UTRECHT UNIVERSITY

Department of Information and Computing Science

Applied Data Science master thesis

Identifying and improving quality issues in Google Semantic Location History DDPs for public transport activities

First examiner:

Erik-Jan van Kesteren

Second examiner:

Thijs Carrière

Candidate:

Daniëlle Bakker

0187410

July 5, 2023

I would like to give special thanks to my supervisors Erik-Jan van Kesteren and Thijs Carrière for their guidance and feedback these past 10 weeks.

Abstract

More and more human life takes place online, resulting in an increasing role of digital privacy in society. New laws are created to protect people's privacy. As a response to these laws, companies now give their users the opportunity to download their personal data as Data Download Packages (DDPs). A recent study used the Google Semantic Location History DDPs to investigate how the COVID-19 pandemic changed travel behaviour. However, these DDP suffer from potential quality issues, influencing the data quality and inferences made on these data. The aim of this project is to identify these potential quality issues, take them into account with data imputation where possible, and see if this makes a difference. This thesis will focus on errors in public transport activity types found in Google Semantic Location History.

A Python script will check if different parts of the data meet set requirements to locate the quality issues. This script will count the number of errors and use data imputation where possible, resulting in a more accurate data extraction. This, in turn, leads to a better understanding of travel behaviours. While multiple steps are still needed to make the extraction as accurate to reality as possible, this is a first step towards improving the accuracy of inferences with Google Semantic Location History data.

Contents

1. Introduction

As an increasing part of human interaction and behavior occurs online, digital privacy and data protection are increasingly central in society. According to the EU's 2018 General Data Protection Regulation (GDPR), users have the right to access their own personal data where the data must be in "a structured commonly used and machine-readable format" [1], [2]. Therefore, companies now give the possibility for their users to download their personal data, typically as a Data Download Package (DDPs) [3]. These DDPs contain all the user's digital trace data collected by a company [4], [5] The GDPR also states that the user is allowed to share their personal data with another party making the process called data donation possible [1], [2].

1.1 Data donation through Data Download Package

Data donation means that the user actively consents to donate their data to researchers. Before data donation, it used to be difficult to contact the user and get consent for using personal data [5], [6]. The use of DDPs has some other advantages compared to other types of data collection. The DDPs contain a full overview of how the user interacted with the platform from the moment the account was created. The companies automatically record all the data stored in DDPs, meaning no extra applications need to be installed limiting researcher bias. Another advantage is that the DDPs are organized in a structured manner in time periods and types of information for example social media activity or Google location or search history [4].
The DDPs can contain privacy-sensitive data or data that the researchers are not interested in. Because of this, software like PORT and OSD2F are developed where the data from DDPs is transferred to the researchers without

sensitive data and only data necessary for the research. The first step of this kind of software is for the participant to request their DDP after which the participant downloads the DDP to their personal device. Before extracting the data, researchers must think about which information they need to answer their research question and what part of the DDP contains that data [5]. The user can decide which part of that data they want to share with the researcher. OSD2F [7] and PORT [8] deal differently with what gets sent to the researcher. OSD2F has the option for users to select which files are sent to the researcher but it does send the chosen parts of the file, after anonymization, where PORT will only extract the relevant information and not the files [7], [8]. With OSD2F, the anonymized files are first loaded on the OSD2F server after which the user can give their consent to send the files to the researcher [7]. With PORT the files will always stay on the user's device and only the relevant data will be sent to the researcher after the user gives consent to donate the data. This will be the first time the data is shared with the researchers. This way the researcher can perform their analysis while maintaining user privacy [8]. Figure 1.1 shows an overview of the different steps of PORT.

1.2 Aim of the project

In a recent study, PORT and Google Semantic Location History DDPs were used to investigate how the COVID-19 pandemic changed travel behavior in the Netherlands using activity frequencies and activity types which can be found in these DDPs. The problem is that these DDPs contain potential quality issues, for example, location accuracy or the wrong predictions for the activity types. This project aims to improve the PORT pipeline by letting it identify these potential quality issues and replace the identified errors with imputing more realistic values. The before and after will be compared to see the effect of these improvements. This thesis report will focus on the train, bus, tram, subway, and plane activity type in The Netherlands.

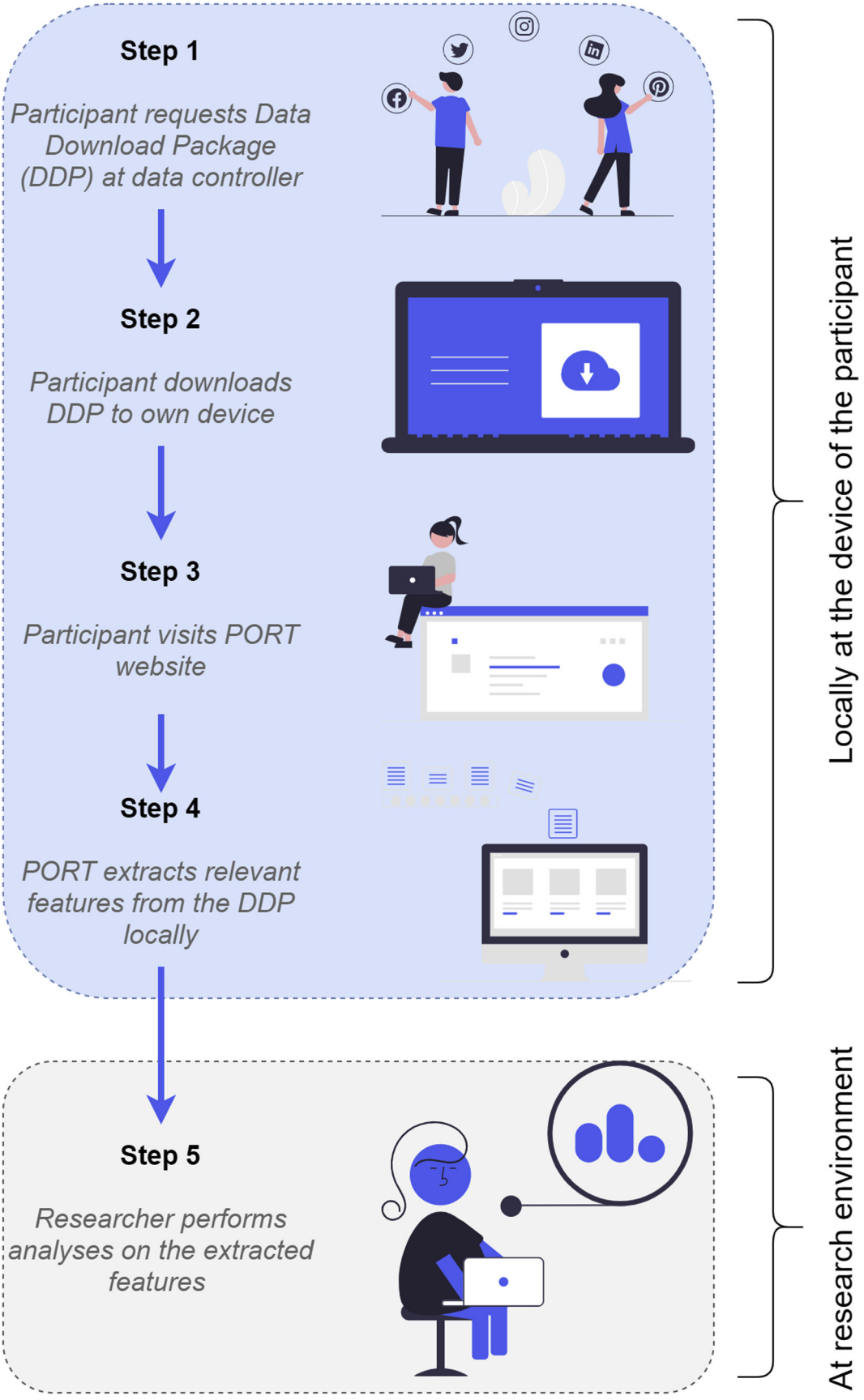

Figure 1.1: The workflow of PORT [8].

1.3 Google Semantic Location History

The settings set by the user influence how Google can collect the location history data. When the user allows the collection of location history data, Google can extract this data by periodically collecting data from GPS, Wi-Fi, mobile networks, and device sensors which are found in the so-called Google Location Service. If the user does not consent to the collection of location data, it is possible to turn this function off [9]–[11].
Google Location Services has three kinds of device sensors; the motion, environmental and position sensors [12]. For this thesis, the motion sensors which measure acceleration forces and rotational forces, and the position sensors that measure the physical position of a device are of importance[12]. This is because these sensors can be used to locate the location and orientation of the device, which in turn can be used to make predictions for the Google Semantic Location History data. A side note to make is that these sensors are used to make games/apps for Android devices. There was no evidence found that Google uses these sensors to predict the activity type let alone the algorithm Google uses to make those predictions with the values from the sensors. However, it is nice to know how Google collects its data and where some errors might come from even if it is based on assumptions.
Google Play offers *ActivityRecognitionClient* to help with activity recognition together with the Transition API to improve accuracy [13]. It automatically identifies activities by regularly collecting the sensor data and processing these with machine learning models [14].
As mentioned before, the motion sensors measure acceleration forces and rotational forces which can be used to determine device movement, for example, tilt, shake, rotation, or swing. This can be used to predict how the user's device is moving. The different sensors that fall under the motion sensors are the gravity sensor, accelerometer, rotation vector sensor, significant motion sensor, step counter, step detector sensor, and gyroscope sensor [15]. The gravity sensor locates the relative orientation of the device. The accelerometer measures the acceleration of the device. The rotation vector sensor measures the rotation of the device. The significant motion sensor

turns on if it measures an event and turns off again. The step counter counts the steps, and the step detector sensor detects when the user takes a step. The gyroscope sensor measures the rotation of the device [15].
There has been some research done on the accuracy of Google's location history. Next to the location history they also had GPS data to compare the actual location with the one predicted by Google. They made a radius around the location Google predicted. If the GPS location fell into that radius, it was a hit and if it fell outside of the radius, it was a miss. They looked at Google's GPS, 3G, 2G, and Wi-Fi predictions. GPS did best with 52% of hits, 3G had 38%, 2G 33% and Wi-Fi did the worst with 7% hits [16].
In the next section, *Data*, the format of the Google Semantic Location History will be introduced together with the privacy-sensitive data it contains. The *Methods* will describe how the different quality issues are found and flagged and how the imputation is done. The *Results* will show the number of found quality issues. The *Discussion and Conclusion* will give a summary of the findings with the implications and the limitations and possible future work.

2. Data

The Google Semantic Location History data used in this project concerns data about the author of this thesis. As a result of this data being available, no additional participants and consent were needed. The time period of the data starts in July 2016 and ends in April 2023. Since that is a lot of data to go through for this 10-week project, the focus will mostly be on September, except for planes which will be in July. Section 2.1 will explain how the data was obtained and what kind of information it contains and the privacy issues that brings. It is important to know the structure of the JSON files, to know where to look for the information that can be found in the data, and to help with identifying the different kinds of errors.
To help with identifying the quality issues, data from OpenStreetMap [17] was collected. The data consist of coordinates from public transport stations and stops in The Netherlands and will be discussed in section 2.2.

2.1 Google DDPs

Google made it possible for their users to download their personal data with DDPs through the *Download your data* page [1]. When downloading the user's personal data there is a choice to include only a selection of the data in the DDP, for example, the data about the location history. There are options in delivery method, export type, and file type [18], [19]. The zip file contains a folder Takeout which in turn contains a folder with the location history data and an HTML to look at the Google account archive. In the Semantic Location History folder, there are folders of different years containing JSON files for each month [20]. The structure of the Takeout folder can be found in Figure 1.2.

[1] https://takeout.google.com/

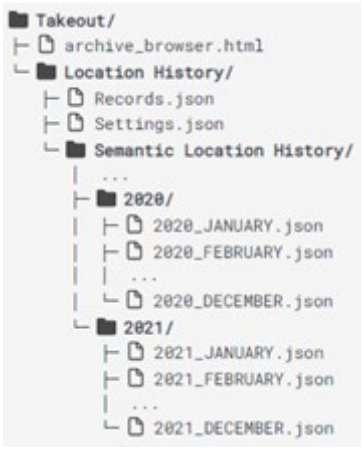

Figure 2.1: The structure of the Google Semantic Location History DDP [20].

The JSON files start with the *timelineObjects*. After this, there are two types of segments. The *activitySegment* and the *placeVisit*. The *activitySegment* consists of *startLocation*, *endLocation*, *duration*, *distance*, *activityType*, *activities*, and *waypointPath*. The *startLocation* and *endLocation* both contain the latitudeE7 and the longitudeE7 of those locations in the WGS84 coordinate reference system [9], [20]. The *duration* has the start and end timestamps in the ISO 8601 datetime format. Distance shows the distance in meters and *activityType* shows the most probable movement type with a probability between 0 and 100. *Activities* shows possible activity types with their probabilities with data from 2018 and before having the top 3 and after 2018 the first 14-16 most likely activity types even if the probabilities are very small. The *waypointPath* has different coordinates of points during the activity [21]. There are a lot more variables in the *activitySegment* but these are of interest for this thesis.
The *placeVisit* segment is less interesting for this thesis since the subject is about the movement of people and not about the places they stay.
Figure 1.3 illustrates the JSON structure for files in the Google Semantic History Location DDP. The coordinates are for privacy reasons removed.

In regard to privacy, the data in Google's location history has some sensitive information about the user. More specifically, the locations of the user can be found in the location history JSON files as coordinates as well as ac-

```
"timelineObjects": [{
  "activitySegment": {
    "startLocation": {
      "latitudeE7": [redacted],
      "longitudeE7":[redacted],
      "sourceInfo": {
        "deviceTag": [redacted]
      }
    },
    "endLocation": {
      "latitudeE7": [redacted],
      "longitudeE7":[redacted],
      "sourceInfo": {
        "deviceTag": [redacted]
      }
    },
    "duration": {
      "startTimestamp": "2018-07-01T07:02:41.690Z",
      "endTimestamp": "2018-07-01T07:40:04Z"
    },
    "distance": 6419,
    "activityType": "IN_PASSENGER_VEHICLE",
    "confidence": "HIGH",
    "activities": [{
      "activityType": "IN_PASSENGER_VEHICLE",
      "probability": 89.10366763462011
    }, {
      "activityType": "WALKING",
      "probability": 7.705263609110605
    }, {
      "activityType": "IN_BUS",
      "probability": 2.3985920743213045
    }],
    "waypointPath": {
      "waypoints": [{
        "latE7": [redacted],
        "lngE7":[redacted]
      }, {
        "latE7": [redacted],
        "lngE7":[redacted]
      }, {
        "latE7": [redacted],
        "lngE7": [redacted]
      }],
      "source": "INFERRED"
    },
```


Figure 2.2: The structure of a JSON file as found in the Google DDP.

tual addresses. The JSON file called *Settings* contains data about the user's phone [21]. The user's day-to-day life can be read from the JSON files which are privacy-sensitive data. This is where PORT can be applied, only receiving the data researchers need and leaving the rest of the data on the user's own device maintaining the user's privacy [8].

2.2 OpenStreetMap

The data collected from OpenStreetMap [17] contains data on public transport stations and stops more specifically the train, tram, bus, subway, and plane. Only the interesting parts for this project were extracted, namely the x and y coordinates and the station's or stop's name. The x and y coordinates are to determine the location of the station or stop. The name is to check with the name in the Google Semantic Location History and with the knowledge of the actual journey to see if the logged coordinates are correct. Only locations located in The Netherlands were collected.

3. Method

As discussed in the introduction, there are potential quality issues in Google Semantic Location History data. In this study, the errors of interest occur in the *activitySegment* namely the *activityType*, the *duration*, and the *distance*. The next section will explain how these quality issues were located in the Google Semantic Location History data. The script and input files can be found on `https://github.com/Danielle222/Thesis`.

A way to identify quality issues is by setting requirements. This is based on the practice of unit testing. This practice is a test to see if a program meets the set requirements [22]. The test would be a function that compares the output to the requirement returning *True* if the requirement is met and *False* if not [23]. Since this study is also interested in potential data imputation of the quality issues, the approach was influenced by the unit testing, but did more than detecting and flagging errors.

3.1 Overview

The *activitySegment* consists of multiple variables but the interest lies with *activityType*, the *distance*, and the *duration*. Figure 3.1 shows the workflow of how the different parts were checked.
Starting with the *activityType*, Google's algorithm distinguishes between 38 different activity types when classifying an activity [21]. To narrow this down, this study only looks at five public transport activity types namely the train, bus, tram, subway, and plane. There were two requirements the activity types must meet to check if the predicated *activityType* was correct. The first was if the start and end locations were public transport stations or stops. The second was if the average speed did not exceed the maximum

Figure 3.1: An flowchart of the process.

speed the vehicles are allowed in The Netherlands.

Next the requirement for the *distance*. To calculate the *distance*, the Google algorithm uses the waypoints from the *waypointPath* [24], [25]. For public transport, there is the *transitPath* instead of the *waypointPath* which shows the different stops the transport makes. The distance calculated by the Google algorithm was compared with the straight-line distance between the start and end locations. The requirement was that these two distances did not differ too much from each other which will be explained later.

As last the *duration*, where the assumption was made that there are no errors in the *startTimestamp* and the *endTimestamp* themselves meaning the time of the timestamps. The timestamps are in ISO 8601 format with, most of the time, milliseconds, and the UTC time zone [26], [27]. There could still be errors in the *duration* in the length of the activity. A continuous predicated *activityType* train with a *duration* of 72 hours is probably too long. The requirement was that the *duration* was no longer than 24 hours.

3.2 Activity type

3.2.1 Public transport station requirement

3.2.1.1 Error detection

The first requirement for the *activityType* was if the start and end locations were public transport stations or stops. This was done by checking if the start and end locations fall into a 500-meter radius around the coordinates of these stations or stops. This is based on the Macarulla et al. [16] who performed an accuracy assessment and error prediction of the Google timeline. For this method, the coordinates of public transport stations and stops in The Netherlands were collected using the community-collected, open-source database OpenStreetMap [17]. With QuickOSM from QGIS 3.28.5 [28] the coordinates of the different stations and stops were obtained using certain queries (3.1). The code for this method was written with *Python 3.8* [29]. The coordinates from OpenStreetMap were loaded into a *GeoDataFrame* from the *GeoPandas 0.13.0* package [30] and transformed from the EPSG 4326 to EPSG 32634 with *shapely* [31] to get the points in meters instead of degrees. The Google Semantic Location History was also transformed from EPSG 4326 to EPSG 32634 to be able to check, with *contains()* if the location was in the buffer. If the start or the end location cannot be found in the buffer, the requirement was not met. When the requirement was not met for the train activity type, the next step was checking if the start or end location can be found in the buffer of a tram or subway station.
This requirement worked differently for the flying activity type. The data about the airports were only locations in The Netherlands, but flying usually includes going abroad. This would mean that only the start or the end location was in The Netherlands. To avoid flagging all the plane activity as errors, only one location had to be found in the buffer around airports to meet the requirement.

Mode of public transport	Query
Train	railway=station
Bus	highway=bus_stop
Tram	railway=tram_stop
Subway	station=subway
Plane	aeroway=aerodrome [17]

Table 3.1: QuickOSM query table.

3.2.1.2 Data imputation

There was also some data imputation involved in this step, specifically for the train *activityType*. When a start or end location was not a train station, but a tram station, this journey would be aggregated to the total tram travels. This assumed that the tram travel was not characterized by Google's algorithm. Instead, the transfer was missed, making it part of the predicted train *activityType*. It would not change anything with the total count of train travel. That would only happen when both the start and end locations were not train stations, meaning one train journey would be subtracted from the total train travel.

3.2.2 Speed requirement

The second requirement for the *activityType* was checking if the average travel speed of the user was lower than the allowed maximum speed. That is, by dividing the *distance* by the *duration* and comparing this with the maximum speed of the *activityType* predicted by the Google algorithm. If the speed was higher, this would mean that the requirement was not met. These maximum speed thresholds can be found in Table 3.2. The number of times the speed was higher than the maximum speed will be counted and added to the result table.

Transport	Max speed
Train	140 km/h [32]
Bus	100 km/h [33]
Tram	70 to 80 km/h [34]
Subway	70 to 80 km/h [35]
Plane	880 to 926 km/h [36]

Table 3.2: The maximum speed of the different modes of public transport.

3.3 Distance

To test if the *distance* met the requirement, the public transport stations and stops were also used. With *haversine* the distance between two public transport stations in EPSG 4326 would be calculated. However, since the distance was calculated in a straight line instead of following the public transport tracks, the true and calculated distance could be different. By looking at the distances found in the Google Semantic Location History and comparing this with the *haversine* calculated distances, the maximum difference of 5 kilometers was chosen. If the difference was bigger than 5 kilometers, the requirement was not met. Some of the *activitySegment* did not have a distance but the calculated distances could give an indication of the possible distance.

To see if there was a difference between the Google Semantic Location History JSON file distance and the newly calculated *haversine* distances both were aggregated. One time with just all the Google predicated distances and one time with data imputation of the missing distances replaced with the calculated *haversine* and also the wrongly predicted distances replaced by the calculated *haversine*.

To validate if the distance in the JSON file was the wrong one, travel planning websites, such as *Omio* and *The Train Line* were used [37], [38]. Also, the NS website has files on the *Tariefeenheden* which are the distances between stations in the Netherlands [39].

3.4 Duration

Deciding on the maximum duration of the travel time was complicated. The average public transport trip takes 57.5 minutes with 30 minutes for train and 21.6 minutes for tram, bus, and subway [40]. However, some train trips, without transfer, in The Netherlands can be multiple hours. When looking at the Google Semantic Location History data, sometimes trips of different activity types were combined meaning one *activitySegment* contained a train travel and a tram, or two different trains. Sometimes the device sensors took a timestamp at the wrong time, in this case, the end timestamp too late causing the duration to be too long.
The maximum duration depended on the error type that needs to be found. Since the first error was found with the requirement of the start and end location being a public transport station or stop, the choice falls on the second type of error. This made the requirement that the duration was no longer than 24 hours to find real outliers of where travel takes multiple days. The number of journeys that exceeded that number was counted and added to the results.

4. Results

The next section will show the results of the different requirement checks and data imputation, starting with the *activityType*, followed by the *distance* and *duration*.

4.1 Activity type

4.1.1 Public transport station requirement

The first requirement of the *activityType* was checking if the start and end locations were public transport stations or stops. The results are the number of travels in September between 2017 and 2022. For the plane activity type the period is between July 2017 and 2022.

4.1.1.1 Error detection

In Table 4.1 the total number of train travels can be found with the total of times the start or end station was not a train station. This table shows a total of 145 train trips with 30 locations not being a train station.
First, the 2022 routes which are routes between Heemskerk and Utrecht Science Park and back. A total of 41 train trips were found with 10 locations not being a train station. When checking which type of station these locations could be, 25 tram stations and 2 no other type were found.
2019, trips between Heemskerk and Leiden Centraal, has 31 train trips with 16 locations not being train stations. This time no other type was found for 15 locations. 4 tram stops and 6 subway stops were found.
The travel by bus, tram, and subway between September 2017 and 2022 can be found in Table 4.2. The total for tram is 7 with 1 not being a tram stop and 7 subway trips but 7 locations not being a subway stop. There were 7

Year	Month	Train travel	Not train station	Tram stop	Subway stop	No other stop
2017	SEP	34	0	0	0	0
2018	SEP	33	2	6	0	1
2019	SEP	31	16	4	6	15
2020	SEP	6	2	0	0	2
2021	SEP	0	0	0	0	0
2022	SEP	41	10	25	0	2

Table 4.1: The results of the station requirement for the train activity type in September. *Train travel* shows the total amount of train trips. *Not train station* is the number of locations that are not a train station. *Tram stop* and *Subway stop* show the number of stops found of that specific type and *No other stop* when no other type was found.

bus trips found and all the locations were bus stops.

Year	Month	Tram travel	Not tram stop	Subway travel	Not subway stop	Bus travel	Not bus stop
2017	SEP	0	0	0	0	0	0
2018	SEP	5	1	0	0	1	0
2019	SEP	0	0	0	0	0	0
2020	SEP	0	0	0	0	3	0
2021	SEP	0	0	0	0	0	0
2022	SEP	2	0	7	7	3	0

Table 4.2: The results of the station/stop check for tram, bus, and subway in September. The *travel* columns are the total trips of that *activityType*. The *not* columns are the total of locations that do not match in public transport station/stop type.

For the flying activity type the month of July between 2017 and 2022 was chosen as can be seen in Table 4.3. There were 5 plane trips found where twice both the start and end location were no airports.

4.1.1.2 Data imputation

Instead of only counting the number of times the station or stop types did not match the predicted *activityType*, these numbers were also used for data imputation. In Table **??** are the times traveled shown per *activityType* without data imputation and with data imputation. For train travel, there was no

Year	Month	Plane travel	Not airport
2017	JULY	3	1
2018	JULY	2	1
2019	JULY	0	0
2020	JULY	0	0
2021	JULY	0	0
2022	JULY	0	0

Table 4.3: The results of the airport check for the flying activity type in July. *Plane travel* is the total number of plane trips. *Not airport* is the number of times that both the start and end location is not an airport.

difference in the total number staying at 145. For tram and subway travel, there are some differences. Tram went from 7 to 16 and subway from 7 to 1. There is no data imputation performed on the bus trips.

Year	Month	Train with imputation	Tram with imputation	Subway with imputation
2017	SEP	34	0	0
2018	SEP	33	6	0
2019	SEP	31	0	1
2020	SEP	6	0	0
2021	SEP	0	0	0
2022	SEP	41	10	0

Table 4.4: The results of the data imputation for train, tram, and subway in September. Each column shows the total number of trips found, for that type, in the data after data imputation.

4.1.2 Speed requirement

The second requirement was that the average speed was not allowed to be higher than the maximum allowed speed of that public transport in The Netherlands. When looking at Table 4.5 and Table 4.6 there is only one instance of this requirement not being met.

Year	Month	Speed train	Speed tram	Speed subway	Speed bus
2017	SEP	0	0	0	0
2018	SEP	0	0	0	0
2019	SEP	1	0	0	0
2020	SEP	0	0	0	0
2021	SEP	0	0	0	0
2022	SEP	0	0	0	0

Table 4.5: The results of the speed check for train, tram, subway, and bus in September. Each column shows the number of times the average speed was higher than the allowed maximum speed of that vehicle.

Year	Month	Speed plane
2017	JULY	0
2018	JULY	0
2019	JULY	0
2020	JULY	0
2021	JULY	0
2022	JULY	0

Table 4.6: The results of the speed check for plane in July. *Speed plane* shows the number of times the logged average speed was higher than the average speed of planes.

4.2 Distance

The requirement for distance was that the difference between the distance from the Semantic Google History Locations JSON file and the *haversine* calculated distance was no more than 5 kilometers. As is shown in Table 4.7 there was a total of 5 wrong distances. When looking into the data, these were all predicted as train. To account for this, the calculated *haversine* distance replaced the original logged distance for these activities in obtaining the total distance. There was a total of 4 missing distances, again all in the train activity type, which was also replaced by the calculated *haversine* distance. The before distance improvement and after can also be found in Table **??**.

Year	Month	Activity distance	Distance missing	Wrong distance	Activity distance imputation
2017	SEP	738.815	0	0	738.815
2018	SEP	804.045	0	0	804.045
2019	SEP	709.593	5	3	765.577
2020	SEP	169.408	0	0	169.408
2021	SEP	0	0	0	0
2022	SEP	1509.16	0	1	1503.843

Table 4.7: The results of the distance check in September. *Activity distance* shows the total logged distance. *Distance missing* is the total number of missing distances and *Wrong distances* is the total of wrong logged distances. *Activity distance imputation* is the new total distance after improving the logged distances.

4.3 Duration

The requirement for the *duration* is that the activity is no longer than 24 hours. As can be seen in Table 4.8 and Table 4.9, none of the travel took longer than the maximum set duration.

Year	Month	Train max duration	Tram max duration	Bus max duration	Subway max duration
2017	SEP	0	0	0	0
2018	SEP	0	0	0	0
2019	SEP	0	0	0	0
2020	SEP	0	0	0	0
2021	SEP	0	0	0	0
2022	SEP	0	0	0	0

Table 4.8: The results of the duration check for train, tram, bus, and subway in September. The columns show per *activityType* the number of activities longer than 24 hours.

Year	Month	Maximum duration
2017	JULY	0
2018	JULY	0
2019	JULY	0
2020	JULY	0
2021	JULY	0
2022	JULY	0

Table 4.9: The results of the duration check for plane in July. *Maximum duration* shows the number of times the plane activity was longer than 24 hours.

5. Conclusion

In an earlier data donation study errors were found in DDPs. The goal of this project was to find these potential quality issues in the Semantic Google Location History, more specific errors for *activityType* train, bus, tram, subway, and plane. The Google algorithm was not always able to log switching between activity types. Instead of the last train station being the end location, the bike ride home was included in the *activitySegment* or train and tram travel is combined into one. These kinds of errors cause quality issues when extracting the data. There will be an overestimation of the actually traveled distance and the tram travel will not be counted to the total of tram travel. The way this study handles these quality issues is with requirement checking and data imputation.

5.1 Activity Type

5.1.1 Public transport station requirement

Starting with the first requirement, some activities did not have the correct type of station or stop. These errors prevent correct inference on travel behavior and should be dealt with. By checking if the start or end location is another type of station or stop, the errors can be identified and improved. The effect of this is that more information is extracted for the data which in turn gives a better image of the travel behavior.
With the timestamp of the found error, it is possible to locate the error in the JSON file. This way types of errors were identified. A common one is that the Google algorithm misses the transfers between activity types. This results in different types of public transport stations between the start and end locations. This error is reflected in the results as the total train trips

in September was 145 before and after the identifying and improving with an increase in tram trips (Table 4.1, 4.4). This means tram trips were found in the same *activitySegment* as the train tips instead of being a separate one. Another reason for the increase in tram trips is the decrease in subway trips. Some of the predicted subway trips were actually tram trips.
Some locations had multiple other types found and some had none. When looking at the JSON file it seems that for the *no other type* the start or end location coordinates were wrong. When the location coordinates were wrong, the activity would be flagged as an error and removed from the results. This means that some information will be lost because of mobile sensor errors, influencing the conclusion on travel behavior. The reason some locations have multiple other types found is that some tram, bus, and subway stations can be found twice in the coordinate list. This is because the line goes two ways and the stops for both ways can be at the same location. Another reason could be that these stops are closer together than the 500-meter radius, resulting in finding more than one inside it.
To summarize, there seem to be two types of errors. One is that the location is wrongly predicted and the other is that the transfer of activity type is missed by the Google algorithm. The results show that it is possible to identify the errors and that some are improved. This means that the accuracy of the data extraction has been improved, which in turn gives a better image of the travel behavior.

5.1.2 Speed requirement

When the average speed requirement was not met, it was linked with an error in the *distance*. An overestimation of the *distance* results in a higher average speed. This means that this requirement cannot be used to determine if the *activityType* is wrongly predicted and will have no influence on the data extraction.

5.2 Distance

An error in distance results in a wrong conclusion about the travel behavior specifically in the total distance traveled. To prevent this, the wrong and missing distances were replaced by the *haversine* calculated distance. This data imputation results in a total distance traveled that was closer to reality. When there were missing distances, there was an underestimation of the total distances traveled, resulting in a higher total traveled distance. However, the wrong distances were usually an overestimation of the actual distance, resulting in a lower total traveled distance.
To conclude, the errors in the logged distance do in fact influence the total traveled distance. The way of data imputation is a starting point and there is a way to improve it, which will be mentioned in the discussion.

5.3 Duration

When looking at the data, all the trips predicted as being public transport met the duration requirement. It seems that the Google Algorithm is not able to predict the *activityType* when the duration is that long, classifying it as *UNKNOWN_ACTIVITY_TYPE*. This issue did not occur after October 2018 which indicates that some sensors have been improved. As of right now, this requirement does not influence the data extraction.

As discussed, it is possible to identify and improve different quality errors. This project is a first step in this process but more information is already found during the data extraction. This also includes the type of errors to look for. There are still multiple steps necessary which will be discussed in the next section.

5.4 Discussion

The Semantic Google Location History data used for this project concerned data about the author of this thesis. This made it easier to obtain the data and removes the issue of privacy-sensitive data. It also helped in knowing the actual route that was taken and recognizing locations in the data to identify what the error is. It is possible to ask participants about their trips. However, this is time intensive which results in participants that stop responding. Especially when you want to know about trips multiple years ago, for example, travel changes around COVID.

5.4.1 Limitations

Coordinates of public transport stations and stops were needed to check if a start or stop location was a station or stop. Since this study focuses on The Netherlands only these coordinates were collected. This means that for participants that live abroad or travel a lot, errors can be identified less correctly, resulting in less accurate results. To maximize the effectiveness of the quality pipeline in real-world studies, ideally, the public transport data should contain all potential stops in the study's scope.
Right now, the radius around the public transport stations and stops is all the same even though the distance between tram, bus, and subway stations might be smaller than between train stations and airports. This could result in finding more possible stops for the start or end location than there actually are. It becomes a problem when two different types of stops are found. It is not possible to know which is the right type, meaning that the one chosen could be wrong. This results in a wrong count for the travel type which in turn could influence the conclusion about travel behavior.
Because of the limited availability of data, some potential errors that might occur in the population data were not encountered and therefore not accounted for. For example, there was not a lot of bus travel or subway travel in the used Semantic Google Location History data meaning that it was not possible to find specific errors for these activity types. The implication of

this would be that errors in these activities will be less easily found, leading to less valid data on the use of these transport types. The common requirements are checked on all the activity types to account for the errors that were found in the data.

5.4.2 Future work

Future studies on data quality should focus on obtaining more Google Semantic Location History data from people with different kinds of backgrounds and travel habits. This way other kinds of errors can be found and accounted for. It would also give a better view of how often these errors happen and if there is a pattern of why they happen.
An *activityType* that seems to be interesting to look at together with public transport is the car. Since the data concerns the author, the different trips are known, making it easier to recognize errors. When looking into the data, some tram travel was characterized as being car travel. If this is happening for tram travel, it might also be happening for bus travel. Thinking of ways to locate these errors in the data could give a more accurate result when extracting the data as right now some of the tram travel is not accounted for.
Another thing that could help with more accurate results is thinking of more requirements per error type. This way errors can be flagged for more than one source, making it more reliable.
When checking the public transport station or stop requirement, the radius is the same for all the activity types due to time restraints. To get more accurate data imputation it might be better to set a different radius for each activity type. This way the chances of finding the station or stop connected with the start or end location are greater than with a general radius resulting in more accurate data imputation.
To improve the distance requirement, the straight line distance between locations was used. This works if the route taken is also close to straight. It is possible to get better values to improve the wrongly logged distance with. Some of the routes contain a *waypointPath* which consists of multiple coordinates during the route. The straight line distance between these coordinates

can be used to get a more accurate distance of the route. The duration requirement is not specific for each activity type also due to time restraint. Right now, it had no influence on the data extraction. Making the maximum time specific for the different activity types could help with finding more errors, helping with more accurate results.

So, in summary, this project is a first step in identifying and improving quality issues in Google Semantic Location History but there are still multiple steps needed to make the data extraction as accurate to reality as possible.

Bibliography

[1] "Art. 15 gdpr – right of access by the data subject - general data protection regulation (gdpr)." (Mar. 28, 2018), [Online]. Available: https://gdpr-info.eu/art-15-gdpr/ (visited on 05/01/2023).

[2] "Art. 20 gdpr – right to data portability - general data protection regulation (gdpr)." (Mar. 28, 2018), [Online]. Available: https://gdpr-info.eu/art-20-gdpr/ (visited on 05/01/2023).

[3] L. Boeschoten, R. Voorvaart, C. Kaandorp, R. van den Goorbergh, and M. de Vos. "Automatic de-identification of data download packages." (May 4, 2021), [Online]. Available: http://arxiv.org/abs/2105.02175.

[4] I. I. Van Driel, A. Giachanou, J. L. Pouwels, L. Boeschoten, I. Beyens, and P. M. Valkenburg, "Promises and pitfalls of social media data donations," *Communication Methods and Measures*, vol. 16, no. 4, pp. 266–282, Sep. 12, 2022. DOI: 10.1080/19312458.2022.2109608. [Online]. Available: https://doi.org/10.1080/19312458.2022.2109608.

[5] L. Boeschoten, J. Ausloos, J. Möller, T. Araujo, and D. L. Oberski, "A framework for privacy preserving digital trace data collection through data donation," *Computational communication research*, vol. 4, no. 2, pp. 388–423, Oct. 1, 2022. DOI: 10.5117/ccr2022.2.002.boes. [Online]. Available: https://doi.org/10.5117/ccr2022.2.002.boes.

[6] A. Skatova and J. Goulding, "Psychology of personal data donation," *PLOS ONE*, vol. 14, no. 11, e0224240, Nov. 20, 2019. DOI: 10.1371/journal.pone.0224240. [Online]. Available: https://doi.org/10.1371/journal.pone.0224240.

[7] T. Araujo, J. Ausloos, W. Van Atteveldt, *et al.*, "Osd2f: An open-source data donation framework," *Computational communication research*, vol. 4, no. 2, pp. 372–387, Oct. 1, 2022. DOI: 10.5117/ccr2022.2.001.arau. [Online]. Available: https://doi.org/10.5117/ccr2022.2.001.arau.

[8] L. Boeschoten, A. M. Mendrik, E. Van Der Veen, *et al.*, "Privacy-preserving local analysis of digital trace data: A proof-of-concept," *Patterns*, vol. 3, no. 3, p. 100 444, Mar. 1, 2022. DOI: 10.1016/j.patter.2022.100444. [Online]. Available: https://doi.org/10.1016/j.patter.2022.100444.

[9] M. G. Moncayo-Unda, M. Van Droogenbroeck, I. Saadi, and M. Cools, "An anonymised longitudinal gps location dataset to understand changes in activity-travel behaviour between pre- and post-covid periods," *Data in Brief*, vol. 45, p. 108 776, Nov. 1, 2022. DOI: 10.

1016/j.dib.2022.108776. [Online]. Available: https://doi.org/10.1016/j.dib.2022.108776.

[10] "How google uses location information." (), [Online]. Available: https://policies.google.com/technologies/location-data?hl=en-US (visited on 05/05/2023).

[11] X. Yu, A. L. Stuart, Y. Liu, *et al.*, "On the accuracy and potential of google maps location history data to characterize individual mobility for air pollution health studies," *Environmental Pollution*, vol. 252, pp. 924–930, Sep. 1, 2019. DOI: 10.1016/j.envpol.2019.05.081. [Online]. Available: https://doi.org/10.1016/j.envpol.2019.05.081.

[12] "Sensors overview." (), [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_overview (visited on 05/09/2023).

[13] "Activityrecognitionclient | google play services." (), [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionClient (visited on 05/09/2023).

[14] "Activity recognition api." (), [Online]. Available: https://developers.google.com/location-context/activity-recognition (visited on 05/09/2023).

[15] "Motion sensors." (), [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_motion (visited on 05/19/2023).

[16] A. C. Rodriguez, C. C. J. M. Tiberius, R. Van Bree, and Z. Geradts, "Google timeline accuracy assessment and error prediction," *Forensic Sciences Research*, vol. 3, no. 3, pp. 240–255, Oct. 2018. DOI: 10.1080/20961790.2018.1509187. [Online]. Available: https://doi.org/10.1080/20961790.2018.1509187.

[17] OpenStreetMap contributors. "Planet dump retrieved from https://planet.osm.org." (2017), [Online]. Available: https://www.openstreetmap.org (visited on 05/22/2023).

[18] "How to download your google data." (), [Online]. Available: https://support.google.com/accounts/answer/3024190?hl=en (visited on 05/04/2023).

[19] Bergillos and Kirchhoff. "Downloading the data." (), [Online]. Available: https://locationhistoryformat.com/guides/downloading/ (visited on 05/04/2023).

[20] Bergillos and Kirchhoff. "General structure - location history format." (), [Online]. Available: https://locationhistoryformat.com/guides/general-structure/ (visited on 05/05/2023).

[21] C. Bergillos and J. Kirchhoff. "Semantic location history - location history format." (), [Online]. Available: https://locationhistoryformat.com/reference/semantic/ (visited on 05/19/2023).

[22] P. Runeson, "A survey of unit testing practices," *IEEE Software*, vol. 23, no. 4, pp. 22–29, Jul. 1, 2006. DOI: 10.1109/ms.2006.91. [Online]. Available: https://doi.org/10.1109/ms.2006.91.

[23] G. P. Sarma, T. B. Jacobs, M. Watts, S. V. Ghayoomie, S. M. Larson, and R. Gerkin, "Unit testing, model validation, and biological simulation," *F1000Research*, vol. 5, p. 1946, Aug. 10, 2016. DOI: 10.12688/f1000research.9315.1. [Online]. Available: https://doi.org/10.12688/f1000research.9315.1.

[24] "Distance matrix api request and response." (), [Online]. Available: https://developers.google.com/maps/documentation/distance-matrix/distance-matrix (visited on 06/03/2023).

[25] "Getting directions through the directions api." (), [Online]. Available: https://developers.google.com/maps/documentation/directions/get-directions (visited on 06/03/2023).

[26] C. Bergillos and J. Kirchhoff. "Settings.json - location history format." (), [Online]. Available: https://locationhistoryformat.com/reference/settings/ (visited on 06/03/2023).

[27] "Iso 8601 – effectively communicate dates and times internationally." (Nov. 29, 2022), [Online]. Available: https://www.ionos.com/digitalguide/websites/web-development/iso-8601/ (visited on 06/03/2023).

[28] QGIS Development Team, *QGIS Geographic Information System*, QGIS Association, 2023. [Online]. Available: https://www.qgis.org.

[29] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[30] K. Jordahl, J. V. den Bossche, M. Fleischmann, *et al.*, *Geopandas/geopandas: V0.8.1*, version v0.8.1, Jul. 2020. DOI: 10.5281/zenodo.3946761. [Online]. Available: https://doi.org/10.5281/zenodo.3946761.

[31] S. Gillies, C. van der Wel, J. V. den Bossche, M. W. Taves, J. Arnott, B. C. Ward, *et al.*, version 2.0.1, Jan. 2023. DOI: 10.5281/zenodo.5597138.

[32] NOS, "Waarom schieten kilometers spoor maar met 140 onder je door?," Apr. 2018. [Online]. Available: https://nos.nl/op3/artikel/2227785-waarom-schieten-kilometers-spoor-maar-met-140-onder-je-door.

[33] "Buses - standard speed limits in europe - traffic regulations in europe - v.en, travel - studentnews.eu." (), [Online]. Available: https://trip.studentnews.eu/s/4086/77069-Buses-standard-speed-limits-in-Europe.htm (visited on 05/26/2023).

[34] AmsTips. "Amsterdam trams | travel by public transport | gvb tram lines." (Feb. 3, 2023), [Online]. Available: https://www.amsterdamtips.com/amsterdam-trams (visited on 05/12/2023).

[35] "Railalert." (), [Online]. Available: https://www.railalert.nl/ (visited on 05/12/2023).

[36] "How fast do commercial planes fly?" (), [Online]. Available: https://epicflightacademy.com/flight-school-faq/how-fast-do-commercial-planes-fly/ (visited on 05/12/2023).

[37] "Treinen van castricum naar utrecht." (), [Online]. Available: https://www.omio.nl/treinen/castricum/utrecht (visited on 05/26/2023).

[38] "Trein ns heemskerk - amsterdam." (), [Online]. Available: https://www.thetrainline.com/nl/treintijden/heemskerk-naar-amsterdam (visited on 05/26/2023).

[39] "Tarieven l prijs van treinkaartjes en abonnementen | ns." (), [Online]. Available: https://www.ns.nl/klantenservice/betalen/tarieven-consumenten.html (visited on 06/03/2023).

[40] M. Sabir, M. J. Koetse, J. Van Ommeren, and P. Rietveld, "Weather and travel time of public transport trips," *ResearchGate*, Jan. 1, 2010. [Online]. Available: https://www.researchgate.net/publication/46434571_Weather_and_Travel_Time_of_Public_Transport_Trips.