BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INTELLIGENT SYSTEMS
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

USER LOCATION INTERPRETATION BASED ON LOCATION DATA

INTERPRETACE POLOHY UŽIVATELŮ NA ZÁKLADĚ DAT O POLOZE

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. ALEXANDRA LIGOCKÁ**
AUTOR PRÁCE

SUPERVISOR **Ing. ANTON FIRC**
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Master's Thesis Assignment

148198

Institut:	Department of Intelligent Systems (UITS)
Student:	**Ligocká Alexandra, Bc.**
Programme:	Information Technology and Artificial Intelligence
Specialization:	Computer Networks
Title:	**User Location Interpretation Based on Location Data**
Category:	Information Systems
Academic year:	2022/23

Assignment:

1. Learn about the processing of user location data and its subsequent interpretation.
2. Survey existing approaches for interpreting significant locations and location-based suggestions. Analyze what methods are currently being used in practise (e.g. Google, Apple, Meta).
3. Propose a method for processing and interpreting significant locations based on location data. Consider putting significant locations into context using current map documents.
4. Implement the proposed method as standalone tool for interpreting collected location data as significant locatios.
5. Create evaluation dataset and perform tests of the implemented method. Focus on the accuracy of significant locations, processing time and interpretability of results.
6. Evaluate the results achieved by your implementation.

Literature:

- Ibrahim, A., Zhang, H., Clinch, S. *et al.* From GPS to semantic data: how and why—a framework for enriching smartphone trajectories. *Computing* **103**, 2763–2787 (2021). https://doi.org/10.1007/s00607-021-00993-z
- Wenchao Li, Xin Liu, Chenggang Yan, Guiguang Ding, Yaoqi Sun, Jiyong Zhang: STS: Spatial-Temporal-Semantic Personalized Location Recommendation. ISPRS Int. J. Geo Inf.9(9): 538 (2020) https://mdpi-res.com/d_attachment/ijgi/ijgi-09-00538/article_deploy/ijgi-09-00538.pdf?version=1599561491
- Li, L., Chen, X., Liu, Q., Bao, Z. (2020). A Data-Driven Approach for GPS Trajectory Data Cleaning. In: Nah, Y., Cui, B., Lee, SW., Yu, J.X., Moon, YS., Whang, S.E. (eds) Database Systems for Advanced Applications. DASFAA 2020. Lecture Notes in Computer Science(), vol 12112. Springer, Cham. https://doi.org/10.1007/978-3-030-59410-7_1

Requirements for the semestral defence:
1-3

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Firc Anton, Ing.**
Head of Department:	Hanáček Petr, doc. Dr. Ing.
Beginning of work:	1.11.2022
Submission deadline:	17.5.2023
Approval date:	3.11.2022

Abstract

The topic of this thesis is to investigate the problem of processing user location data with the aim of mining semantically interesting places. Geolocation data represents great potential for advertising systems, recommending regional news, reaching users for targeted advertising, suggesting popular locations, location-based suggestions, and much more. This work explores key principles in geolocation data acquisition, processing and interpretation. The thesis further investigates the feasibility and challenges involved in extracting a user's home and work location from raw GPS data collected from GPS-enabled devices. Additionally, it explains the need for semantic enrichment of users' locations. The main challenges discussed include identification of stops from GPS traces, identifying locations with a higher level of significance for users, extracting visited places and their semantic enrichment and interpretation using current map bases.

Abstrakt

V tejto práci sa skúmajú kľúčové princípy získavania, spracovania a interpretácie geolokačných údajov s cieľom získania sémanticky zaujímavých miest používateľa. Geolokačné dáta majú obrovský potenciál pre rôzne aplikácie, vrátane reklamných systémov, odporúčaní miest a podobne. Práca sa tiež zameriava na identifikáciu výziev pri extrahovaní domovskej a pracovnej polohy používateľov zo surových GPS dát zozbieraných z GPS zariadení. V práci sa ďalej vysvetľuje dôležitosť sémantického obohatenia miest používateľa. Medzi hlavné diskutované výzvy patrí detekcia zastávok z GPS stôp, identifikácia miest s vysokým významom pre používateľov, extrakcia navštívených miest a ich sémantické obohatenie a interpretácia pomocou aktuálnych mapových podkladov.

Keywords

geoprocessing, geolocation, geospatial analysis, geocoding, data enrichment, clustering algorithms, significant places, location-based services

Kľúčové slová

geoprocessing, geolokácia, geopriestorová analýza, geokódovanie, obohacovanie údajov, zhlukovacie algoritmy, významné miesta, lokalizačné služby

Reference

LIGOCKÁ, Alexandra. *User Location Interpretation Based on Location Data.* Brno, 2023. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Anton Firc

Rozšírený abstrakt

S nárastom počtu zariadení s GPS, ako sú smartfóny, sa zvyšuje aj objem údajov o užívateľoch. Tieto údaje umožňujú užívateľom sledovať ich aktivity a pohyb prostredníctvom GPS trajektórií. Údaje o polohe užívateľov sú ďalej využívané v rôznych odvetviach vrátane sociálnych médií, elektronického obchodu, dopravy a zdravotnej starostlivosti a môžu byť prínosom pre používateľov a poskytovateľov služieb.

V tejto práci sa zameriavame na spracovanie polohových dát o užívateľoch. Pojem polohové dáta vyjadruje polohu zariadenia používateľa spojenú s časových identifikátorom. Polohové dáta sú získavané prostredníctvom technológií určovania polohy zahŕňajúce vnútorné a vonkajšie polohovanie. Najbežnejším spôsobom vyjadrenia polohy je zemepisná šírka a výška, pretože tieto súradnice môžeme použiť na presné určenie miesta na zemskom povrchu. Existujú aj ďalšie spôsoby vyjadrenia a interpretácie polohy avšak väčšina z týchto reprezentácií je nevhodná z pohľadu bežného človeka pri používaní aplikácií orientovaných na využívanie lokalizačných služieb. Pre väčšinu užívateľov sú najviac známe aplikácie využívajúce lokalizačné služby, ako napríklad Google Maps, Uber, Facebook a iné. Tieto služby sú určené na pomoc ľuďom pri hľadaní miest alebo navigácii do konkrétnych cieľov. Namiesto používania súradníc majú ľudia tendenciu používať opisné výroky o svojej polohe a priraďovať miestam významovú a opisnú sémantiku. Takéto opisy sa dajú vyjadriť ako "som doma" alebo "som v práci", často označujú navštevované miesta a sú spojené s konkrétnymi súradnicami alebo geografickými oblasťami. Na riešenie tohto problému bolo navrhnutých niekoľko techník na odvodzovanie sémantického významu miesta. Sémantické obohacovanie miest zahŕňa priradzovanie miestam rôzne popisy týkajúce sa typu miesta, názov, adresa alebo vzťah daného miesta k užívateľovi. Úroveň sémantickej bohatosti sa môže líšiť v závislosti od cieľovej aplikácie. Niektoré aplikácie využívajú jednoduché princípy založené na technikách geokódovania a iné si vyžadujú komplexnejšie prístupy zahŕňajúce ďalšie údaje zo zdrojov tretích strán. V našej práci sa zameriavame najmä na extrakciu sémanticky dôležitých miest pre užívateľa zo stôp GPS, ako je domov a práca. Okrem týchto význačných lokalít sa zameriavame aj na vyhľadávanie a pridávanie širšieho kontextu k miestam, ktoré používateľ navštívil.

V oblasti automatického vyhľadávanie miest záujmu používateľov je niekoľko výziev a prekážok, ktoré sťažujú celý proces. Jednou z hlavných výziev je získavanie údajov. Zhromažďovanie údajov o polohe môže byť náročné z dôvodu ochrany súkromia, ako aj technických problémov, ako je rušenie signálu alebo nízka výdrž batérií mobilných zariadení. Okrem toho je nevyhnutné zabezpečiť, aby zhromaždené údaje boli presné a reprezentatívne pre skutočný pohyb používateľa. Ďalšou výzvou je detekcia miesta pobytu. Hoci sa koncept bodu pobytu zdá byť jednoduchý, existujú rôzne faktory, ktoré môžu komplikovať jeho identifikáciu, ako napríklad šum v údajoch, variabilita pohybov používateľa a prítomnosť viacerých bodov pobytu v tesnej blízkosti. Na odfiltrovanie šumu a identifikáciu zmysluplných bodov pobytu sú potrebné sofistikované algoritmy. Ďalšou výzvou je variabilita priestorových súradníc. Aj keď sa používateľ nachádza na rovnakom fyzickom mieste, jeho zariadenie môže hlásiť mierne odlišné súradnice v dôsledku faktorov, ako je drift signálu alebo rušenie. Výsledkom môže byť viacero pobytových bodov s mierne odlišnými súradnicami, ktoré v skutočnosti predstavujú tú istú fyzickú polohu. Tento problém môžu pomôcť riešiť techniky, ako je zhlukovanie alebo sémantické obohacovanie. Sémantické obohacovanie je ďalšou výzvou v oblasti automatického vyhľadávania miest záujmu používateľov. Identifikácia sémantického významu bodu pobytu, napríklad či predstavuje domov, pracovisko alebo rekreačnú oblasť používateľa, si vyžaduje ďalšie informácie nad rámec údajov o polohe. Techniky, ako je začlenenie obsahu vytvoreného používateľom alebo využitie ex-

terných zdrojov údajov, môžu pomôcť poskytnúť bodom pobytu ďalší kontext. Napokon, interpretácia výsledkov môže byť náročná vzhľadom na komplexnú a dynamickú povahu správania používateľov. Ten istý bod pobytu môže mať rôzny význam v závislosti od dennej doby, dňa v týždni alebo kontextu používateľa. Na získanie zmysluplných poznatkov z veľkých objemov údajov generovaných automatickým dolovaním miest záujmu používateľov sú potrebné účinné techniky vizualizácie a analýzy údajov.

Na základe znalostí o problémoch a krokoch pri extrakcii sémanticky zaujímavých miest používateľa sme navrhli metódu pre nájdenie polohy domova, práce a ďalších miest, ktoré užívateľ navštevuje. Náčrt metódy je nasledovný: metóda využíva nespracované údaje GPS zozbierané zo smartfónu, extrahuje miesta, kde používateľ strávil významné množstvo času, agreguje ich do významných pobytových oblastí, na základe časových charakteristík z určených pobytových oblastí odvodzuje miesta domova a práce používateľa a ostatným extrahovaným miestam záujmu pridáva sémantický význam. Hlavné prínosy nášho prístupu sú:

- Navrhovaná metóda sa dá použiť nad rôznymi zdrojmi údajov GPS, pretože ako vstupné údaje sú potrebné len surové údaje GPS.
- Metóda dokáže automaticky odvodiť miesta bydliska a práce bez akýchkoľvek označených vstupných údajov.
- Extrahujú sa ďalšie miesta, ktoré používateľ navštívil, odhaduje sa čas odchodu a vypočíta sa trvanie návštev miest.
- Náš prínos spočíva vo využití technológií a údajov s otvoreným zdrojovým kódom na sémantické obohatenie, čo zvyšuje účinnosť a použiteľnosť nášho prístupu.
- Výstup metódy je v dvoch formátoch. Jeden z nich je vhodný na budúce použitie extrahovaných navštívených miest a druhý prezentuje získané výsledky v interaktívnej, pre človeka prívetivej vizualizácii pomocou rôznych mapových dlaždíc a iných vizuálnych prvkov.
- Vytvorili sme vzorový dataset zodpovedajúci skutočnému používateľovi a vykonali sme testy a vyhodnotenia.

Vykonali sme experimenty s použitím reálnych údajov, vybrali sme najlepšie algoritmy a ich parametre tak, aby boli dostatočne robustné, dosahovali dobré výsledky výkonu a presne extrahovali miesta. Výsledky sme vyhodnotili porovnaním s miestami vyexportovanými zo služby Google Timeline. Na základe tejto validácie sme zistili že navrhnutá metóda správne nájde o označí lokácie domova a práce užívateľa. Získavanie bodov záujmu užívateľov a ich sémantické obohatenie dosahuje presnosť okolo 90%, pričom v niektorých prípadoch poskytuje vyššiu úroveň sémantického obohatenia ako výsledky z Google Timeline.

User Location Interpretation Based on Location Data

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Anton Firc. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Alexandra Ligocká
May 15, 2023

Acknowledgements

I would like to thank my supervisor Ing. Anton Firc, for his help and encouragement. Furthermore, I would like to thank my family and friends for their support during my studies.

Contents

List of Figures

Chapter 1

Introduction

Data has become the commodity that sustains much of the digital ecosystem. The global adoption of smart devices has grown at incredible speed in the last decade, especially smartphones are turned into a gateway to users' privacy and provide us with information about their life, where they go, what activities they do, etc. GPS-enabled devices, such as smartphones, allow users to track their outdoor activities through GPS trajectories and researchers to infer further knowledge about the behaviour of mobile users.

In this work, we will focus mainly on data related to the user's location - geolocation. This term reflects the position of a user's device associated with a time identifier, whether that's a home computer, laptop, smartphone, or fitness tracker [24]. This device data is assumed to correlate to a person and is often aggregated to provide significant scale insights into a user's movement.

Latitude and longitude are the most common way to represent position data, as these coordinates can be used to specify the precise location of a place on the Earth's surface [44]. Geohashes are another common format for storing and transmitting position data, as they are easy to generate and can be efficiently searched and indexed. In addition to these formats, there are many other formats that can be used to store and transmit position data between applications, such as GeoJSON, which is a standard for encoding geolocation data as a JSON object, and KML (Keyhole Markup Language), which is an XML-based format commonly used in the context of Google Earth and other mapping applications [15].

However, these representation formats are inappropriate from the point of view of an ordinary person when using applications oriented to the use of location-based services, which are designed to help people find places or navigate to specific destinations. Instead of using coordinates, humans tend to use descriptive statements about their location and attach meaningful and descriptive semantics to locations. Such descriptions can be expressed as „I am at home“, „I am at work“, or „I am at a supermarket“, and are often called places and can be associated with specific coordinates or geographic areas. To address this issue, several techniques to infer semantic meaning to a place were discovered. The level of semantic richness may vary depending on the target application. Some applications use simple principles based on geocoding techniques and others require more complex approaches involving additional data from third-party sources [37].

There are several ways to gather geolocation data, including GPS, WiFi triangulation, and cell tower triangulation. GPS is the most precise method, but it requires a device with built-in GPS capabilities. WiFi triangulation and cell tower triangulation are less precise but can be used with devices that do not have GPS capabilities. Almost every application

working with geolocation data is heavily dependent on the source of the data and the choice of the positioning system [24, 65].

One of the key attributes of geolocation data is time. Besides location and the places themselves, we should be interested in arrival time, departure time, and total duration of the visit. It allows for the creation of a temporal context for the location data. This can be particularly useful for understanding life patterns of movement and behaviour over time. For example, a business might use geolocation data to understand how people are using a particular location at different times of the day or on different days of the week. This can help the business to optimize its operations and to make informed decisions about how to best serve its customers [71].

Nowadays, location data is widely used by social networks, e.g. Facebook and Instagram, search engines - Google and its applications such as Google Maps or Google Earth, as well as by various other mobile applications, including fitness tracking apps and transportation services like Uber and Lyft.

In our work, we focus mainly on extracting semantically important locations from GPS traces, as well as mining and adding broader context to places the user has visited. While the ability to accurately track and extract meaningful location data is a key component of many modern location-based services and applications, there are several challenges associated with this process. Some of these challenges include [27]:

- Positioning, accuracy and precision
- Indoor Location Tracking
- Privacy Concerns
- Billing Issues
- Data Management
- Power consumption and battery drain

The automatic mining users' places of interest are a key part of tasks when studying human behaviour for building new applications. Some potential applications may include [42]:

- Prediction of user movements
- Location and activity recommendations
- Assisting mobile devices in determining their behaviour (e.g. turn on silent mode based on the location where noise is inappropriate)

Another potential application of automatic mining of users' places of interest is in the field of urban planning. By analyzing the spatial distribution of user activities, city planners can gain insights into how people move and interact within urban environments. This information can inform decisions on transportation infrastructure, zoning regulations, and public space design.

One of the main challenges in the field of automatic mining of users' places of interest is data acquisition. Collecting location data can be difficult due to privacy concerns, as well as technical challenges such as signal interference or low battery life of mobile devices. Additionally, it is essential to ensure that the data collected is accurate and representative of the user's actual movements.

Another challenge is stay point detection. While the concept of a stay point seems straightforward, there are various factors that can complicate its identification, such as noise in the data, variability in the user's movements, and the presence of multiple stay points in close proximity. Sophisticated algorithms are needed to filter out the noise and identify meaningful stay points.

The variability of spatial coordinates is another challenge [42]. Even when a user is in the same physical location, their device may report slightly different coordinates due to factors such as signal drift or interference. This can result in multiple stay points with slightly different coordinates that actually represent the same physical location. Techniques such as clustering into regions with the same semantic meaning can help address this issue [65].

Semantic enrichment is another challenge in the field of automatic mining of users' places of interest. Identifying the semantic meaning of a stay point, such as whether it represents a user's home, workplace, or recreational area, requires additional information beyond just the location data. Techniques such as incorporating user-generated content or utilizing external data sources can help provide additional context to stay points [20].

Finally, result interpretation can be challenging due to the complex and dynamic nature of user behaviour. The same stay point may have different meanings depending on the time of day, day of the week, or the user's context. Effective visualization and data analysis techniques are needed to extract meaningful insights from the large volumes of data generated by the automatic mining of users' places of interest.

In this work, we propose a method to extract semantically interesting user's significant locations and other places user visits. The outline of the method is as follows: the method uses raw GPS data collected from a smartphone, extracts locations where the user spent a significant amount of time, aggregates them into meaningful stay regions, infers the user's home and work locations based on temporal features from determined stay regions and adds semantic meaning to other extracted places. The main contributions of our approach are:

- The proposed method can be used over a variety of GPS data sources because only raw GPS data is needed as input data.
- The method can automatically infer home and work locations, without any labelled input data.
- Other visited places by the user are extracted, the leaving time is estimated, and the duration of the places visits is calculated.
- Our contribution lies in leveraging open-source technologies and data for semantic enrichment, which enhances the effectiveness and usability of our approach.
- Output of the method is in two formats. One of them is suitable for future use of extracted visited places and the other one presents results obtained in interactive, human-friendly visualisation using different map tiles and other visual features.
- We created a sample input dataset corresponding to a real user and performed tests and evaluations.

We performed experiments using real-world data, selected the best algorithms and their parameters to be robust enough, to reach good performance results and to accurately extract places. The results were evaluated by comparing them to places exported from Google Timeline.

Chapter 2

Location data and its processing

Location data, also known as geospatial data refers to information collected about the location of objects, elements or people present in geographic space. In general, data processing is the method of collecting raw data and transforming it into usable information. Any position gathered from the real world is a kind of measurement and all measurements contain some degree of error. We need to process location data as well in order to gain usable information and get rid of errors caused by different sources. Typical errors in measurements are included by human errors, caused by environmental characteristics or errors caused by instruments or measuring probes.

This chapter discusses what location data is, explains key terms in geodata processing and analysis and provides an introduction to spatial data quality and interpretation.

2.1 What is spatial data?

Spatial data refer to the real-world geographic objects of interest, such as streets, buildings, lakes, and countries, and their respective locations [63]. In addition to location, each of these objects also possesses certain traits of interest, or attributes, such as a name, number of stories, depth, or population. There are various technologies used to determine the location, generally called positioning systems. One of the most known and commonly used is GPS - a global positioning system. Spatial data can be represented in various forms, including maps, aerial photographs, satellite imagery, and digital data layers [12]. In geographic information systems (GIS), spatial data is typically stored and managed in a spatial database that allows users to analyze and visualize the data in a spatial context [14]. Spatial data is used in a variety of industries and applications. In urban planning, spatial data can be used to analyze population density, transportation patterns, and land use to optimize city design. In environmental management, spatial data can be used to monitor changes in vegetation cover, track animal migration patterns, and identify areas at risk for natural disasters. In transportation planning, spatial data can be used to optimize routes for logistics and delivery, reduce traffic congestion, and improve public transportation systems.

2.2 Accuracy, precision and errors

The error, accuracy, and precision of the location data are often used terms when we process data from various sources - government, open source datasets or commercial sources.

First, we need to define the terms *data accuracy* and *data precision*. Data accuracy describes how close a measurement is to its actual value and is often expressed as a probability. Accuracy can be horizontal or vertical. The accuracy reported in location datasets is almost always the horizontal accuracy, however, vertical accuracy is used as well. Accuracy can be negatively affected by near buildings, trees and can get worse by weather conditions [26]. Precision refers to the variance of a value when repeated measurements are taken. It is a mathematical term represented by the number of digits after the decimal point, ranging from 1-9 and it is represented using latitude and longitude [57]. Precise data may be inaccurate because it may be exactly described but inaccurately gathered. The concept of accuracy and precision is shown in Figure 2.1.

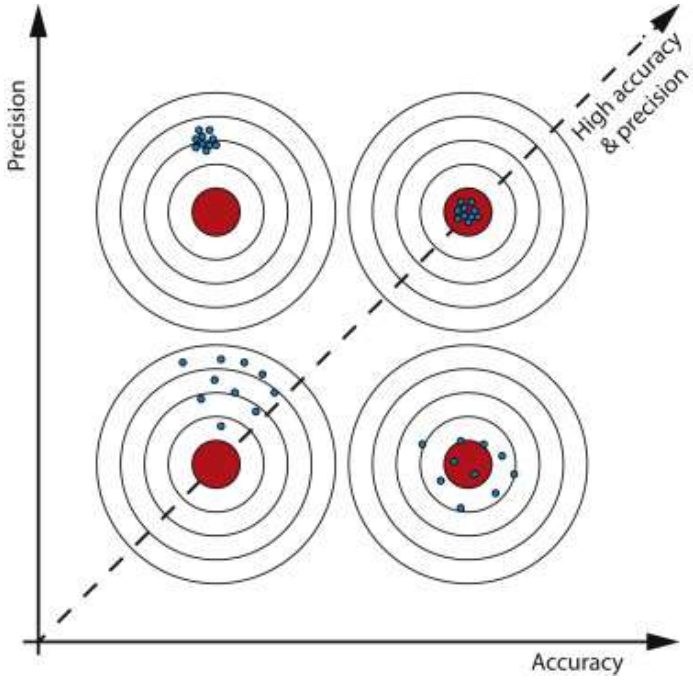


Figure 2.1: Relationship between data accuracy and precision [56].

2.3 Positioning technology systems

Positioning technologies are the technologies used to determine the location of a device or object. These technologies can be classified into two main categories: indoor and outdoor positioning systems.

Well known outdoor positioning system is global positioning system (GPS) [12], which is a satellite-based navigation system that provides location and time information anywhere on or near the Earth. Figure 2.2 shows the GPS architecture, which includes satellites orbiting the Earth, ground control stations, and GPS receivers. The figure demonstrates how these components interact to enable accurate location tracking. GPS operates on the basis of a network of 24 to 32 satellites that orbit the Earth and send signals to GPS receivers on the ground. These signals contain information about the satellite's position and time, as well as the exact time the signal was transmitted. The distance to the satellite is calculated by measuring the time between sending and receiving the signal.

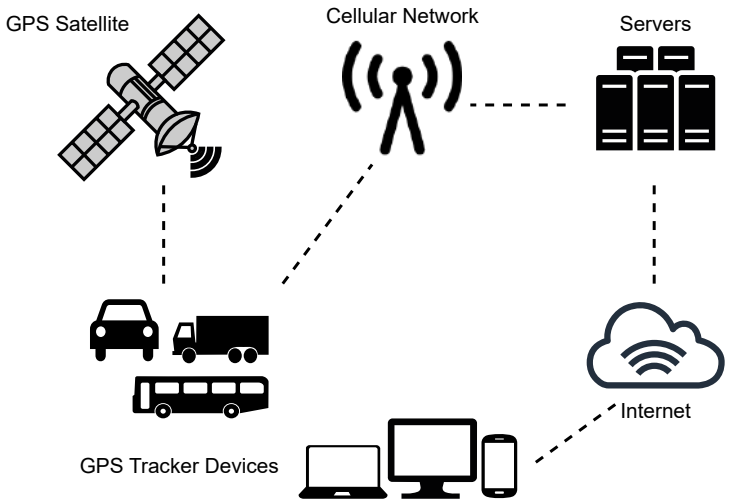


Figure 2.2: Global positioning system architecture.

Signals from multiple satellites allow the receiver to triangulate its position on the Earth's surface to improve positioning accuracy. However, there are also various errors in GPS due to different causes. Types of GPS errors are shown in Figure 2.3. Position errors can be caused by some of the following sources [43]:

- User errors: most GPS errors are caused by user error, either typographical errors when entering coordinates into the receiver or incorrect reference data. These errors can cause errors up to many kilometres away and the GPS receiver has no possibility to identify and correct them. Also, the human body itself can cause signal interference, especially when facing south due to the location of most satellites facing the southern hemisphere of the Earth.

- Satellite clock errors: these errors lead to inaccuracies in the GPS signals and result in errors in position calculations however are monitored and can be corrected by the Master Control Station.

- Orbit errors: the orbit of the satellite is related to the altitude, position and velocity of the satellite and may vary due to gravity and solar pressure variations.

- Ionospheric interference: the effect of ionospheric inference is expressed by the acceleration or the slowdown of the signals due to their refraction during passage through the Earth's atmosphere. Correction of these errors is only possible to half of the possible 70 nanosecond delay, which on the ground can cause up to a ten-metre horizontal error. This kind of error can be reduced using the Wide Area Augmentation System (WAAS), which is a space-based and ground-based extension of GPS.

- Tropospheric interference: this type of error is mainly affected by changes in temperature, pressure and humidity caused by weather changes in the troposphere. The greatest influence on the travel time of GPS signals is due to changes in the electromagnetic field caused by water vapour in this layer of the atmosphere. The noise caused can be mitigated by remote antennas but cannot be corrected by the GPS receiver.

- Multipath interference: this type of interference is caused by reflections of signals from surfaces near the receiver. Common examples of problematic surfaces are car bodies, buildings, power lines, water, and others. An example of the result of multipath propagation is the ghosting image that appears on older televisions equipped with rabbit ear antennas. It is difficult to identify a degraded signal and sometimes impossible to correct, but most signal interference when using GPS in a car can be mitigated by placing an external antenna on the roof of the vehicle.

- Satellite geometry: the relative position of the satellites at a given time is important for proper signal spreading and distance calculation. The ideal geometry exists if the satellites are positioned at a large angle to each other. An example of geometry causing errors is the positioning of satellites in a line or tight grouping.

Other sources of GPS errors are selective availability when signals and their interference are modified intentionally, the number of visible satellites and random-based nature errors.

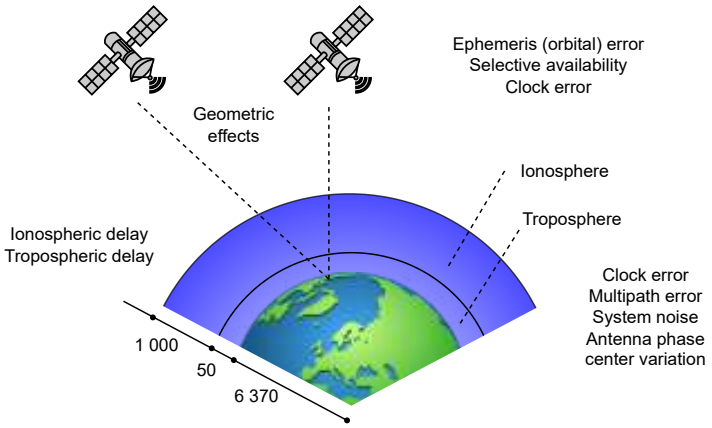


Figure 2.3: GPS errors.

The accuracy issue of GPS can be solved using differential GPS (DGPS). DGPS is designed for applications requiring greater accuracy, such as aircraft or maritime transport applications. To satisfy this requirement, DGPS provides correction for errors caused by delays in signals and multipath errors. Differential GPS comprises a network of land-based reference stations at fixed locations on heavily surveyed sites where a very accurate position can be determined. The position is calculated on reference stations by GPS receivers and compared to the actual known position. The comparison is then used to calculate an error correction factor for each satellite and then broadcasted. In Figure 2.4, we present the architecture of the DGPS system. The figure illustrates the components of the system, including the GPS receiver, the base (reference) station, and the correction data stream.

Among other outdoor positioning systems are the well-known Global Navigation Satellite System (GLONASS), cellular and wireless networks. Global Navigation Satellite System (GNSS) technology has become an integral part of outdoor positioning systems. Besides GPS, GLONASS is another GNSS system that is widely used for outdoor positioning. GLONASS, developed by Russia, is similar to GPS because it provides global coverage and accurate positioning information.

In addition to GNSS systems, cellular and wireless networks are also used for outdoor positioning. Mobile networks such as 4G and 5G use triangulation to determine a device's

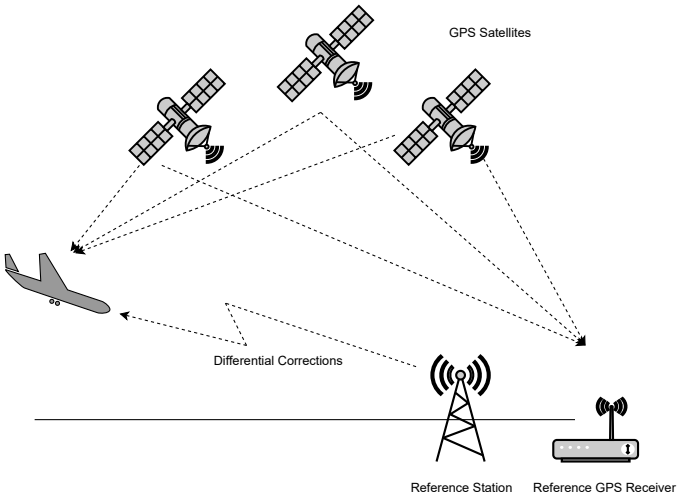


Figure 2.4: Differential GPS architecture.

location based on the signal strength of nearby cell towers. The use of multiple outdoor positioning systems can improve the accuracy and reliability of location information. For example, combining GPS and GLONASS can improve positioning accuracy in areas with obstructed GPS signals. Additionally, cellular and wireless networks combined with GNSS systems can provide positioning information in areas where GNSS signals are weak or obstructed, such as urban canyons or indoor environments near windows.

The second category, indoor positioning systems (IPS), are technologies and methods used to determine the location of an object or person inside buildings or other enclosed spaces where GPS signals may not be available or accurate. These systems use a variety of technologies, including Wi-Fi, Bluetooth Low Energy (BLE), Ultra-Wideband (UWB), Inertial Measurement Units (IMUs), and magnetic positioning [17].

Bluetooth [6] is a wireless standard for Wireless Personal Area Networks (WPANs) and operates in the 2.4 GHz Industrial, Scientific and Medical (ISM) band. Nowadays, this technology is equipped in most devices, such as mobile phones, laptops, etc. Bluetooth's primary purpose is to enable wireless communication between devices within a short range, typically up to 10 meters. However, it can also be used for indoor positioning by detecting Bluetooth beacons' signal strength throughout a building. The use of Bluetooth in location sensing has several advantages, including its widespread availability in devices, low cost, and low power consumption. These advantages make Bluetooth a popular choice for indoor positioning systems, particularly in settings where high accuracy is not critical. One of the main disadvantages of Bluetooth-based positioning systems is their limited accuracy, which typically ranges from 2-3 meters. This level of accuracy may be sufficient for some applications, but it may not be precise enough for others, such as navigation in large indoor spaces. Additionally, Bluetooth-based positioning systems may have a delay of about 20 seconds, which can affect the responsiveness of location-based services. Among the BLE technology is the well-known standard iBeacon developed by Apple, which provides location-based information and services to iPhones and other iOS devices based on proximity measurements.

Wi-Fi-based positioning systems [39] measure the signal strength of nearby Wi-Fi access points to determine a user's location indoors. Wi-Fi access points emit signals that Wi-Fi-enabled devices can detect. These signals are broadcasted with a unique identifier known

as the Service Set Identifier (SSID). One of the main advantages of this technology is the possibility of localizing the position of wide-range devices without needing extra software. A Wi-Fi-based system's accuracy depends on the density of wireless router deployments. In areas with fewer access points, it may be difficult to accurately triangulate a user's position, leading to reduced accuracy. Additionally, other factors, such as interference from other radio signals, can also impact the accuracy of these systems.

2.4 Data models

A data model is an abstract model of elements corresponding to real-world entities. Although there are several data models applicable to location data, we can divide them into three groups - raster data model, which uses grid and cells to represent the spatial variation of a feature; vector data model, representing real-world entities using primitives (points, lines and polygons) and triangulated irregular network (TIN) data model representing continuous surface using triangular facets [32].

Raster data model uses grid-cell data where the geographic area is divided into cells identified by row and column. Raster data includes additional information such as elevation, species type, density and others.

Vectors are the most common type of spatial data used while making maps. In contrast to the raster data model, it uses points described by X, Y coordinate pairs. Fundamental vector data types are points - zero-dimensional objects containing a single coordinate pair, lines - one-dimensional features composed of multiple, explicitly connected points and polygons - two-dimensional features created by multiple lines that loop back to create a "closed" feature, sometimes called areas [55].

The differences between raster and vector representations are illustrated in Figure 2.5, which displays examples of each type of data representation in the context of real-world features like land cover.

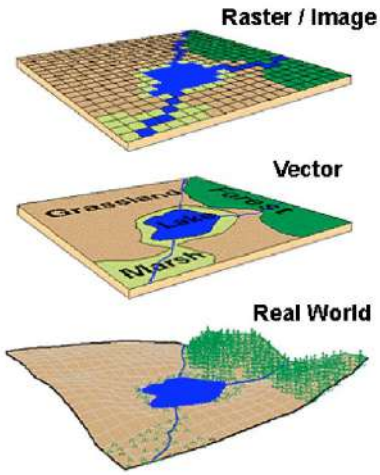


Figure 2.5: Raster and vector representation on real world [53].

Triangulated irregular network models are based on vector models. In this data model, each point is connected by non-overlapping lines to create planes of contiguous triangles of varied sizes. When compared to the raster format, TINs are more efficient in terms of data storage, especially when representing smooth terrains. Another advantage of TIN models

is that they are adaptable to varying data scales, allowing the use of smaller triangles in complex areas and larger triangles in simpler areas, as shown in Figure 2.6.

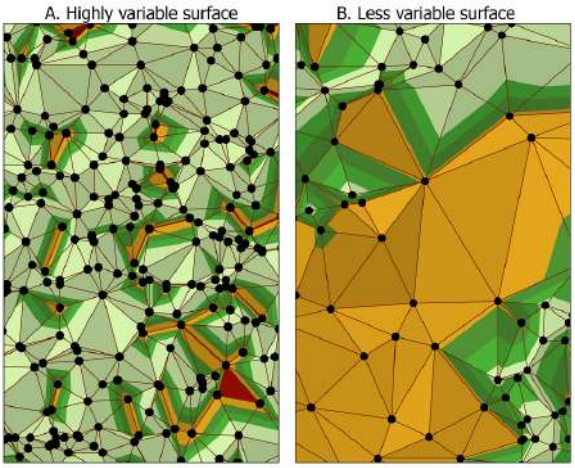


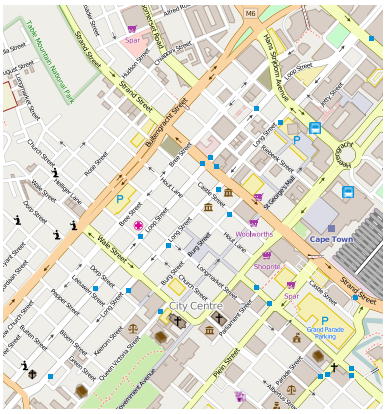
Figure 2.6: Triangulated irregular network [32].

2.5 Location data representation

Location data interpretation is an important aspect of location-based services and applications. The interpretation of location data depends on several factors, such as the intended recipient, the device used to view the data, the data's purpose, and the level of detail required. The most accurate method to represent location data is to use a coordinate system, regardless of the positioning method used to determine the location. However, this method may not be understandable to most users and may require more user-friendly representations of location data.

One approach to make location data more understandable is to describe a position relative to a known location, such as „100 meters from the post office.“ However, graphical representations are the most logical and easily understandable option for many location-based applications [12]. Maps are a common graphical representation of location data, with various types of maps available, such as topographical maps, street maps, and satellite imagery. The difference between these types of maps is shown in Figure 2.7. These maps usually contain several layers, each grouping objects of the same type, such as roads, buildings, landmarks, and points of interest. Users can choose to view one or more layers depending on their needs, making this representation method flexible and customizable. Heat maps are another popular graphical representation of location data, especially for large datasets. Heat maps show the density of data points within a specified area, with warmer colours indicating higher density. They are often used to visualize customer or user behaviour, such as where they spend their time or what areas they avoid.

(a) Topographic map [70]



(b) Street map [69]

(c) Satellite imagery [68]

Figure 2.7: Examples of different map types.

Charts and graphs can also be used to represent location data, providing a quick overview of trends or patterns. For example, a bar chart may show the number of sales in each region, or a line graph may show the change in sales over time for a particular location.

In summary, the interpretation of location data depends on several factors, including the intended recipient, the device used to view the data, and the level of detail required. While the most accurate method is to use a coordinate system, graphical representations, such as maps, heat maps, and charts, are often the most logical and easily understandable options for location-based applications.

2.6 Location data processing

Geo-processing is one of the GIS terms frequently used to describe processes performed by specialized software tools to compare, analyze or manipulate with geographical data [14]. A typical geoprocessing operation takes an input dataset, performs an operation on that dataset, and returns the result of the operation as an output dataset.

Using GPS data to derive human behaviour needs to involve some kind of enrichment process. Some common methods include:

1. *Data cleaning and preprocessing* - GPS data can be noisy and may require cleaning and preprocessing to ensure accuracy and usefulness. This may involve techniques such as filtering out erroneous data points, interpolating missing data or data transformation and standardization techniques to modify data to make it easier to analyze or visualize, for example, a transformation between different projection systems. According to Lun Li et al. [29], there are several ways to perform cleaning on location data. One common approach is applying a map-matching algorithm on location data to identify the true path, consisting of a sequence of blocks matching a known map network [41]. Figure 2.8 shows the true path superimposed on a map, with GPS points matched to the nearest road segment using a map-matching algorithm. In practice, map-matching algorithms apply likelihood or scoring functions to candidate paths. The calculation of these functions can be based on geometric and topological approaches. For example, a naive approach to calculating a geometric score might be to invert the distance of each GPS point from the candidate route or, more complexly, based on probabilities as a nonlinear function of distance based on a structural model of GPS error. The topological score can be based on the difference between the network distance along the candidate path and the Euclidean or great circle distance between GPS points. We can also use routing algorithms such as Dijkstra. Another approach involves using features extracted from the trajectory, such as distance, velocity, direction, time or density.

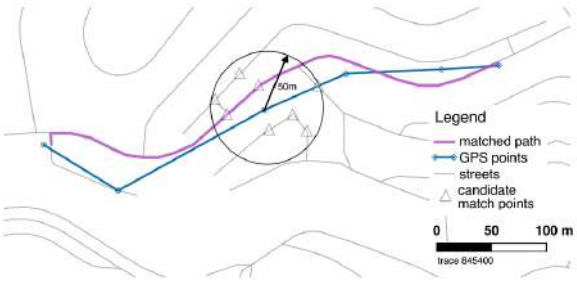


Figure 2.8: Example map-matched trace [41].

Another challenge in cleaning GPS data is random noise. GPS signal consists of noise from several sources, e.g. errors caused by ionospheric effects, multipath interference, satellite clock errors, and receiver noise. To smooth out the GPS data and get rid of random noise, we can use regression models. Kernel smoother statistical techniques are widely used in various applications. The technique estimates a real-valued function as the weighted average of observed data. Kernel refers to a window function and defines the weight. Popular kernels used not only in the task of smoothing data are Gaussian, Epanechnikov, cosine or triangular. For example, Yan et al., in their

work proposed a Gaussian kernel-based local regression model to smooth out the GPS points to handle random noise. Filtering is another approach to obtain correct data. To filter out data points that are irrelevant to the analysis, Parth Bhawalkar et al. [7] preprocessed GPS data by removing all points with speed less than the given value to gain information about when the user is in motion.

2. *Combining GPS data with other data sources* - This approach can help to provide a more complete picture of a particular area depending on the application of data. For example, Jasinski and Baldo [21] used weather conditions, speed limit and important place surroundings obtained from Google API. There are many more types of data which can be used, for example, demographic data, traffic data or data obtained from social media.

3. *Using machine learning techniques* - Machine learning techniques are very popular nowadays, and we can use them for various purposes, even in the context of positional data. Different methods can be used for different purposes, e.g. for behavioural analysis [67], predictions from location data in any context (e.g. prediction of delivery time [19], annotation data [33]) or classification. Popular algorithms include naive Bayes, logistic regression, K-nearest neighbours, support vector machines, classification trees, random forests and neural networks.

4. *Using GIS systems and other tools for visualisation and analysis* - The term GIS refers to geographical information system frequently applied to geographically oriented computer technology, integrated systems used in substantive applications [38]. GIS systems provide a wide range of functions for storing, visualizing, analyzing spatial data and the relationships between them. In current approaches, GIS may be usually used in combination with other technologies. A GIS organizes data into layers, and users can choose the necessary data for a particular project or task [61]. Figure 2.9 demonstrates how a GIS can organize geographic data to create thematic maps with layers of information that can be added to a basemap, highlighting its potential for manipulating spatial information to support decision-making in environmental and spatial planning.

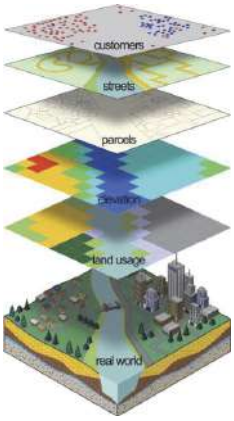


Figure 2.9: Layers of GIS model [61].

Chapter 3

Location-based services and suggestions

When we focus on the user's location and behavioural profile built upon his location history, we can see that such a model reflects his everyday experiences and decisions. The rise of location-based technologies has led to the emergence of various services that use our significant locations to provide personalized recommendations, notifications, and experiences. Such services can be helpful in everyday life, but they also carry threats we should be aware of. Users' location data belongs to the group of sensitive data because it allows the identification of the specific individual who provided it and can be misused for attacks on their privacy or security.

In the previous chapters, we covered the basics of location data, including its definition, how it is collected, and how it can be represented. In this chapter, we will delve into the concept of location-based services, which utilize location data to provide personalized recommendations, notifications, and experiences to users. Additionally, we discuss the most common approaches to implementing recommendation systems. We will also explore the idea of users' significant locations, and we will discuss the security and privacy challenges associated with location-based services, including the potential for data breaches and misuse of location data.

3.1 Location-based services

Location-based services (LBS) rely on the location data obtained from a mobile device to provide users with relevant and customized services. The mobile device's ability to provide precise location data rely on positioning technology and makes it possible for LBS to deliver location-specific content and services to users.

LBS can be categorized based on the type of customer needs they serve and the information they provide in a specific space-time setup [52]. These needs could include mass-market consumers, speciality niche markets, businesses, and various inventory and object tracking in physical space. The first type of demand comes from consumers who require location and navigation information. This includes queries such as „Where am I?“ and „How can I get there?“. LBS that cater to these needs include maps, driving directions, directories, yellow page listings, and business descriptions. The second type of demand relates to the delivery of personalized information at the point of need. This includes information on new or interesting products, services, and promotions based on advanced knowledge of customer

profiles and preferences. Service providers need access to customers' preference profiles to deliver this information transparently and efficiently, either through proprietary databases or arrangements with LBS providers. Other demands include, for example, specialized applications aimed at specific market segments or industrial and corporate applications used to track materials or people. Some of the most popular location-based services today include:

- Google Maps: Google Maps is one of the most widely used location-based services around the world. It provides users with turn-by-turn directions, real-time traffic updates, and information on nearby businesses and attractions.
- Uber/Lyft: Ride-sharing apps like Uber and Lyft use GPS to track the location of their drivers and riders, making it easy for users to hail a ride and for drivers to find their passengers.
- Yelp: Yelp is a platform that provides users with reviews and ratings of businesses in their local area, such as restaurants, bars, and shops. It uses location data to suggest nearby businesses and helps users find the best places to go.
- Weather apps: Weather apps like AccuWeather and The Weather Channel use location data to provide users with up-to-date information about local weather conditions.
- Social media apps: Social media platforms like Facebook and Instagram use location data to provide users with location-specific content, such as posts and stories from friends and businesses in the area.
- Fitness apps: Fitness apps like Strava, MyFitnessPal and Runkeeper use location data to track users' workouts and provide them with data on personalized recommendations for their fitness routines.

3.2 Security and privacy challenges in location-based services

As mentioned in the introduction of this chapter, location data can be misused, which can have an intrusive impact on users' privacy and personal freedom. It is, therefore, important that this data is protected and processed concerning the principles of data protection and privacy.

If we consider only users' views on privacy in LBS, we can identify three main independent parts, namely identity privacy (i.e. Who you are), location privacy (i.e. Where you are), and query privacy (i.e. What you want to know) [50]. The most popular category of LBS privacy is location privacy and thus, several methods have been proposed to protect it, e.g. spatial and temporal cloaking (anonymisers and pseudonymisers) or individual and collaborative location obfuscation. Most methods use trusted third parties to proxy users and providers to hide their real identities or collaboration to hide real locations.

Examples of malicious activities are tracking threats when an intruder identifies the user's mobility pattern and predicts his future locations; identification threats, using the user's locations to infer his identity from an anatomized database; and profiling threats, an intruder uses the user's points of interest to profile him in a variety of sensitive information, such as political view and health condition [49].

There are many more location security problems, generally, we can divide them into two categories, namely direct and indirect location privacy problems [11]. Mostly pseudonymized users are exposed to indirect location privacy problems. Intruders usually study location information and patterns in location sequences to create quasi-identifiers and apply methods to disclose user identity. Examples of attacks to reveal users' identity are called front garden or writing desk, when attackers misuse knowledge about places where a person typically stays, e.g. a residential house (for outdoor scenarios) or office room (for indoor scenarios) and place where LBS request comes from in combination with information from public sources (e.g. maps or telephone books). Another example of an attack is a commuter attack, when attackers use the temporal-spatial sequence of places visited by a user, e.g. a commuter's daily way to get to work.

3.3 User's significant locations

Significant locations are a term often used to refer to a list of places that are meaningful or important to a particular user. This list typically contains places that a person frequently visits or has visited in the past. These locations may be important to a person for a variety of reasons, such as work, home, or social activities [36].

This list is typically compiled based on data from the user's device, such as location information from GPS, Wi-Fi, and cellular networks. The list may include the user's home, work, and other places that the user frequently visits, such as coffee shops or gyms.

Significant locations are typically tracked using geolocation data and can be accessed through a user's location history. This information can be used by applications and services to provide location-based features and recommendations.

3.3.1 Significant locations on iPhone

Location services on iPhone and other Apple products use different sources of user's location. The device's approximate location is collected using GPS and Bluetooth, if available, in combination with Wi-Fi hotspot and cell tower locations [2]. The device may also use other sensors, such as the accelerometer and gyroscope, to help improve the accuracy of the location information. In addition to location data, apple uses information from map searches, calendar data and other apps.

Collected data is encrypted using end-to-end encryption (E2EE) [3], which prevents third parties from accessing data while it's transferred from one end system or device to another. The data is encrypted on the sender's device and can only be decrypted on the recipient's device, using a secret key that is known only to the sender and recipient. This ensures that the data is secure and private, even if it is intercepted by a third party during transmission, in this case, to Apple's servers [35].

In addition to the significant locations mentioned above, Apple uses location-based system services for other purposes, for example, routing and traffic or location-based suggestions. When the user disables location services on the device, the IP address is used to match the geographic region.

It is not clear which algorithms Apple uses to create a list of significant locations as this information is not publicly disclosed. It is likely that the feature uses a combination of machine learning algorithms and statistical techniques.

3.3.2 Significant locations on Google

Similar to Apple's iPhones' significant locations, the techniques they use are not publicly discussed. There are several Google applications using location data - the most common are Google Maps and Google Maps Timeline. Google Maps Timeline allows users to enable gathering location data and modify it.

Google uses various algorithms in the backend of Google Maps to provide accurate and up-to-date information about locations and directions. Some of the methods include image recognition, machine learning techniques geospatial data analysis. They also include various sources of data. Real-time data is gathered from an IP address or device location, and also the user's past activity on Google sites and services [16]. To create personalized ads, they use location history, which is anonymized and aggregated and identifying information is not shared with advertisers.

3.4 Location-based recommendations

Location-based recommendation system is one application of a recommendation system which incorporates users' location data to provide more relevant and personalized recommendations. Generally recommendation system refers to some kind of decision-making process, such as what product to purchase, what music to listen to, or what online news to read and is applicable in various areas. Location-based services incorporate location information about users. Those systems may profit from information gathered from of Location-Based Social Networks (LBSNs), e.g. Yelp, TripAdvisor and Foursquare, which allow users to add check-ins, together with reviews, ratings and other metadata. The most common approaches to building such a model are collaborative filtering, content-based filtering and hybrid recommendations approaches.

3.4.1 Collaborative filtering

Collaborative filtering is a technique used to identify relationships between user data and items used in recommender systems. The base of collaborative filtering algorithms is a database of user votes used to predict the utility of items to a particular user [9]. Generally, we can divide them into two categories - *memory-based* and *model-based* algorithms. The basic difference between memory-based and model-based collaborative filtering is shown in Figure 3.1.

Memory-based algorithms work with an entire database consisting of a set of votes corresponding to the vote for users on given items. Examples of this approach are neighbourhood-based collaborative filtering and item-based/user-based top-N recommendations. In user-based approaches, the value of ratings user a gives to the item j is calculated as an aggregation of some similar users' ratings of the item. The predicted vote for an active user for a given item can be represented as a weighted sum of the votes of other users from a collaborative filtering database. The predicted vote can be calculated using the following equation 3.1,

$$p_{a,j} = \bar{v}_a + \kappa \sum_{i=1}^{n} w(a,i)\,(v_{i,j} - \bar{v}_i) \tag{3.1}$$

where $p_{a,j}$ represents weighted sum of the votes for active user a and item j, n is the number of users in database with nonzero weights $w(a,i)$ and κ is a normalizing factor. The weights

reflect the similarity between each user i and the active user a. Vote for user i on item j is represented by $v_{i,j}$,$\bar{v}_i$ defines the mean vote for user i and $\bar{v}_a$ is the average rating of user a for all the items from the database. Similarity computation is an important step in memory-based filtering. Widely used is correlation-based similarity computation using Pearson correlation. Other correlation-based similarities include modifications of Pearson correlation - constrained Pearson correlation; Spearman rank correlation using ranks instead of ratings, or Kendall's τ correlation using relative ranks to compute correlation. Another approach to the calculation of similarity adopted the idea of measuring the similarity of documents from the field of information retrieval. This group is called vector cosine-based similarity. The similarity between two items i and j is computed from matrix R with size $m \times n$ as the cosine of the n dimensional vectors corresponding to the ith and jth column of matrix R.

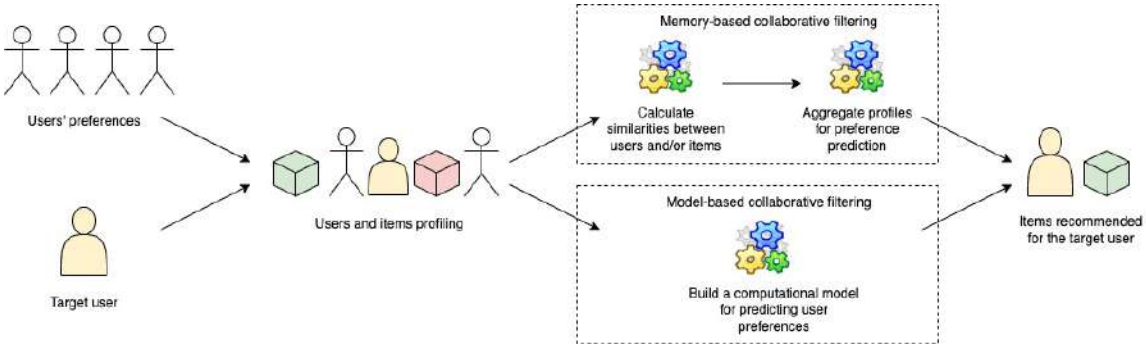


Figure 3.1: A basic difference between memory-based and model-based CF [72].

To achieve better prediction performance and overcome the shortcomings of memory-based collaborative filtering algorithms, model-based approaches have been investigated. They operate over a user database as well, but use it to train a model that can predict a user's preference for items. The model can be a data mining or machine learning algorithm, including Bayesian belief nets, clustering collaborative filtering models, Markov decision process-based or latent semantic collaborative filtering systems [59]. Other model-based techniques may include other approaches such as two-stage order learning models, association rule-based or maximum entropy approaches. One of the problems associated with memory-based models is performance decrease in the case of sparse data. Model-based models try to alleviate this problem using dimensionality reduction techniques - singular value decomposition or principal component analysis to compress the user-item matrix into a low-dimensional representation in terms of latent factors. Many popular services across the web use this approach, such as Reddit, YouTube, Amazon or Linked In [64].

3.4.2 Content-based filtering

The term content in a content-based recommender system refers to the descriptive attributes of items that are used to make recommendations [1]. Content-based filtering methods bring advantages in cases when rating data for a given item is not available or is submitted by just a few users. Ratings of items with similar features can be used instead. In Figure 3.2, the concept of content-based filtering is depicted.

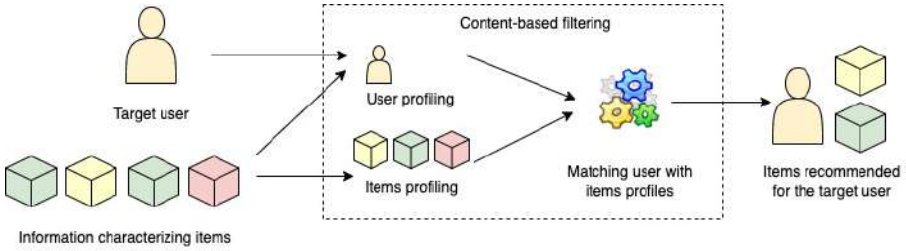


Figure 3.2: Content-based filtering [72].

Such systems are built on user and item profiles. To build a user profile, the system focuses mostly on a model of the user's preference and the history of the user's interaction with the recommender system. An item profile is described as a set of discrete attributes and features characterizing the item within the system. Widely used is vector space representation, also called term frequency and inverse document frequency (tf-idf) algorithm [66]. Using this algorithm we compute weights denoting the importance of each feature. Simple approaches may also use the average values of the rated item vector. Other, more sophisticated approaches, use machine learning methods Bayesian Classifiers, cluster analysis, decision trees, or artificial neural networks.

3.4.3 Hybrid recommendations approaches

Both, collaborative and content-based filtering approaches come with bottlenecks. To overcome the main disadvantages hybrid recommendation approaches are being used instead. In cases where the is a variety of input data, we can combine both methods for the same task. According to Burke [10], hybrid recommender systems can be classified into the following categories:

- Weighted - The final score is computed as the weighted aggregate of scores from both collaborative and content-based filtering methods. The aggregation function may be a simple sum or average computation, or it may involve more sophisticated techniques such as matrix factorization or neural networks.
- Switching - The algorithm involves switching between recommender systems based on certain criteria and task needs. For example, in earlier stages can be used recommender system which can better handle cold-start issues. In later stages, other systems may be used to reach better accuracy.
- Mixed - This type of recommender is not combining scores as in weighted systems, but recommendations from more than one technique are presented together. This approach is quite different from the others and is applicable in complex item domains, e.g. recommendations for a program of television viewing.
- Cascade - This hybrid technique includes a staged process. One recommender system refines recommendations given by another system. The efficiency of this method comes from cascade's second step focused only on items for which additional discrimination is needed.
- Feature augmentation - The architecture of the feature augmentation hybrid model consists of stages similar to the cascade model, but the difference is that the output of

one recommender system is used to create input features for the next one. This kind of hybridization brings advantages mainly in classification tasks and offers performance improvements.

- Feature combination - Additional feature data may also be helpful to build a merger recommender system. This kind of approach uses a single recommendation algorithm, either content or collaborative, and features from different data sources are combined together. The combination of data reduces the overall sensitivity of the system.
- Meta-level - Another way of combining two recommendation systems is using a model generated by one system as the input of another. This approach differs from other approaches in the way we use the whole model as input, not only computed features. A typical combination being used is that a collaborative system is modified to use the content features. The benefit of such a system comes from this approach is that the model is a compressed representation of the user's profile and the collaborative mechanism operates on information-dense representation.

Chapter 4

Semantic enrichment of location data

In previous chapters we discussed the amount of data collected on users' is increasing. This data is utilized by various location-based services across various sectors, including social media, e-commerce, transportation, and healthcare. However, raw location data often lack the semantic richness required to provide additional contextual information and deeper insights into user behaviour. Despite the growing quantity and precision of location data obtained from mobile devices, its semantic quality remains a concern. In this chapter we discuss concept of semantic enrichment and the most common tasks and approaches.

4.1 Semantic enrichment

Semantic enrichment of location data refers to the process of adding additional context or meaning to raw location data to make it more useful, easier to understand, and can enable more advanced analyses and applications such as location-based search and recommendations. Semantic enrichment also provides several benefits to end users, such as personalization, better decision-making, time savings, and overall contributes to an improved user experience by providing more accurate, personalized, and relevant results.

According to Ibrahim et al. [20] process of semantic enrichment consists of one or more of the following tasks: segmentation, annotation and behaviour recognition.

4.2 Segmentation

Segmentation is the process of dividing location data into segments or regions, typically based on some common characteristics or attributes of the locations within each segment. Segments may also be called *episodes* and may reflect user behaviour in location data, e.g. divide trajectory into stay-points and move-points.

The choice of segmentation algorithm will depend on the specific goals of the analysis, as well as the characteristics of the data itself. There are a few characteristics of data which may influence the choice of algorithm including the level of granularity, size and complexity of the dataset or desired level of interpretability. The most common algorithms for this task are clustering algorithms (e.g. k-means [4, 7], hierarchical clustering [75], DBSCAN [21, 60]), rule-based approaches or machine learning techniques.

Clustering algorithms are powerful tools that can provide insights into the structure and patterns in data and we can classify them into groups of unsupervised machine learning methods, so we do not need any labels in the input dataset to be able to use any clustering algorithm. Another advantage of these methods is that we can use them for data reduction which can lead to better results in the next steps of analysis. We can divide them into categories centroid-based, density-based, distribution-based and hierarchical clustering.

4.3 Annotation

Annotation refers to the term of adding more context to our data. In the context of location data, we may be interested in addresses consisting of more information, e.g. city, state, and street. To gain more information, we can use various sources of data, especially external. Those may include Open street map API, Foursquare API, Google Places API, LinkedGeoData and many more. Thanks to this information we can add more descriptive data labels like the type of place (e.g. restaurant).

Internal sources employ contextual data collected explicitly or implicitly alongside the GPS data. To annotate episodes based on explicitly collected data; users are required to annotate the extracted segments, and then a classification task is conducted to train a model that utilises additional features (e.g. temporal features).

The first approach (external data) mostly does not require interaction from users, but contextual data sources providers may require authentication, login or using specially generated API keys and the accuracy and amount of additional data of results may differ. The second approach (internal sources) usually requires human collaboration or the creation of annotations to train the models used.

4.4 Behaviour recognition

Behaviour recognition is the process of identifying and classifying activities on the basis of observations from different sources, such as sensor data streams or any other source of spatio-temporal data [18]. The requirements of the process are defined by the application domain. According to research [20], we can divide behavioural analysis into two groups - *episode-based* and *trajectory-based*. The main difference between the two approaches is the target of mining features. Episode-based behavioural analysis focuses on trajectory components and studies how these components and their latent features correlate with each other to form a behaviour. The knowledge extraction may focus on episodes of similar types, such as counting the frequency of similar episodes to get the number of visits to a specific place or study the inter-relations across different episodes to investigate preferred types of transportation between places. Trajectory-based approaches use the entire trajectory to extract behavioural knowledge. The behavioural habits may be extracted based on aggregating similar trajectories and performing sequential patterns analysis. An example of using this method is the extraction of daily habits from multiple daily trajectories.

Chapter 5

Analysis of current approaches

As mentioned earlier, location data has many uses. It can be used in a variety of applications ranging from social networking, mobile devices and location-based services such as maps and navigation to user profiling and advertising and suggestions based on the user's location. Several related papers and researches have been studied in order to investigate the different types of methods used for data preprocessing, extraction of points (regions) of interest, exploration of user location changes and prediction of user movement.

This chapter describes the approaches and experiments of researchers' work in areas focusing on user location data in order to obtain, process, analyze, interpret, or predict user location. Some of the available approaches for each task are discussed in the previous chapters.

5.1 Stay points

The term stay-point reflects the location of the user with which he is in some relationship, e.g. he has spent a certain amount of time in a given place. Stay-points are important in many applications that rely on location-based data, such as location-based advertising, urban planning, and transportation planning.

Formally we can define stay-point by latitude and longitude coordinates, arrival and departure time. We can calculate it from a set of time consecutive GPS points

$$\mathbf{P} = \{p_i, p_{i+1}, \ldots, p_j\},$$

where p_i is the first and p_j the last location point. Each point consists of longitude, latitude coordinates and time information. When stay-points are retrieved, we can use them to define location history sequence $S = s_1, s_2, s_3, \ldots, s_n$, where each stay-point s_i consists same information as a GPS point.

As explained earlier, time and geographical area are important, so we need to set thresholds for time and distance. To constrain the size of the point's geographical area, Equation 5.1 can be used.

$$\text{distance}\,(p_i, p_x) \leq \theta_d, \forall i < x \leq j \tag{5.1}$$

Basically, the equation says the distance between the first location point and any other contained in set $\mathbf{P}$ must be shorter than θ_d. To ensure the condition with minimum time spent in one place, we can use Equation 5.2. This equation says the time interval between

the timestamps of the first location point (p_i) and the last one (p_j) is greater or equal to threshold value θ_t.

$$|p_j.ts - p_i.ts| \geq \theta_t \tag{5.2}$$

From the set of points satisfying both conditions, we select the representative position as the mean value of longitude and latitude coordinates. Figure 5.1 shows the process of transforming points defined by latitude, longitude and timestamp into stay points.

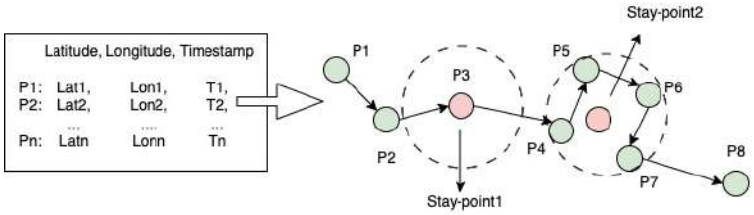


Figure 5.1: Stay-points from GPS traces.

According to [51, 65] we can divide popular methods to compute stay-points from trajectories of time-sequenced GPS points into three major categories:

- Geometric strategies
- Fingerprinting strategies
- Inertial sensor-based strategies

5.1.1 Geometric detection strategies

The geometric strategies rely on distances between physical locations. We can further divide them into three categories: clustering-based, differential-based and probabilistic approaches. Geometric strategies are commonly used with GPS, but also applicable in positioning based on Wi-Fi.

Clustering-based strategies rely on grouping raw GPS coordinates based on their proximity in space and time. There are several clustering algorithms commonly used for this task, among them K-Means clustering [5] or Density-Based Spatial Clustering of Applications with Noise (DBSCAN). DBSCAN method uses input parameters $minPts$ and ϵ. Parameter ϵ defines the radius of the neighborhood around each data point and specifies the maximum distance that two points can be from each other and still be considered part of the same cluster. Parameter $minPts$ is used to specify the minimum number of points that must be present within a given point's epsilon neighborhood in order for that point to be considered a core point. Every point which has at least $minPts$ in a ϵ radius is considered resulting in GPS points that are within a specified spatial distance threshold and have a minimum number of neighbouring points within that distance. We classify data points using $minPts$ and ϵ parameters into three categories, namely core points, border points and noise points [58]. A point is classified as a core point if it has at least $minPts$ neighboring points within a distance of ϵ. Core points are at the center of a cluster and are used to build the clusters. A point is classified as a border point if it is within the ϵ radius of a core point, but has fewer than $minPts$ neighbors within that radius. Border points are part of the cluster, but are not at the center of it. Finally, points that are neither

core nor border points are considered as noise points, which do not belong to any cluster. Example of DBSCAN clustering with *minPts* set to 4 and three types of points is shown in Figure 5.2.

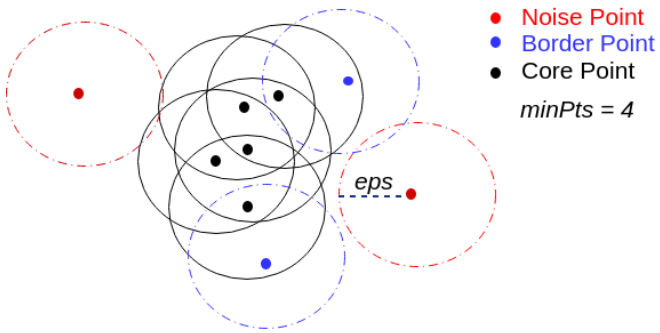


Figure 5.2: DBSCAN clustering with core, border and noise points [58].

Modifications of DBSCAN algorithms can be used as well. Density-and-Join-based algorithm (DJ-Cluster) [77] extends DBSCAN by incorporating the time dimension into the clustering process, ensuring that points within a specified time threshold are included in the same cluster.

The algorithms mentioned above are considered to be point-based clustering methods. On the other hand, trajectory-based algorithms cluster locations by taking advantage of the temporal continuity of trajectories. For example, algorithm CB-SMoT [48] also involves time dimension and is based on the speed variation of the trajectory.

Previous algorithms deal with stay-point detection in trajectories. Another possible method Spatial-Temporal DBSCAN (ST-DBSCAN) deals with both space and time features. Similarly to the DBSCAN algorithm, it uses parameter ϵ. To evaluate similarities between cluster points it uses distance metric parameters $Eps1$ and $Eps2$ to define the similarity of spatial and non-spatial values. The $Eps1$ parameter is used to measure the spatial similarity between points, while the $Eps2$ parameter is used to measure the temporal similarity between points [8].

Differential-based stay-points detection strategies focus on time and space as well as clustering approaches but are based on differences between individual GPS points. All of the differential-based approaches utilize temporal and spatial parameters together, but each algorithm has a unique method for evaluating parametric differences. These methods are unsuitable for processing long user location histories due to the computationally intensive nature of feature changes. They are thus more suitable for online processing due to the inherent streaming nature of the data.

Algorithms in category of probabilistic approaches use probabilistic modelling to identify locations where a user has spent a significant amount of time. Probabilistic modelling involves modelling the distribution of GPS points using a probability density function (PDF) and then identifying locations where the PDF is significantly concentrated. Zhang et al. [73] proposed a Gaussian mixture model to cluster GPS points into stay points using an online algorithm to incrementally estimate the model and then evaluate the learned Gaussian mixture to identify the most probable components corresponding to important locations, based on a priori probabilities.

Another probabilistic-based approach was proposed by Petteri et al. [45]. They introduced a Bayesian model using the Markov chain Monte Carlo Gibbs sampling algorithm, which sequentially updates each parameter and samples from a probability distribution conditioned on the values of the other parameters.

5.1.2 Fingerprinting detection strategies

Fingerprinting is a popular approach for indoor localization that involves creating a database of wireless signal fingerprints at known locations and then matching the signal pattern of an unknown location to the closest matching pattern in the database to estimate its location.

The fingerprinting approach relies on finding similar patterns between the received signal strength values of wireless signals (such as Wi-Fi or Bluetooth) at the target location and those in the database [65]. The similarity can be calculated using different metrics, such as the number of beacons the fingerprints have in common or similarity parameters like Tanimoto or Sørensen's similarity index. However, one of the disadvantages of the fingerprinting approach is that the learned patterns lack a direct reference to physical location. To address this issue, some fingerprinting-based systems combine the fingerprinting approach with other techniques to obtain accurate location information. For example, GPS can be used to obtain location information for the fingerprint database, which can then be used to estimate the location of an unknown device based on the similarity of its wireless signal pattern to the patterns in the database.

Vidmar et al. [65], used fingerprinting approach to find Wi-Fi stay points. For more accurate detection they utilize method of sliding windows. To further tolerate an unstable radio environment, the method uses a scan window, which determines the duration of the scanning process for each access point. By using a scan window, the method is able to cope with signal fluctuations and obtain more stable location data. The similarity between two fingerprints $F1$ and $F2$ is then computed using Tanimoto coefficient 5.3.

$$T\left(F_1, F_2\right) = \frac{F_1 * F_2}{\left(|F_1|\right)^2 + \left(|F_2|\right)^2 - F_1 * F_2} \tag{5.3}$$

In Equation 5.3, $F1$ and $F2$ are vectors of compared scan windows, $F1 * F2$ is a scalar product between both vectors, and $|F1|$ the Euclidean norm of the vector $F1$.

5.1.3 Inertial sensor-based detection strategies

The inertial sensor-based approach most commonly relies on accelerometer data to detect changes in motion, such as movement or stillness, to determine stay points, which are locations where a device remains stationary for a certain period of time [65].

Inertial sensors are typically less power-hungry than other localization technologies, such as Wi-Fi or Bluetooth, and can provide location information even in areas with poor or no wireless signal coverage [25]. However, the location information obtained from inertial sensors is generally less accurate than that obtained from other technologies because they do not directly provide physical location information. Instead, the location information must be inferred based on the device's movement and stay points. We can use the same techniques as in fingerprinting detection strategies to obtain more accurate location information from an inertial sensor-based approach.

5.2 Stay regions

Aggregation of stay points into stay regions is commonly used when stay points are obtained using geometric detection strategies to find stay points. The term stay regions refers to a geographic area that encompasses multiple stay points. Stay region then holds the same semantic meaning to a user and can be represented using average x and y coordinates. Aggregating stay points into stay regions typically involves clustering techniques such as K-means [4], density-based [75], or grid-based clustering [74]. The concept of stay regions is applied for several reasons. The concept of stay regions is useful because previous research has demonstrated that even when multiple measurements are taken at the same location using location systems like GPS, errors and variations in the measured phenomena can result in different coordinates. This remains true even if the user stops at precisely the same point every day. Consequently, two stay points with the same semantic meaning may not necessarily have the same coordinates.

5.3 Mapping stay-points to POIs

A point of interest (POI) is a specific place or location point on a map that someone might find interesting or useful. Mapping GPS points to POIs is one form of semantic enrichment of positional data. Points of interest are usually defined by their geographical coordinates along with additional attributes such as name, category, address, opening hours and contact details. According to Furletti et al. [13] process of enriching GPS points with corresponding POI objects involves two steps: preprocessing phase, when we need to gather GPS data and transform them into stay points since we are interested in places where user spends a larger amount of time and the second phase when we obtain the most probable activities associated to the POIs. Corresponding POIs can be filtered by both spatial and temporal rules. Spatial rules can include criteria such as proximity to the POI, the size of the POI, the shape of the POI, and other spatial attributes, e.g. a GPS point must be within a certain distance of the POI to be considered a match. Temporal rules include criteria such as the time of day, day of the week, or month of the year, e.g. the temporal compatibility of the arrival and departure to the stop with the opening hours of the POIs.

There are several POI data providers, among them the most well-known Google Places API and OpenStreetMap (OSM). The Google Places API is generally considered to be the most accurate and complete source of POI data. However, the use of data from the Google Places API may be restricted by its license terms, which can be difficult to comply with and can bring billing issues. Google categorises the use of its APIs into different categories and some of these categories may be chargeable depending on the number of API calls and the amount of data that is required. Their API is one of the most popular and publicly available alternatives to Google Places API for retrieving POI data. OpenStreetMap is a crowdsourcing platform for digital maps where users contribute their own data on geolocated objects, including POI. The OSM API provides access to these data and enables developers to create applications and services based on them.

Compared to Google Places API, the OSM API may be less accurate and contain less information on POI. However, it is freely available and provides greater flexibility in the use of the data, making it an attractive option for many developers and projects. It offers a wide range of features for retrieving and processing POI data, such as POI searches based on keywords and criteria such as category, distance, and geographic area. It also provides data on public transport operations and cycle routes, which can be useful for many applications.

In addition, OSM displays physical and cultural features on the map, which are divided into several categories. When extracting users' significant places and their examples, the most interesting ones are shown in Table 5.1 [46].

Table 5.1: Map features divided into categories with examples.

POI group	POI category	Examples
Amenities	Sustenance	bar, cafe, fast food, food court, pub, restaurant
	Education	college, driving school, kindergarten, language school, library, training, school, university
	Transportation	bus station, car wash, fuel, parking
	Healthcare	baby hatch, clinic, dentist, doctors, hospital, pharmacy, veterinary
	Entertainment	arts centre, cinema, events venue, fountain, nightclub, theatre, planetarium, music venue
Buildings	Accommodation	apartments, barracks, dormitory, hotel, houseboat
	Commercial	hotel, house, industrial, kiosk, office, supermarket, retail
	Civic, amenity	bakehouse, bridge, college, fire station, hospital, train station, kindergarten
	Sports	grandstand, sports hall, stadium
Crafts	Craft	fatelier, bakery, brewery, candlemaker, carpenter, clockmaker, distillery, floorer, photographer
Shops	Food, beverages	alcohol, bakery, butcher, greengrocer, tea, spices, seafood
	General store	general, mall, supermarket
	Clothing, shoes, accessories	bag, clothes, fabric, shoes, tailor, watch
	Discount store, charity	second hand, charity
	Health, beauty	cosmetics, hairdresser, herbalist, massage, optician, tattoo
	Household, gardening	florist, garden centre, glaziery, locksmith, appliance
	Furniture, interior	bed, candles, carpet, curtain, tiles, doors, antiques
	Electronics	computer, mobile phone, telecommunication
	Outdoors	bicycle, hunting, fishing, golf, ski
	Art, music	art, camera, craft, video, musical instrument

5.4 Extracting personally interesting places

A personally interesting place is a location that holds some level of significance or importance to an individual based on their personal experiences, preferences, and interests. This can be a place that they visit frequently or a location that is related to a specific activity or interest, such as a favourite restaurant or a museum. Such a place carries a semantic meaning (e.g., home, work, etc.) and therefore, it is not preferable to interpret it by a GPS point described by longitude and latitude. According to Lv et al. [37], we can divide the problem of obtaining semantic places into two phases: physical place extraction and semantic place recognition.

5.4.1 Physical place extraction

As mentioned earlier in this work, we can obtain location data from different sources. According to the source of data, we can divide approaches to physical place extraction into two categories: fingerprint-based approaches, where the source of data are radio beacons (e.g., cell towers, WiFi APs, etc.) and geometry-based approaches. We focus mainly on GPS data, so physical location is interpreted by GPS coordinates and in geometry-based approaches a place is represented by points, circles or polygons. Most of the existing geometry approaches apply clustering algorithms to identify places. Point-based and trajectory-based clustering approaches include using clustering algorithms.

Ashbrook and Starner [4] used a modification of the K-Means clustering algorithm for obtaining locations. This quite simple clustering approach was adopted by Jiang et al. [22]. In addition, they used extracting *Region-of-Interest (ROI)*, referring to the integrated urban areas with specific functionalities that attract people's attention and activities [60]. Lv et al. [37], introduce the method of extracting physical places based on a two-step clustering by taking into account the signal loss problem of GPS. Approaches to solve whether multiple points correspond to the same physical location can be solved using a distance threshold [23] or finding intersections [48].

5.4.2 Semantic place recognition

Semantic place recognition can be viewed from two perspectives: publicly interesting places and personally interesting places. Manual labelling of locations is one approach to adding semantic meaning to locations, but this approach requires user cooperation with the system and is not sustainable as the number of locations increases. Techniques for automatic inference of semantic meaning have been presented in many papers. Physical location can be transformed using a reverse geo-coding technique, as proposed in [34]. However, this approach returns the postal address for a given location, which may be insufficient. Another approach to obtaining semantic meaning to a location is using POI databases, as described in Section 5.3. Inferring the personal semantic meaning of places requires analysis of user mobility patterns, spatial relationships, and temporal relationships. For example, querying the POI database for a place returns a restaurant, which to one user means that he dines there, but to another user, this place represents his workplace. An approach proposed by Lv et al. [37] incorporates the classification of extracted physical places into pre-defined types. To classify places into categories, they use temporal features on visit and place levels. Visit level features are extracted from each visit point, including the day of the week, time of the day, duration and response rate. Place-level features include statistical values calculated for a physical place, such as mean, min, max, variance, frequency and others. Liao et al. [31] introduced a probabilistic temporal model to extract high-level activities from sequences of GPS points using conditional random fields. Authors in their work [76] proposed a classifier based on K-Nearest Neighbor(KNN) classifier and a decision tree based on the C4.5 algorithm. Other ways can be used, including hidden Markov models (HMMs), Bayesian networks, other regression or classification models or combinations with POI databases and geo-coding features.

Chapter 6

Design

Based on the analysis of existing approaches and methods for the analysis and modelling of GPS data, we have designed an application for the retrieval of important user locations and their semantic labelling. As described in the previous chapters, similar systems consist of several layers since this task is not trivial and requires several steps. In this chapter we describe the single parts of the system and the importance of their existence in the proposed platform.

The main task of the whole system is to extract semantically significant locations of the user, including the user's home and work locations, as well as other key locations that can provide insights into the user's behavior, such as restaurants, bars, cafes, and supermarkets. The system is designed to specifically focus on these types of locations.

6.1 Significant locations extraction platform

Firstly, we designed the application to process raw GPS data so that each data point consists of three items - latitude, longitude and timestamp, which provide enough information to analyse the user's behaviour. The system comprises several layers, each with a specific purpose, specially pre-processing layer, stay-point detection, segmentation layer, semantic enrichment and location labelling layer. An overview of the whole system and its layers is described in Figure 6.1.

The task of the data pre-processing layer is to process the raw GPS data and provide a suitable format for further analysis. This involves mainly converting coordinates and processing time data. Time information will be used in all layers, but plays a significant role in semantic enrichment layer, thus we need to process it correctly and include information about the timezone. The next layer, the stay point detection layer, identifies points in data which are more interesting than others. This means we are interested in places where the user spends more time. In Section 5.1 we provided a definition of stay point and the most common detection strategies. In our work, we will use both, time and distance rules, in order to detect stops where the user have stopped for an amount of time or wandered around the target place at a given distance. These two types of stops are shown in Figure 5.1. When the stay points are extracted into a sequence of stops each point corresponds to some location and represents the user's location history. However, this sequence is still not suitable for extraction of significant locations, due to the variability of stay points' spatial coordinates caused by the imprecision of location data. Even if two stay points correspond to the same place, they may not have identical spatial coordinates due to location noise. For example,

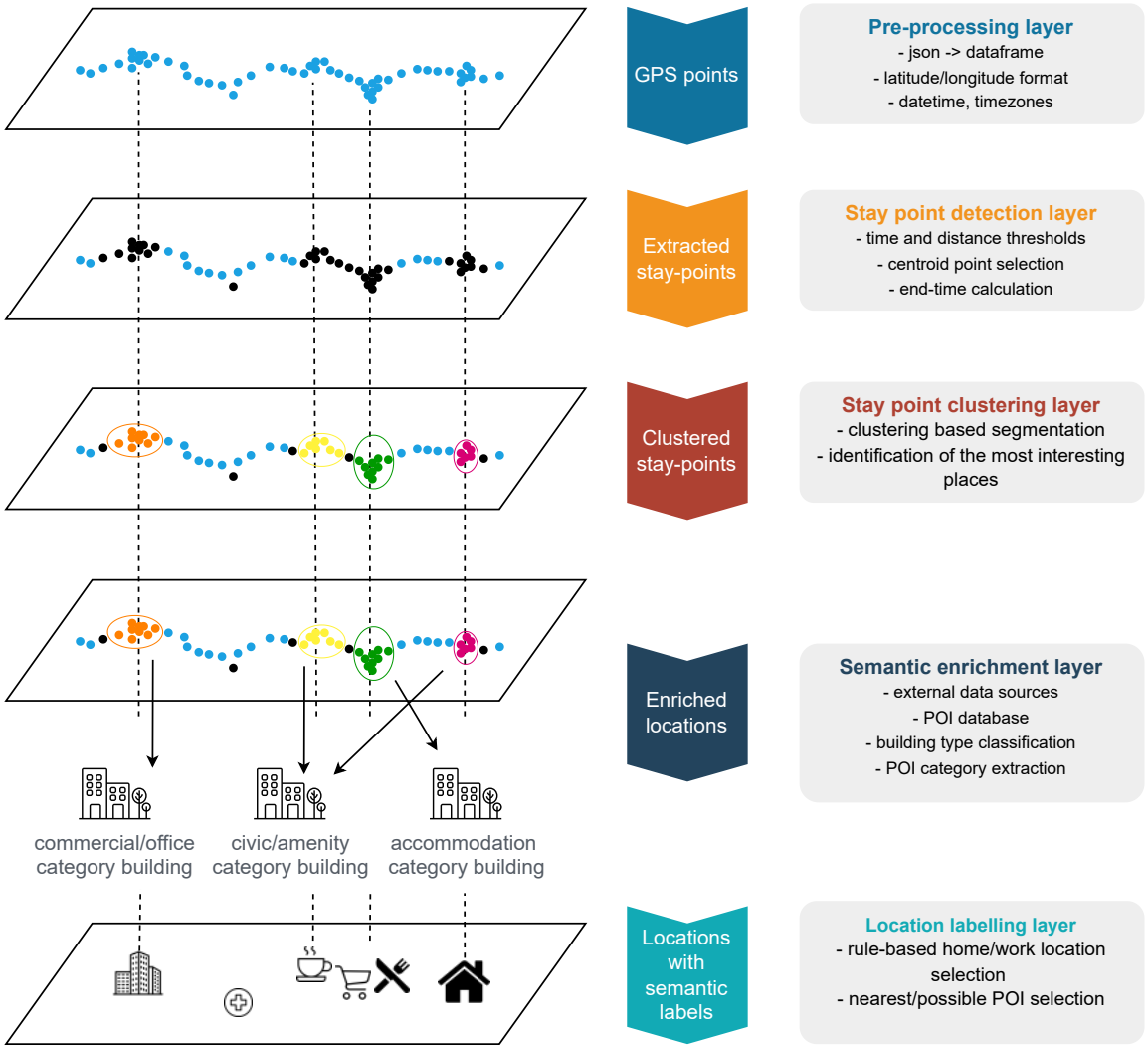


Figure 6.1: Platform layers overview.

stay points for the work location on different days may be very close to each other but not identical. To address this issue, we utilize clustering similar stay points into groups of stay points. We use a density-based clustering algorithm DBSCAN which allows us to set two parameters, a minimum number of points in the cluster and a parameter for the distance between two points for them to be considered as part of the same cluster. The biggest advantages of density-based approaches are that we are able to create clusters with irregular structure, the setting of minimum points in a cluster allows us to find significant clusters that are dense enough, while disregarding clusters that are too sparse and setting optimal values for the distance parameter allow us to identify location with valuable meaning to users, such as restaurant, shops and other places. Output from this layer is used to retrieve the semantic meaning of places.

As described in Section 5.4.2, a place can be interesting from the user's perspective as well as publicly interesting. In the semantic enrichment layer, we use third-party sources of data to gather information about the clusters of places and determine whether they appear to be in areas where people typically live, work, or visit for other purposes. By analyzing this

information, we can identify which clusters correspond to home and work locations. In this study, we utilized data from the OpenStreetMap (OSM) database to extract information about the building types within a given cluster by accessing a publicly available API. By analyzing the prevalence of different building types, specifically civic/amenity, commercial, and accommodation, we are able to classify clusters as either industrial or residential zones with a high degree of confidence. In instances where clusters have a low degree of confidence in representing the location of home or work, they are instead considered to be locations where the user spends leisure time or visits for other purposes, e.g. shopping or dining. In such cases, we gather information about the surrounding points of interest. We use the categories listed in Table 5.1.

The last part of the framework is based on defined rules - assumptions about usual times and time duration in work/home locations, maximum distances and the possibility of visiting a given POI based on its opening hours.

6.2 Detailed design description

In Section 6.1 we aimed to provide a high-level introduction to the proposed method. In this section, we provide a detailed description of the steps of data processing of the proposed method and description on used algorithms and methods.

6.2.1 Stay-point detection layer

As described earlier in this work, there are several techniques to extract stay points from the sequence of GPS points. In our work, we utilized a differential-based stay-point detection algorithm described in Algorithm 1. Stay points can be extracted from the user's GPS points using the algorithm by seeking the spatial region within a given radius where the user spent a given amount of time. The algorithm was inspired by an algorithm proposed by Li et al. [30]. According to the evaluation of similar stay-point detection algorithms performed in work by Lee et al. [28] this algorithm exposes low computing requirements and achieves competitive accuracy when compared against other algorithms.

The input to the proposed algorithm is an array of GPS points, along with distance and time thresholds δ_d and δ_t, respectively. The algorithm then outputs a set of stay points with the same attributes as the input GPS points. Formally, there exists a stay point from a location point p to location point q if the two following constraints are satisfied:

$$\begin{aligned} distance(p, q) &< \delta_d \\ timeDifference(p, q) &> \delta_t \end{aligned} \tag{6.1}$$

Algorithm 1 Stay-point detection algorithm

Input: an array of GPS points *data*, distance threshold δ_d, time threshold δ_t
Output: a set of stay points Π

$\Pi \leftarrow \emptyset$
$N \leftarrow |data|$
$p_0 \leftarrow data[0]$
$points \leftarrow p_0$
for $i \leftarrow 0$ to N **do**
    $p_1 \leftarrow data[i+1]$
    if $distance((p_0.\text{lat}, p_0.\text{lon}), (p_1.\text{lat}, p_1.\text{lat})) > \delta_d$ **then**
        if $abs(p_0.\text{ts}, p_1.\text{ts}) > \delta_t$ **then**
            if $len(points) > 0$ **then**
                $\pi.\text{lat} \leftarrow \frac{\sum_{n=0}^{|points|} p_n.\text{lat}}{|\text{points}|}$
                $\pi.\text{lon} \leftarrow \frac{\sum_{n=0}^{|points|} p_n.\text{lon}}{|\text{points}|}$
                $\pi.\text{start\_ts} \leftarrow p_0.\text{ts}$
                $\pi.\text{end\_ts} \leftarrow p_1.\text{ts}$
                $\Pi \leftarrow \Pi \cup \pi$
            end if
            $p_0 \leftarrow p_1$
            $points \leftarrow \emptyset$
        else
            $p_0 \leftarrow p_1$
            $points \leftarrow \emptyset$
        end if
    end if
    $points \leftarrow points \cup p_1$
end for
return Π

First, the algorithm keeps two pointers p_0 and p_1, meaning the start and end points of a subset of GPS points trajectory. The algorithm iterates over input data and accumulates more points into a trajectory based on time and distance conditions. Second, once the accumulated points in the trajectory exceed the specified time and distance thresholds, the algorithm identifies the centroid of the points in the trajectory and adds it to the set of stay-points. This process is repeated for each subset of points until all the input GPS points have been processed. The resulting set of stay-points represents the locations where the user has spent a significant amount of time within a given radius.

Time and distance thresholds affect the number of extracted stay-points. Since it is not possible to determine in advance the desired number of resulting stay points, in our experiments, which will be described later, we tried different combinations of thresholds.

The distance threshold determines the minimum distance between two consecutive points for them to be considered part of the same stay-point. A smaller distance threshold will result in more granular stay-points, while a larger threshold will result in fewer but larger stay-points.

The time threshold determines the minimum duration that a user must stay within the distance threshold in order for the location to be considered a stay-point. A shorter time

threshold will result in more numerous but shorter stay-points, while a longer threshold will result in fewer but longer stay-points.

6.2.2 Stay-point clustering layer

To address the issue of variability of stay points' spatial coordinates we utilize a stay-point clustering layer which performs clustering of stay-points extracted using the previous layer. There are several clustering algorithms, among the most well-known K-means, agglomerative clustering, Gaussian Mixture Model (GMM), DBSCAN, and HDBSCAN clustering algorithms. For example, the ST-DBSCAN and CB-SMoT algorithms are specifically designed for clustering spatiotemporal data, such as GPS data, and may be useful for identifying clusters of stay points that occur at specific times of day or days of the week.

The choice of clustering algorithm can have a significant impact on the quality of the clustering results. In our work, we implemented K-means, DBSCAN, and HDBSCAN clustering algorithms into the processing pipeline and performed an evaluation of clustering results. Each of the algorithms has its advantages and disadvantages, K-means is a popular algorithm that can be computationally efficient, but it may not perform well on data with complex cluster structures or varying cluster sizes, DBSCAN clustering is a density-based algorithm that can identify clusters of varying shapes and sizes but may be sensitive to the choice of parameters and HDBSCAN clustering (an extension of DBSCAN) that can automatically determine the number of clusters and their shapes but may be computationally expensive.

We decided to use density-based clustering due to its ability to manage spatial characteristics. Another benefit of this approach is that we can overcome many of the limitations of K-means, such as the detection of noisy points in data or the discovery of clusters with different shapes and sizes. Although the settings of parameters for density-based algorithms may be challenging, they are more robust and can achieve better results compared to setting the parameter, a number of clusters, for K-means. Additionally, HDBSCAN, in particular, can automatically determine the number of clusters and their shapes, which can save time and resources in the clustering process.

6.2.3 Semantic enrichment and location labelling layer

Semantic enrichment layer process clustered stay points. The main task of this layer is to infer semantic labels to all places extracted by the layers above. In our work, we are considering three groups of locations, thus home location, work location and POI location. To gain more information about places we utilize third-party sources of data to gather additional information. Enrichment of POI locations is provided by the next layer, and thus in this layer, we rather focus on the extraction of home and work locations. We designed an algorithm to identify the types of buildings that are present within a cluster of locations, based on their geographic coordinates and determine the prevalence of different types of buildings within a cluster based on the proportion of buildings that belong to different categories. The algorithm is described in Algorithm 2.

Algorithm 2 Building categories enrichment

Input: a list of clustered locations C

Output: a list of clustered locations enriched with building categories

$categories \leftarrow$ ['accommodation', 'commercial', 'civic/amenity']
for $c \in C$ **do**
    $building\_categories \leftarrow \emptyset$
    $bounding\_box \leftarrow ((\text{min\_lat}(c)), (\text{min\_lon}(c)), (\text{max\_lat}(c)), (\text{max\_lon}(c)))$
    $buildings \leftarrow query(bounding\_box)$
    for $building \in buildings$ **do**
        $building\_categories \leftarrow building\_categories \cup \text{category}(building)$
    end for
    for $category \in categories$ **do**
        $c.category \leftarrow \frac{building\_categories.\text{count}(category)}{|building\_categories|}$
    end for
end for
return C

First, the algorithm takes clustered locations as input, each cluster corresponds to a set of geographic coordinates and outputs the same list of clustered locations enriched with building categories prevalence. We divided building types into three categories: accommodation, civic/amenity and commercial. Second, for each cluster, we build a bounding box of coordinates within the cluster to cover the whole area of the cluster. Next, we use an API query to retrieve a list of buildings that are located within this bounding box. For each building in the list, the algorithm determines its category (from building tags) and adds it to a set of building categories. Finally, the algorithm calculates the proportion of buildings in each category within the set and assigns proportions for each category defined above. This algorithm is useful for identifying whether a cluster corresponds to a residential or commercial area based on the prevalence of different types of buildings.

Once the proportions of each category for each cluster are calculated we can infer home and work location. Our approach was inspired by Lv et al. [37]. We take advantage of temporal features extracted from each visit point, including the day of the week, time of the day, and visit duration. Inferring these locations is then based on three conditions:

- Categories proportion: This involves the proportion of each category for a given cluster calculated using the Algorithm 2 above. For example, the proportion of the „accommodation“ category in a cluster must be higher than the proportion of the „commercial“ category to be considered as a potential home location.
- Arrival and departure time: Typically, users arrive at work locations in the morning and leave in the evening, while they arrive at home locations in the evening and leave in the morning. For example, if a user arrives at a location early in the morning and departs in the evening, it is more likely to be a work location.
- Length of stay: The length of stay at a cluster is used to filter out locations that are not suitable for home or work locations. For instance, a user is unlikely to have a work location that they stay at for less than an hour.

By analyzing these three conditions, we can determine which clusters correspond to the user's home and work locations and filter out other places. Once the home and work

clusters are found, we can focus on other locations which are more likely to be mapped to some kind of point of interest. To assign a POI to a place, we adopted the approach proposed by Furletti et al. [13]. We utilised an Algorithm 3 to find the nearest available POI and other candidate POIs. This algorithm takes two conditions into account when assigning a POI to a place:

1. Maximum distance: The first condition is that only POIs within a given maximum distance from a stop point are considered. This means that the algorithm will only look for POIs that are located within a certain radius of the stop point.

2. Accessibility in visit time: The second condition is that the algorithm checks the accessibility of a POI during the visit time. This means that the algorithm will only consider POIs that are open or available for a visit during the time that the user is expected to be at the stop point. We consider POIs without defined accessibility as accessible.

By taking these two conditions into account, the algorithm is able to narrow down the list of candidate POIs to those that are both nearby and likely to be accessible during the visit time.

Algorithm 3 POI selection algorithm

Input: a stop point s, with latitude, longitude and timestamp, distance threshold δ_d
Output: nearest POI, list of other possible POIs

$poi\_list \leftarrow \text{getPOIs}(s, \delta_d)$
$selected\_poi \leftarrow \emptyset$
$available\_pois \leftarrow \emptyset$
for $poi \in \text{sortByDistance}(s, poi\_list)$ **do**
    if $(s.\text{ts} \in poi.\text{opening\_hours}) \mid (poi.\text{opening\_hours} = \emptyset)$ **then**
        $available\_pois \leftarrow available\_pois \cup poi$
    end if
end for
if $available\_pois \neq \emptyset$ **then**
    $selected\_poi \leftarrow available\_pois[0]$
    $available\_pois \leftarrow available\_pois - selected\_poi$
end if
return $selected\_poi, available\_pois$

When obtaining POI information about a place we need to use third-party data. We decided to use OpenStreetMap (OSM) database and query publicly accessible read-only API Overpass. We also use only a subset from the query result, as we are interested in places like restaurants, bars or shops listed in Table 5.1 divided into categories.

Chapter 7

Experimental results and evaluation

Individual parts of the system were described in Chapter 6. This chapter describes the experiments conducted on the proposed system for mining user's significant locations. The experiments were designed to test the system's ability to extract locations and correctly label them with semantic labels, home and work, and the ability to find other interesting locations for user.

This chapter is structured as follows. First, we provide a description of the input dataset, including the data collection process and the characteristics of the data. Next, we describe the experiments in detail for each part of the system.

Overall, the experiments presented in this chapter provide a comprehensive evaluation of the proposed system and demonstrate its effectiveness in mining user significant locations. By analyzing the results in detail, we provide insights into the strengths and limitations of the system, as well as opportunities for future research.

7.1 Dataset description

In this section, we provide a detailed description of the dataset used in experiments. The dataset was created using data obtained from Google, which provides the location history of its user. The dataset includes the location history of one user over a period of four months. We performed experiments on the entire input dataset as well as its subsets. The dataset covers primarily the area of the city of Brno, located in the Czech Republic, but also includes some more distant regions, such as the northern part of Slovakia.

The location history provided by Google includes latitude, longitude, timestamp information and more attributes, which allows us to track the movements of users over time. We designed the platform to be able to extract user's significant locations from raw GPS input without additional information. Input data are stored in a structured format, with each record consisting of the following information:

- `latitudeE7`: The latitude of the location visited by the user in E7 format, which is a format that represents the value as an integer by multiplying it by 10^7.
- `longitudeE7`: The longitude of the location visited by the user in E7 format, which is a format that represents the value as an integer by multiplying it by 10^7.
- `accuracy`: The estimated accuracy of the latitude and longitude values in meters.

- `altitude`: The estimated altitude of the location in meters.
- `verticalAccuracy`: The estimated accuracy of the altitude value in meters.
- `deviceTag`: A unique identifier assigned to the device used to collect the data.
- `platformType`: The type of platform used to collect the data.
- `serverTimestamp`: The timestamp of when the data was received by Google's server in ISO 8601 format.
- `deviceTimestamp`: The timestamp of when the data was collected by the device in ISO 8601 format.
- `batteryCharging`: Indicates whether the device was charging at the time of the data collection.
- `formFactor`: The type of device used to collect the data.
- `timestamp`: The timestamp of the visit, indicating the time at which the user arrived at the location in ISO 8601 format.

An example of the input data structure is given in Listing 7.1.

```
{
    "latitudeE7": 492318058,
    "longitudeE7": 165700256,
    "accuracy": 35,
    "altitude": 286,
    "verticalAccuracy": 19,
    "deviceTag": -1210809530,
    "platformType": "IOS",
    "serverTimestamp": "2022-11-27T17:03:45.104Z",
    "deviceTimestamp": "2022-11-27T17:03:44.861Z",
    "batteryCharging": false,
    "formFactor": "PHONE",
    "timestamp": "2022-11-27T16:41:26.521Z"
  }
```

Listing 7.1: Example input data record.

7.2 Input data preprocessing

As described in Chapter 6, we designed the system to only require latitude, longitude, and timestamp data. This approach is advantageous for several reasons. First, it reduces the amount of data that needs to be processed and analyzed, which in turn reduces the computational resources needed to run the system. Second, the use of only latitude, longitude, and timestamp data makes the system more widely applicable and as input dataset can be used any set of data containing these information. Many location data sources, such as GPS-enabled devices and mobile phone apps, already collect this information by default. This means that the system can be easily applied to a wide range of data sources, without

requiring additional data collection efforts. Additionally, it reduces the storage requirements for the collected data. Additional attributes, such as device information or battery status, can add significant amounts of data to each record, which do not bring significant improvement in mining user's significant locations.

In order to create desired input in form of sequence of latitude, longitude points with timestamp, we converted latitude and longitude into decimal degrees format. Converting the latitude and longitude from the E7 format is necessary in order to make the location data more easily usable in standard mapping and geospatial analysis tools. Next, we added time zone offset to ISO format timestamp. Adding the time zone offset to the ISO format timestamp is important in order to accurately represent the local time at each location visited by the user.

Example of processed input data record, containing timestamp, latitude and longitude is given in Listing 7.2.

```
2022-11-27 17:41:26.521000+01:00,49.2318058,16.5700256
```

Listing 7.2: Example of processed input data record.

7.3 Stay-points extraction

To extract stay-points from input data we used the algorithm described in Subsection 6.2.1 and a subset of the input dataset with one-month duration. The selection of parameters δ_d (distance threshold) and δ_t (time threshold) affects the number of stay points found.

Figure 7.1 shows a matrix where the x-axis represents the distance threshold parameter values and the y-axis represents the time threshold parameter values. Each cell in the matrix represents the number of stay points detected for the corresponding combination of distance and time thresholds. The colour of each cell indicates the number of stay points, with darker colours indicating a higher number of stay points. We can also see how changes in the distance and time thresholds affect the number of stay points.

By analyzing the matrix described above, we can select the optimal combination of parameters. We used the knee method to compute the optimal number of stay points for setting parameter values and later analysis. The Knee Point Detection Method (also known as the „knee“ method) is a technique for identifying a knee point or an elbow point in a curve, where the curve starts to deviate significantly from a straight line. The method was first introduced in the paper „Finding a “kneedle„ in a haystack: Detecting knee points in system behavior“ by Satopaa et al. [54]. The basic idea of the knee method is to find the point on the curve where adding an additional data point to the set would result in a relatively small decrease in the curvature of the curve. To identify the knee point, we use the KneeLocator algorithm, which takes the sorted distances and uses a „knee finding“ algorithm to identify the point where the curvature of the curve changes the most.

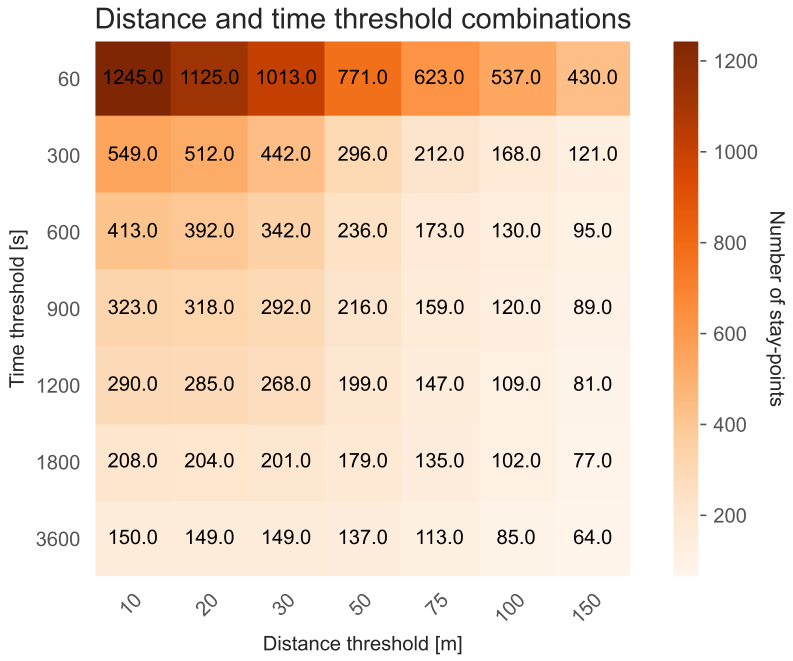


Figure 7.1: Number of stay points detected using combinations of thresholds parameters.

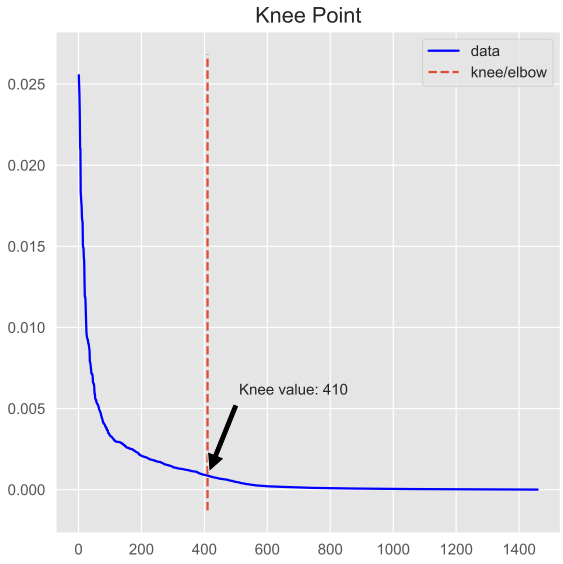


Figure 7.2: Visualized result of knee method to estimate optimal number of stay points.

The knee method recommended a number of stay points of 410, as shown in Figure 7.2, which corresponds to 10 meters as the distance threshold and 10 minutes as the time threshold according to distance and time thresholds matrix.

However, this parameter setting was not optimal for further analysis, so we set the time threshold to 15 minutes and the distance threshold to 50 meters, resulting in 216 stay points detected. We decided to change the recommended values because the number of stay points was too high, which would result in computation time increase in later steps of processing.

Another reason is that by choosing higher value of distance threshold we can reduce noise in the data and simplify the data and make it easier to analyze. While this number is lower than the knee method's recommendation, it is still a sufficient amount for our analysis purposes and also reduces the overall computation time. Our future experiments proved that the smaller number of stay points does not lower down the accuracy of place detection.

In other words, if the user stays over 15 minutes within a distance of 50 meters, a stay-point is detected. By fine-tuning the parameters, we were able to distinguish significant locations, such as restaurants, coffee shops, or shopping malls, from insignificant places, such as traffic lights or stop signs. This enabled us to focus our analysis on places that are of greater importance to the user.

The output of the stay point detection algorithm with selected parameters is visualized in Figure 7.3. The x-axis represents longitudes and the y-axis latitude points. In addition, we can see two marginal plots on the top and right sides of the plot. These marginal plots show the distribution of the data along each axis separately, using a kernel density estimate (KDE) plot.

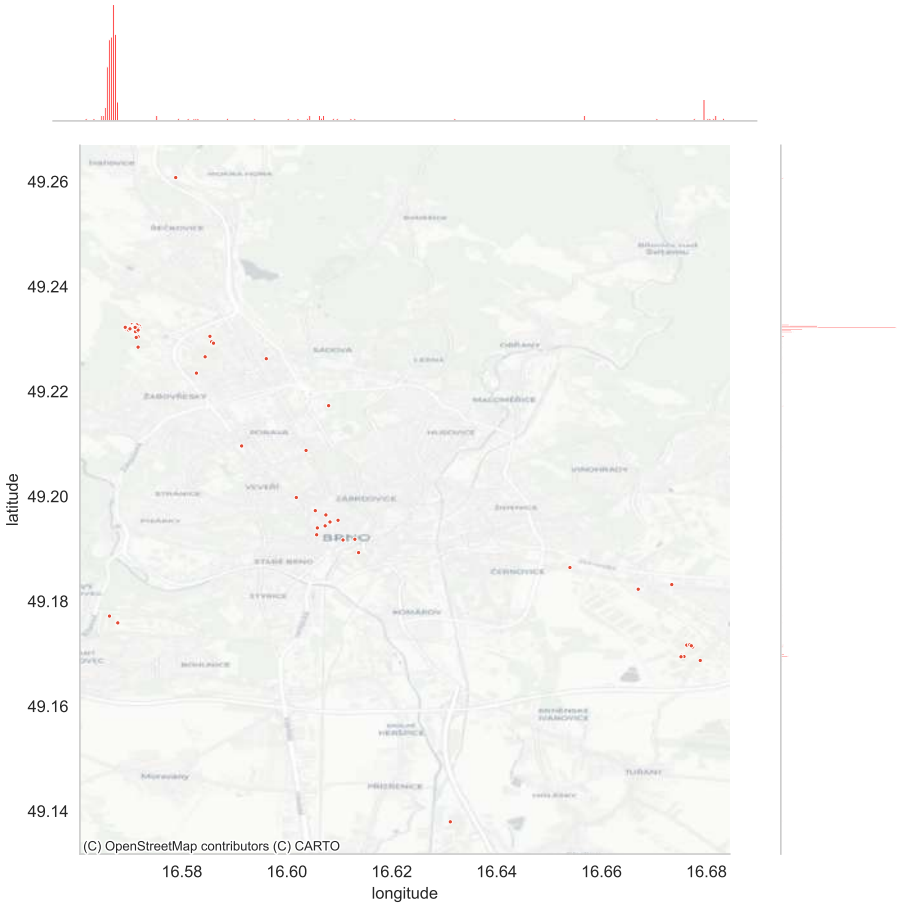
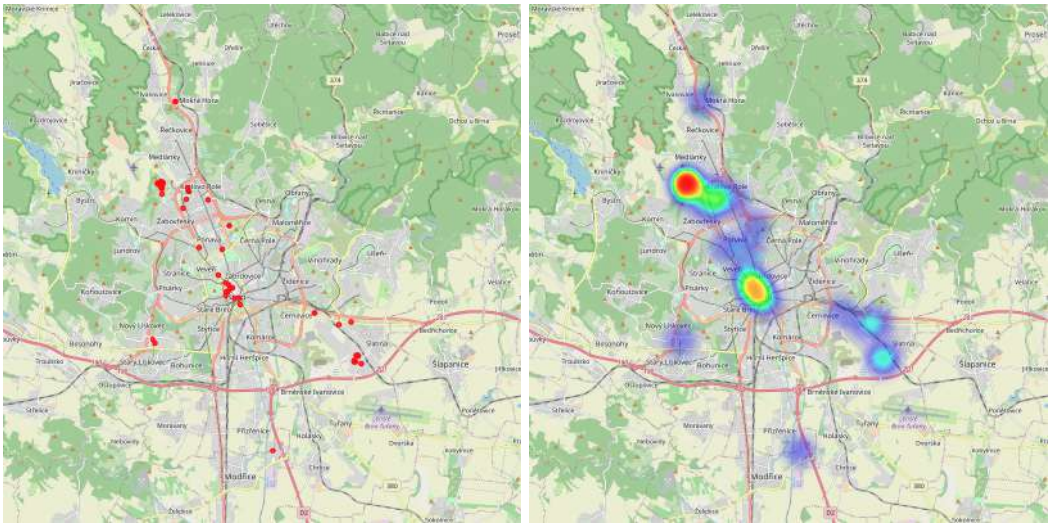


Figure 7.3: Spatial distribution of points extracted using stay-point detection algorithm.

Figure 7.4 shows extracted stay points and their visualization using a heat map. Sub-figure 7.4b allows us to study the density of GPS points, and identify areas where is a high concentration of GPS points, which may indicate popular user's locations.



(a) Extracted stay points (b) Heat map

Figure 7.4: Visualization of extracted stay points.

7.4 Stay-point clustering

The experiments on the stay-point clustering layer were conducted with the primary objective of selecting the most appropriate clustering method and its optimal parameters for our task of mining user's significant locations. As mentioned earlier, there are several clustering algorithms available, each with its own advantages and disadvantages. Based on this review of commonly used approaches and considerations when designing the application, we selected K-means, DBSCAN, and HDBSCAN as the clustering algorithms to be evaluated in our experiments.

Next, we implemented these algorithms in our processing pipeline and performed clustering on our stay-point dataset. We evaluated the clustering results using various performance metrics, including silhouette score and others, as well as visual introspection of the results.

7.4.1 Centroid-based clustering

K-means clustering requires setting a parameter for the optimal number of clusters into which the data may be clustered. To estimate this value, we used the elbow method, which is a commonly used technique for estimating the optimal number of clusters in k-means clustering. The elbow method (also known as the knee method) involves plotting the values of distortion or inertia, which is the sum of squared distances between each data point and its assigned cluster centroid, as a function of the number of clusters k, and identifying the point at which the rate of decrease in distortion or inertia slows down significantly. To implement the elbow method, we used k-means clustering with a range of values of k, calculated the distortion and inertia for each value of k, and plotted these values against a given range.

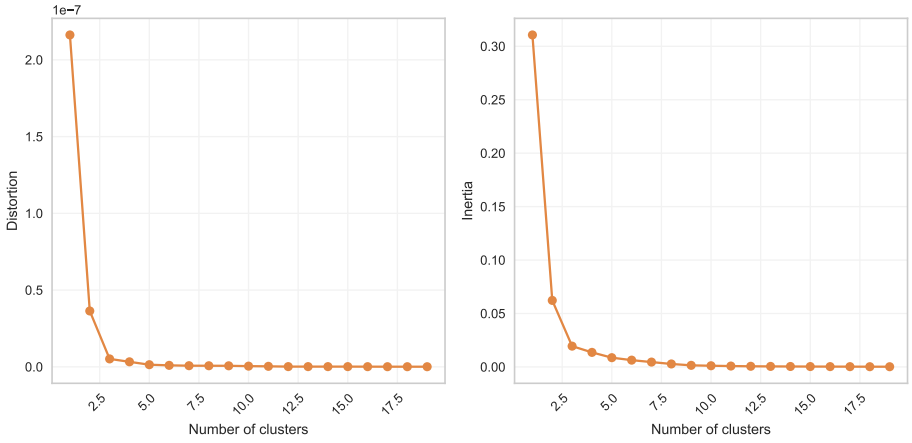


Figure 7.5: Visualization of elbow method on values of distortion and inertia.

Figure 7.5 shows the distortion and inertia values against a number of clusters graphs, where we can see that the curve starts to level off at around K=2 or K=3. This suggests that the optimal number of clusters for our dataset is likely to be between 2 and 3. Distortions, the average of the squared distances from the cluster centres of the respective clusters, were calculated using different distance functions - euclidean, cosine and city-block (manhattan). The figure above shows distortions calculated using the cosine distance function since the results obtained using other functions were comparable. However, since the elbow method is considered to be a heuristic approach, we utilized the silhouette method to validate our results.

Silhouette method is used to determine the optimal number of clusters in a clustering algorithms. It measures the quality of the clustering output by evaluating how well each data point fits into its assigned cluster, and how different it is from other clusters. Silhouette analysis produces a silhouette coefficient value for each data point. The silhouette coefficient is a measure of how well each data point fits into its assigned cluster. It takes into account both the cohesion, how close the data point is to other points in its cluster, and the separation, how far the data point is from points in other clusters. The coefficient ranges from -1 to 1, where a score of 1 indicates a data point is well-matched to its cluster and poorly matched to other clusters, a score of -1 indicates the opposite, and a score of 0 indicates that the data point is equally close to two different clusters [62]. The silhouette value $s(i)$ of single data point i in the cluster $C_i (i \in C_i)$ can be calculated using Equation 7.1.

$$
\begin{aligned}
s(i) &= \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \\
a(i) &= \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j) \\
b(i) &= \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j),
\end{aligned}
\tag{7.1}
$$

where $a(i)$ is the average distance between data point i and all data points in the same cluster C_i, $b(i)$ is the minimum average distance between data point i to all data points in other clusters C_k, where $C_k \neq C_i$. Finally, silhouette coefficient (SC) is computed using

Equation 7.2 for the optimal value of the average $s(i)$ over all data points in the dataset with a specific number of cluster k.

$$SC = \max_{k} s(k) \tag{7.2}$$

Visualization of the silhouette method on our dataset is shown in Figure 7.6 with silhouette values for cluster sizes in range <2, 22>.

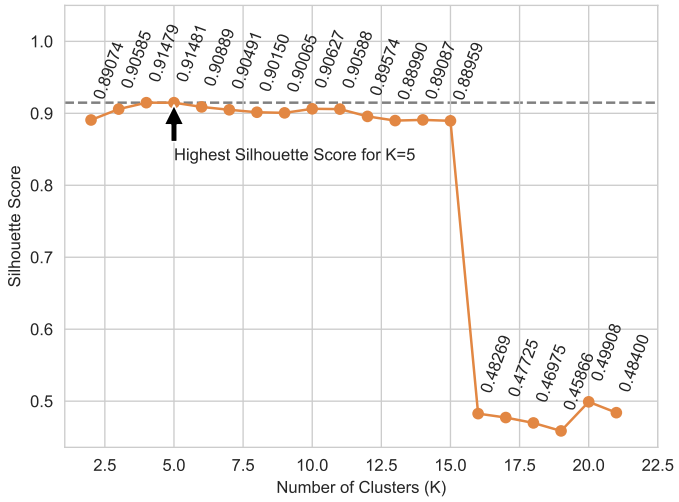


Figure 7.6: Silhouette method.

The silhouette analysis revealed that the optimal number of clusters for our dataset is 5. Although the elbow method suggested a lower number of clusters, the silhouette method is a more robust approach that considers the quality of the clusters as well as their quantity.

(a) K-means clustering result with K=3

(b) K-means clustering result with K=5

Figure 7.7: K-means clustering result with a different number of clusters.

Figure 7.7 shows the clustering of our dataset using the k-means algorithm with two different values of K, namely 3 and 5. The figure shows how the data points are assigned to different clusters based on their proximity to the cluster centres. From both subfigures, we can see that the clusters are quite well separated but points on larger distances are grouped together, thus this approach is not suitable for our task of mining significant locations.

7.4.2 Density-based clustering

Density-based clustering algorithms, like DBSCAN and HDBSCAN require setting more parameters compared to centroid-based (K-means) clustering algorithms. Both clustering algorithms named require distance metrics, we can use various metrics - euclidean, hamming, manhattan or haversine distance metrics. In our experiments, we used the haversine distance metric based on the great-circle distance between two points on a sphere given their longitudes and latitudes. It is commonly used to calculate distances between points on the surface of a sphere, such as the distance between two points on the Earth's surface. This distance can be computed using Equation 7.3.

$$\begin{aligned} a &= sin^2(\Delta lat/2) + cos(lat1).cos(lat2).sin^2(\Delta long/2) \\ c &= 2.atan2(sqrt(a), sqrt(1-a)) \\ d &= R.c \end{aligned} \tag{7.3}$$

where $lat1$ and $long1$ are the latitude and longitude of the first point, $lat2$ and $long2$ of the second point, Δlat is the difference in latitude, $\Delta long$ is the difference in longitude, R is the radius of the sphere (in our case the Earth), and a, c, and d are intermediate variables used in the calculation.

DBSCAN clustering algorithm requires setting ε parameter, the maximum distance between two samples for one to be considered as in the neighbourhood of the other and $min\_samples$ parameter, the number of samples in a neighbourhood for a point to be considered as a core point. In our first experiments, we used elbow method to estimate value of ε using k-nearest neighbours. We set parameter k in k-nearest neighbours to 5, since it is recommended to use 4 neighbours for two-dimensional data and we also want to include the point itself. Similarly to finding an optimal number of clusters in k-means clustering, we are looking for an elbow in the plot, which is the point of maximum curvature. This point corresponds to the optimal value of ε.

Figure 7.8 shows the sorted k-NN (k-nearest neighbours) distances for each point in the dataset and knee point in the graph, which can be used to estimate the appropriate value for the ε parameter.

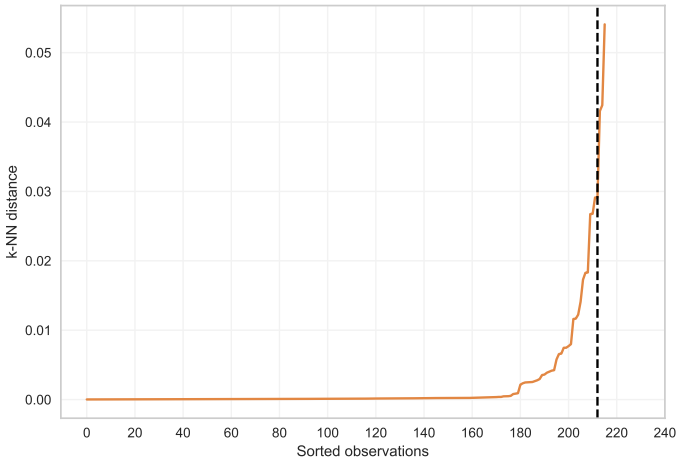


Figure 7.8: Elbow method for ε parameter selection in DBSCAN.

The value of the *min_samples* parameter is recommended to be set to 4, as we work with two-dimensional data, but generally, the larger the data set, the larger the value of *min_samples* should be.

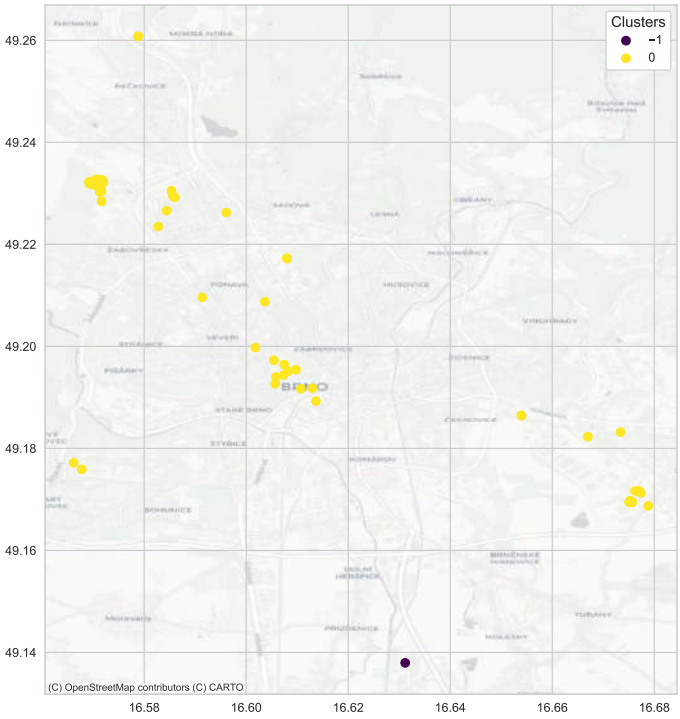


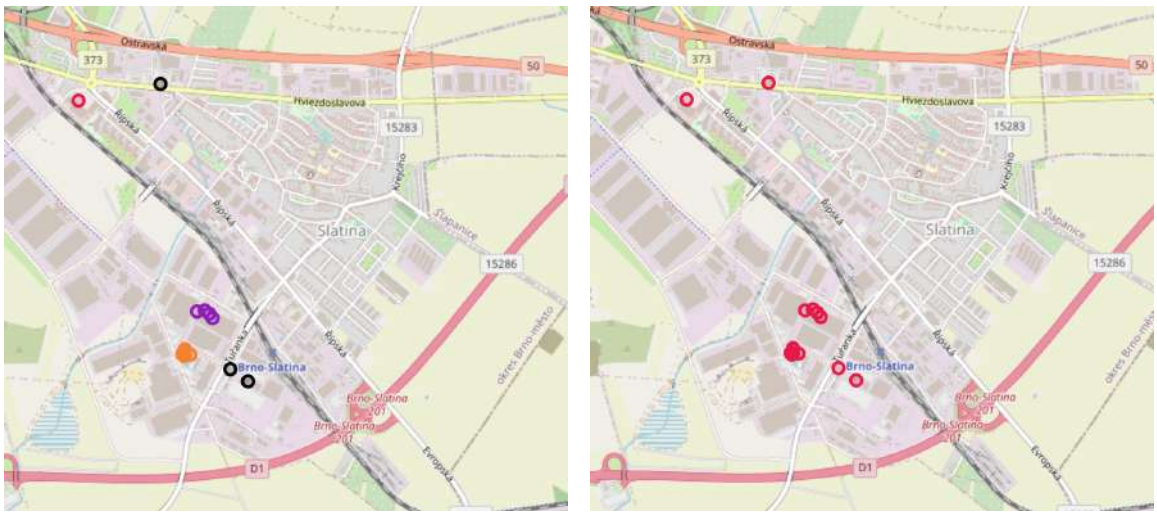
Figure 7.9: DBSCAN clustering result.

Putting all together, Figure 7.9 show the result of applying DBSCAN clustering on stay points obtained from the previous stage. As we can see there are only two different colours, the algorithm found only one cluster with label 0 and one outlier with label -1, so this result does not provide sufficient information for further analysis as it does not identify any meaningful clusters.

In our initial experiments, DBSCAN achieved the best results for clustering the data. However, when we scaled up to larger datasets, this approach proved to be inappropriate. This could be due to the fact that DBSCAN is sensitive to the choice of hyperparameters, such as ε and *min_samples*, and these hyperparameters may not generalize well to larger datasets.

HDBSCAN clustering algorithm is similar to DBSCAN clustering and requires setting more parameters to perform clustering. The advantage of this algorithm is that we do not need to set all parameters but the algorithm can automatically determine them.

According to the documentation provided by `HDBSCAN` Python implementation documentation [40], it is not recommended to set value for parameter *alpha*, a threshold for the minimum cluster stability, so we will let it set by default. The setting of the *min_cluster_size* parameter, the smallest size grouping that we consider a cluster, is important and affects the result of clustering. Next, we need to set ε similarly to DBSCAN clustering. Again, we can use the method of finding a knee in a k-distance plot but we adopt another approach to setting this parameter. When using this parameter in HDBSCAN, we can ensure that clusters below the given threshold are not split up any further, so we compute it using the distance threshold. We experimented with settings of this parameter, and in conclusion, we ended with 10 meters as the best option to be used.



(a) *min_cluster_size*=3 (b) *min_cluster_size*=4

Figure 7.10: HDBSCAN clustering result with different size of cluster.

Result of HDBSCAN clustering is shown in Figure 7.10, as well as the difference when using different cluster size parameter. Figure is zoomed to place, where the difference is obvious. Setting the *min_cluster_size* to 3 allows us to find more smaller clusters, compared to setting to value 4. In Subfigure 7.10b we can see that all the stay points were clustered into one location, which is incorrect because not all data points belong to the same area or building. Setting *min_cluster_size* to 3 allowed us to correctly separate points that are too far away from each other and correctly group points that belong to the same area.

7.5 Semantic enrichment

As described in Subsection 6.2.3 we divide the semantic enrichment stage into two steps. First, we enrich clusters with building categories and their proportions, next, we classify clusters as home or work locations using defined conditions, second, we enrich other locations with POI information and use the proposed algorithm to obtain the corresponding POI to a place.

7.5.1 Classifying buildings

To obtain information about building type we use OpenStreetMap (OSM) database accessed through a publicly open API Overpass. This API serves queries in various formats, e.g. OverpassXML or Overpass QL. We are working with queries written in Overpass Query Language since OQL provides a more powerful and flexible querying syntax compared to OverpassXML. An example query written using OQL is shown on Listing 7.3. The query retrieves information about buildings in a specific geographical area specified by latitude and longitude coordinates. The geographical area is calculated from GPS points for each cluster representing aggregated stay points into the stay area.

Figure 7.11: Visualization of input and output for classifying buildings.

Figure 7.11 explains the proposed method for classifying buildings. The geographical area is represented as a green rectangle calculated from red points. The result obtained from the OSM query is visualized as orange points and black-framed buildings. The query consists of three main statements separated by semicolons, which are executed in order:

- `[out:json];`: This statement sets the output fromat to JSON.
- Body inside parentheses: This statement retrieves all nodes, ways, and relations with the „building“ tag within the specified geographical area. The latitude and longitude coordinates specify the bounding box of the area.
- `(._;>;);`: Building information may be stored in a relation not directly on a node or way, so we need to expand the result set to include all nodes and ways that are part of the retrieved relations.

The result from this query contains

```
[out:json];
    (
        node["building"](49.1694278, 16.6751178, 49.1697298, 16.6757214);
        way["building"](49.1694278, 16.6751178, 49.1697298, 16.6757214);
        relation["building"](49.1694278, 16.6751178, 49.1697298, 16.6757214);
    );
(._;>;);
out;
```

Listing 7.3: Example OQL query

(a) Visualized result

```
"tags": {
    "addr:city": "Brno",
    "addr:housenumber": "100",
    "addr:street": "Tuřanka",
    "building": "commercial",
    "building:levels": "4",
    "building:ruian:type": "14",
    "name": "Honeywell (CZ27)",
    "smoking": "no",
    "source": "cuzk:ruian"
}
```

(b) Part of result in JSON format

Figure 7.12: Result from query in Listing 7.3.

Result of the query above is shown in Figure 7.12. The research conducted in this work confirms that the OSM API is effective in retrieving building types within a specified location. The experimental results demonstrate that the proposed query successfully identifies buildings in the target area. Despite the existence of alternative providers, such as the Google Places API, the OSM API was selected for its ease of use and query clarity, as explained earlier in this study. Additionally, the use of public instances seems to be sufficient enough. According to documentation [47], public instances are able to handle about 1 million requests per day, and a typical request has a run time of less than 1 second. In our case, the average run time was 0.2 seconds.

7.5.2 Mapping to point of interest

As described in the previous chapter, we designed the Algorithm 3 to map a place to a point of interest. First, we need to find POI objects at a given distance, second, use additional information about POIs with the combination of features of a visit to find the most probable POI visited. To obtain information about nearby POIs, we use similar queries as those used to retrieve information about building types in a given area. However, there is a difference in that the query does not include a bounding box that covers the area of all points of a particular cluster. Instead, the query is structured to retrieve all nodes within a defined radius of a specified point.

To achieve this, we specify the categories that we wish to focus on in the query. Specifically, we target nodes that are tagged with at least one of the following categories: amenities, shops, beauty, sports, and leisure. By doing so, we can lower down the computation time, as we are not querying for all types of tags and the amount of data to be processed.

In our experiments, we discovered that our Algorithm 3 requires the correct time zone of the user to be set, as the opening hours of POIs are obtained from the OSM database, where they are stored in the local time of the area. Additionally, we encountered an issue when a user arrives at a POI a few minutes before its opening time, e.g. appointment at a doctor's office or another arranged meeting. To mitigate the impact of this issue, we have implemented a solution where the expected time of the user's visit to the POI is rounded to the nearest half-hour.

Figure 7.13 shows the visualization of the proposed method to obtain the nearest POIs to a given place using a circular buffer (blue) at a given distance from the location (blue marker) to be mapped. The red points represent POIs obtained from an OSM query.

Figure 7.13: Visualization of the proposed method to obtain nearest POIs.

```
[out:json];
    node(around:100,
        49.20980159013491,
        16.5913388167351)->.all;
        (
            node.all["amenity"];
            node.all["shop"];
            node.all["beauty"];
            node.all["sport"];
            node.all["leisure"];
        );
out center;
```

(a) Query example

(b) Visualized query result

Figure 7.14: Query example and visualized result.

Figure 7.14 shows the example query to obtain POI information and visualized query result. The information included in the resulting JSON may vary depending on the type of POI. To limit the search distance, we set a maximum radius of 100 units since this threshold provides enough results. Opening hours are optionally stored under `opening_hours` tag in the result JSON in a human-readable format. For instance, opening hours might be specified as „Mo-Sa 07:00-21:00; Su 08:00-21:00“.

7.5.3 Assigning address information

Since our framework relies on information obtained from API queries, they may not always contain all the necessary information, especially the full address. To overcome this issue, we used a combination of geocoding and reverse geocoding provided by Nominatim geocoder.

7.5.4 Result interpretation using interactive maps

In this work, we wanted to create output in a human-readable format. Basic Python plotting libraries are not very visually pleasing, so we decided to use tools that are specifically designed for this purpose. We decided to create interactive maps using `Folium` library, which is built on top of `leaflet.js` and allows us to create interactive maps with various features such as markers, popups, and custom tiles. Our interactive map output provides users with several map layers containing:

- Different map tiles
- Home location
- Work location
- Visited places (POIs) visually distinguished using icons according to POI category
- Visited POIs clustered into areas, which may help identify parts of the city where the user visited the most places

7.6 Results

We conducted experiments on a subset of the input dataset. We choose a sample with a total duration of one month. As described in this work, input data were collected using Google Maps location history, which provides a semantic history of locations aggregated with granularity for one month. Google's semantic location history is divided into activity segments and place visits. We are interested in place visits in our work, so we parsed location history to gain only a subset of information. Next, we used this data to validate the results of our work.

7.6.1 Home and work locations

Figure 7.15 shows obtained home and work locations by the proposed framework. According to the user, both home and work locations were found correctly and annotated with correct name and address.

Figure 7.16 shows the home location and nearby places the user visited. This figure confirms that even places several tens of meters away are recognized and correctly assigned to corresponding POIs.

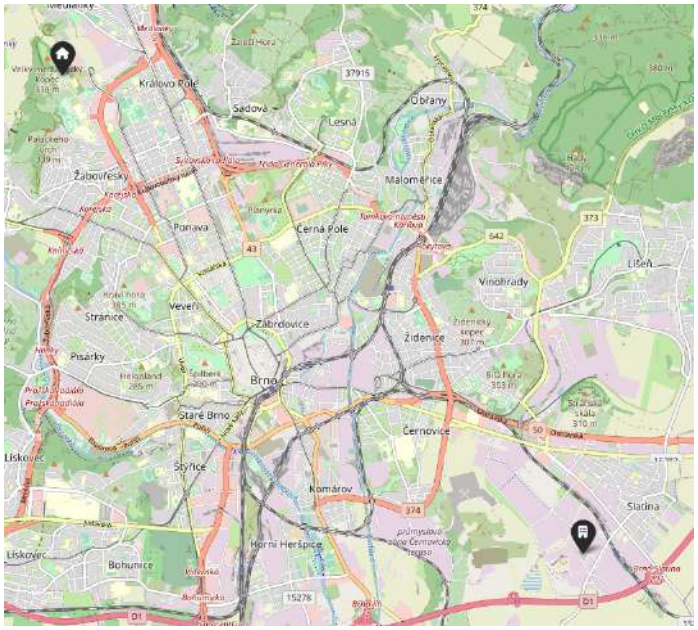


Figure 7.15: Extracted home and work locations.

Figure 7.16: Home locations and nearby visited places.

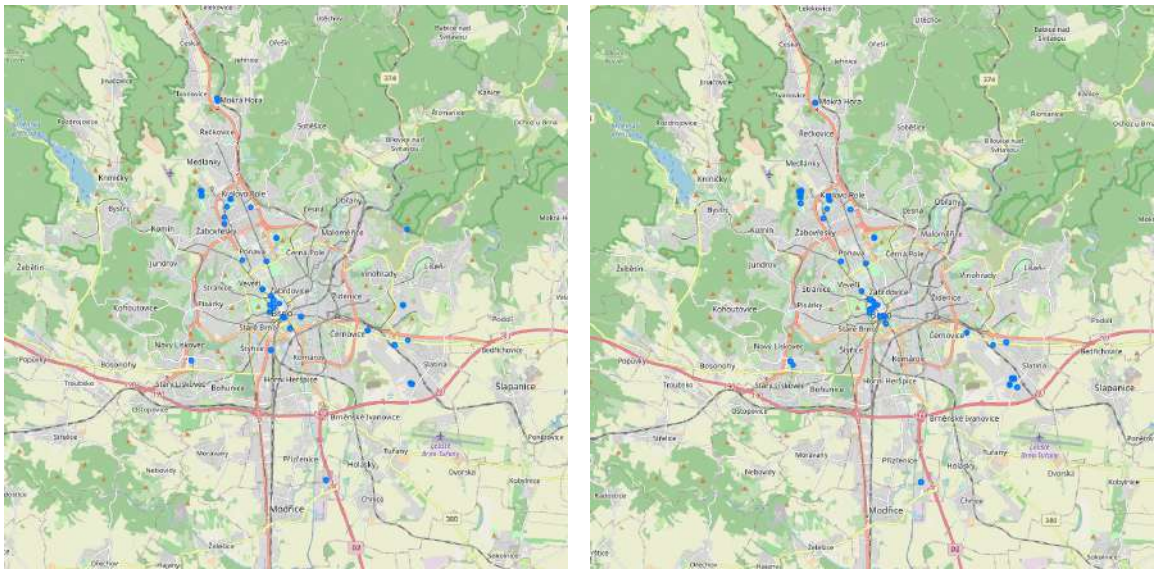
7.6.2 Place visits

Table 7.1 shows the number of places extracted, the number of missed places and the number of correctly and incorrectly mapped places to POIs resulting in 90,0763 % accuracy.

Table 7.1: Results evaluation.

	Proposed framework
Number of found places	128
Number of correct places	118
Number of incorrect places	10
Number of missed places	3
Accuracy	90,0763 %

Visual comparison of places extracted using the proposed method and places from the Google Maps platform are shown in Figure 7.17. In this Figure we can easily identify the three places, located in right upper quarter of the figure, which the proposed method failed to extract as mentioned in Table 7.1. However, there are some places which Google Maps missed and user visited them. Example of this situation is visualised in Figure 7.18. The result obtained from Google Maps platform identified two locations in one building, our method interprets this locations as one, since it represents work location and is associated with one meaningful point of interest. The place, which Google Maps missed is located near user's work location, across the road by the building, meaning restaurant „Pauzza", which user visits for lunch on workdays.



(a) Places extracted from Google Maps platform (b) Places extracted using the proposed method

Figure 7.17: Visual comparison of places extracted using the proposed method and result from Google Maps platform.

(a) Google Maps platform (b) The proposed method

Figure 7.18: Example of correct place missed by Google Maps platform.

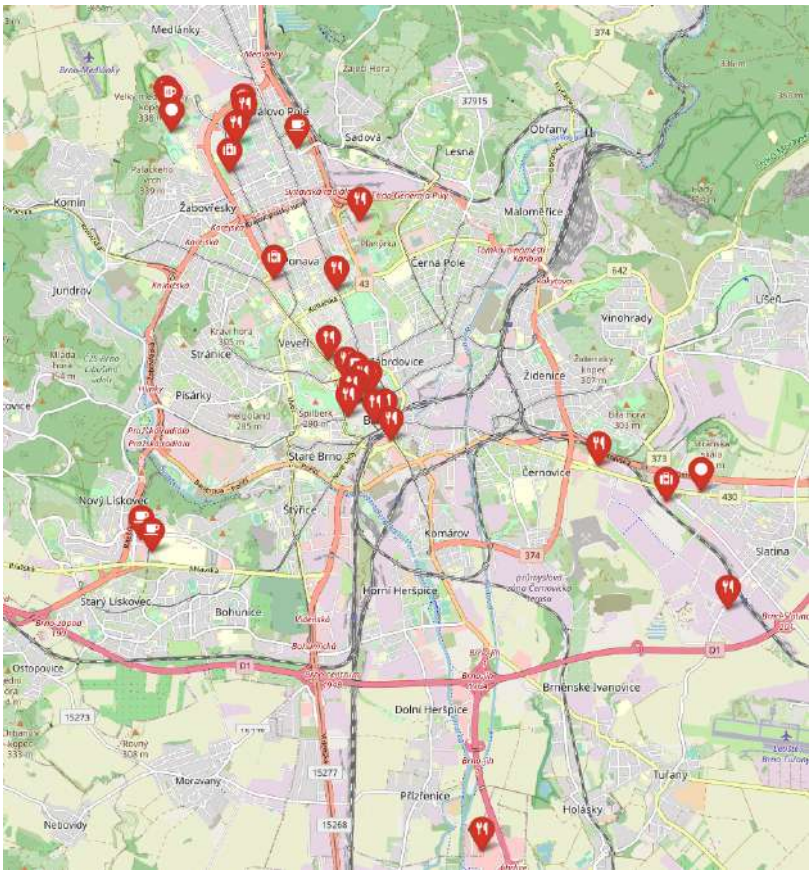


Figure 7.19: Visited places visualized with icons representing different amenity categories.

One of the map layers from created output consists of extracted places mapped to POIs, as shown in Figure 7.19. We used different display icons for better visualization according to POI type and category.

7.6.3 Performance

To evaluate the performance of our solution, we used the cProfile profiling tool for the Python programming language, in which the proposed method is programmed. Execution times of each step of data processing are shown in Table 7.2.

As described in the previous chapters, the method heavily relies on several external API calls, such as queries to the OSM database to obtain information about buildings in a given neighborhood, queries to find POIs within a given distance, and queries to the geocoding service Nominatim to apply reverse-geocoding techniques. Unfortunately, we cannot optimize these calls, but we tried to set the distance thresholds so that the retrieval of the results does not overload these services, but on the other hand, we get enough results for further processing. As we can see from Table 7.2, address information assignment takes the most time, which is dependent on Nominatim geocoder; the second most time-consuming operations are mapping to POIs and building category mapping, which are dependent on Overpass API queries. The data processing steps that rely on external resources and are beyond our control are indicated in italicized text.

Another part, which is not possible to optimize and can cause performance issues when processing larger amounts of data, is the visualization of the results using Folium library for interactive map outputs. The performance of Folium can depend on several factors, such as the size and complexity of the map, the amount of data being displayed, and the capabilities of the user's computer or device since it generates HTML, CSS, and JavaScript code to display the maps in a web browser.

The parts of the code that work with larger amounts of data, especially the step of finding stay points, are optimized by using numpy arrays. Initial experiments have shown that this modification achieves approximately tenfold speedup on the same input dataset.

Table 7.2: Execution time in each step of data processing.

Part of data processing pipeline	Execution time [seconds]
Input data preprocessing	0.419
Stay points detection	0.570
Stay points aggregation	0.030
Building categories mapping	5.166
Home and work locations extraction	0.222
Mapping to nearest POIs	10.815
Assign address info	32.007
Interactive map output	2.216
JSON format output	0.023

Overall, processing time in our experiments is under 60 seconds on an input dataset of 1460 entries, representing a subset of input data with a duration of one month.

Chapter 8

Conclusion

In this work, we have investigated the issues of location data, its collection, cleaning, analysis, processing and possibilities of its further use. We have reviewed existing approaches and solutions to the problem of mining user's significant places. Based on the existing research, we proposed and implemented a method to extract both visited places and significant locations. The proposed method covers main challenges, namely finding stop points in GPS traces, extracting stay regions, identification of home and work locations, mapping visited places to the point of interest and labelling them with the name of the POI, address and category, and finally visualizing the data using icons and different map tiles to represent different amenity categories and provide user-friendly output.

The results of our study showed that the proposed method is effective in accurately identifying visited places and significant locations, mapping places to POIs and categorizing them into different amenity types. Possible use cases of this information include predicting the next user's location, location and activity recommendations, and urban planning.

Our proposed method uses a differential-based stay point detection algorithm and a clustering algorithm HDBSCAN to extract stay points and regions. We designed an algorithm to extract home and work locations based on building category proportions and temporal features extracted from visit points. Mapping of visited places to POIs is done by finding the nearest available POI to a place. Additional data about buildings and POIs in given areas are obtained using OpenStreetMap API. To enhance the usability of the extracted location data, we also developed a user-friendly visualization approach using icons and different map tiles to represent different amenity categories. Reverse geocoding techniques to find names and addresses of POIs were adopted.

We created a dataset corresponding to a real user, performed tests and validated the results. The results were evaluated against place visits exported from Google Maps Timeline. Our experiments have shown that the proposed method can extract and correctly label visited places with an accuracy of 90 %. According to user validation, home and work locations were extracted correctly. Validation of the obtained results has shown that the method can extract visited places with a quite good accuracy and, in some cases, may work better than Google Maps algorithms. We explored improvements in the processing time of the implementation and optimized the proposed algorithms. Overall, the processing time ranges from 50 to 60 seconds on the dataset of 1460 entries, depending mostly on the utilized APIs.

As described in Section 7.6, the missed places pose opportunities for further study and improvements. Expanding the dataset to include more users can also help validate the results of the proposed method and identify any potential limitations or biases. Additionally,

developing a multi-user application can enable the analysis of location data for multiple users simultaneously, allowing for more comprehensive insights since the current solution is implemented as a single-user application. In terms of processing time, several potential strategies could be explored to improve efficiency. The proposed method relies on the availability and accuracy of POI databases, which may not be available or up-to-date in all regions. A good choice with the world-best and most covered areas is Google Places API. More extensive experiments could propose a further use of the obtained results. Location data provide the potential to extract more information about users to provide valuable insights into users' mobility patterns. Our method focuses on visited places. To further extend the scope of this research, methods to extract move segments from trajectories and infer the transportation modes could be incorporated.

Bibliography

[1] Aggarwal, C. C. *Recommender systems: The textbook*. SPRINGER, 2018 [cit. 2023-03-05]. Available at: https://link.springer.com/book/10.1007/978-3-319-29659-3.

[2] Apple. *Legal - Location Services & Privacy - Apple*. [cit. 2023-01-05]. Available at: https://www.apple.com/legal/privacy/data/en/location-services/.

[3] Apple. *Privacy - features*. [cit. 2023-01-05]. Available at: https://www.apple.com/privacy/features/.

[4] Ashbrook, D. and Starner, T. Learning significant locations and predicting user movement with GPS. In: *Proceedings. Sixth International Symposium on Wearable Computers,*. 2002, p. 101–108 [cit. 2023-01-05]. DOI: 10.1109/ISWC.2002.1167224.

[5] Ashbrook, D. and Starner, T. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*. 2003, vol. 7, no. 5, p. 275–286, [cit. 2023-03-09]. DOI: 10.1007/s00779-003-0240-0.

[6] Batistić, L. and Tomic, M. Overview of indoor positioning system technologies. In: IEEE. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2018, p. 0473–0478 [cit. 2023-04-10].

[7] Bhawalkar, P., Bigio, V., Davis, A., Narayanaswami, K. and Olumoko, F. ScheduleNanny: Using GPS to Learn the User's Significant Locations Travel Times and Schedule. *CoRR*. 2004, cs.AI/0409003, [cit. 2022-20-12]. Available at: http://arxiv.org/abs/cs.AI/0409003.

[8] Birant, D. and Kut, A. St-DBSCAN: An algorithm for Clustering Spatial–Temporal Data. *Data & Knowledge Engineering*. 2007, vol. 60, no. 1, p. 208–221. DOI: 10.1016/j.datak.2006.01.013.

[9] Breese, J. S., Heckerman, D. and Kadie, C. M. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *CoRR*. 2013, abs/1301.7363, [cit. 2023-03-04]. Available at: http://arxiv.org/abs/1301.7363.

[10] Burke, R. *User Modeling and User-Adapted Interaction*. 2002, vol. 12, no. 4, p. 331–370, [cit. 2023-03-06]. DOI: 10.1023/a:1021240730564.

[11] Decker, M. Location Privacy-An Overview. In: *2008 7th International Conference on Mobile Business*. 2008, p. 221–230 [cit. 2023-04-20]. DOI: 10.1109/ICMB.2008.14.

[12] D'Roza, T. and Bilchev, G. An overview of location-based services. *BT Technology Journal*. 2003, vol. 21, no. 1, p. 20–27, [cit. 2023-03-19]. DOI: 10.1023/a:1022491825047.

[13] Furletti, B., Cintia, P., Renso, C. and Spinsanti, L. Inferring human activities from GPS tracks. *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*. 2013, [cit. 2023-03-24]. DOI: 10.1145/2505821.2505830.

[14] GIS Wiki. *Geoprocessing*. [cit. 2022-20-12]. Available at: http://wiki.gis.com/wiki/index.php/Geoprocessing.

[15] GISGeography. *The ultimate list of GIS formats and Geospatial File Extensions*. Mar 2023. Available at: https://gisgeography.com/gis-formats/.

[16] Google LLC. *How google uses location information – privacy & terms*. Google, n.d. [cit. 2023-01-09]. Available at: https://policies.google.com/technologies/location-data?hl=en-US#how-find.

[17] Gu, Y., Lo, A. and Niemegeers, I. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications Surveys & Tutorials*. 2009, vol. 11, no. 1, p. 13–32, [cit. 2023-04-10]. DOI: 10.1109/SURV.2009.090103.

[18] Guesgen, H. W. and Marsland, S. *Recognising human behaviour in a spatio-temporal context*. IGI Global, Jan 2011 [cit. 2023-01-06]. Available at: https://www.igi-global.com/chapter/recognising-human-behaviour-spatio-temporal/54670.

[19] Hughes, S., Moreno, S., Yushimito, W. F. and Huerta Cánepa, G. Evaluation of machine learning methodologies to predict stop delivery times from GPS data. *Transportation Research Part C: Emerging Technologies*. 2019, vol. 109, p. 289–304, [cit. 2023-01-11]. DOI: https://doi.org/10.1016/j.trc.2019.10.018. ISSN 0968-090X. Available at: https://www.sciencedirect.com/science/article/pii/S0968090X18314645.

[20] Ibrahim, A., Zhang, H., Clinch, S. and Harper, S. *From GPS to semantic data: How and why-A framework for enriching smartphone trajectories - computing*. Springer Vienna, Aug 2021 [cit. 2022-20-12]. Available at: https://link.springer.com/article/10.1007/s00607-021-00993-z.

[21] Jasinski, M. G. and Baldo, F. *A method to identify aggressive driver behaviour based on enriched GPS Data Analysis*. 2017 [cit. 2022-20-12]. Available at: http://personales.upv.es/thinkmind/dl/conferences/geoprocessing/geoprocessing_2017/geoprocessing_2017_6_20_38002.pdf.

[22] Jiang, J., Pan, C., Liu, H. and Yang, G. Predicting human mobility based on location data modeled by Markov chains. In: *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*. 2016, p. 145–151 [cit. 2023-01-06]. DOI: 10.1109/UPINLBS.2016.7809963.

[23] Kang, J. H., Welbourne, W., Stewart, B. and Borriello, G. Extracting places from traces of locations. *ACM SIGMOBILE Mobile Computing and Communications Review*. 2005, vol. 9, no. 3, p. 58–68, [cit. 2023-03-25]. DOI: 10.1145/1094549.1094558.

[24] Khan Academy. Khan Academy. Available at: https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:online-data-security/xcae6f4a7ff015e7d:user-data-tracking/a/geolocation.

[25] Kim, D. H., Kim, Y., Estrin, D. and Srivastava, M. B. SensLoc: Sensing Everyday Places and Paths Using Less Energy. In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. New York, NY, USA: Association for Computing Machinery, 2010, p. 43–56. SenSys '10. DOI: 10.1145/1869983.1869989. ISBN 9781450303446. Available at: https://doi.org/10.1145/1869983.1869989.

[26] Kumar, A. A. *How precisely do you understand geolocation accuracy?* UniverCity.ai, Jun 2022 [cit. 2022-06-11]. Available at: https://univercity.ai/how-precisely-do-you-understand-geolocation-accuracy/.

[27] KumarTyagi, A. and N.Sreenath, N. Future challenging issues in location based services. *International Journal of Computer Applications*. 2015, vol. 114, no. 5, p. 51–56, [cit. 2023-03-20]. DOI: 10.5120/19978-1921.

[28] Lee, C., Yoon, G. and Han, D. A Probabilistic Place Extraction Algorithm Based on a Superstate Model. *IEEE Transactions on Mobile Computing*. 2013, vol. 12, no. 5, p. 945–956, [cit. 2023-04-10]. DOI: 10.1109/TMC.2012.64.

[29] Li, L., Chen, X., Liu, Q. and Bao, Z. A Data-Driven Approach for GPS Trajectory Data Cleaning. In: Nah, Y., Cui, B., Lee, S.-W., Yu, J. X., Moon, Y.-S. et al., ed. *Database Systems for Advanced Applications*. Cham: Springer International Publishing, 2020, p. 3–19 [cit. 2023-02-24]. ISBN 978-3-030-59410-7.

[30] Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W. et al. Mining User Similarity Based on Location History. In:. New York, NY, USA: Association for Computing Machinery, 2008 [cit. 2023-04-10]. DOI: 10.1145/1463434.1463477. ISBN 9781605583235. Available at: https://doi.org/10.1145/1463434.1463477.

[31] Liao, L., Fox, D. and Kautz, H. Extracting places and activities from GPS traces using hierarchical conditional random fields. *The International Journal of Robotics Research*. 2007, vol. 26, no. 1, p. 119–134, [cit. 2023-03-25]. DOI: 10.1177/0278364907073775.

[32] Lim, J. and Pilesjö, P. *Triangular Irregular Network (TIN) Models*. University Consortium for Geographic Information Science (UCGIS), Apr 2022 [cit. 2023-01-11]. DOI: 10.22224/gistbok/2022.2.7. Available at: http://dx.doi.org/10.22224/gistbok/2022.2.7.

[33] Liu, F., Janssens, D., Wets, G. and Cools, M. Annotating mobile phone location data with activity purposes using machine learning algorithms. *Expert Systems with Applications*. 2013, vol. 40, no. 8, p. 3299–3311, [cit. 2023-01-11]. DOI: https://doi.org/10.1016/j.eswa.2012.12.100. ISSN 0957-4174. Available at: https://www.sciencedirect.com/science/article/pii/S0957417412013425.

[34] Liu, J., Wolfson, O. and Yin, H. Extracting semantic location from outdoor positioning systems. *7th International Conference on Mobile Data Management (MDM'06)*. 2006, [cit. 2023-03-25]. DOI: 10.1109/mdm.2006.87.

[35] Lutkevich, B. *What is end-to-end encryption (E2EE) and how does it work?* TechTarget, june 2021 [cit. 2023-01-05]. Available at: https://www.techtarget.com/searchsecurity/definition/end-to-end-encryption-E2EE.

[36] Lv, M., Chen, L. and Chen, G. Discovering Personally Semantic Places from GPS Trajectories. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2012, p. 1552–1556. CIKM '12. DOI: 10.1145/2396761.2398471. ISBN 9781450311564. Available at: https://doi.org/10.1145/2396761.2398471.

[37] Lv, M., Chen, L., Xu, Z., Li, Y. and Chen, G. The discovery of personally semantic places based on trajectory data mining. *Neurocomputing*. 2016, vol. 173, p. 1142–1153, [cit. 2023-03-25]. DOI: https://doi.org/10.1016/j.neucom.2015.08.071. ISSN 0925-2312. Available at: https://www.sciencedirect.com/science/article/pii/S0925231215012916.

[38] Maguire, D. J. *An overview and definition of GIS - Universidad Nacional del Sur*. [cit. 2023-01-11]. Available at: http://lidecc.cs.uns.edu.ar/~nbb/ccm/downloads/Literatura/OVERVIEW%20AND%20DEFINITION%20OF%20GIS.pdf.

[39] Mainetti, L., Patrono, L. and Sergi, I. A survey on indoor positioning systems. In: *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. 2014, p. 111–120 [cit. 2023-04-10]. DOI: 10.1109/SOFTCOM.2014.7039067.

[40] McInnes, L., Healy, J. and Astels, S. Hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*. 2017, vol. 2, no. 11, p. 205, [cit. 2023-04-15].

[41] Millard Ball, A., Hampshire, R. C. and Weinberger, R. R. Map-matching poor-quality GPS data in urban environments: the pgMapMatch package. *Transportation Planning and Technology*. Routledge. 2019, vol. 42, no. 6, p. 539–553, [cit. 2023-02-24]. DOI: 10.1080/03081060.2019.1622249. Available at: https://doi.org/10.1080/03081060.2019.1622249.

[42] Montoliu, R. and Gatica Perez, D. Discovering Human Places of Interest from Multimodal Mobile Phone Data. In: *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*. New York, NY, USA: Association for Computing Machinery, 2010. MUM '10. DOI: 10.1145/1899475.1899487. ISBN 9781450304245. Available at: https://doi.org/10.1145/1899475.1899487.

[43] New Hampshire Division of Historical Resources. *Introduction to global positioning system*. 2013 [cit. 2023-03-19]. Available at: https://www.nh.gov/nhdfl/documents/introduction-to-global-positioning-system.pdf.

[44] Nurmi, P. and Bhattacharya, S. Identifying Meaningful Places: The Non-parametric Way. In: Indulska, J., Patterson, D. J., Rodden, T. and Ott, M., ed. *Pervasive Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, p. 111–127 [cit. 2023-05-11]. ISBN 978-3-540-79576-6.

[45] Nurmi, P. and Bhattacharya, S. Identifying Meaningful Places: The Non-parametric Way. In: Indulska, J., Patterson, D. J., Rodden, T. and Ott, M., ed. *Pervasive Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, p. 111–127 [cit. 2023-03-19]. Available at: https://doi.org/10.1007/978-3-540-79576-6_7.

[46] OpenStreetMap contributors. *Map features*. [cit. 2023-05-10]. Available at: https://wiki.openstreetmap.org/wiki/Map_features.

[47] Overpass API. *Commons*. N.d. [cit. 2023-04-16]. Available at: https://dev.overpass-api.de/overpass-doc/en/preface/commons.html.

[48] Palma, A. T., Bogorny, V., Kuijpers, B. and Alvares, L. O. A clustering-based approach for discovering interesting places in trajectories. *Proceedings of the 2008 ACM symposium on Applied computing*. 2008, [cit. 2023-03-09]. DOI: 10.1145/1363686.1363886.

[49] Partovi, A., Zheng, W., Jung, T. and Lin, H. Ensuring Privacy in Location-Based Services: A Model-based Approach. *CoRR*. Mar 2020, abs/2002.10055, [cit. 2023-03-20]. Available at: https://arxiv.org/abs/2002.10055.

[50] Pérez Martínez, P. A. and Solanas, A. W3-privacy: the three dimensions of user privacy in LBS. In: *12th ACM Intl. Symp. Mobile Ad Hoc Networking and Computing*. 2011 [cit. 2023-03-27].

[51] Pérez Torres, R., Torres Huitzil, C. and Galeana Zapién, H. Full On-Device Stay Points Detection in Smartphones for Location-Based Mobile Applications. *Sensors*. MDPI AG. Oct 2016, vol. 16, no. 10, p. 1693, [cit. 2023-03-07]. DOI: 10.3390/s16101693. ISSN 1424-8220. Available at: http://dx.doi.org/10.3390/s16101693.

[52] Rao, B. and Minakakis, L. Evolution of mobile location-based services. *Communications of the ACM*. vol. 46, no. 12, p. 61–65, [cit. 2023-03-20]. DOI: 10.1145/953460.953490.

[53] Saab, D. *Conceptualizing Space: Mapping Schemas as Meaningful Representations*. [cit. 2023-01-11]. Dissertation.

[54] Salvador, S. and Chan, P. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: *16th IEEE International Conference on Tools with Artificial Intelligence*. 2004, p. 576–584 [cit. 2023-05-10]. DOI: 10.1109/ICTAI.2004.50.

[55] Saylor Academy. *4.2 Vector Data Models*. Saylor Academy, 2012 [cit. 2023-01-11]. Available at: https://saylordotorg.github.io/text_essentials-of-geographic-information-systems/s08-02-vector-data-models.html.

[56] Schaefer, M. and Pearson, A. Chapter 19 - Accuracy and precision of GNSS in the field. In: Petropoulos, G. p. and Srivastava, P. K., ed. *GPS and GNSS Technology in Geosciences*. Elsevier, 2021, p. 393–414 [cit. 2022-06-11]. DOI: https://doi.org/10.1016/B978-0-12-818617-6.00002-0. ISBN 978-0-12-818617-6. Available at: https://www.sciencedirect.com/science/article/pii/B9780128186176000020.

[57] Skyhook. *The difference between location accuracy and precision.* [cit. 2022-06-11]. Available at: https://www.skyhook.com/blog/advertising/the-difference-between-location-accuracy-and-precision-and-why-you-need-to-know.

[58] Srivastava, A. *DBSCAN clustering algorithm.* Dec 2021. Available at: https://blog.knoldus.com/dbscan-clustering-algorithm/.

[59] Su, X. and Khoshgoftaar, T. M. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence.* 2009, vol. 2009, p. 1–19, [cit. 2023-03-04]. DOI: 10.1155/2009/421425.

[60] Sun, Y., Zhu, H., Zhuang, F., Gu, J. and He, Q. *Exploring the urban region-of-interest through the analysis of online map search queries: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* Jul 2018 [cit. 2023-01-06]. Available at: https://dl.acm.org/doi/pdf/10.1145/3219819.3220009.

[61] Sánchez Lozano, J. M., Teruel Solano, J., Soto Elvira, P. L. and Socorro García-Cascales, M. Geographical Information Systems (GIS) and Multi-Criteria Decision Making (MCDM) methods for the evaluation of solar farms locations: Case study in south-eastern Spain. *Renewable and Sustainable Energy Reviews.* 2013, vol. 24, p. 544–556, [cit. 2023-01-11]. DOI: https://doi.org/10.1016/j.rser.2013.03.019. ISSN 1364-0321. Available at: https://www.sciencedirect.com/science/article/pii/S1364032113001780.

[62] Tambunan, H. B., Barus, D. H., Hartono, J., Alam, A. S., Nugraha, D. A. et al. Electrical Peak Load Clustering Analysis Using K-Means Algorithm and Silhouette Coefficient. In: *2020 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP).* 2020, p. 258–262 [cit. 2023-05-10]. DOI: 10.1109/ICT-PEP50916.2020.9249773.

[63] Techopedia. *Spatial Data* [https://www.techopedia.com/definition/871/spatial-data]. [cit. 2023-05-11]. Accessed: May 13, 2023.

[64] Turing. *How collaborative filtering works in Recommender Systems.* Turing Enterprises Inc, Jul 2022 [cit. 2023-03-04]. Available at: https://www.turing.com/kb/collaborative-filtering-in-recommender-system.

[65] Vidmar, L., Štular, M., Kos, A. and Pogačnik, M. An automatic Wi-Fi-based approach for extraction of user places and their context. *International Journal of Distributed Sensor Networks.* 2015, vol. 11, no. 3, p. 154958, [cit. 2023-03-07]. DOI: 10.1155/2015/154958.

[66] Wang, D., Liang, Y., Xu, D., Feng, X. and Guan, R. A content-based recommender system for Computer Science Publications. *Knowledge-Based Systems.* 2018, vol. 157, p. 1–9, [cit. 2023-03-05]. DOI: 10.1016/j.knosys.2018.05.001.

[67] Wang, G. Machine learning for inferring animal behavior from location and movement data. *Ecological Informatics.* 2019, vol. 49, p. 69–76, [cit. 2023-01-11]. DOI: https://doi.org/10.1016/j.ecoinf.2018.12.002. ISSN 1574-9541. Available at: https://www.sciencedirect.com/science/article/pii/S1574954118302036.

[68] Wikimedia Commons. *Cape Town by SPOT Satellite*. [cit. 2023-05-09]. Available at: https://upload.wikimedia.org/wikipedia/commons/c/c3/Cape_Town_SPOT_1186.jpg.

[69] Wikimedia Commons. *Long and Loop Street map*. [cit. 2023-05-09]. Available at: https://commons.wikimedia.org/wiki/File:Long_and_Loop_Street_map.svg#/media/File:Long_and_Loop_Street_map.svg.

[70] Wikimedia Commons. *Topographic map of the Caucasus in English*. Jul 2016 [cit. 2023-05-09]. Available at: https://commons.wikimedia.org/wiki/File:Caucasus_topographic_map-en.svg#/media/File:Caucasus_topographic_map-en.svg.

[71] Ye, Y., Zheng, Y., Chen, Y., Feng, J. and Xie, X. Mining Individual Life Pattern Based on Location History. In: *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*. 2009, p. 1–10. DOI: 10.1109/MDM.2009.11.

[72] Yera, R. and Martínez, L. Fuzzy tools in recommender systems: A survey. *International Journal of Computational Intelligence Systems*. 2017, vol. 10, no. 1, p. 776, [cit. 2023-03-05]. DOI: 10.2991/ijcis.2017.10.1.52.

[73] Zhang, K., Li, H., Torkkola, K. and Gardner, M. Adaptive Learning of Semantic Locations and Routes. In: Hightower, J., Schiele, B. and Strang, T., ed. *Location- and Context-Awareness*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, p. 193–210 [cit. 2023-03-19].

[74] Zheng, V. W., Zheng, Y., Xie, X. and Yang, Q. Collaborative Location and Activity Recommendations with GPS History Data. In: *Proceedings of the 19th International Conference on World Wide Web*. New York, NY, USA: Association for Computing Machinery, 2010, p. 1029–1038. WWW '10. DOI: 10.1145/1772690.1772795. ISBN 9781605587998. Available at: https://doi.org/10.1145/1772690.1772795.

[75] Zheng, Y., Zhang, L., Xie, X. and Ma, W.-Y. *Mining interesting locations and travel sequences from GPS trajectories: Proceedings of the 18th International Conference on World Wide Web*. Apr 2009 [cit. 2023-01-06]. Available at: https://dl.acm.org/doi/pdf/10.1145/1526709.1526816.

[76] Zhou, C., Bhatnagar, N., Shekhar, S. and Terveen, L. Mining Personally Important Places from GPS Tracks. In: *2007 IEEE 23rd International Conference on Data Engineering Workshop*. 2007, p. 517–526 [cit. 2023-03-25]. DOI: 10.1109/ICDEW.2007.4401037.

[77] Zhou, C., Frankowski, D., Ludford, P., Shekhar, S. and Terveen, L. Discovering personally meaningful places. *ACM Transactions on Information Systems*. 2007, vol. 25, no. 3, p. 12, [cit. 2023-03-09]. DOI: 10.1145/1247715.1247718.

Appendix A

Contents of the included storage media

The root folder contains following directories:

- `significant_locations`: source codes for the proposed method, including folder `experiments` with source codes to the conducted experiment described in Chapter 7
- `dataset`: input dataset discussed in Section 7.1, including output from Google Maps platform used for validation
- `latex_src`: source codes for this work