highest similarity as compared to driving and walking mode routes. In Table 1, we can see that, the similarity percentage of driving routes is highest for two out of three sample users. This is mainly because OSRM returns highest number of candidate routes in case of driving mode. This means that, there are more options of shortest candidate routes to travel by car as compared to cycle or walk. Our sample user 7 achieved the lowest similarity percentage in all three travel modes. The reason of lowest performance for user 7 is because this user has the least straight routes. Therefore, it was most difficult to predict routes for user 7.

Table 1: Travel modes performances for three sample users.

Users	🚗	🚲	🚶
1	45%	48%	34%
7	20%	19%	14%
50	46%	44%	40%

In Figure 24, we have shown the *Best Candidate's* similarity (BC) of routes in all three travel modes by our models. We have presented our models' performances in comparison with BC in graph of Figure 24. Our *normalized-personalized model* (NPM) has achieved the highest similarity in all three travel modes as compared to our simple *personalized model* (PM) and the *universal model* (UM). We see that, in case of driving and walking, the performance of PM and UM are very close to each other. However, UM shows slightly better performance than PM in both modes. This might be possible because we include the universal user data for UM, and it tends to give more candidate routes for cycling and walking and thus, gives better similarity with the original route.

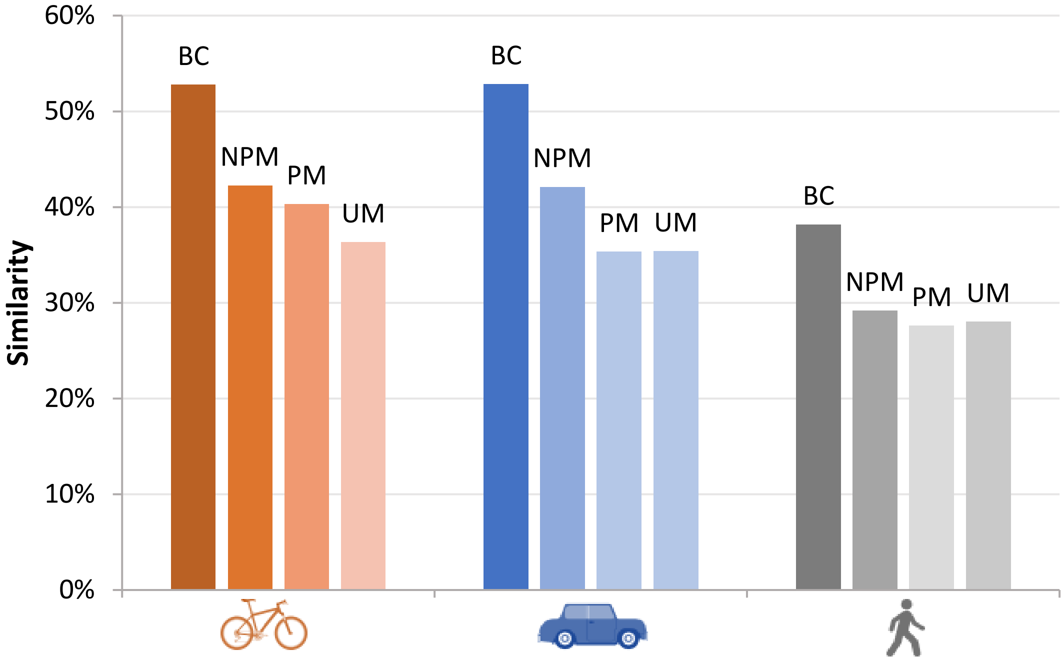


Figure 24: Similarity performance by personal models for three travelling modes.

The graph in Figure 25 shows the similarity trend folowed by our personal models with respect to the range of different route lengths in our data. The highest similarity percentage is achieved by all three models for the routes withing 0 to 5 kilometers range. The similarity decreases as the route length increases. We evaluate that, the similarity reaches close to 0% for the routes ranging from 25 to 30 kilometers because, there are minimum candidate routes available for such long range. We also analyzed that, UM and PM have achieved better similarity percentage than NPM. This shows that UM and PM perform better than NPM for very long routes.

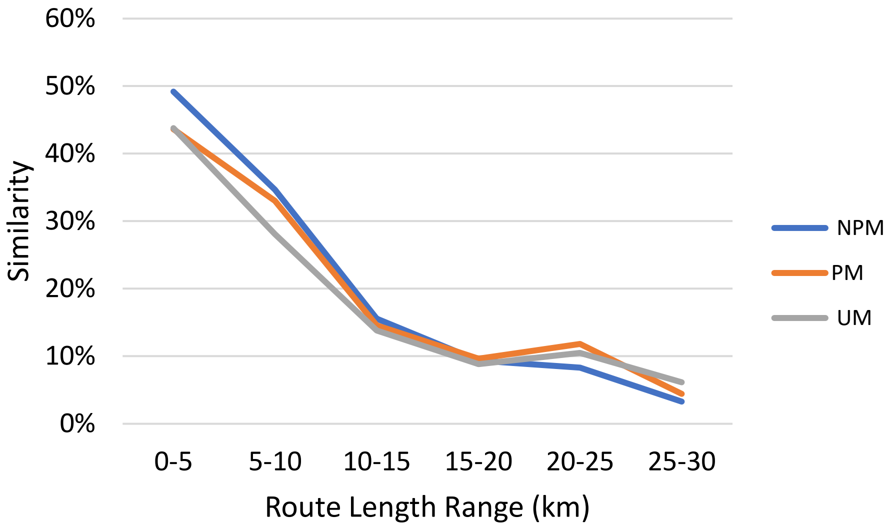


Figure 25: Similarity achieved by our models with respect to route length.

Table 2 summarizes the overall performances by all models with reference to BC and linear interpolation (LERP) for all three sample users and travel modes. The percentages shown in bold letters, represent the performances of our three personal models. The results show that our models achieve much better results than the LERP method. Also, the results achieved by NPM are very close to the *Best Candidate's* similarity.

Table 2: Model Performance by Users and Travel Modes.

Travel Mode	**User**	**BC**	**NPM**	**PM**	**UM**	**LERP**
🚲	User 1	63%	**51%**	**51%**	**44%**	16%
	User 7	35%	**23%**	**22%**	**21%**	10%
	User 50	60%	**52%**	**48%**	**44%**	14%
🚗	User 1	74%	**60%**	**52%**	**45%**	29%
	User 7	31%	**21%**	**21%**	**20%**	15%
	User 50	66%	**59%**	**41%**	**51%**	25%
🚶	User 1	51%	**41%**	**33%**	**36%**	29%
	User 7	21%	**14%**	**15%**	**16%**	7%
	User 50	59%	**49%**	**50%**	**46%**	17%

We also evaluated the performance of our models in terms of BC. The results show that NPM achieves ~90% of the BC when the user travels within the range of 0 to 5 kilometers per hour (km/hr). The percentage trend followed by models with respect to the speed is irregular between different speed ranges (Figure 26). We analyzed that all models show same similarity percentage for travel speed of ~25 km/hr. The same similarity percentage at this speed range is possible due to same candidate routes.

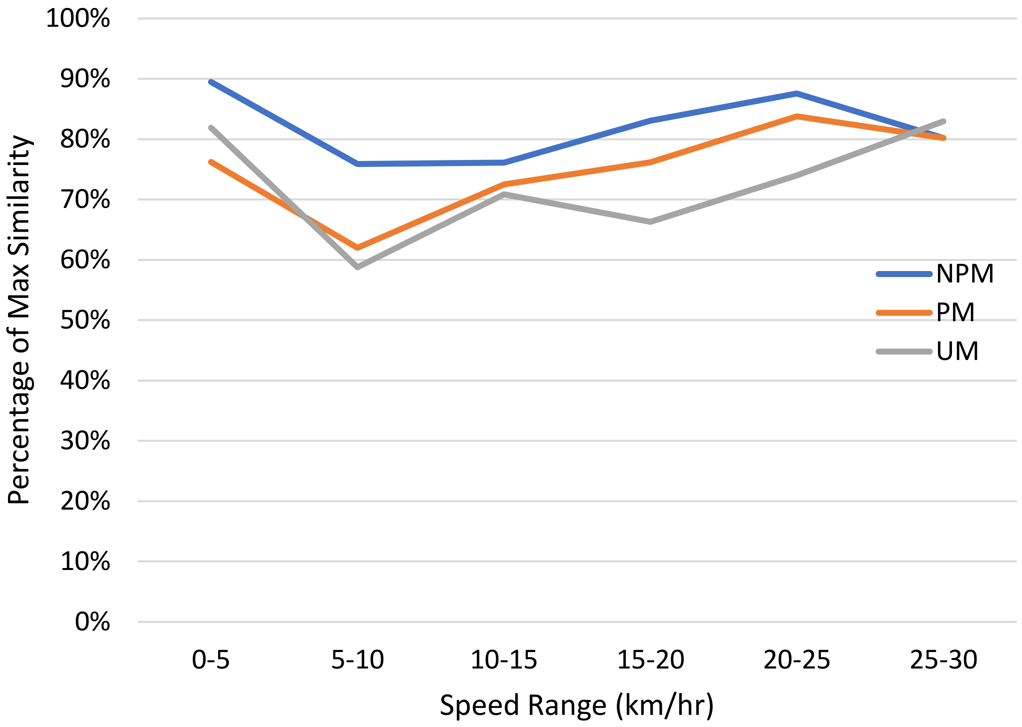


Figure 26: Similarity with respect to speed of route.

In Figure 27, graph shows the performance of our personal models, OSRM and linear interpolation model in comparison to the best candidate. This tells us how much accurately the model selects a candidate route so that the predicted route in closest to best candidate.

Moreover, we analyzed that OSRM returns same routes for all travel modes in 30% of the cases that is why the graph in Figure 27 shows various model accuracies close to each other. It can be seen in the graph that LERP selects the candidate route with least accuracy.

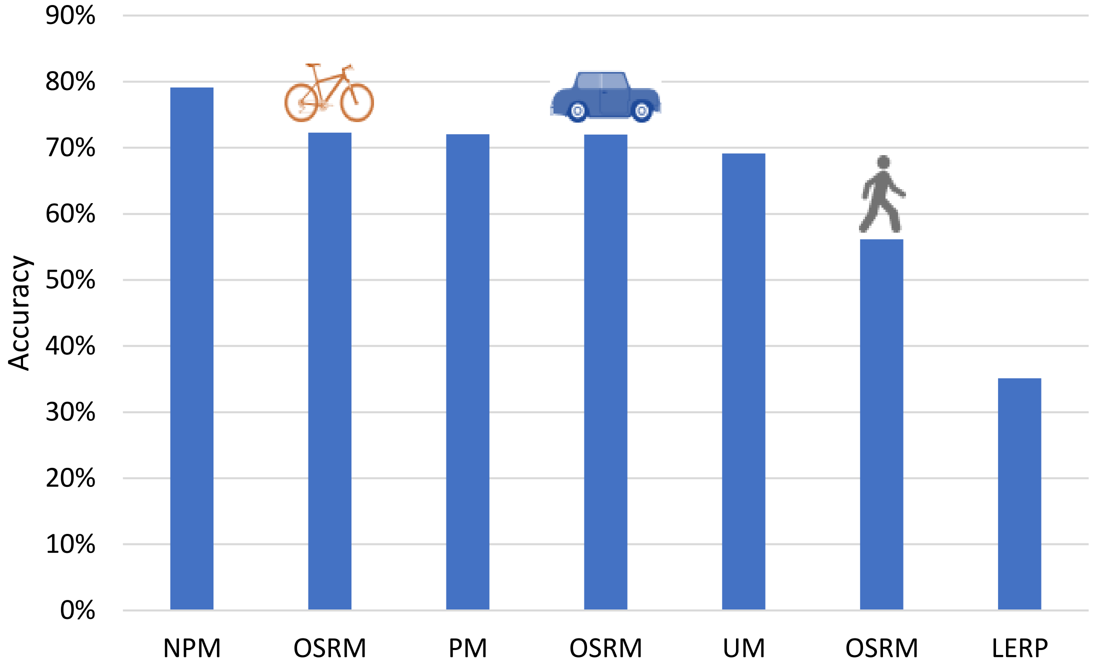


Figure 27: Performance accuracy of all models.

5 Conclusions

In this study, we predicted routes for mobile users based on their travel history in Joensuu region. Three different prediction methods are presented to address the challenge of missing or broken routes in user data. Our personalized model uses travel history of sample users to predict their missing routes. Normalized-personalized model is used to get unbiased prediction results. Universal model uses the travel history of all sample users to study their travel trends and to find out if their routes have some similarity.

We have used the C-Sim route similarity measure [1] in our models to identify route similarity. This is a fast and intuitive similarity measure, that uses the grid-based representation of routes to findout similarity between roots.

Our method successfully predicts the shortest route that the user is most likely to take, after considering the previous short routes taken by that user between any two points. It is easier and faster to predict short and straight routes by our prediction method. Our normalized model improves the prediction performance for short length of routes (0 to 10 km). The prediction results are not affected significantly by the speed of user.

Our model could not predict correctly for the circular or closed routes. However, it can perform better if rich travel data is available with more number of routes to be predicted. The model performance can also be improved by knowing the additional information about travel time of routes. This prediction method can be performed for any region on the map if rich travel data of users is available for that region. We used this method only for Joensuu region because of rich travel data of our sample users available by Mopsi in that area.

REFERENCES

[1] R. M. Istodor, P. Fränti. "Grid-Based Method for GPS Route Analysis for Retrieval". ACM Trans. on Spatial Algorithms and Systems, 3 (3), 8:1-28, September 2017.

[2] G. Saud. "Global positioning system (gps)". 2015.

[3] P. V. Klaine, M. A. Imran, O. Onireti, R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks", *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2392-2431, 4th Quart. 2017.

[4] C. Song, Z. Qu, N. Blumm, A.-L. Barabási, "Limits of predictability in human mobility", *Science*, vol. 327, no. 5968, pp. 1018, 2010.

[5] A. Aljadhai, T. F. Znati, "Predictive mobility support for QoS provisioning in mobile wireless environments", *IEEE J. Sel. Areas Commun.*, vol. 19, no. 10, pp. 1915-1930, Oct. 2001.

[6] S. Tabbane, "An alternative strategy for location tracking", *IEEE J. Sel. Areas Commun.*, vol. 13, no. 5, pp. 880-892, Jun. 1995.

[7] H. Georgiou, S. Karagiorgou, Y. Kontoulis, N. Pelekis, P. Petrou, D. Scarlatti, Y. Theodoridis, "Moving Objects Analytics: Survey on Future Location & Trajectory Prediction Methods". July. 2018.

[8] R. Wu, G. Luo , J. Shao, L. Tian, and C. Peng, "Location Prediction on Trajectory Data: A Review". Big data mining and analytics, ISSN 2096-0654 02/06 pp108–127 Volume 1, Number 2, June 2018.

[9] J. J. Ying, E. H. Lu, W. C. Lee, T. C. Weng and V. S. Tseng. “Mining user similarity from semantic trajectories”. In Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (ACM SIGSPATIAL GIS '10), San Jose, CA, USA, 19-26. 2010.

[10] S. Shang, R. Ding, B. Yuan, K. Xie, K. Zheng and P. Kalnis. “User oriented trajectory search for trip recommendation”. In Proceedings of the 15th ACM International Conference on Extending Database Technology, Berlin, Germany, 156-167. 2012.

[11] M. R. Evans, D. Oliver, S. Shekhar and F. Harvey. “Fast and exact network trajectory similarity computation: a case-study on bicycle corridor planning”. In Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing (UrbComp '13), Chicago, IL, USA, 9. 2013.

[12] X. Ying, Z. Xu and W. G. Yin. “Cluster-based congestion outlier detection method on trajectory data”. In Proceedings of the 6th IEEE International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '09), Tianjin, China, 243-247. 2009.

[13] James D. Hamilton. “Time series analysis” (Vol. 2). Princeton: Princeton university press. 1994.

[14] R. Agrawal, C. Faloutsos and A. Swami. “Efficient similarity search in sequence databases”. In Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms (FODO '93), Chicago, Illinois, USA, 69-84. 1993.

[15] M. Vlachos, G. Kollios and D. Gunopulos. “Discovering similar multidimensional trajectories”. In Proceedings of the 18th IEEE International Conference on Data Engineering (ICDE '02), 673-684. 2002.

[16] L. Chen, M. T. Ozsu and V. Oria. “Robust and fast similarity search for moving object trajectories”. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data and Symposium on Principles Database and Systems (SIGMOD/PODS '05), Baltimore, MD, USA, 491-502. 2005.

[17] L. X. Pang, S. Chawla, W. Liu and Y. Zheng. “On detection of emerging anomalous traffic patterns using GPS data”. Data & Knowledge Engineering (DKE), 87, 357-373. 2013.

[18] D. Zhang, N. Li, Z. H. Zhou, C. Chen, L. Sun and S. Li. “iBAT: detecting anomalous taxi trajectories from GPS traces”. In Proceedings of the 13th ACM international conference on Ubiquitous Computing (UbiComp '11), Beijing, China, 99-108. 2011.

[19] L. Y. Wei, Y. Zheng and W. C. Peng. “Constructing popular routes from uncertain trajectories”. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '12), Beijing, China, 195-203. 2012.

[20] V. W. Zheng, Y. Zheng, X. Xie and Q. Yang. “Collaborative location and activity recommendations with gps history data”. In Proceedings of the 19th ACM International Conference on World Wide Web (WWW '10), New York, NY, USA, 1029-1038. 2010.

[21] T. Bao, H. Cao, Q. Yang, E. Chen and J. Tian. “Mining significant places from cell id trajectories: A geo-grid based approach”. In Proceedings of the 13th IEEE International Conference on Mobile Data Management (MDM '12), Bengaluru, India, 288-293. 2012.

[22] J. Krumm and E. Horvitz. “Predestination: Inferring destinations from partial trajectories”. In Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp '06), Orange County, CA, USA, 243-260. 2006.

[23] J. Taylor. “Three steps to put Predictive Analytics to Work”. Decision Management Solutions, USA. 2014.

[24] N. D. Oye and J. Nathnaiel. “Location Awareness Using Mobile Application”. International Journal of Trend in Research and Development, Nigeria, 2394-9333. 2018.

[25] R. Bandakkanavar. “Location Based Services using Global Positioning System”. International Technical Papers, USA. 2015.

[26] T. A. Herring. “The Global Positioning System”. Scientific American, USA. Vol.274. 44-50. 1996.

[27] H. Stark. “Since You Asked, Here's How Google Maps Really Works”. Forbes, USA. 2017.

[28] M. Law and A. Collins. “Getting to Know ArcGIS”. Environmental Systems Research Institute, Inc. 2015.

[29] S. O. SMITH. DMA Technical Manual, Chapter 3. Datums, Ellipsoids, Grids, and Grid Reference Systems. 1990.

[30] Global Positioning System. Digital Image. NOAA. 2017.

[31] B. M. Williams. “An overview of GPS concept”. Digital Image. 2019.

[32] Google Maps Developers Guide. https://developers.google.com/maps/documentation/javascript/tutorial. Accessed March, 2020.