Master's Thesis

Mining Travel Patterns from Mobile Ticket Applications

Author:
Saeed Zanderahimi

Supervisors:
Morten Goodwin,
Ph.D., Associate Professor
Andreas Häber,
Ph.D., System Engineer

Faculty of Engineering and Science
Department of Information and Communication Technology (ICT)

June 2014
Grimstad, Norway

UNIVERSITY OF AGDER

Abstract

Faculty of Engineering and Science
Department of Information and Communication Technology (ICT)

Mining Travel Patterns from
Mobile Ticket Applications

by Saeed Zanderahimi
Master's Student in ICT
saeedzr@gmail.com

Customers' travel patterns are highly interesting for transportation companies due to the insight it gives over the use for their services. Logs of the customers' location data is an important source for such companies. However, such data is not collected and is privately owned by the individual customers. To find the customers' travel patterns, their location data requires to match a coded map of transportation network.

This paper introduces a novel solution that automatically collects location and time data from the customers without the need for the customers to actively submit the data. This paper presents an innovative pre-processing and post-processing of location data. The pre-process, cleans and anonymizes the parts of the data that are addressable to the individual customers, and optimizes the data for later processes. The proposed solution, by using classification techniques, is able to match the customers' location data to coded transportation networks and yield high accuracy of travel matching, even with anonymized incomplete data. Examples are provided on the location data collected in Arendal area, Norway. The matching accuracy of the presented solution is relatively high which is indeed promising results.

Acknowledgements

This paper presents a Master's Thesis carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn.

I would like to use this opportunity to express my gratitude to a number of people. Firstly, my supervisors Associate Professor Doctor Morten Goodwin and Doctor Andreas Häber. They provided me with full-time support and helped me throughout the whole project. I hope to get the chance of working with this great team again. This project is proposed by Intelecom Group AS in Arendal, Norway. I would like to thank them for giving me the opportunity of this cooperative project and working with them at their local office. This Master's Thesis is a part of our cooperations' results.

Last but most certainly not least, I want to appreciate the support of my dear wife, Anna, who kept me motivated, always believed in me and gave me the extra energy to study and work hard during the recent two years of my life, especially while I was working on this project. I dedicate this paper to you. I love all of you with all of me.

Contents

Abstract i

Acknowledgements ii

Contents iii

List of Figures v

List of Tables vi

1 Introduction 1
- 1.1 Problem Statement 3
- 1.2 Problem Solution 4
- 1.3 Importance of the Topic 4
- 1.4 Background 5
  - 1.4.1 Classifiers 6
    - 1.4.1.1 Decision Tree 6
    - 1.4.1.2 Support Vector Machine 6
    - 1.4.1.3 *k*-Nearest Neighbors 6
  - 1.4.2 The Weka Java Library 7

2 State of the Art 8
- 2.1 The Sources of Errors in Satellite Location Sensors 8
- 2.2 Outliers in Spatial Data 9
- 2.3 Spatial Data Anonymization 9
- 2.4 Spatial Data Mining 10
  - 2.4.1 Hot Routes 10
  - 2.4.2 Finding Popular Routes from Spatial Data 11
  - 2.4.3 Learning Significant Locations from GPS 12
- 2.5 A Data Mining Approach on Route Planing 13
- 2.6 The Problem of Map-Matching 13
  - 2.6.1 Some Map-Matching Algorithms for Personal Navigation 14
  - 2.6.2 Map-Matching of Large GPS Datasets 15

3 Solution 17
- 3.1 Data Collection 19
  - 3.1.1 Data Store 19

3.2 Acquiring the Data for Classification 20
3.2.1 Training Data 21
3.2.2 Testing Data 22
3.3 Route Mapping 23
3.3.1 Removing Outliers from the Data 23
3.3.2 Data Anonymization 25
3.3.3 The Different Approaches Towards Classification 26
3.3.3.1 Employing Classification Algorithms 26
3.3.3.2 Map Matching to Pre-Coded Maps 27

4 Results 28
4.1 Extracted Data for Classification 28
4.1.1 Training Data Results 28
4.1.2 Testing Data Results 30
4.2 Results of Applying the Solution 31
4.2.1 Removing the Outliers 31
4.2.2 Anonymizing the Data 33
4.2.2.1 Anonymization on All of the Data 33
4.2.2.2 Anonymization on Individual tracks 35
4.2.3 Applying the Classification Algorithms 35
4.2.3.1 Ideal Classifiers' Performance 36
4.2.3.2 Classifiers' Performance Including Unwanted Data 37
4.2.3.3 Classifiers' Performance After Applying the Proposed Algorithms 38
4.2.4 Map-Matching with Open Street Maps 39

5 Discussion 40
5.1 The Proposed Algorithms' Performance 40
5.1.1 Outliers Removal 41
5.1.2 Anonymization 41
5.2 Map-Matching 42
5.2.1 Classification Algorithms' Performance 42
5.2.2 TrackMatching's Performance 45

6 Conclusion 47

7 Future Work 49

Bibliography 51

List of Figures

1.1 A mobile ticket application selling a season ticket. 3

3.1 Flowchart of the prototype solution. 18
3.2 Screen shot of the developed mobile application 20
3.3 End-to-end architecture of the solution. 21
3.4 An example of sorting the bus stations 22

4.1 The five extracted bus stations in Arendal, sorted in by the bus' visiting order. 29
4.2 The three routes that data is collected on them. 30
4.3 Before applying the outlier removal algorithm. 32
4.4 Outliers removal with $n = 3$. 32
4.5 Outliers removal with $n = 2$. 32
4.6 Outliers removal with $n = 1$. 32
4.7 Outliers removal with $n = 1/2$. 32
4.8 Outliers removal with $n = 1/3$. 32
4.9 Outliers removal with $n = 1/4$. 33
4.10 The data series without outliers. 33
4.11 Before applying the anonymization algorithm. 34
4.12 Anonymization with $n = 1 1/2$. 34
4.13 Anonymization with $n = 1$. 34
4.14 Anonymization with $n = 1/2$. 34
4.15 Anonymization with $n = 1/3$. 34
4.16 Anonymization with $n = 1/4$. 34
4.17 Before applying the anonymization algorithm. 35
4.18 After applying the anonymization algorithm with $n = 1$ 36
4.19 Comparison of the accuracy of applying the k-NN algorithm with different values of k. 37
4.20 Visualization of the DTree. 37
4.21 An example of TrackMatching. 39

5.1 The precision and recall of the classification results after applying the outliers removal and anonymization algorithms. 44
5.2 Comparison of precision and recall for k-NN algorithm. 45
5.3 Comparison of precision and recall for DTree algorithm. 45

List of Tables

3.1 Equation 3.1 explanation. 19
3.2 Algorithm 3 variables explanations. 25

4.1 Number of bus stations in each line. 29
4.2 Information about collected data. 31
4.3 The effect of value n in removing outliers. 31
4.4 The effect of value n on each route in removing the outliers. 32
4.5 The effect of value n on each route in anonymization. 34
4.6 The effect of value n in anonymization on all the data series. 34
4.7 Classifiers' performance in the ideal situation. 36
4.8 Classifiers' performance including the unwanted data. 38
4.9 Classifiers' performance after applying the outliers removal and anonymization algorithms. 38

5.1 Confusion matrix for DTree without considering the unwanted data. . . . 43
5.2 Confusion matrix for k-NN without considering the unwanted data. 43
5.3 Confusion matrix for DTree considering the unwanted data. 44
5.4 Confusion matrix for k-NN considering the unwanted data. 44

Chapter 1

Introduction

The introduction of the new generation of smart phones, such as iPhones and Android phones, is changing the way that their consumers behave in their daily life. For instance, how they accomplish their tasks and how their habits change. Such smart phones are taking more responsibilities day by day and new applications are released for them on daily basis. Billions of individuals are the users of smart phones and the population of these individuals is growing hourly. The number of installed sensors on decent smart phones is not countable by the fingers of two hands. These sensors are growing in numbers and they become more powerful as their new generations take over. The smart phones' ability of observing the environment around them, opens up a research area with huge potential of finding novel information.

Location sensors are amongst the oldest members of the smart phones' sensors club. As of possibly any other sensor, extracting data from them can result in new solutions capable of making the people's lives easier. E. g., predicting a route that an individual travels on a certain weekday and giving directions before the user asks for it. Online traffic information and estimated travel time are a few instances of useful information that the user can be provided with. However, continuous operation of sensors yields in more power consumption (less battery life) and location sensors are not an exception. The more online data that can be collected from the users, the better information quality they can be provided it. But currently, this is a tradeoff between power consumption and consistent data collection.

Transportation companies are interested to find their customers flow within their transportation network. Such companies want to know how a customer gets from point A to B. To elaborate, they want to know how a customer arrives to point A, and what are the potential in between connections to reach the point B. As one can realize, location and time are the basic characteristics of such information. In this domain of research, location and time data are referred as spatial and temporal data [1]. Mobile ticket applications are of the software solutions that transportation companies offer. The customers can use such mobile applications for purposes such as finding the available services around them and buying their desired tickets online.

Season tickets are of tickets that a customer can buy. Such tickets are valid for a particular period of time and the owner can travel unlimitedly within that period. Figure 1.1[2] shows a mobile ticket application selling a season ticket (valid for 24 hours). The season ticket owners, need to repurchase whenever their season ends. The customers who buy such tickets, become anonymous the transportation system. Transportation companies cannot afford to lose track of such customers. Because the customers that travel often, buys such tickets and this is the biggest share of transportation companies' sales. Thus, tracking the mentioned customers becomes a challenge. It is very valuable to know the zones that are used frequently. The importance of this fact is elaborated in section 1.3.

There exists solutions to this problem that are worth to be mentioned here. Some operators require the customers to validate their tickets on every hop of their travel. The tickets are validated by techniques such as swiping or showing an animation on the mobile phone. Especially for subway transportation, the exit and entrance gates only open up by providing a valid ticket. One can count the customers and find the information about their ticket by extracting data from such validation techniques. Moreover, a new solution for customers counting, namely automated passenger counting, is implemented in some trains in Norway. This solution counts the customers when they enter the train (pass the train's door) but it is expensive to implement and it cannot distinguish the ticket type [3].

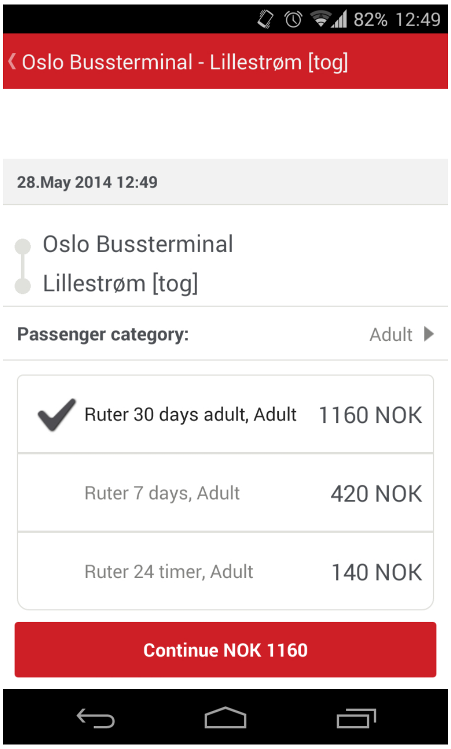
There are new concerns for the users to employ new software solutions, especially when the new software deals with the customers' privacy and security. The users must trust the service provider and grant the permission to them for accessing their data. The users must be assured that their data is kept confidential and their privacy is respected. The private spatial data of the users is very important to them. Service providers must keep the users anonymous and untraceable [4]. For example, users do not desire that unwanted people have the possibility of finding their home/work location. Such places' location must stay confidential to the users. The assurance of confidential data handling, from the service providers, encourages the customers' will to trust the service providers.

[1] In this paper, these terms are often used from now on.

[2] Image is chosen just as an instance, it can be from any transportation company offering periodic tickets.

[3] http://www.tu.no/it/2014/01/29/matematisk-verktoy-skal-gi-kortere-stopp-pa-togstasjonene

[4] Norway's customer privacy policies and industry standard for e-tickets:
http://www.datatilsynet.no/Teknologi/Sporing/



82% 12:49

‹ Oslo Bussterminal - Lillestrøm [tog]

28.May 2014 12:49

Oslo Bussterminal

Lillestrøm [tog]

Passenger category: Adult ▶

✓ Ruter 30 days adult, Adult	1160 NOK
Ruter 7 days, Adult	420 NOK
Ruter 24 timer, Adult	140 NOK

FIGURE 1.1: A mobile ticket application selling a season ticket.

In addition to the mentioned assurance, one can offer incentives to users in order to encourage them to use new software solutions. For instance, free membership of premium services for a period of time.

This paper is written in 7 chapters. The rest of the current chapter, discusses the problem of this project and the approach towards solving it. Importance of the topic and a background of the information, that are often referred in this paper, are followed respectively. Chapter 2 explores the related work in this research domain. Chapter 3 gives detailed information of the proposed solution. Chapter 4 shows the results of applying the solution on collected spatial data. Chapter 5 discusses the achieved results and provides an overview of the solution's performance. The contributions and the paper conclusion are brought in chapter 6. And finally, the way to move forward from this project is presented in chapter 7.

1.1 Problem Statement

The problems to study and solve in this project are:

1. How to automatically collect spatial and temporal data from customers without neither special equipment nor interaction from the customers?

2. How to make the customers anonymous and untraceable?
3. How to automatically match the routes that customers are travelling on based on the collected data?

1.2 Problem Solution

In order to tackle the first problem, customers must carry a sensor that is capable of communicating with resources that report spatial data such as location satellites. Thanks to ubiquitousness of smart phones equipped with such capability, customers do not need to carry an additional device. To solve the first problem, a lightweight mobile application is developed. This application is capable of fetching spatial and temporal data and sending it to a central database where it can be accessed later.

In second problem, all the data collected from the customers should not be addressable back to them. They should stay anonymous so that their privacy is respected. This policy, in addition to providing extra security, can be also an incentive for the customers to grant the permission of collecting their data.

Last but not least, the proposed solution matches this data to the actual transportation tracks. To perform the mentioned matching, the solution employs classification algorithms. It is desirable to find the optimal classification algorithm that are suitable for this study.

Note that this study deals with the data that matches the routes that transportation companies offer their services on. Individual and private travel patterns are not within the scope of this project. However, mining individual travel patterns can lead to creative applications. A few examples of such applications are mentioned in the next chapter.

1.3 Importance of the Topic

People prefer to carry less number of equipment with them. They tend to carry one device that is capable of performing many tasks. Since the invasion of mobile phones market by smart phones, one can see the new payment methods that a smart phone carries out. The goal of these methods is to replace physical payment methods such as credit cards and cash money. Google Wallet and PayPal Wallet are instances of such methods. Mobile ticketing applications are developed for the same purpose. To replace physical payment methods such as tickets, pre-paid and periodic cards. They make the purchasing procedure quicker as the customers select their desired plan of transportation before they get on the transportation vehicle. Depending on the transportation system, there may exist a ticket verification point to check the validity of the tickets. This process can be carried out by using QR codes or the NFC sensor on smart phones. In addition, the transportation vehicles operators also want their customer to use such methods. They do not want to watch the cash money in their bags all the time to keep it away from robbers. Therefore, as mobile ticket applications become more popular, they provide a novel area of research that calls for innovative ideas.

The scenario that basically conveys this project is the challenging task of finding the most used travel routes, especially when the customers are using season tickets. The pattern of the most used routes can help the transportation companies to realize the flow of their customers. This kind of information, assist them to allocate their resources (vehicles) to the routes that demand the most. In addition to that, such information can be employed for marketing purposes in the sense of offering new schemes to the customers that yields in more profit for the companies. But the reason that makes the season tickets more valuable to track is the fact that the customers who buy such tickets, become unknown to the system. They can travel wherever they want within a region and use such the ticket as much as they want. Moreover, such ticket owners are usually the majority of the customers that transportation companies' lives depend on.

1.4 Background

This project develops a *third-party* application that runs on smart phones with a particular operating system. The application can be installed on smart phones and it collects spatial and temporal data and sends it to a remote database. This mobile application fetches spatial data from location reporting resources including location satellites. Example of location satellites are GPS[5] and GLONASS[6] that are available to public and are popularly used by the mobile phone manufacturers.

The terms *outlier* and *anonymization* are widely used in this paper. In statistics, an *outlier* is an observation point that is distant from other observations [7]. And *anonymization* means the process of making the observations anonymous and not addressable to individual customers.

This rest of this section describes the classification algorithms that are employed in this paper. Various classification algorithms are tested and the best one are chosen. These

[5] Global Positioning System, http://www.loc.gov/rr/scitech/mysteries/global.html

[6] Globalnaya navigatsionnaya sputnikovaya sistema or Global Navigation Satellite System, http://www.astronautix.com/craft/glonass.htm

[7] http://en.wikipedia.org/wiki/Outlier

algorithms are k-NN, DTree and SVM and are explained respectively. This section ends with a summary of the employed library to apply the algorithms.

1.4.1 Classifiers

In machine learning [1], determining the class membership of an object is called classification. Classification is either done by using a training set with labeled objects or by defining new classes based on (dis)similarities between objects. To implement classification, an algorithm called classifier is used. In this paper, the algorithms are applied with supervised learning approaches (provided with training data.).

1.4.1.1 Decision Tree

DTree is a predictive model that maps observations to conclusions for an item on its target value. In this tree, class labels and conjunction of the extracted features are represented as leaves and branches respectively. Decision tree is one of the most successful algorithms used in supervised classification learning. For numeric values, which are the case in this study, decision tree defines a numeric limit on its leaves, the values that are larger than this limit go to one branch and the smaller ones go to another.

1.4.1.2 Support Vector Machine

SVMs are amongst supervised learning models with an associated learning algorithm to analyze data and recognize patterns. They are applicable in classification and regression analysis tasks. Basic SVM is a non-probabilistic binary linear classifier that takes a series of input data and assigns each given input in one of the two possible classes as the output form. In this basic form, classes are separated by a line in 2D space with a gap as wide as possible. The line that defines the boundary is called hyperplane, and every hyperplane has a margin that defines the ranges of its boundary.

1.4.1.3 k-Nearest Neighbors

This algorithm is amongst the simplest machine learning algorithms. It predicts the objects' class memberships based on the closest examples in training series. The value k, determines the number of neighbors that vote for an object's class membership. k-NN is an instance-based or lazy type of learning, i.e, all the computation is postponed until classification. Determining the best k is dependent on the structure of data. This algorithm, for our data series will measure the Euclidean distance (extracted feature from raw data) between the point to be predicted and k points from the training series. Then, assigns the asked point to a class with the majority of neighbor points of that class. Larger values of k neutralize the effect of noise on the classification, but they yield to less distinct boundaries. If k=1, the algorithm is called nearest neighbor.

1.4.2 The Weka Java Library

The Weka [2] Java library is developed by the University of Waikato. It has a collection of machine learning algorithms for data mining tasks. This library provides various means to classify or cluster data series based on various algorithms.

Chapter 2

State of the Art

In this chapter, the prior work in the domain of this research is reviewed. Moreover, the existing solutions on the problems that this paper encounters are discussed. Section 2.1 explains the reliability of location reporting sensors. Section 2.2 shows how the potential unreliable data (outliers) affects the quality of the data series. Section 2.3 presents the most common approaches on spatial data anonymization. Section 2.4 reviews the prior work on using data mining techniques on spatial data. Section 2.5 describes a paper that employes data mining techniques for navigation applications. And finally, section 2.6 explains the problem of map-matching and reviews some existing solutions on it.

2.1 The Sources of Errors in Satellite Location Sensors

Satellite location sensors have never been errorless, their performance depend on factors such as the number of satellites they can find at time. To date, GPS satellites have been the most popular location reporting satellites. However, in addition to GPS satellites, descent location reporting devices use GLONASS satellites that are in operation for a few years. Combining the received signals from both satellites yields to more accurate location information. Moreover, assisted GPS, that is now widely used especially in mobile phone devices, helps the location reporting devices to get a quicker GPS fix by exchanging online data via a network [3].

In [4], the authors overview the GPS measurements errors. The reported coordinate has an accuracy approximation of about 300 meters. Various factors affect the accuracy such as: 1- the capabilities of the user's GPS receiver, 2- the number of satellites in view and their distribution position regarding the receiver's location, 3- the particular position that the users are roaming and 4- how real-time the position is requested to be reported.

However, even if one is provided with an ideal situation and thus mentioned errors are minimal, there are still measurement errors that can be divided into three groups: 1- satellite related errors, 2- propagation medium and 3- computation errors of receiver [5].

2.2 Outliers in Spatial Data

As reviewed in section 2.1, location reporting sensors can report inaccurate data. Such errors may confuse the data mining algorithms. A research study in 1994 [6] addresses this problem. The authors propose clustering data mining algorithms to mine spatial data. The algorithms mine large databases of spatial data to find interesting patterns. As the authors are aware of the negative effect of outliers on the results, they find and remove them before the data is mined. They remove all the objects that are more distant than a particular threshold. The threshold is dependent on the number of partitions that the spatial data is divided into. All the data that exceeds the threshold is removed from the data series, and the cleaned data from outliers, is inputted into data mining algorithms.

Moreover in [7], that is a study on spatial data mining, the authors focus on detecting outliers as features for data mining. Although outliers generally represent inconsistent data, they can lead to find interesting patterns. For example, severe weather conditions or voting irregularity. Regardless the reason of finding the outliers, they have to be found initially. The mentioned study, maps the data points on a distribution diagram. The points lying outside of the mean value of the distribution, plus and minus two times of standard deviation are considered outliers.

2.3 Spatial Data Anonymization

L. Sweeney in 2002 [8], proposed a concept namely k-anonymity that attempts solving the following problem: "Given person-specific field-structured data, produce a release of the data with scientific guarantees that the individuals who are the subjects of the data cannot be reidentified while the data remain practically useful." [8]. He defines two methods to solve the problem: 1- suppression and 2- generalization. The first method replaces the data that is addressable to an individual with asterisks while the second method replace such data with a broader category such as an age range. The bigger the value of k is, the more anonymization is applied to the data.

A paper published in 2005 [9], applies the k-anonymity technique to spatial data. The cited paper proposes two methods for spatial data anonymization: 1- perturbing the spatial data by replacing it with a large spatial region (range) and 2- by delaying the message containing the spatial data (only for avoiding the real-time tracking). However, generalizing too much of the spatial data that has a low resolution yields to losing the wanted data. The authors of the mentioned paper propose an algorithm that transforms the spatial data into a graph with nodes representing the coordinates. Then, depending on the assigned value to k, the nodes that are closer than a particular value to each other are merged into one node. The transitions between nodes represent the direction of the moving objects. Despite the simplicity of this algorithm, it can provide satisfactory data anonymization in many case scenarios. There exists several papers that approach the anonymization by using similar techniques mentioned in this paragraph [10][11][12].

2.4 Spatial Data Mining

In general, data mining (also called knowledge discovery) is finding the hidden patterns and features of a data series. Similarity or dissimilarity of a series of data can be a pattern. In spatial data mining, one is interested in finding the ir/regular routes. Such findings can lead to many valuable information [13]. This section reviews some existing work in this domain that has lead to innovative solutions. Section 2.4.1 describes a paper that employes a data generator to generate moving objects in a network of routes. The authors propose a solutions that identifies the busiest routes (hot routes) with the focus on efficiency and speed of the solution. Section 2.4.2 reviews a similar paper that employs real spatial data from users and finds the most popular routes. Section 2.4.3 represents a study that proposes a solution towards finding significant places.

2.4.1 Hot Routes

A paper published in 2009 [14] proposes a solution towards finding the most used routes (hot routes) in a road network. Various sensors for reporting the location of moving objects are employed to track them. Depending on the desired accuracy of these reports and the goal of the solution, the sensors can be different. For instance, for automatic pay tolls RFID sensors installed on the cars are employed. In some countries, police uses GPS sensors to track the trucks and to monitor their speed and etc. A potential application of this information can be finding the most used routes. Moreover, a system capable of real-time analysis, can tell the online traffic density. Such information can be used for emergency departments to find out about the accidents.

The mentioned paper focuses on motorways network, especially their intersections where the vehicles want to change their routes. The solution discovers the chains of these intersections as they are most likely to be hot routes. Because the authors did not have access to real data, they used a network-based data generator provided by [15]. This generator generates moving objects in a real-world city. The maximum speed and capacity of the road are the factors that are taken into account for generating the objects.

The generated data is from the cities in the US that are modelled with the data generator. By running the proposed algorithm on the generated data, the paper identifies the hot routes in these cities. However, the detailed results of the algorithm's behavior on intersections are not fully discussed. The algorithm finds a hot area and puts it as the starting point. It then searches for other hot areas and continues until it captures all of them within a defined radius. The minimum traffic is the other important factor in discovering such areas. These hot areas are chained together with the best estimation of the route shape and are introduced as a hot route.

2.4.2 Finding Popular Routes from Spatial Data

The pervasiveness of location reporting technologies such as GPS has provided large datasets of raw data. There is an opportunity for discovering valuable information about the behavior of moving objects that send out spatial data. The authors of [16] add semantic meanings to raw GPS data in order to mine interesting patterns from them: 1- Connecting a series of GPS points to each other which forms a trajectory. 2- Marking the geographic region that the users stay for a while as a stay point. 3- Recording the locations that are visited over an interval as location history. 4- Clustering the spatial dataset into regions in a hierarchical graph.

The proposed solution of the mentioned paper consists of three main components: 1- representation of location history, 2- exploration of user similarity and 3- location recommendation. However, the novelty and contribution of this paper is exploring similarities between the users who send the GPS data logs. The first component, parses the spatial data and forms the trajectories for each user. It extracts the stay points of the users' trajectories and clusters them into different spatial regions in a hierarchical graph. Each user holds an individual graph. The second component, which is the most important one, performs offline user similarity exploration. It searches for the same graph nodes in two hierarchical graphs that represent two users. It then retrieves a sequence containing the graph nodes including the information of arrival and leaving time of each stay point. The time interval between the two nodes, each on a graph, is extracted from the sequence. The described process finds similar sequences that have the same visit sequence of places with a similar time interval. The longer the sequences are, the more similar they are to each other and the higher score they get.

The last component of the mentioned solution, represents the users with the similar sequences to themselves, which can be interesting to them. When users send requests to the system, it retrieves a set of sequences with the closest scores to the requests. This set of sequences are ranked and represented to the users. The authors of this paper employ actual collected GPS data for the experiments. Over a 6 month period, 65 volunteers use the solution and according to the authors' claims, their solution outperforms traditional systems for location recommendation.

2.4.3 Learning Significant Locations from GPS

The technologies that are designed to assist the individuals must be intelligent to be truly assistive. They must learn the users' actions and predict them. A paper by D. Ashbrook and T. Starner [17], demonstrates how significant locations can be learned from GPS data. Significant locations are the locations which a user frequently visits.

The experiments in the mentioned paper have been done in separate stages with different scales of time and data volume. The authors, in their first experiment in 2001, create a location modelling system based on the GPS data of a single user. The data is collected in four months. In the next stage, they collect GPS data for six individuals over a period of seven months. An important fact about learning systems is that one must input some data to them. And based on this data, the algorithms of the system produce the results. Therefore, one can never guaranty that the results are entirely true. However, the algorithms developed in the first stage of the described paper proved to be effective in the second stage too.

In mentioned paper, similar to what described in 2.4.2, stay points are extracted from the GPS data. Stay points are most likely to be indoors where GPS signals are not available. The authors use this fact to find the significant places such as home and work. A stream of data are recorded before the user enters an indoor place, then there will be a gap in time. This gap is between the last GPS signal and the next GPS signal when user exists the indoor place. The places with a time gap of over ten minutes are considered significant here. This time value is the result of the analysis of the GPS data that the mentioned paper is provided with. Smaller values can mislead the solution to wrongly identify urban canyons and tunnels, which can result in a signal loss up to five minutes, as significant places. The significant places, that are just meters away from each other, create a location. The authors employ a Markov model and place the locations as nodes. The transitions of these nodes represent the probability of moving between them. When the users send queries to the system, based on their location in the model, the next locations with the highest probabilities are shown to them.

2.5 A Data Mining Approach on Route Planing

Traditional route planing systems such as Google Maps give directions based on real-time data. For example, they obtain traffic information from online resources and calculate the fastest path based on that. An innovative research study, published in 2007 [18], uses historical spatial data to give the best directions. According to the mentioned study, the routes that the end-users of navigation systems have used are reasonably chosen. The history of such routes provide a useful database to give directions to the users who currently want to use the navigation system. For instance, at the night time, users prefer to avoid the routes that are crime prone. If one mines such data to give directions to the real-time users, they will be given directions in a route that avoids crime prone areas. Moreover, factors such as driving speed, weather conditions and local rush hours will be considered in such navigation system.

A new solution, explained in the mentioned approach in paragraph above, needs to learn from historic traffic data. The traffic data must be collected over an adequate period of time from users to shape a reliable source for data mining. Authors in [18] develop a path-finding method based on mining the traffic. The method, from the historic traffic data, mines speed and driving models. Whenever a new request of directions including a start point and an end point is given to the method, considering the factors such as time and weather, the method calculates the fastest route. The novelty of this method is taking into account the frequent behavior of the users who have chosen a similar route before. However, the real-time factors such as online traffic data are also considered.

The mentioned study is able to use generated data on predefined route networks as well as the data collected from the cars that are tagged for toll collection. Such database provided, the road network is partitioned into several large regions and then the large regions are divided into sub regions. This is to organize the road network around a well-defined hierarchy of roads. Each region potentially has different characteristics than the others. The driving patterns of all the regions are extracted by using data mining techniques. The study's solution, extracts various patterns for different times in a year and different times of the days. Then by considering the online traffic data, suggests the fastest directions to the user. Not only the suggested route is the fastest path, but it also reflects the observed driving preferences in particular regions.

2.6 The Problem of Map-Matching

According to [19], "The general purpose of a map-matching algorithm is to identify the correct road segment on which the vehicle is travelling and to determine the vehicle location on that segment." Depending on topological characteristics of a region, map-matching algorithms perform different than each other. A paper by A. Quddus et al. [20], reviews many existing map-matching algorithms, groups them into 4 categories and compares them to each other.

The four categories, represented by A. Quddus et al., are as following: 1- geometric analysis, 2- topological analysis, 3- probabilistic algorithms and 4- advanced algorithms. The performance of map-matching algorithms, regardless of what category they belong to, depends very much on the quality of the spatial data. As the map-matching algorithms advance, they become more accurate in terms of correct link (route) identification and two-dimensional horizontal accuracy (the distance from the actual point.). Section 2.6.1 explores the potential approaches on the map-matching problem. It proposes a number of algorithms that solve the map-matching problem via all the explored approaches. Last but not least, section 2.6.2 describes a project with actual collected data in Zurich. The authors of this paper propose a solution that match a user's location to pre-coded maps. In this paper, the solution proposed in section 2.6.2, is employed and discussed in chapters 3 and 4.

2.6.1 Some Map-Matching Algorithms for Personal Navigation

Third generation personal navigation assistants, are a type of devices that provide a map, the user's location and directions towards a destination. However, to provide directions from point A to point B, the user's location must coincide with a route. The research study [21], describes some of the algorithms in this domain. The accuracy levels of map-matching classifies the algorithms. Ideally, this process should be performed quickly and accurately. The usual approaches towards map-matching are, 1- point-to-point 2- point-to-curve and 3- curve-to-curve matching.

Moreover, [21] views the problem of map-matching as either a 1- search problem or a 2- statistical estimation. The first view is to solve the problem of matching a point to the nearest node in the network. The latter view considers fitting a curve to a sequence of points. This curve is limited to lie on the network (actual map). The cited paper aims to combine these two views and as the result, four algorithms are proposed. The first three algorithms are very similar to each other. Finding the closest distance between a GPS point and a node on the map is the basis of all of them. However, the authors fail to mention the usage of nearest neighbor algorithm. They are basically using the same logic without mentioning it. The last algorithm has more of complexity as it uses the curve-to-curve matching approach.

To evaluate the performance of the aforementioned algorithms in this paper, the authors construct their own testing data consisting of four routes. These routes have different lengths and network complexity. The matching rates are relatively high in all the routes using these four algorithms (more than 60%), however, there is no algorithm that entirely outperforms the other three. Three different algorithms score the best for four different routes. This shows the the performance of the algorithms relies on the routes' structure. It is concluded by the authors that they can not know whether the proposed algorithms will have the same results for other route networks and this fact requires further study.

2.6.2 Map-Matching of Large GPS Datasets

A study in Zurich [22], proposes a solution to match the streams of GPS coordinates to a coded map of transportation network. The proposed solution identifies the routes that were taken. Because the amount of data can get relatively large, it is very important to consider the aspects of the algorithms that affect the computational time. In the mentioned study, GPS loggers record the spatial displacements of users every second and the stored raw data is downloaded and processed offline (at the server side).

The network (actual map) that the spatial raw data is mapped to, consists of sets of nodes that are connected by directional links. The goal is to find the best estimation of the sets of connected links (path) that the user actually takes. The authors' focus is on solving the problem of matching a path to the closest GPS points in with minimum computation time. It is claimed that the intuitive methods such as nearest-node search of nearest-link search do not insure the consistency. These methods do not take into account the correlation between continuous coordinates. Therefore, the authors introduce a new algorithm. The first step towards their solution is finding the closest distance between a point and a road segment. However, they fail to use or mention the existing algorithms such as k-NN that deal well with this problem. Indeed, the logic of this part of their solution is very close to k-NN. To find the distance between a GPS coordinate and a path, they measure the euclidean distance between that coordinate and the closest point on that path. If the coordinates set is on any point of the path, the distance is zero. Due to potential interruptions in the GPS data streams, usually the complete route will not be match to a single continuous path. Therefore, it is instead matched to a sequence of paths. Connecting these sequences is not of the authors' concerns.

By using commercial coded networks (maps), the paths that a car takes are identified. The results of the mentioned study are focused on the map-matching's computation time. It is claimed in this paper that the results are not comparable to other studies since the resolution of the coded networks vary. If they are to compare, other solutions should apply on the exact same network. The accuracy of the results in an experiment collecting 2.5 million of GPS points depend on the resolution of the employed coded network. In high resolution networks, the maximum accuracy is about 10 matches per path. The accuracy is dependent on factors such as network resolution and collected GPS points. In high resolutions networks, the GPS points are mapped with more details. As the number of the set candidates (GPS data fed to algorithm each time.) increases, the computations time gets longer. According to the results, limiting the set candidates to 30 yields to a balance between computation time and accuracy. The computation speed of this algorithm is about 1000 times quicker than the collection time. This shows that map-matching of a car trip can be performed each second and the entire trip can be processed in a reasonable time.

Chapter 3

Solution

This chapter explains the solution towards collecting data, pre-processing the data for optimality of classification, anonymizing the data and classifying it by various algorithms. The solution starts from collecting spatial data from mobile devices and storing them in a remote database. This part, provides adequate data to use as testing data for applying data mining techniques. It is explained how the bus stations' coordinates are used as training data. The proposed algorithms in the next sections, describe how by comparing the testing and training data, one can remove the outliers from testing data and how data becomes anonymous and not addressable to the customers. This chapter also describes how the classification algorithms are employed to match the testing data to training data.

Figure 3.1 illustrates how this paper tackles the problem. The solution should provide the classification algorithms with training and testing data. In "Data collection", the solution collects spatial data from customers while they are using the developed mobile application. Subsequently, in "Data store", this data is sent to a database on a remote server through Internet (section 3.1). At this stage, the solution can remove the outliers from this data series (section 3.1.1). This is an optional step and can be avoided to make the solution quicker. But only if the data is collected in perfect situations with reliable sensors. In the next step (Label the data), the data is labeled with the corresponding route that it belongs to. The solution employs this labeled data as the testing data for classifiers.

The steps highlighted with red squares on the right side of the figure 3.1 represent how the solution prepares the data for training the classifiers. The bus stations' coordinates, labeled with the correspondent region they reside in, are extracted and employed as the training data for classifiers. This is elaborated in section 3.2.1. In "Anonymize data?" step, the solution can perform data anonymization by comparing the provided testing and training data (section 3.3.2). One can choose not to anonymize data as if such data should be used in analyzing individual travel patterns. A comparison between two data series before and after anonymization is written in chapter 4. In the next step, various classification algorithms perform the classification task (section 3.3.3) and finally the results are visualized.

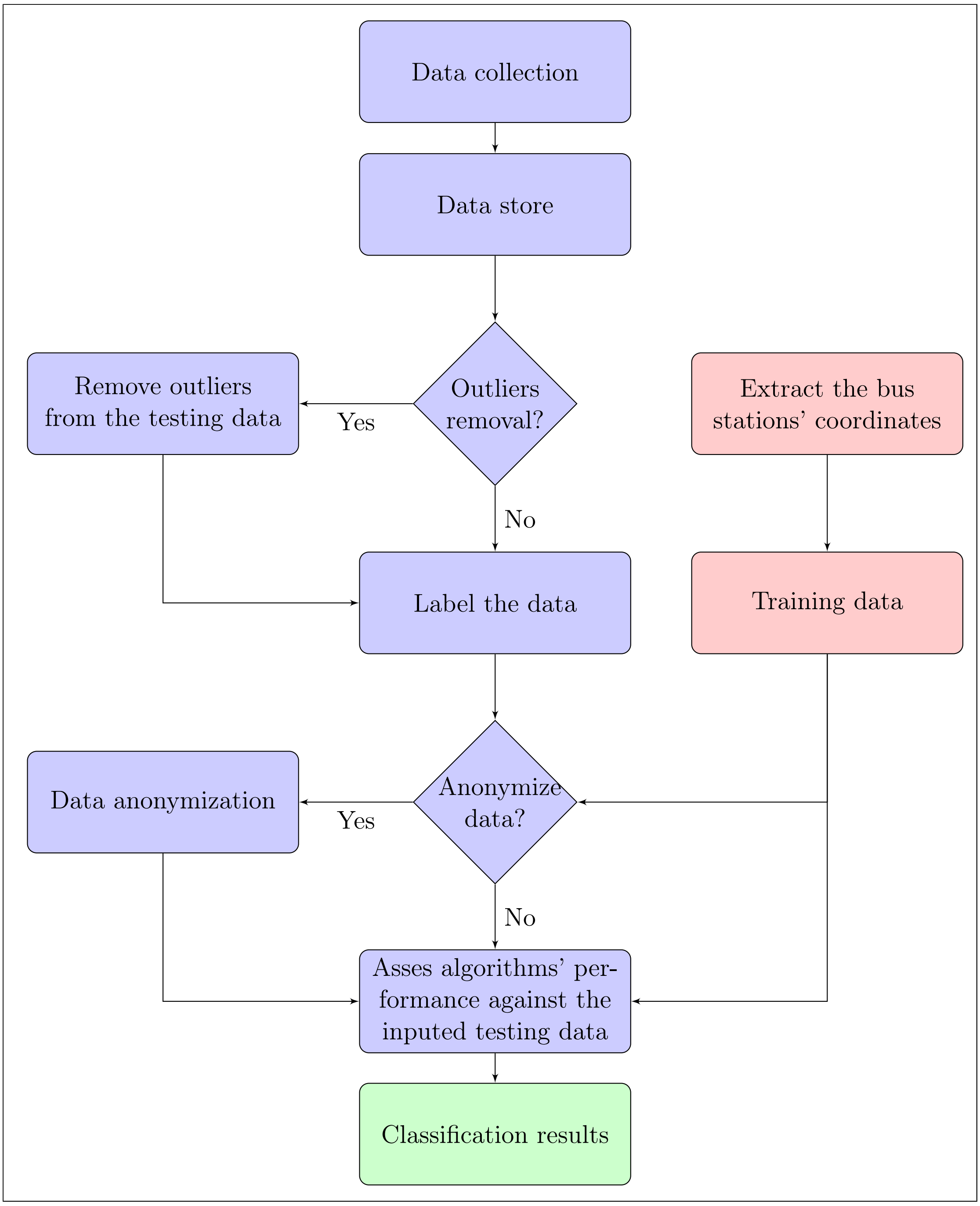

Data collection
Data store
Outliers removal?
Yes
Remove outliers from the testing data
No
Extract the bus stations' coordinates
Label the data
Training data
Anonymize data?
Yes
Data anonymization
No
Asses algorithms' performance against the inputed testing data
Classification results

Figure 3.1: Flowchart of the prototype solution.

Variable	Explanation
y_i	Latitude
x_i	Longitude
i_i	Identification field for labelling the sent data
t_{1i}	Time on the device
t_{2i}	Time on the database server

Table 3.1: Equation 3.1 explanation.

3.1 Data Collection

In order to collect spatial data for this project, an Android mobile application is developed. This lightweight application communicates with Google Play location services and fetches the coordinates. The advantage of using Google Play location services rather than communicating directly to the GPS sensor is that it can fetch the coordinates from various resources that are: 1- GPS satellites (also GLONASS satellites if the smart phone supports it), 2- network antennas and 3- wireless networks. However, as it was experienced in data collection phase, while GPS sensor is not available, the coordinates can become inaccurate. It also takes more time to report the coordinates when GPS sensor is offline. By utilizing this feature, the application can still send data while the smart phone is on a subway track or tunnel. But the reliability of this data must be measured.

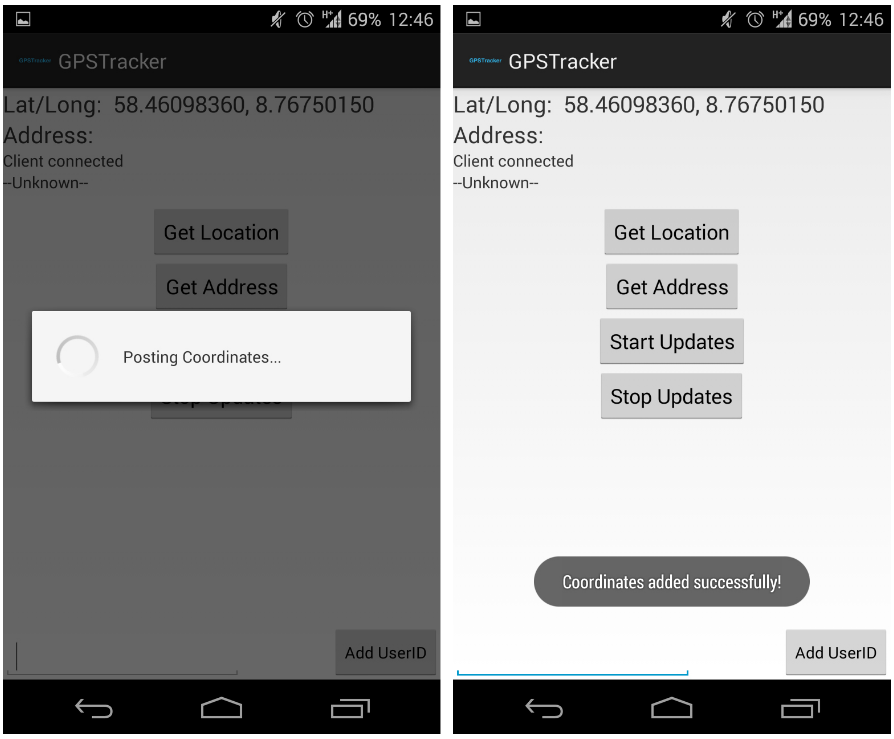
3.1.1 Data Store

The mobile application collects data and sends it to a remote database. The application sends a tuple of

$$(y_i, x_i, i_i, t_{1i}, t_{2i)} \tag{3.1}$$

every 5 seconds to the database. Refer to table 3.1 for the explanations of the variables. A mySQL database server is employed for this part of the solution. To keep it secure, it resides on a private server and is only accessible, from the outside, by a VPN connection. I. e. to extract information from the database server, one must connect to it via a VPN connection with SSL protocols that makes the database less vulnerable.

The data from the mobile application is sent to the database via a web sever. A PHP script deployed on a Apache web server takes the responsibility of receiving the data from the mobile application and sending it to the database. The web sever and the database server are on the same network infrastructure and thus they can see each other. The data is stored in a table in the database with correspondent rows to the mentioned tuple. On a successful trial of sending the data to the database, the web server sends a confirmation back to the mobile application.



Lat/Long: 58.46098360, 8.76750150
Address:
Client connected
--Unknown--

Posting Coordinates...

Lat/Long: 58.46098360, 8.76750150
Address:
Client connected
--Unknown--

Coordinates added successfully!

FIGURE 3.2: Application sends the coordinates (left) and receives a confirmation (right).

The status of successfully submitting the data to the database is shown on the mobile phone's screen can be seen in 3.2. Figure 3.3 shows the end-to-end architecture of the solution. As one can see, the mobile phone receives the spatial data from resources such as location satellites, Wi-Fi networks and mobile network antennas. The mobile application sends the data to the database via a web server and receives confirmation upon successful delivery. The solution, proposed in this paper, dumps the required data from the database, applies the outliers removal and anonymization algorithms, and finally applies the classification algorithms to the data and provides the results. The users of the solution, must be able to realize the data, thus the results must be visualized to them by employing adequate tools. For instance, a web application can fetch the results and illustrate them by the aid of diagrams and etc [1].

3.2 Acquiring the Data for Classification

The extracted data is stored in Attribute-Relation File Format (.ARFF) file format. This file format contains extra headings in comparison to Comma Separated Values file format (.CSV), otherwise, they are identical. The Weka Java library that performs the classification, handles .ARFF files and can output in this format as well.

[1] Developing such web application is not within the scope of this project. More can be found in chapter 7.

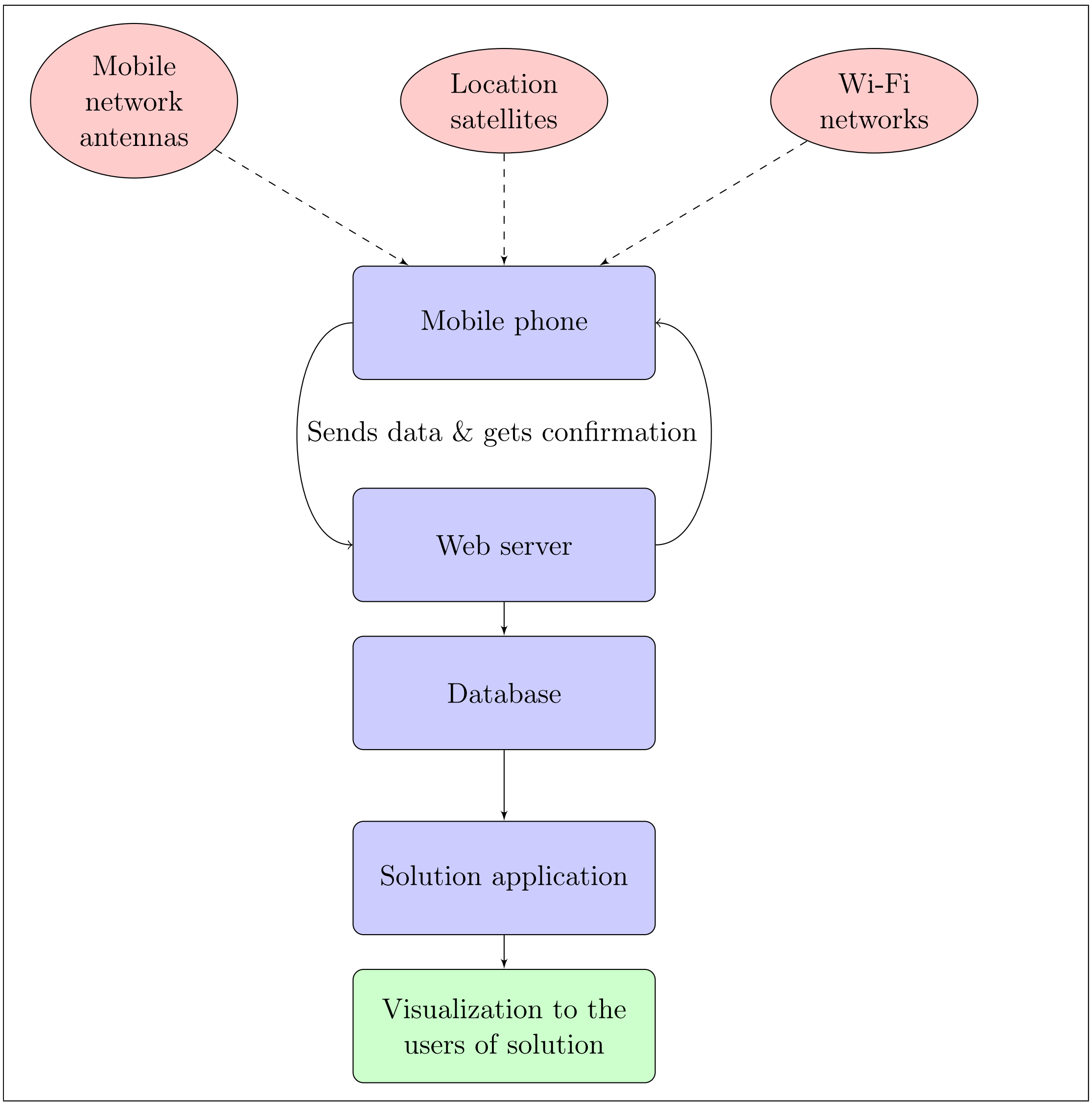


Figure 3.3: End-to-end architecture of the solution.

3.2.1 Training Data

This project is granted with the permission to access the official transportation data provided by the Norwegian Travel Information *(Norsk Reiseinformasjon AS* [2]). The coordinates of all the bus stations in Norway are available on this data centre. These stations are sorted in a big plain text file by provinces and cities. However, for each city, they are sorted alphabetically and it is not possible to find the sequence that a transportation vehicle visits them. Manually sequencing these coordinates is very time consuming. But it is inevitable as it is the best resource available at the time of doing this research.

[2] http://www.reiseinfo.no/

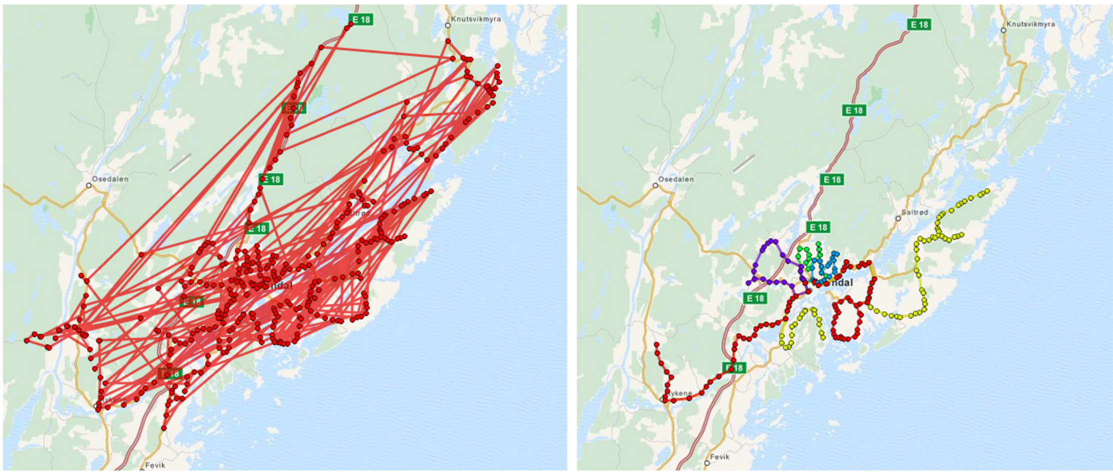


FIGURE 3.4: Raw bus stations coordinates (left) and sorted bus stations (right). Each colour on the right picture represents a bus line.

This study maps the stations' coordinates of the target study place (city) on an actual map. The bus routes data is available to the public. Therefore, based on the bus routes data, one can sequence the stations one after another. In this process, non-operative bus stops are filtered. Figure 3.4 illustrates an example of the output of this process and compares it with the raw data. This study employs the stations' coordinates for training the classification algorithms.

3.2.2 Testing Data

For this study, in addition to spatial data, the identification field is extracted from the database. The temporal data is not extracted as the implementation of the solution does not deal with that. However, this data can be useful for future works that are explained in chapter 7. The tuple structure is illustrated in 3.2.

$$(y_i, x_i, i_i) \tag{3.2}$$

Each tuple is saved in a new line in the .ARFF file. The value y_i is the latitude, x_i longitude and i_i represents what route each tuple belongs to. The i_i is validated manually to make sure it is representing the right route. This validation is possible by mapping the coordinates on a map and comparing them to the route they are on. This process makes it possible to measure the precision of the classification algorithms. It can also be replaced by a question mark if predicting the route's name is desired. These files are to measure the precision of the trained classification algorithms.

3.3 Route Mapping

This section explains the proposed algorithms that are developed in this paper. Section 3.3.1 describes a pre-process on the data that optimizes the further steps of the solution. Section 3.3.2 explains how this paper makes the data generic and removes personal information related to customers. Section 3.3.3 describes how the solution tackles the map-matching problem and lastly, section 3.3.3.2 explains how an alternative solution that is developed by prior researchers is applied to the data of this project.

3.3.1 Removing Outliers from the Data

Considering that location sensors are not errorless, they may report incorrect coordinates especially while they are trying to initiate a fix to satellites. In the application developed for the sake of this project, location can be reported even when the GPS sensor is not working. If this happens, the probability of incorrect location reporting increases. Therefore, based on the quality of the testing data, the solution's user can choose to use this feature before classification starts. Another application of this part of the solution is that if one desires to limit the data to a particular region. By having the central point of the desired region, one can choose to remove the data that lies outside of a defined radius.

The solution removes the coordinates that have a longer distance than a particular limit from the mean of all coordinates in the testing data. At this stage of the solution, training data is not employed. The algorithm 1[3] calculates the standard deviation (σ) of the testing data by using the equation shown in equation 3.3.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - \overline{z})^2} \tag{3.3}$$

Where N is the number of all the instances in the testing data, z_i represents the latitude or longitude of the instances and $\overline{z}$ is the mean of all the latitudes or longitudes in the testing data. The algorithm 1 uses equation 3.3 to calculate the σ for the latitudes and the longitudes of the given data series.

The algorithm 2, that follows after the algorithm 1, employs a for loop and removes any set of coordinates that lies outside of $n * \sigma_y \pm \overline{y}$ and $n * \sigma_x \pm \overline{x}$. The value of n, which is a ratio for σ, determines the extent of the target data set. For the definition of other

[3] Note that this algorithm is not a contribution of this study by itself. The purpose of explaining it is that how this study calculates two standard deviation for latitudes and longitudes.

Algorithm 1 Standard deviation calculation

```
procedure Standard deviation($\sigma_y, \sigma_x$)
    $sum_y \leftarrow 0$                                  ▷ Initialisation of variable $sum_y$
    $sum_x \leftarrow 0$                                  ▷ Initialisation variable $sum_x$
    $N \leftarrow$ number of instances
    for $j \leftarrow 1, N$ do                            ▷ $j$ is an instance of tuple $(y_i, x_i, i_i)$
        Select $y_i$ from the tuple $j$
        $sum_y \leftarrow sum_y + y_i$
        Select $x_i$ from the tuple $j$
        $sum_x \leftarrow sum_x + x_i$
    end for
    $^{sum_y}/_N \leftarrow \overline{y}$                 ▷ The mean value of latitudes
    $^{sum_x}/_N \leftarrow \overline{x}$                 ▷ The mean value of longitudes
    $SS_y \leftarrow 0$                                   ▷ Initialisation of variable $SS_y$
    $SS_x \leftarrow 0$                                   ▷ Initialisation of variable $SS_x$
    for $j \leftarrow 1, N$ do                            ▷ $j$ is an instance of tuple $(y_i, x_i, i_i)$
        Select $y_i$ from the tuple $j$
        $(y_i - \overline{y})^2 \leftarrow S_y$
        $SS_y = SS_y + S_y$
        Select $x_i$ from the tuple $j$
        $(x_i - \overline{x})^2 \leftarrow S_x$
        $SS_x = SS_x + S_x$
    end for
    $\sqrt{^{SS_y}/_N} \leftarrow \sigma_y$               ▷ Standard deviation of latitudes
    $\sqrt{^{SS_x}/_N} \leftarrow \sigma_x$               ▷ Standard deviation of longitudes
end procedure
```

values refer to section 3.2.2. The smaller the n is, the less sets of coordinates remain. This procedure counts as an optional pre-process in the prototype solution and can be deactivated by the user. However, this pre-process makes the solution more optimized as it removes some unwanted data. But of course, makes the whole solution to consume more time to run.

Algorithm 2 Outliers removal function

```
procedure Outliers removal(From the data)
    $N \leftarrow$ number of instances
    for $j \leftarrow 1, N$ do                            ▷ $j$ is an instance of tuple $(y_i, x_i, i_i)$
        Select $y_i$ from the tuple $j$
        Select $x_i$ from the tuple $j$
        if $(y_i > \overline{y} + n * \sigma_y)$ or $(y_i < \overline{y} - n * \sigma_y)$        ▷ $n$ sets the sensitivity
        and $(x_i > \overline{x} + n * \sigma_x)$ or $(x_i < \overline{x} - n * \sigma_x)$ then
            Remove instance $j$                           ▷ Removes the outlier from the instance series
        end if
    end for
end procedure
```

Variable	Explanation
$distance_{max}$	The longest distance between any coordinate in testing and training data.
$distance_{min}$	The shortest distance between any coordinate in testing and training data.
$\overline{distance}$	The average distance between all coordinates in testing and training data.
σ	The standard deviation of all the distances between all coordinates in testing and training data.

Table 3.2: Algorithm 3 variables explanations.

3.3.2 Data Anonymization

This project explicitly aims to keep only the data that does not interfere with the customer's privacy. Any data outside of the routes, that transportation companies offer their services on, is removed [4]. Differentiating between personal and public data is a difficult and sensitive job. One should not access the individual customer's location data. Though one can use security techniques such as IP hiding to make the customers unknown, yet it is not guaranteed that the location data is not addressable to individual customers anymore. This is because one can still identify the individuals by finding their starting point on the map which for instance addresses the customer's home or office location. Therefore, this study goes beyond this and removes such private data.

For this part of the solution, the euclidean distance between each instance of the training data and each instance of the testing data is calculated. A for loop inside another one, each iterating through testing and training data, performs this operation and calculates various characteristic values of these distances. These values are represented in table 3.2.

Algorithm 3 shows the approach of the prototype solution in making the data anonymous. The calculation of σ value is not illustrated in this algorithm because of a similar approach shown in algorithm 1. Similar to algorithm 2, n represents the ratio of σ and sets the sensitivity of anonymization [5]. The Algorithm 3, removes all the data from the testing data (private data) that lies outside of a radius of each training data instance (bus stations' coordinates). The results and examples of applying the aforementioned algorithms are explained in chapter 4.

[4] Private routes might be of an interest for other studies, especially the projects that deal with mining individual travel patterns.

[5] All of the mentioned algorithms in this chapter are implemented by Java programming language.

Algorithm 3 Anonymization function

```
procedure ANONYMIZATION OF THE TESTING DATA(data cleaning)
    $N \leftarrow$ number of instances
    for $j \leftarrow 1, N_j$ do        ▷ $j$ is an instance of testing data tuples $(y_i, x_i, i_i)$
        Select $y_i$ from the tuple $j \leftarrow y_j$
        Select $x_i$ from the tuple $j \leftarrow x_j$
        $N \leftarrow$ number of instances
        for $k \leftarrow 1, N_k$ do        ▷ $k$ is an instance of training data tuples $(y_i, x_i, i_i)$
            Select $y_i$ from the tuple $k \leftarrow y_k$
            Select $x_i$ from the tuple $k \leftarrow x_k$
            $y_j - y_k \leftarrow distance_y$
            $x_j - x_k \leftarrow distance_x$
            $\sqrt{(distance_y)^2 + (distance_x)^2} \leftarrow distance$
            if $distance_{max} < distance$ then
                $distance_{max} = distance$
            else if $distance_{min} > distance$ then
                $distance_{min} = distance$
            end if
        end for
        if $(distance_{min} \geq \overline{distance} + \sigma * n)$ or $(distance_{min} \leq \overline{distance} - \sigma * n)$ then
            Remove instance $j$        ▷ Removes the instance from testing data
        end if
    end for
end procedure
```

3.3.3 The Different Approaches Towards Classification

This paper applies two solutions to the collected data. Section 3.3.3.1 describes the approach that is developed in this project. In this approach, classification algorithms match the collected data to the **custom built map** that consists of the bus stations's coordinates. In section 3.3.3.2, the collected data is matched to existing coded maps.

3.3.3.1 Employing Classification Algorithms

When the training and testing data are processed and available, the chosen classification algorithms in this study match the testing data to the training data. The goal of classification is to find the best match for the testing data. In other words, one should be able to realize which actual routes the customer is travelling on. This project approaches the map-matching problem as a machine learning problem and tackles to solve it with classification algorithms.

Various classification algorithms are employed to perform the matching. The Weka Java library, that is used in this project, provides many algorithms that can be used for classification. The prototype solution trains the classification algorithms on the training data. For the testing data, it inputs either the modified or the raw testing data to the algorithms. The solution, measures the performance aspects of the classification algorithms and prints them out as the output of it. The results of the experiments are explained in chapter 4.

3.3.3.2 Map Matching to Pre-Coded Maps

This project, in addition to the developed solution, assesses using an externally developed solution that is not particularly designed for the purposes of this study. This solution matches the inputted test data to the widely developed open source Open Street Maps (OSM)[6]. This is considered as an alternative solution for this study. This external solution is accessible via a REST[7] API. One can send a series of coordinates together with their timestamps in XML formatted file to this API. This API, in return, provides the user with standard OSM routes formatted in XML. The OSM routes include a source, a destination and a node ID. This solution estimates the best matched routes based on an algorithm [8]. Fortunately, it only considers the main routes, i. e., the alleys are not considered (supported). This fact, anonymizes the data intuitively and as the transportation services are only offered on the main routes, still the goal of finding the actual routes based on customers' spatial data is achieved. The results of using this alternative solution are explained in chapter 4.

[6] http://www.openstreetmap.org/

[7] http://en.wikipedia.org/wiki/Representational_state_transfer

[8] Refer to section 2.6.2 for more details of this solution.

Chapter 4

Results

This chapter describes the results of applying the aforementioned algorithms and solutions proposed in chapter 3. Section 4.1 shows the mapped training and testing data and gives some information about them. Section 4.2 demonstrates the results of applying the outliers removal and anonymization algorithms and how they affect the collected data. Section 4.2.3 presents the results of map-matching by using the classification algorithms and shows how the outliers and anonymization algorithms affect the classification. And lastly, section 4.2.4 describes the results of map-matching by using an external solution.

4.1 Extracted Data for Classification

This section shows how this project prepares the data to be employed by the classification algorithms. Sections 4.1.1 and 4.1.2 explain the extracted data to be employed as training and testing data respectively. It is important to notice that the training data is considerably less than the testing data. Indeed, the amount of the training data for a region is constant while the amount of the testing data varies. This fact, makes the classifiers job more difficult as their knowledge of data is limited but they are asked to answer unlimited classification questions.

4.1.1 Training Data Results

There is a need to provide the classifiers with training data. As discussed in the previous chapter, this project intends to employ the bus stations' coordinates as the training data. The Norwegian Travel Information company, provides this project with a text file that contains all the bus station coordinates in Norway. This project uses data from Arendal city in Norway, therefore, all the stations that belong to this city are extracted from the

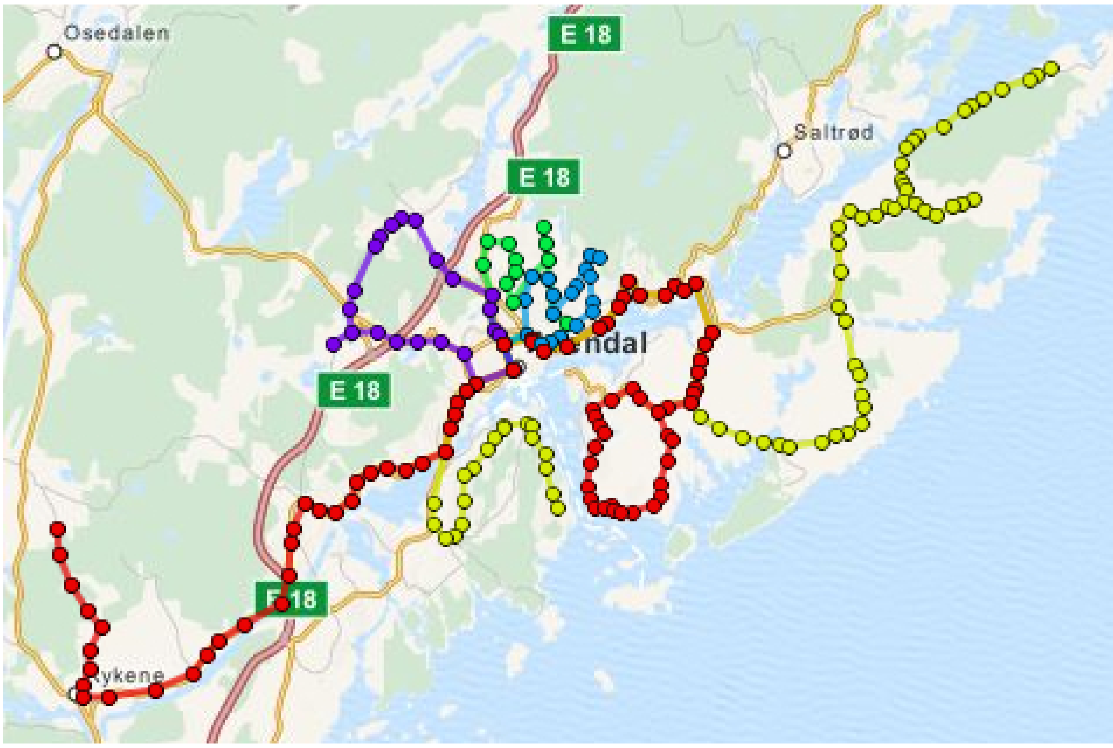


FIGURE 4.1: The five extracted bus stations in Arendal, sorted in by the bus' visiting order.

Line number	Number of points
L2	140
L3	100
L10	30
L11	25
L12	25

TABLE 4.1: Number of bus stations in each line.

mentioned text file. A particular prefix code is tied to all the stations that represents the city that a station belongs to. Thus finding all the city's bus stations is a straight forward task. However, as it is stated in section 3.2.1, these stations are not sorted in visit order. In this project, the five most popular bus routes' stations are extracted. By sorting them in the order that they are visited, one can estimate how a bus travels on actual routes. The extracted routes are illustrated in figure 4.1. Note that in this figure many of the extracted stations are shared between various lines and they overlap on each other. For the sack of simplicity, they are called by their names from now on. The lines 2, 3 ,10, 11 and 12 are respectively represented by L2, L3, L10, L11 and L12. There are 320 coordinates in these five routes, the number of points in each of them is provided in table 4.1.

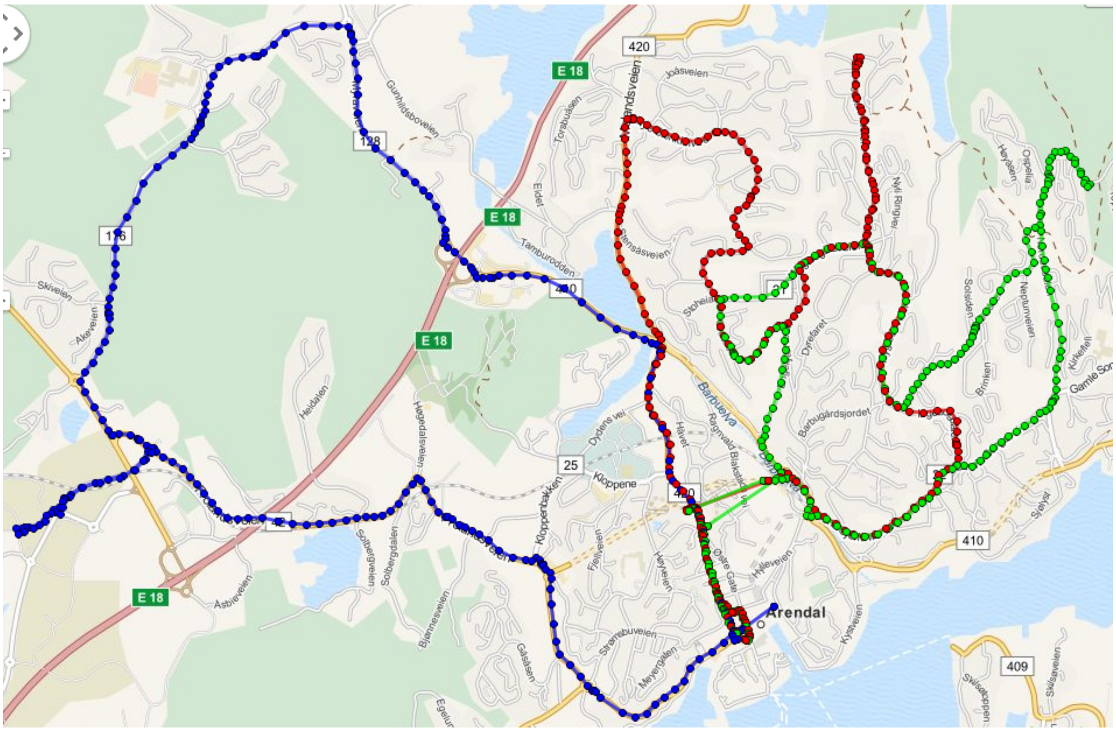


FIGURE 4.2: The three routes that data is collected on them. R10 is red, R11 is green and R12 is blue.

4.1.2 Testing Data Results

In order to test the classifiers' performance, they must be provided with testing data. This project employs its developed mobile phone application to collect data (used as testing data) and stores it in the database. The application is installed on a smart phone and data is collected for three different bus lines. The three bus lines are chosen from those that the training data is provided for them. This project calls the collected data on each line a "route" and they are represented R10, R11 and R12. This project does not collect data for all the available training data to determine how the algorithms behave in assigning points to this unused data. The numbers in each route number correspond to the actual bus lines, for instance, R10 is collected on L10. A similar approach can be employed to collect data from other transportation vehicles such as trains.

The collected data consists of 878 coordinates, some information about this data is represented in table 4.2. Figure 4.2 illustrates the three routes that spatial data is collected on them. The data is collected while the application sends the coordinates to the database every 5 seconds.

Route number	Number of points	Length (km)
R10	302	13.8
R11	262	11.8
R12	314	10.7

TABLE 4.2: Information about collected data.

Value of n	Number of points in data series
Not applied	1118
3	1100
2	1077
1	1001
$1/2$	847
$1/3$	756
$1/4$	634
Without outliers	878

TABLE 4.3: The effect of value n in removing outliers.

4.2 Results of Applying the Solution

This section explains the results of applying the developed solution in this project. These results are considered as the main contributions of this project. Section 4.2.1 shows how the outlier removal algorithm removes the outliers and optimizes the collected data. Section 4.2.2 shows how the anonymization algorithm makes the the data anonymous and not addressable to the customers. Last but not least, section 4.2.3 shows the performance of the classification algorithms considering how each of the developed algorithms affect the classification.

4.2.1 Removing the Outliers

The collected data in our experiments does not have outliers because it is only collected while one was taking the bus. I. e, data collection starts when one enters the bus and stops when s/he exists the bus. Therefore some dummy outliers are added to it to test the performance of the outliers removal algorithm. The value n, that sets the sensitivity in removing the outliers, is assigned various numbers to observe the performance difference. This projects adds the stations' coordinates of the bus lines that actual data is not collected on them (L2 and L3). Because these bus lines do not operate in the city centre, they are distant from L10, L11 and L12 and therefore can be potentially outliers. The goal in this section is removing the coordinates that are not collected in the city centre.

Figures 4.3 to 4.9 show how the value n, which is the ratio of σ, affects the sensitivity of the outliers removal algorithm. Figure 4.10 shows how the data should optimally look.

Route number	Not applied	$n = 3$	$n = 2$	$n = 1$	$n = 1/2$	$n = 1/3$	$n = 1/4$
R10	302	302	302	302	302	299	242
R11	262	262	262	262	219	192	172
R12	314	314	314	297	236	212	182
Outliers series 1	140	124	110	84	46	26	20
Outliers series 2	100	98	89	56	44	27	19

TABLE 4.4: The effect of value n on each route in removing the outliers.

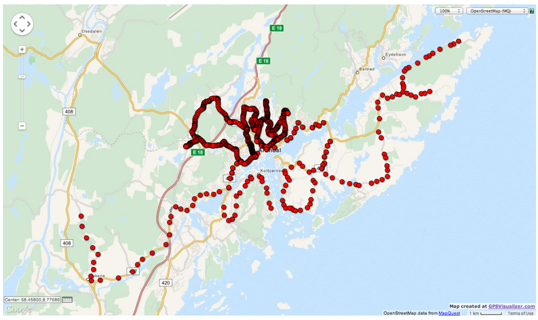


FIGURE 4.3: Before applying the outlier removal algorithm.

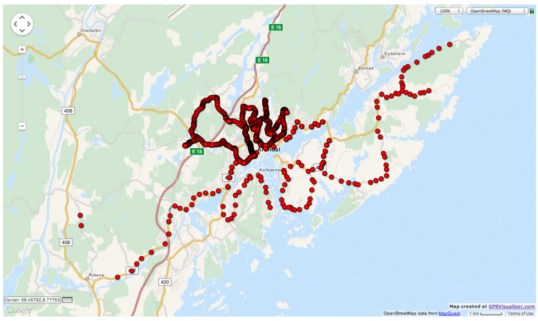


FIGURE 4.4: $n = 3$

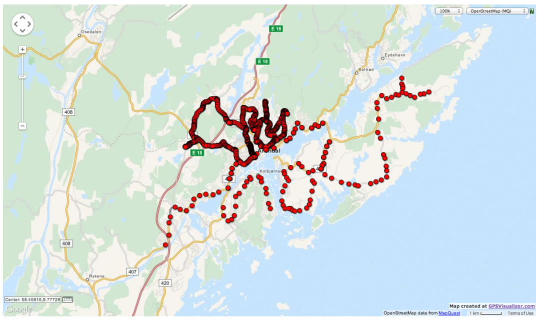


FIGURE 4.5: $n = 2$

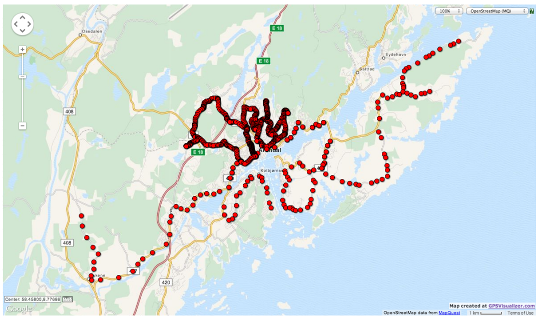


FIGURE 4.6: $n = 1$

Table 4.3 shows the number of removed coordinates from all the data and table 4.4 shows the number of removed coordinates in each route.

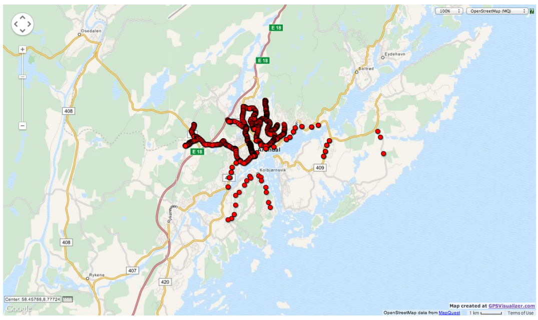


FIGURE 4.7: $n = 1/2$

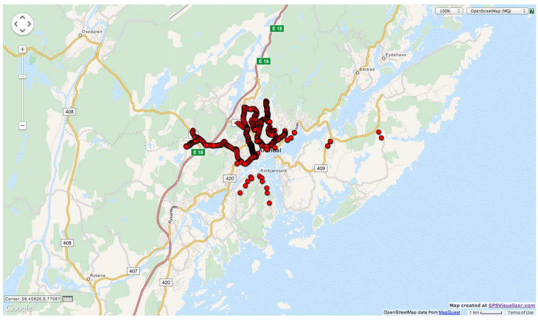


FIGURE 4.8: $n = 1/3$

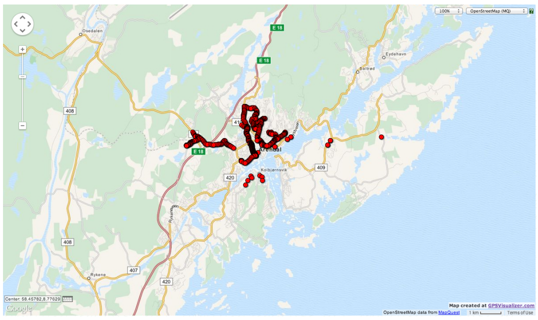


FIGURE 4.9: $n = {}^{1}/_{4}$

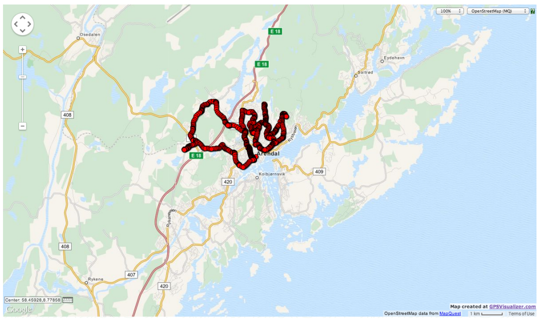


FIGURE 4.10: Without outliers.

4.2.2 Anonymizing the Data

This section explains the results of applying the anonymization algorithm. As explained in section 3.3.2, the anonymization algorithm, employs both training and testing data to clean the data. Because the training data consist of the actual bus stations' coordinates, one can elaborate the solution as follows: Every bus station acts like a radar, and the coordinates that are within the range of each radar are kept. The range of the radar is defined by the user. In this project, the ratio (n) of the value σ defines this range. In this way, a proper value based on the characteristics of the data series is defined. This fact yields in a flexible solution applicable to other data series. For a better demonstration of the anonymization results, section 4.2.2.1 explains the affect of the algorithm on all the data. In section 4.2.2.2, the results of applying the algorithm on real-life scenarios are described.

4.2.2.1 Anonymization on All of the Data

This section explains the affect of applying the anonymization algorithm on all the collected data in this project. Ideally, one should keep all the collected data from customers in order to achieve the best classification results. Figure 4.11 shows the data before applying the anonymization. But removing some wanted data is expected and the value of n determines the relative amount of wanted data to be removed. The more sensitivity of anonymization, the more useful data is removed. Figures 4.12 to 4.16 illustrate the results of anonymization on all the collected data by using different values for n. Note that, the illustrated data in this section, is only collected while buses were taken around the city. I. e., no real private data exists in this data series. The purpose is to show how the anonymization harms the wanted data. Tables 4.6 and 4.5 illustrate the number of removed data points in all data series and in each route respectively.

Route number	Not applied	$n = 1^1/_2$	$n = 1$	$n = ^1/_2$	$n = ^1/_3$	$n = ^1/_4$
R10	302	302	300	231	160	115
R11	262	262	255	189	143	98
R12	314	303	257	187	144	130

Table 4.5: The effect of value n on each route in anonymization.

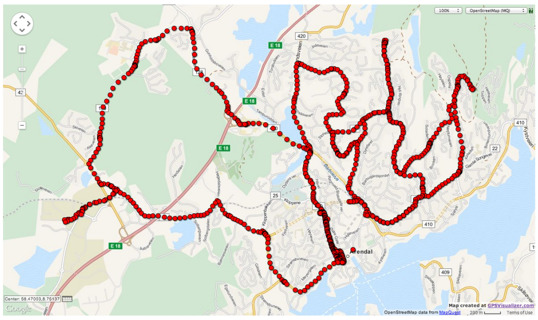


Figure 4.11: Before applying the anonymization algorithm.

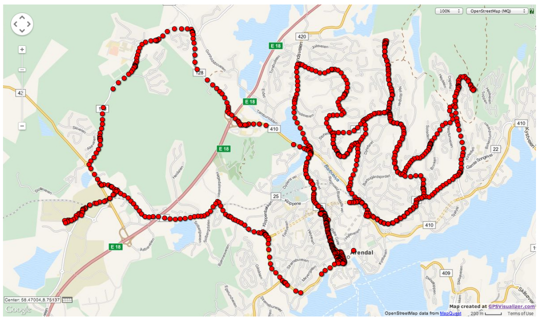


Figure 4.12: $n = 1^1/_2$

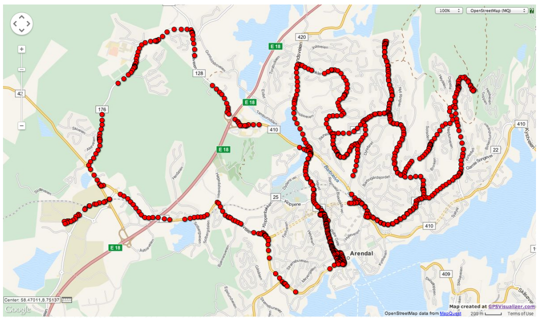


Figure 4.13: $n = 1$

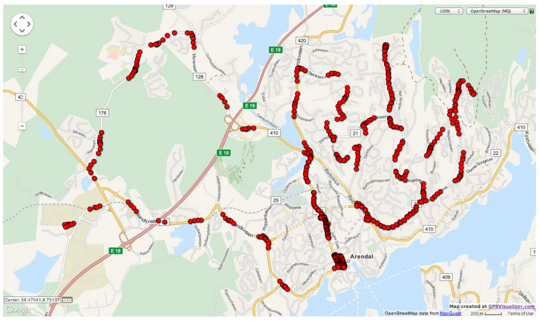


Figure 4.14: $n = ^1/_2$

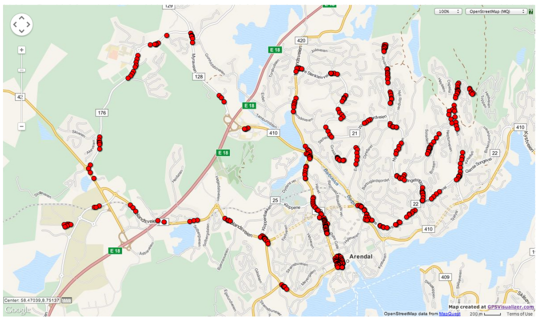


Figure 4.15: $n = ^1/_3$

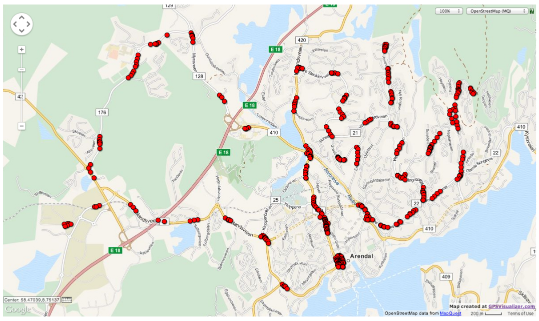


Figure 4.16: $n = ^1/_4$

Value of n	Number of points in data series
Not applied	878
$1^1/_2$	867
1	812
$^1/_2$	607
$^1/_3$	447
$^1/_4$	343

Table 4.6: The effect of value n in anonymization on all the data series.

4.2.2.2 Anonymization on Individual tracks

This section applies the anonymization algorithm to the data collected simulating a real-life scenario. A customer starts the application and exits his home to reach the bus station. Ideally, the data collected before the customer arrives at the station should be cleaned. Figure 4.17 shows a series of data. The blue circles (coordinates) are collected on the bus and the red ones are from the customer's home place to the bus station. Figure 4.18 shows the affect of applying the anonymization algorithm with value $n = 1$ on figure 4.17. As one can see, the red circles representing the the customer's private route are considerably removed. The number of red circles is reduced from 70 to 19. In addition, one can see that the blue circles, representing the collected data on the bus, still demonstrate their belonging route. The number of blue circles is reduced from 116 to 85.

4.2.3 Applying the Classification Algorithms

This section represents the results of applying the classification algorithms. The employed algorithms are k-Nearest Neighbours, Support Vector Machine and Decision Tree. It shows how much of the collected data (from the mobile application) is correctly matched to the custom built map (bus stations' coordinates). In section 4.2.3.1, the best matching results for each classifier are explained. I. e., No outliers exists in the collected data and no anonymization is performed. This is to show the highest matching percentage that can be achieved under the ideal circumstances. Section 4.2.3.2 adds the outliers and private collected data (e. g. from customers' home to the bus station) to the collected data and represents the classifiers' performance. And finally in section 4.2.3.3, the outliers removal and anonymization algorithms are applied and the classifiers' performance is represented.

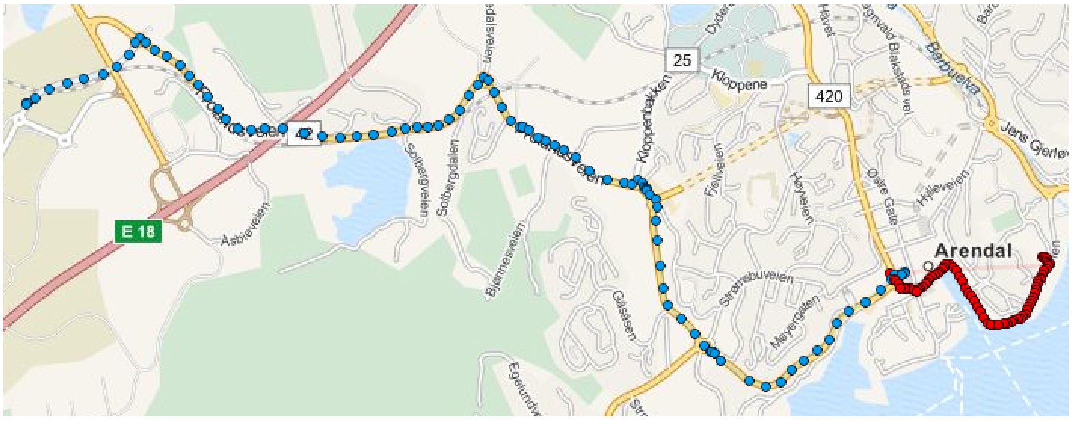


FIGURE 4.17: Before applying the anonymization algorithm.

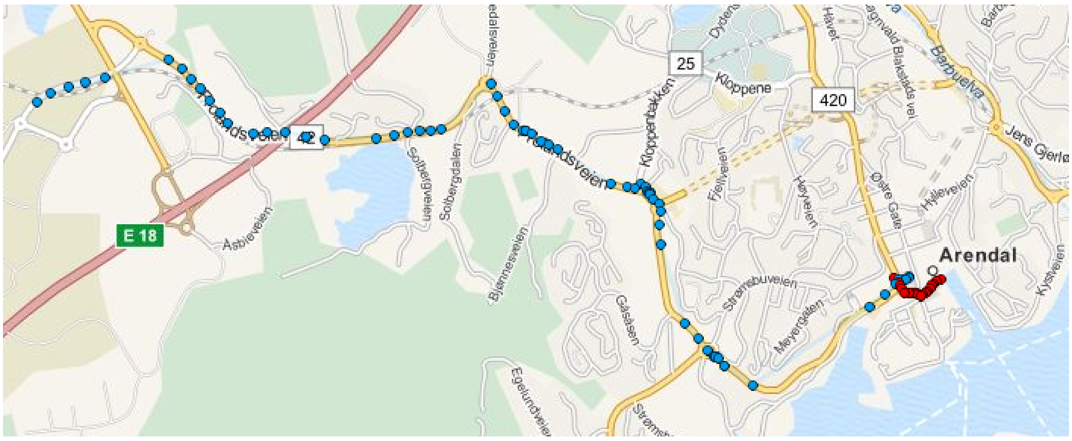


FIGURE 4.18: After applying the anonymization algorithm with $n = 1$

Classifier	Correctly matched points (number)	Correctly matched points (%)
SVM	523	59.63
DTree	637	63.54
k-NN	663	75.48
Total points	87	

TABLE 4.7: Classifiers' performance in the ideal situation.

4.2.3.1 Ideal Classifiers' Performance

There are 878 points (coordinates) in the data series that are only collected on bus routes. I. e., this is the perfect data series without any outliers and it does not contain addressable data to any customer. The classification algorithms are trained with 80 labeled points (bus stations) and their performance is tested by seeing if they can match the testing data, correctly, to the training data. Table 4.7, for each classifier, shows the percentage of correctly matched points from the training data to the testing data. Because the best possible situation is considered here, it is expected that the classifiers perform their best. As one can realize, k-NN outperforms the other two classification algorithms by correctly matching the data with about 75% of accuracy.

The number of neighbours in k-NN algorithm is determined by the value of k. By experience, increasing this number to higher values does not yield better results. Figure 4.19 shows the k-NN classifier accuracy with different values of k. As one can see, when $k = 1$, the accuracy is at its highest. From now on in this paper, the value of k is set to 1 as it yields the best results.

Technically, the great performance of k-NN classifier can be expected. As it classifies each point to the its closest neighbor. But DTree performs surprisingly well, especially after the anonymization phase (see section 4.2.3.3.). Figure 4.20 illustrates the tree that classifies the data in four levels.

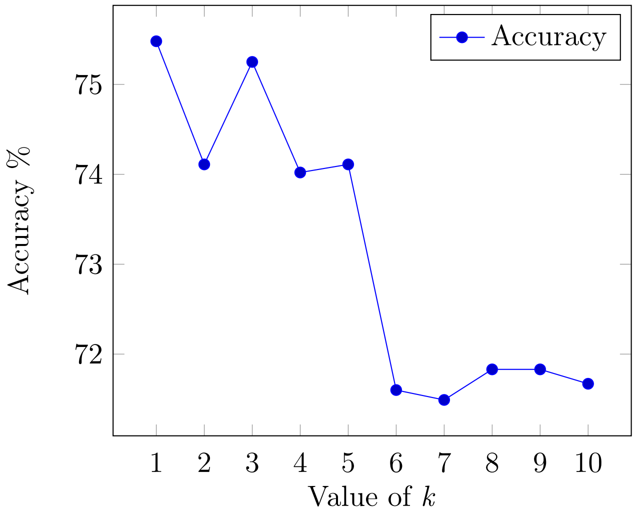


FIGURE 4.19: Comparison of the accuracy of applying the k-NN algorithm with different values of k.

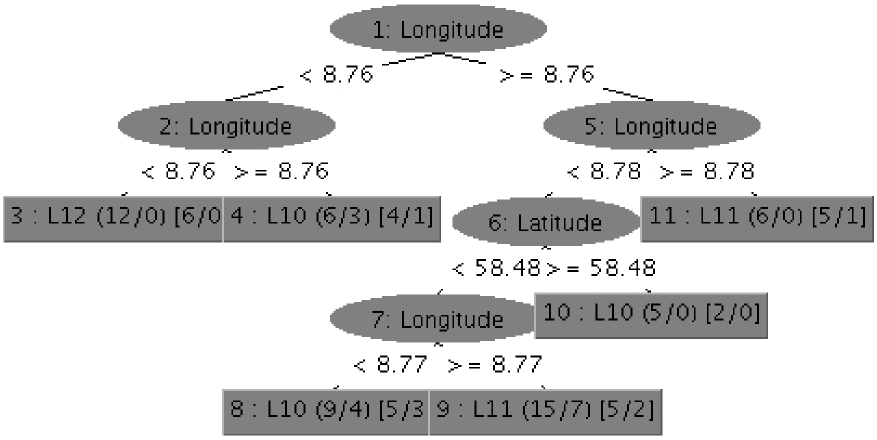


FIGURE 4.20: Visualization of the DTree.

4.2.3.2 Classifiers' Performance Including Unwanted Data

This section represents the classifier's performance taking into account that the collected data contains outliers and private collected data. One must consider as some unrelated data is added to the testing data, classifiers are expected to perform their worst. Table 4.8 shows the classifiers' performance in the described situation. There are 312 extra points that are added in the testing data to deliberately lower the classifiers' performance. These points are added to simulate the outliers and private data in real-life scenarios. In table 4.8, it is apparent that the classifiers' performance is considerably lower compared to table 4.7. One can see that k-NN and DTree perform similar to each other (about 55%) and outperform the SVM.

Classifier	Correctly matched points (number)	Correctly matched points (%)
SVM	447	37.59
DTree	644	54.16
k-NN	664	55.84
Total points	1189	

Table 4.8: Classifiers' performance including the unwanted data.

Classifier	Correctly matched points (number)	Correctly matched points (%)	Correctly matched points (number)	Correctly matched points (%)
	$n_o = 1/2$, $n_a = 1$		$n_o = 1/3$, $n_a = 1$	
SVM	345	51.18	311	52.18
DTree	448	66.46	383	64.26
k-NN	443	65.72	387	64.46
Survived points	674		596	
	$n_o = 1/2$, $n_a = 1/2$		$n_o = 1/3$, $n_a = 1/2$	
SVM	222	48.47	200	50
DTree	308	67.24	261	65
k-NN	311	67.90	266	66.5
Survived points	458		400	
Total points	1189			

Table 4.9: Classifiers' performance after applying the outliers removal and anonymization algorithms. n_o and n_a represent the sensitivity of the outliers removal and anonymization algorithms respectively.

4.2.3.3 Classifiers' Performance After Applying the Proposed Algorithms

In this section, the outliers removal and anonymization algorithms are applied to the data series with unwanted data. Table 4.9 illustrates each classifier's performance while the outliers removal and anonymization algorithms are applied with various sensitivities (the value n). In table 4.9, n_o and n_a represent the sensitivity of the outliers removal and anonymization algorithms respectively. The optimal effect of the mentioned algorithms is when they can remove the outliers, anonymize the data and still, match the data with a performance close to what is described in section 4.2.3.1. One can realize from table 4.9 that k-NN and DTree perform the best in all the situations and they perform similar to each other with a accuracy of about 65%. Comparing to what illustrated in table 4.8, the classifiers' performance is better here.

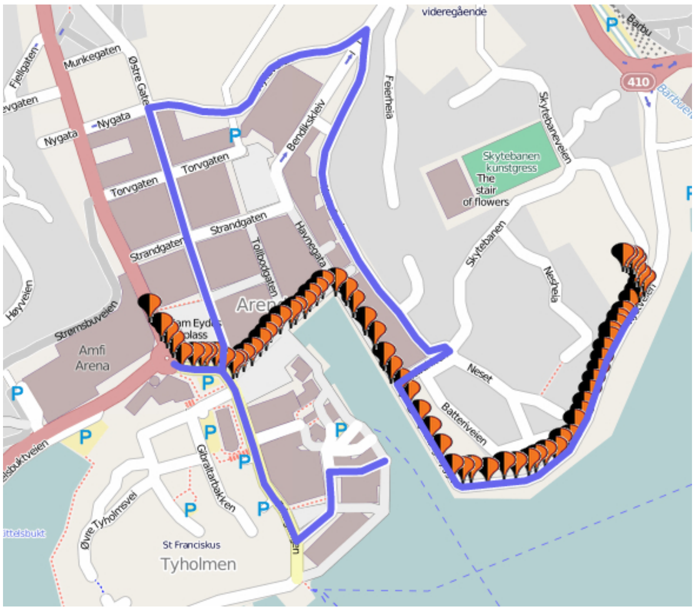


FIGURE 4.21: An example of TrackMatching.

4.2.4 Map-Matching with Open Street Maps

This projects applies the collected data to an application namely TrackMatching[1]. TrackMatching can not deal with outliers but it intuitively deals with the desired anonymization in this project. TrackMatching matches the data to the OSM and because of the fact that TrackMatching only accounts the main roads such as motorways, thus, no private data remains in the results. Therefore, only the outliers removal algorithm is applied to the data and then the data is applied to TrackMatching[2].

TrackMatching is mainly developed to match the inaccurate GPS data to the OSM but it is partly applicable for the purposes of this project as well. TrackMatching employs a REST API to communicate with the user. Figure 4.21, for the sake of demonstration, uses the demo tool of TrackMatching to show an example of its results. TrackMatching requires temporal data in addition to spatial data to operate. Therefore the t_{1i} values (coordinates' collection time from the mobile device) are used as timestamps. Figure 4.21 shows the result of map-matching by TrackMatching. The purple highlighted lines are the output and the orange and black balloons are the input data. In this example, the private data is used as the input and one can see that the output is not optimal and many unrelated routes are incorrectly matched, and yet, the data is not fully anonymised.

The outputs of TrackMatching (routes highlighted with purple in figure 4.21) are actually OSM routes. OSM uses three node IDs to represent a route that consists of a source ID, a middle point ID and a destination ID.

[1] https://mapmatching.3scale.net/
[2] https://test.roadmatching.com/

Chapter 5

Discussion

The goal of this project is to realize the travel patterns of customers. In order to reach this goal, the customers' location must match a map. This map can be from already existing maps on the market or one can make its own map. However, the information that represents the transportation companies' operative routes must be extractable from this map. This project builds a custom map network that consists of bus stations' coordinates and maps the spatial data on it. A mobile application is developed that collects the spatial data from the customers. A pre-process on this spatial data (outliers removal) detects the potentially existing outliers in the spatial data and removes them. This pre-process optimizes the spatial data. The anonymization algorithm cleans the spatial data from information that can be addressed to the customers. And lastly, classification algorithms match the spatial data to the custom built map (map-matching). The classification algorithms perform the map-matching with a accuracy up to about 65%. However when the spatial data is perfect i. e., without any outliers or private data, the accuracy gets up to 75%. In this chapter, a comparison between the classifiers is given and the their errors are discussed. This chapter discussed the guidelines in finding the optimal values of n in the experiments of this project

5.1 The Proposed Algorithms' Performance

This section discusses the performance of the outliers removal and the anonymization algorithms. Both of these algorithms remove data points from the collected spatial data. The characteristics of the collected spatial data may vary based on region they are collected. Because different regions can be geographically different, one can be a dense city with routes circling around the centre while other cities might be wide with long routes with different directions. Considering the mentioned fact, a solution must be flexible enough to be applicable to various cities. This project tackles this fact as a statistical problem and separately calculates the mean values and standard deviations (σ) for the latitudes and longitudes. The $n * \sigma$ is employed to make an arbitrary boundary, from the mean value, in the data series and remove all the data that resides outside of this boundary. Generally, collecting spatial data from customers in different areas yields to different data structures because it depends a lot on the geographic specifications of that particular area. Given this fact, the proper value of n may vary for different areas and is to be found by the solution's users. Sections 5.1.1 and 5.1.2 discuss this fact in more details.

5.1.1 Outliers Removal

The spatial data series, used to test the outliers removal algorithm performance, contains 1189 points (coordinates). At this stage, the project deliberately adds 312 points to the data series to act as outliers. One can realize that the count of related points is 878. Ideally, the algorithm should remove these 312 points. In this algorithm's settings, the smaller the value n is, the more points are removed. However, n should not get smaller than a certain value as it harms the related data. In this project's experiments, with $1 < n \leq {}^{1}\!/_{3}$, the remaining data points come close to 878 but it is noticeable that not all the deleted points are removed from the **unrelated data** (actual outliers). Despite the algorithm's simplicity, it performs pretty well in identifying the outliers (see figures 4.3 to 4.9).

5.1.2 Anonymization

One of most important primary goals of this project is to make the customers not trackable after they share their spatial data. To the best of our knowledge, the anonymization algorithm proposed in this project is novel. Not only it does anonymize the spatial data, it yields to improved map-matching results. However, sometimes the private spatial data confuse the map-matching solutions as they have to match some points to the routes that do not exist in coded maps. Considering that this project employes its custom built map, and as the map data exists in particular places (bus stations, see section 4.1.1 for more info), the private spatial data leads to more errors in matching. On the other hand, one can identify a transportation sub-line between two stations without having all the data in between them. Therefore, the spatial data far from the stations is redundant and cleaning such data yields to quicker map-matching.

The anonymization algorithm creates an imaginary circular boundary around each point on the built map and it only keeps the spatial data that lies within this boundary. The radius of this region plays the key role on the decision of cleaning the redundant data. As explained in section 5.1, one can find the best radius based on the structure of the spatial data. Complexity of the transportation system can be another factor, e. g., different bus stations that are close to each other and operate for different lines.

Similar to the approach in the outliers removal algorithm, the value of $n * \sigma$ decides the radius of the aforementioned boundary. σ is the standard deviation of the euclidean distances between all the points in the collected spatial data and all the points in the built map. In this project's experiments, while $1 \leq n \leq {}^1\!/_2$, an expectable amount of anonymization is performed. In other words, enough amount of private spatial data is cleaned that makes the customers not trackable while the related spatial data is not harmed and the classification algorithms can still perform close to their best.

5.2 Map-Matching

This section discusses the two approaches of this project on map-matching: Classification and TrackMatching. Section 5.2.1 explains the achieved results of this project's solutions that are completely developed by this project. It is discussed, by investigating the classification errors, why the achieved results are very promising and why the errors are negligible. Moreover, the good performance of the outliers removal and anonymization algorithms is discussed. Section 5.2.2 discusses the results of applying the collected spatial data in this project to the TrackMatching solution.

5.2.1 Classification Algorithms' Performance

This project views the map-matching problem as a data mining problem and employs various classification algorithms to match the spatial data to its custom built map. This is a new novel approach to the map-matching problem that fits the main use case scenario of this project very well. This section discusses the performance of the classifiers and explains why they perform in the way they do.

The tested classifier algorithms, k-NN and DTree [1], classify with the best accuracy amongst many other tested classifiers. They perform with a accuracy close to each other that reaches about 75% under the ideal circumstances. Ideal circumstances are when there exists no outliers and private data in the spatial data. One may argue that the error rate of 25%, even in ideal circumstances, is too high. Generally, it is a valid argument, but the investigations in the areas where the errors occur show that they are

[1] This project has tested various classification algorithms and only the results of their best are provided.

A	B	C	Classified as
209	93	0	A = R10
87	175	0	B = R11
58	2	253	C = R12

TABLE 5.1: Confusion matrix for DTree without considering the unwanted data.

A	B	C	Classified as
277	25	0	A = R10
135	127	0	B = R11
55	0	258	C = R12

TABLE 5.2: Confusion matrix for k-NN without considering the unwanted data.

negligible. The errors only happen on the routes that two bus lines overlap each other, however, as long as the spatial data is classified in either of these lines, the errors does not matter. This is because of the fact that one can realize that the mentioned route is a popular route and this is exactly what this project desires to find. Looking to this fact from the users' perspective (potentially transportation companies), the required number of buses operating on that route needs to be found, not the exact bus line number. Technically, the employed classifiers can not go wrong for reasons other than this.

Tables 5.1 and 5.2 [2], describe the classification confusion matrixes of k-NN and DTree algorithms. The fact is that errors only happen on overlapping routes. In an ideal confusion matrix, all the rows except the main diagonal from high to low should equal zero. As one can see in the mentioned confusion matrixes, the errors are shared between lines 10 and 11 and these lines overlap in most of their routes (refer to figure 4.2). It is worth to mention that all the lines overlap in certain parts, but the most overlap is shared between lines 10 and 11.

This project deliberately adds 312 points to the spatial data as the outliers and private data to test the classifiers performance. This growth of spatial data lowers the classifiers performance in correctly classifying the spatial data from 75% to about 55%. This 20% is a considerable drop in accuracy. The outliers removal and anonymization algorithms perform very good in optimizing the spatial data and they can together bring up the accuracy by more than 10%. These are very promising results considering that the size of spatial data series is increased by 35% and more importantly, this 35% is unrelated data. Tables 5.3 and 5.3 show the confusion matrixes of the classification algorithms after the outliers removal and anonymization algorithms are applied to the spatial data. The value of n is set to $1/2$ for both algorithms. The investigation of the errors shows that once again they occur on the points that the bus lines overlap each other.

[2] The SVM classification algorithm is not discussed in more details as it performs poorly. See section 4.2.3.1 for more details.

A	B	C	Classified as
193	0	0	A = R10
87	37	0	B = R11
38	0	78	C = R12

TABLE 5.3: Confusion matrix for DTree considering the unwanted data.

A	B	C	Classified as
187	6	0	A = R10
78	46	0	B = R11
38	0	78	C = R12

TABLE 5.4: Confusion matrix for *k*-NN considering the unwanted data.

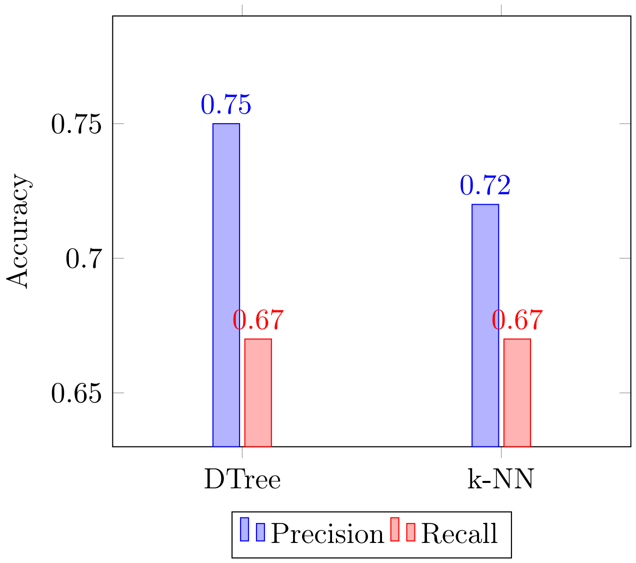


FIGURE 5.1: The precision and recall of the classification results after applying the outliers removal and anonymization algorithms.

In figure 5.1, a comparison between precision and recall for the *k*-NN and DTree classifiers is illustrated. Precision is sum of the fraction of the retrieved points that are relevant, while recall is sum of the fraction of relevant instances that are retrieved. In other words, precision only considers the points of one class that are classified incorrectly. But recall also takes into account the points from other classes that are wrongly classified into that class. In general, precision and especially recall are more reliable than accuracy.

Figures 5.2 and 5.3, for the *k*-NN and DTree classifiers, illustrate the ideal performance in precision and recall (A). In these figures, one can see how adding the outliers and private data lowers the classifiers performance (B). And it is shown how applying the outliers removal and anonymization algorithms considerably raises the classifiers performance (C).

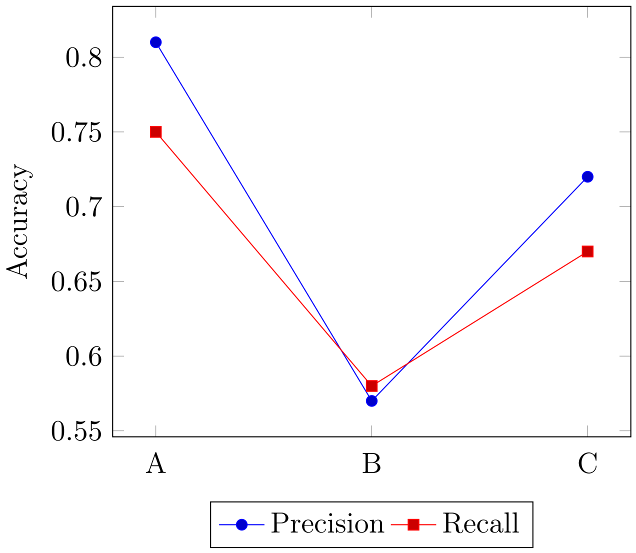


FIGURE 5.2: Comparison of precision and recall for k-NN algorithm. A = Ideal performance, B = Performance after adding outliers and private data, C= Performance after applying the outlier removal and anonymization algorithm.

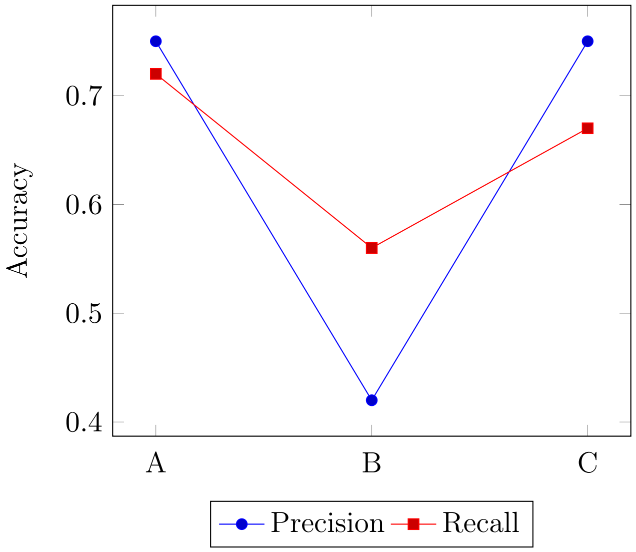


FIGURE 5.3: Comparison of precision and recall for DTree algorithm. A = Ideal performance, B = Performance after adding outliers and private data, C= Performance after applying the outlier removal and anonymization algorithm.

5.2.2 TrackMatching's Performance

It is certain that TrackMatching does not the outliers removal feature and it does not intentionally anonymizes the data. If the data series, that is sent to TrackMatching, is affected by the outliers removal and anonymization algorithms, the results of matching are not satisfactory [3]. This is because the remaining points in the data series are only in certain places (bus stations that build the custom map, see section 4.1.1) and there may exist many possible routes between two stations. Therefore, the structure of the

[3] Due to lack of time, this project cannot fairly compare its solution with TrackMatching.

points confuses the TrackMatching and it outputs the closest possible route between each two points. This fact is expectable because TrackMatching is not developed for mutual purposes with this project. However, if one inputs a perfect series of spatial data to TrackMatching, it outputs the matched routes very accurately. Unfortunately, this project cannot measure this accuracy because of time shortage.

Chapter 6

Conclusion

This project has developed new and novel solutions that are able to solve the following problems: 1- How to automatically collet spatial and temporal data from customers without neither special equipment nor interaction from the customers, 2- How to make the collected data from the customers anonymous and untraceable and 3- How to automatically match the routes that the customers are travelling on based on the collected data.

To tackle the first problem, an Android mobile application has been developed to explicitly collect spatial and temporal data from the customers. The application is very lightweight and therefore can be integrated into the mobile ticket application of the transportation company that desires to employ this project's solution. The application fetches the spatial data from location reporting resources such as GPS and GLONASS satellites, adds the temporal data fetched from the mobile device, and finally sends them via a web server to a database every 5 seconds.

Two new algorithms have been developed to solve the second problem. Both of them share the advantage of using a ratio of the standard deviation to remove data which yields in a flexible solution. Because the collected spatial data, based on the geographic characteristics of the collection region, constructs a different data series. Despite the relative simplicity of the algorithms, they proved to be very successful in accomplishing their missions.

The first algorithm, basis itself on **outliers removal** to remove misleading spatial data from the customer, such as data collected from walking to the public transportation stations. The algorithm separates the latitudes and longitudes into two data series. Subsequently, it removes any data that lies outside the mean value of each of these series plus and minus a ratio of the standard deviation of that data series. The user of the solution sets the ratio based on the desired sensitivity and this fact results in a flexible algorithm.

The second algorithm, namely **anonymization**, is capable of making the collected data from the customers anonymous and untraceable. This algorithm, matches the collected spatial data to a map, that consists of transportation stations, and cleans any data that lies outside of a defined radius of each station. The algorithm finds the mentioned radius by calculating the mean value of the distances between all the data points in the collected spatial data and all the transportation stations. The algorithm also calculates the standard deviation of the mentioned distances and cleans any data that resides outside of the mean value plus and minus a ratio of the standard deviation. The mentioned ratio is set be the solution's user and gives the algorithm the flexibility to deal with different collected spatial data series.

The most important task and contribution of this project comes in solving the third problem. The so called map-matching problem, has been approached in various ways by prior researchers. This project has approached the map-matching problem by employing data mining techniques which is a new and novel approach towards map-matching. In this project, many classification algorithms have been put to test and the best ones are chosen. The k-NN classifier has proven to be the best in the majority of situations with 75% of accuracy. However, the errors are negligible because they only happen on overlapping routes that do not matter to transportation companies.

The experiences in this project have shown that how unwanted data (outliers and private spatial data) affects the performance of classifiers and lowers the accuracy rate by 20%. But **outliers removal** and **anonymization** algorithms have shown their power in dealing with the unwanted data by raising the accuracy level up to 10% yielding in an overall accuracy of 65%. This project, intuitively and by the inspiration of reviewing the state-of-the-art, have chosen to employ the k-NN classifier. But in the trial of various classifiers, decision tree has performed surprisingly well, especially after applying the **outliers removal** and **anonymization** algorithms.

Chapter 7

Future Work

Much work has been done in this project and promising results have been achieved, but clearly there can always be improvements. This project is no exception and recommends the below items to be followed:

- Measuring the computations time and subsequently the efficiency of the developed solutions and algorithms. This becomes more important when it comes to applying the solution to larger data series. The solution must be applied frequently and maybe even in real-time. Therefore, it must be quick enough to run a real-time application.
- The visualization of the results to the clients (most likely transportation companies). I. e., a visualization tool such as a web portal can be developed to show the results to the clients who desire to employ the developed solution. The results, including the hot routes, must be visualized in an understandable way to the clients.
- At this stage of the project, all parts of the solution are applied in the server-side. This means that some redundant data is collected from the customer and then it is removed. One can consider the adoptability of the developed solutions in the client-side (mobile application).
- This project collects both spatial and temporal data, however, temporal data is only used for sorting purposes. Mining the temporal data can potentially lead to interesting applications.
- There can be improvements in realizing the customers' travel patterns (e. g., travelling by car or walking). Identifying the customers' transportation mode could

be a mean yielding in a better understanding of travel patterns. This fact, especially can help the clients to realize how their customers travel. There exists much research in this domain such as [23].

Bibliography

[1] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. The MIT Press, 2010.

[2] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[3] F. S. T. Van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, 2009.

[4] P. Misra, B. P. Burke, and M. M. Pratt, “Gps performance in navigation,” *Proceedings of the IEEE*, vol. 87, no. 1, pp. 65–85, 1999.

[5] R. B. Langley, “The gps error budget,” *GPS world*, vol. 8, no. 3, pp. 51–56, 1997.

[6] R. T. Ng and J. Han, “Efficient and effective clustering methods for spatial data mining,” in *Proc. of*, 1994, pp. 144–155.

[7] S. Shekhar, P. Zhang, Y. Huang, and R. R. Vatsavai, “Trends in spatial data mining,” *Data mining: Next generation challenges and future directions*, pp. 357–380, 2003.

[8] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[9] B. Gedik and L. Liu, “Location privacy in mobile systems: A personalized anonymization model,” in *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*. IEEE, 2005, pp. 620–629.

[10] N. Adrienko and G. Adrienko, “Spatial generalization and aggregation of massive movement data,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 2, pp. 205–219, 2011.

[11] T. Iwuchukwu and J. F. Naughton, “K-anonymization as spatial indexing: Toward scalable and incremental anonymization,” in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 746–757.

[12] C. A. Cassa, S. J. Grannis, J. M. Overhage, and K. D. Mandl, "A context-sensitive approach to anonymizing spatial surveillance data impact on outbreak detection," *Journal of the American Medical Informatics Association*, vol. 13, no. 2, pp. 160–165, 2006.

[13] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques.* Morgan kaufmann, 2006.

[14] X. Li, J. Han, J.-G. Lee, and H. Gonzalez, "Traffic density-based discovery of hot routes in road networks," in *Advances in Spatial and Temporal Databases.* Springer, 2007, pp. 441–459.

[15] T. Brinkhoff, "A framework for generating network-based moving objects," *GeoInformatica*, vol. 6, no. 2, pp. 153–180, 2002.

[16] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems.* ACM, 2008, p. 34.

[17] D. Ashbrook and T. Starner, "Using gps to learn significant locations and predict movement across multiple users," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, 2003.

[18] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. P. Sondag, "Adaptive fastest path computation on a road network: a traffic mining approach," in *Proceedings of the 33rd international conference on Very large data bases.* VLDB Endowment, 2007, pp. 794–805.

[19] J. S. Greenfeld, "Matching gps observations to locations on a digital map," in *National Research Council (US). Transportation Research Board. Meeting (81st: 2002: Washington, DC). Preprint CD-ROM*, 2002.

[20] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 5, pp. 312–328, 2007.

[21] C. E. White, D. Bernstein, and A. L. Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transportation Research Part C: Emerging Technologies*, vol. 8, no. 1, pp. 91–108, 2000.

[22] F. Marchal, J. Hackney, and K. W. Axhausen, "Efficient map matching of large global positioning system data sets: Tests on speed-monitoring experiment in

zürich," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1935, no. 1, pp. 93–100, 2005.

[23] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding transportation modes based on gps data for web applications," *ACM Trans. Web*, vol. 4, no. 1, pp. 1:1–36, Jan. 2010.