Modelling and visualizing macroscopic movements

by

Ananya Chowdhury
Matriculation Number TODO

Piotr Mrówczyński
Matriculation Number 387521

Gabriel Vilén
Matriculation Number TODO

A project documentation submitted to

Technische Universität Berlin
School IV - Electrical Engineering and Computer Science
Department of Telecommunication Systems
Service-centric Networking

June 27, 2017

Supervised by:
Bianca Lüders
Friedhelm Victor
Prof. Dr. Axel Küpper

Service-centric NETWORKING

Technische Universität Berlin

1 Introduction

1.1 Macroscopic City Movements - problem definition

Here we should describe what problem project is about to solve. In the 1st term presentation I mentioned that we want to answer two questions:

Given a location, where and in what proportion people move to another location - Movement Graph?
Where and how long do people stay at certain location? - but we need to be more precise here.

1.2 Localization-based services and positioning technologies

TODO: we need here more details about what our Sudanian guy were supposed to research

1.2.1 Positioning technologies

TODO: http://geoawesomeness.com/knowledge-base/location-based-services/location-based-services-technologies/

1.2.2 Localization-based services

We should describe what are proactive and reactive LBS- https://www.move-forward.com/gps-everywhere-location-based-services/. Here we should describe how the data is obtained, and emphasise the fact that location updates are based on MOVEMENTS, so each next point is triggered by some mobile phone movement. Point updates are not triggered while being stationary.

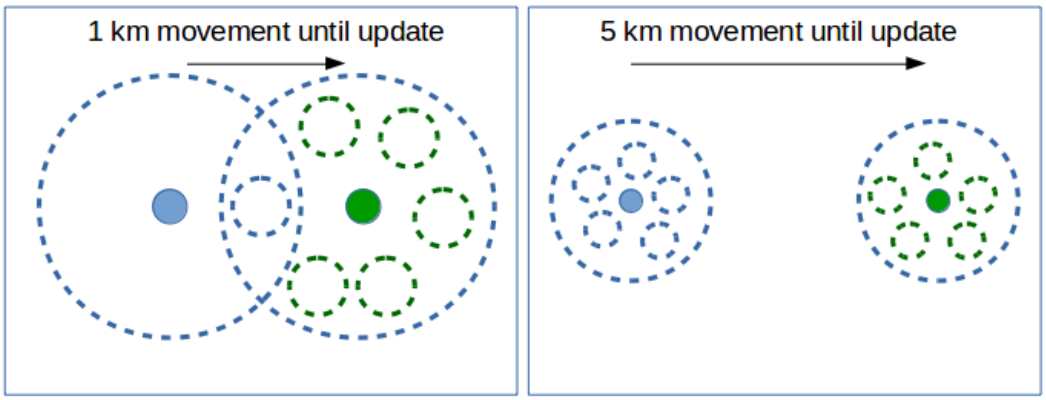

Figure 1.1: Continuous and Discontinuous location update after movement to section lacking area details

Because of the characteristics of GPS and other localization methods, in so called urban canyons and the like, signal loss with re–acquisition times of 45 seconds to five minutes can be observed. This could lead to discontinuous recording of the location - Figure 1.1

2 Related Work

2.1 Human Mobility within the city

According to research [**HumanMobility1**][**HumanMobility2**][**HumanMobility3**], the average walking speed in the city for man aged 40 is 1.4 m/s (5 km/h), and slowest one is for elderly (with walking impairment) below 1 m/s (3.6 km/h).

TODO: elaborate a bit more

2.2 Approaches to stop detection for localization services

2.2.1 Stop Detection analyzing repetitive appearance at location

2.2.2 Stop Detection based on continuous localization

2.2.3 Stop Detection based on mobility index

In the paper [**MobilityIndexGIS**], movement of mobile devices between cells (handovers) is considered. To detect the periods of slow movements of stops at the specific location, parameter called Mobility Index is considered.

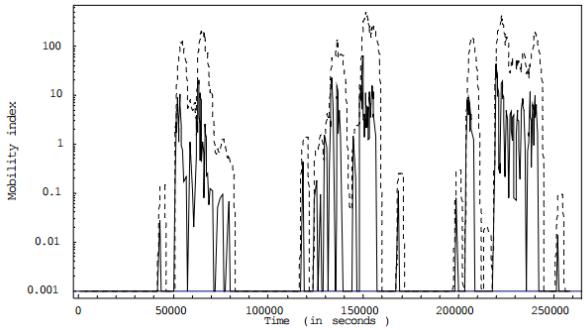

Figure 2.1: Mobility Index. Source: [**MobilityIndexGIS**]

In a populated area, with many GSM cells, the mobile terminal can change from one cell to another within seconds or after several minutes. Given a set of consecutive records, the mobility of a user can be estimated by calculating the Mobility Index over a pre-defined time period (sliding window). The Mobility Index is defined as the sum of the "distances" between each record and the previous ones, where the "distance" is the inverse of the time spent on each cell. If the value of MI is below certain threshold, it is assumed that the object stopped or moved

slowly. In the publication, sliding window of 10 minutes and a mobility index threshold value of 6 has been used for certain area. The obtained results were in agreement with actual movements during more than 90% of the time.

Rapid drop in mobility index represents a set of points (restricted area) in which user has been moving slowly or stopped. Slight decrease in mobility index means that user changed position from restricted area and is moving.

3 Concept and Design

3.1 Stop Location Detection

3.1.1 Stop Detection for proactive localization based services

Analyzing different approaches for stop detection based on localization data (ref. section 2.2), we decided to test and verify algorithms based on human behavior regarding mobility within the cities (ref. section 2.1) and reusing the concept of Mobility Index (ref. subsection 2.2.3)). Stop Location Detection Algorithm have to be able not only identify the time of start stop, but also the stop location.

The approaches also need to consider that because of the method of receiving data in proactive localization services, if human will enter a urban canyon, building or move underground his signal might need to be re-obtained after a movement to new area and this can take certain period of time, where points possibly can be spanned over mid-range/long distances. On the other hand, points received over a short distance might mean that person has been already moving with a good reception of signal.

3.1.2 Mobility index based approach

In related works about mobility index discussed in subsection 2.2.3, the approach to stop detection using mobility index is bound to the way the data is being sampled (handover from one GSM cell to another). The data is recorded whenever mobile device changes its GSM cell. In case of the approach discussed in this paper the situation is similar, however instead of GSM cells we have artificially created area around the location, and while new location is obtained at some distance after movement, new area might need to be updated and point is being recorded.

In a populated area, the user can move from one area to another after several seconds or only after minutes. Thus, if within a certain period of time (called mobility index time window) the user stays in the same area or changes the areas with small pace (mobility index is below certain threshold) it can be assumed that the user is immobile or approaching stop location and having stay somewhere within that time window (restricted area).

3.1.3 Human Behavior based approach

This approach targets the condition for movement between two points, which has to be satisfied, so that it can be considered as a stay within the certain distance.

The section 2.1 discusses, that in average, walking human is traveling with the speed of over 3km/h up to 5km/h. However for movement on long distances, user will most likely use city or private transport, and thus his speed can vary much in that time/distance period.

This approach states, that if movement between two points has been recorded in very short distance - as described in subsection 3.1.1, and movement of speed is below a certain threshold, it might mean that user has stopped or has been moving very slowly within a restricted area, which is called possible stop for that location between these points.

For movements between points over mid/long distances recorded, user might have stopped at first location, moved with different speeds between points and continuing movement (proactive localization is used). As an example, movement over 6km, might require up to 10 minutes with transportation. If the user stopped at previous location for 20 minutes, thus, total movement between point took 30 minutes, which in average is around 12 km/h. Universal speed threshold is not able to detect stops in these kind of examples, since with travels over higher speeds with stop at previous location would result in relatively high average speed.

3.2 Clustering of Stops

Clustering can be described as "*the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters)*" **[clustering]**.

After our stop detection algorithm has detected the proper stops we want to perform some analysis on these. Our goal is to get the overall movement patterns of the population of a certain area, which means we are interested in **macroscopic** movements and not microscopic (individual) movement. If we would consider every individual our analysis algorithms would get fed with a lot of noise (outliers) which would make our data mining and prediction difficult and inaccurate. Meaning, *Macroscopic Movements* are not interesting in tracking individuals but rather seeing the big picture.

Once our clustering algorithm has been applied, and we have the overall picture, we can see which stops are most popular, where people are moving from them, and how long they stay at a certain location. One could then perform some interesting graph analysis on these locations, answering questions such as "what is the probability that a person moves from point A to B?", or "what is the average stop time at location A?".

To achieve these clusters, which represents our interesting stop locations, we need to apply some clustering algorithm. There are different types of clustering (such as centroid, distribution, or density-based), and different algorithms with parameters that can be applied (such as K-means, DBScan, or OPTICS). However, they all work to achieve the same goal of structuring (clustering, grouping) similar objects and neglecting outliers from these groups. The algorithm and parameters are application sensitive and our choice will be handled in the *evaluation* section.

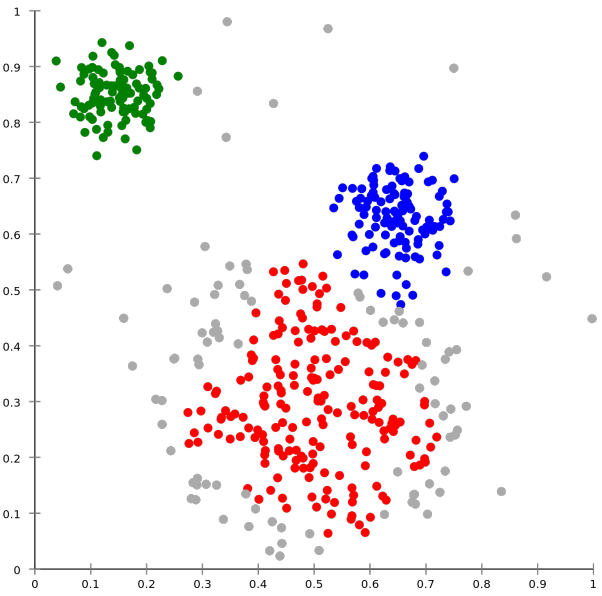

Figure 3.1: Clustering applied to a dataset.

3.3 City Movements Graph

4 Implementation

4.1 Scripts for data analysis - usage

TODO:

4.2 Stop Location Identification - batch processing in Apache Spark

TODO: final implementation and algorithm should be discussed here

4.2.1 Calculation of the index for verification of mobility index approach - python implementation

There are two approaches to calculate mobility index, having equally spaced windows in time, or sliding window.

In the first approach, for a single user, minimum and maximum timestamp of sorted sequence is being found. Having value of MobilityIndexWindow (e.g. 30 minutes), function is creating equaly spaced bins of that value within maximum and minimum timestamps. Each point is being visited and is assigned to the proper timestamp bin according to its own timestamp and duration to the previous point. However, danger of this approach is that points and their duration are randomly placed in bins and could lead to represent mobility of fixed window, instead of mobility history for points.

In the second approach, for a single user, each point is being visited and temporarly marked as "current position". For each current position, all points which are distant in time by MobilityIndexWindow in the past, are assigned to the sliding window. This approach represents mobility history in the past time window for each point.

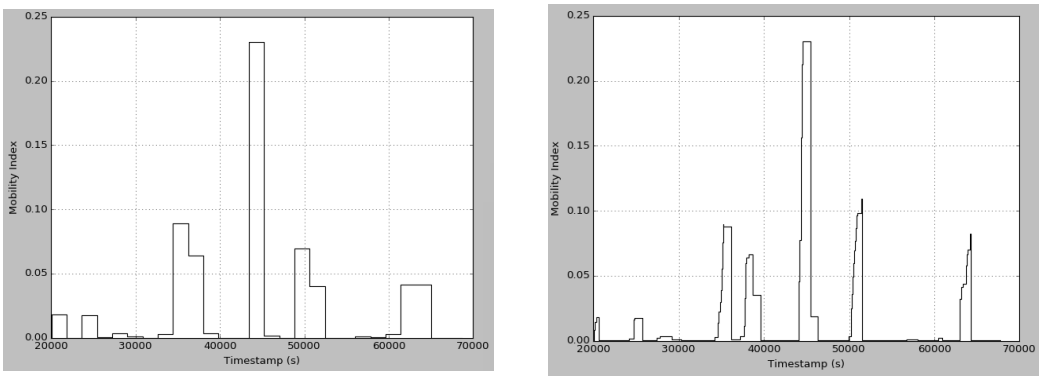

Figure 4.1: Mobility Index calculation using equally spaced windows in time (left) and using sliding window over time (right). Time Window of 30 minutes using the same data set for one unique user

At the end, mobility index, as defined in subsection 2.2.3, is being calculated over the assigned values - Figure 4.1.

4.2.2 Algorithm for stop detection

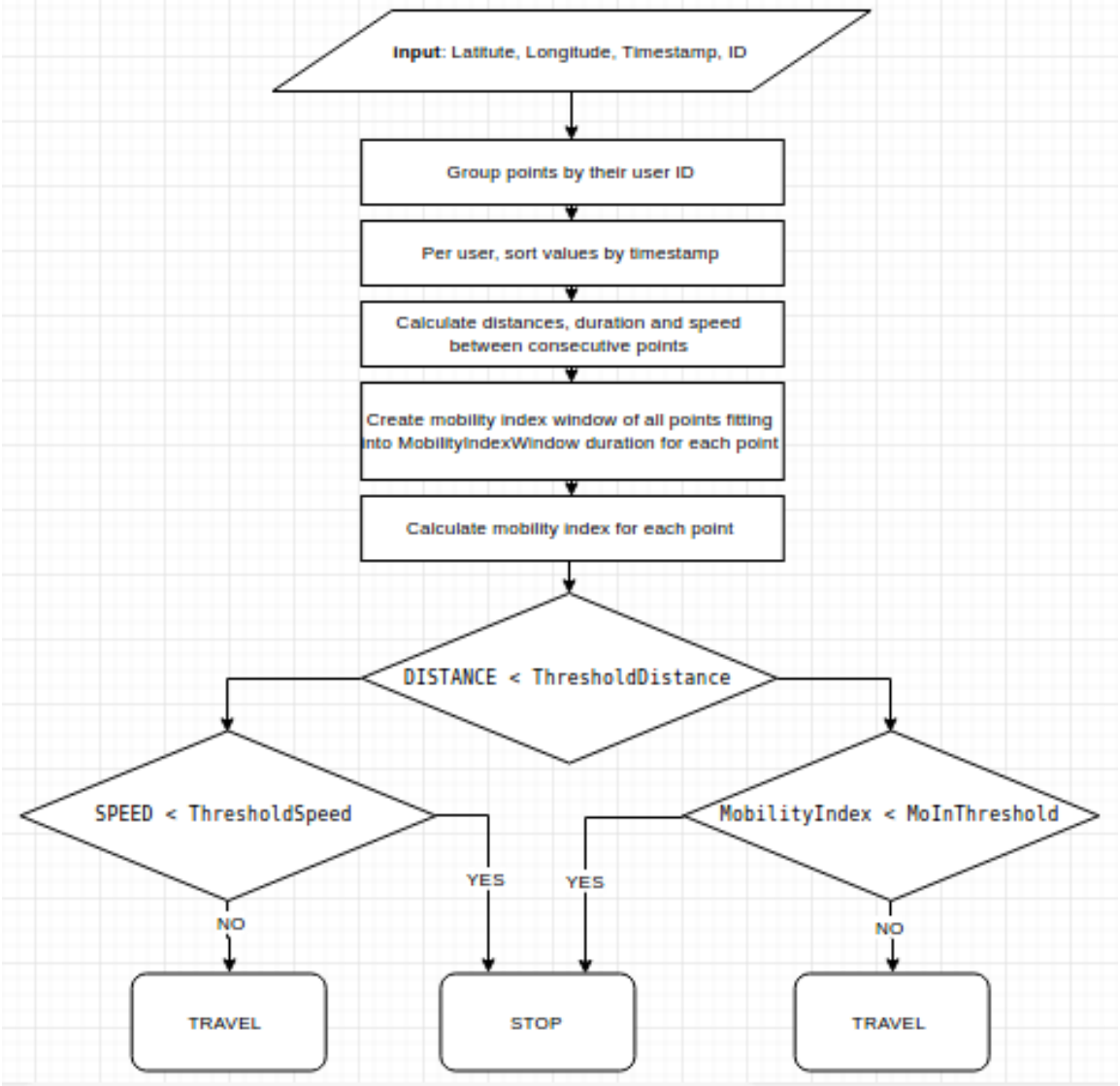

Figure 4.2: Stop detection algorithm for proactive localization services.

4.3 Clustering algorithm

For clustering we started off with DBScan, described in **??**. The difficulty with DBScan is figuring out the two parameters, the minimum points per cluster (*minPts*) and the radius of the clusters (*eps*).

4.3.1 DBSCAN on Spark

We wanted a scalable implementation of DBSCAN that could handle large data sets, preferable in parallel. We decided to use Apache Spark, a scalable engine for large-scale data processing [**spark**]. For this we used an implementation by Irving Cordova, based upon an algorithm called MR-DBSCAN built for the MapReduce framework [**dbscan_on_spark**]. The implemen-

tation of DBScan runs in parallel by splitting the data space into boxes, using the number of boxes as a cost estimator for the algorithm. Each box then grows to include one *eps* in it. After each box and its points has been determined the traditional DBScan algorithm is run on the points in each box. Finally it examines the intersection points betweeen boxes and merges the result together [**vis_dbscan_on_spark**]. The figures below visualises the execution steps.

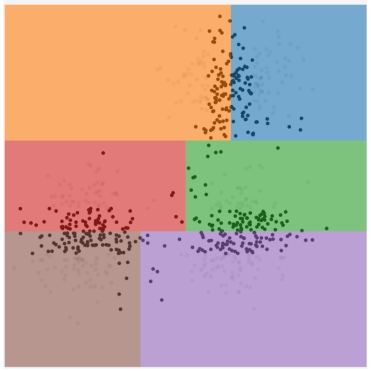

Figure 4.3: Step 1. DBSCAN on Spark assign the data space into boxes.

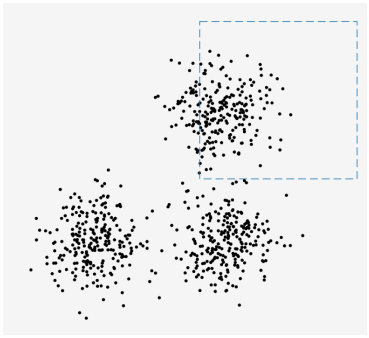

Figure 4.4: Step 2. Each box grows to include the points that are within one *eps* of it.

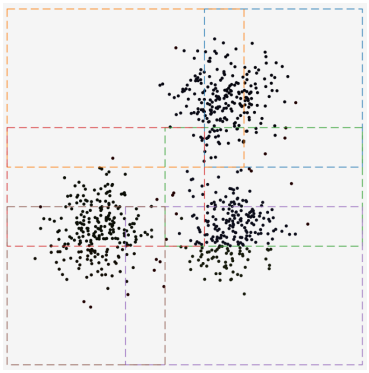

Figure 4.5: Step 3. Traditional DBScan algorithm is applied in parallel for each box. Each different color represents a different cluster.

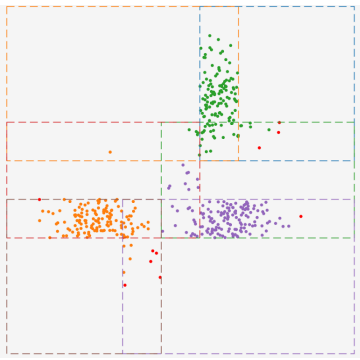

Figure 4.6: Step 4. After DBSCAN is done, all points witihin the borders of two clusters are examnied. If they are part of a cluster within two boxes they are merged to one cluster.

Figure 4.7: Step 5. Finally all remaning points are labeled with the global identified cluster and we are done. (red points are noise points).

5 Evaluation

5.1 Location data characteristics for Berlin area

5.1.1 General overview

Movement triggered data from mobile devices are spanned across the Germany as shown on Figure 5.1.

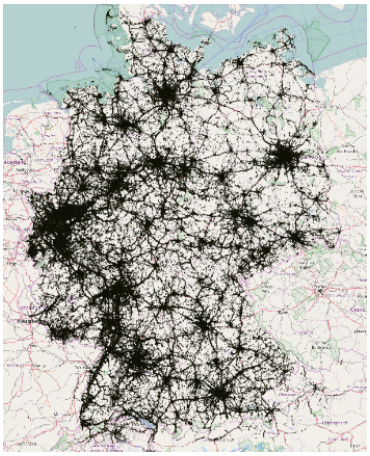

Figure 5.1: Visualisation of data points location in the geographical map of Germany

It points out, that most of the detected movements from one point to another are mostly gathered on the highways and inside the cities.

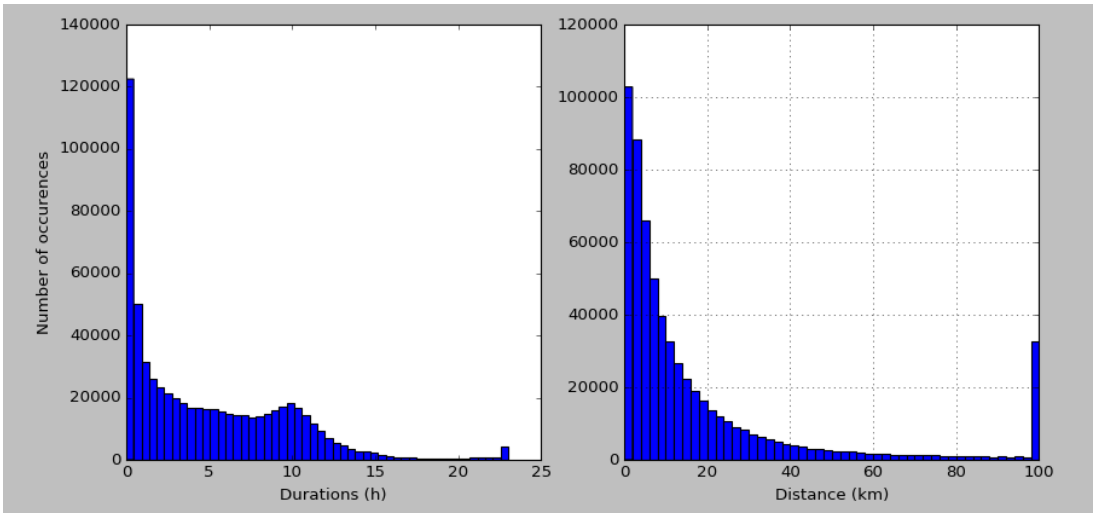

Figure 5.2: Histograms of durations and distances between consecutive points under and cumulatively over 100 km for the area of Germany

The given data batch has been gathered in the interval of 24h starting from 3am in the morning German time. It consists of over 675000 data points, belonging to 47300 unique mobile devices. Figure 5.2 shows, that for unique mobile devices, the durations between consecutive points in time are most frequent within 30 minutes and most of the data is in the interval 1-10h. Regarding distances, consecutive points are most frequent below 2 km and most data is in the interval 0-10km. Big chunk of distances is also over 100km.

Figure 5.3: Heatmap of data point densities within the area of Berlin

Figure 5.3 shows the heatmap of data points for Berlin area. We can see that most of the movements detected are on major train/tram/metro stations, highways and major living areas.

5.1.2 Data analysis

Statistics performed using tool written in Python (https://github.com/mrow4a/macroscopic-movements-algorithm-prototypes) shows, that for users at least once visiting Berlin, had in average 17 points gathered because of their movements in the period of 24h, harmonic mean of 4 points and maximum 100 of points.

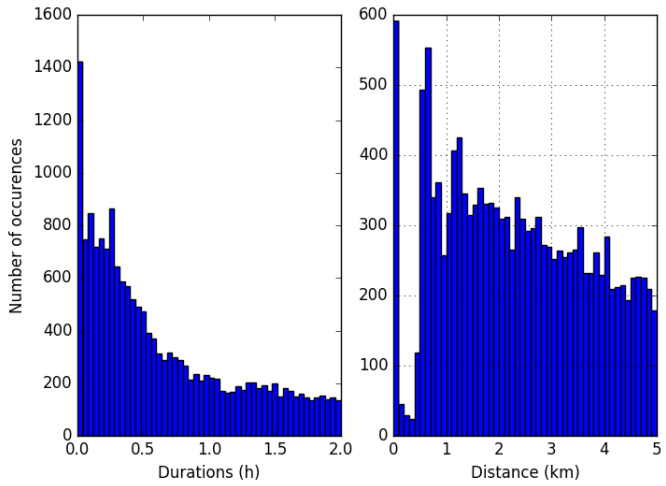

Figure 5.4: Histograms of durations and distances between consecutive points under 5 km and under durations of 2h for the area of Berlin

Furthermore, majority of durations between points is within interval of 30 minutes. For distances between points, there is significant "jump" at 500-1000m and 1100-1500m, which might mean that at these intervals, continuous points been gathered, and distances over that values might be discontinuous (ref. Figure 1.1). It is due to the fact that these distance intervals are most frequent and that might suggest that this is an average "update" distance for the moving mobile devices, as described in section 1.2. Less frequent values might suggest that updates were obtained with lower accuracy (e.g. by presence inside the building, metro line or simply mobile device lag in obtaining its location) and are result of discontinuity between consecutive updates.

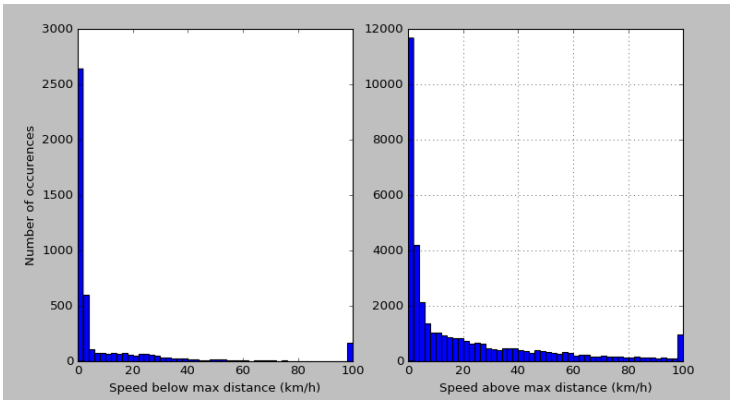

Figure 5.5: Histogram of speed between points with corresponding distance above or below 1.5 km

Figure 5.5 shows that considering the registered distances within range of 1500m, the wast majority of points are between 0-2 km/h, and also significant interval at 2-6 km/h. The rest of the values is sparse distributed in interval 6-100+ km/h. In case of registered point above range of 1500m, we observe that indeed most frequent occurrence is at interval of 0-4 km/h, however most are sparse distributed above 4 km/h, with most points being in interval of 4-20

km/h.

5.2 Pre-filtering of anomalies

Figure 5.4 shows that there is a fraction of points which duration or distance rapidly changed, thus resulting in very high speeds between the points. Points having speeds above 300 km/h and distances above 100km can be considered as flights, however the ones below 100km are anomalies. Figure 5.6 shows that number of flights in Germany has not been significant (around 100) compared to number of speed anomalies (over 25000).

Furthermore, due to the methodology of obtaining points (movement triggered), there are some minimum distances and durations at which points can be collected, and values not matching stop or travel expectation according to gathered statistics, have to be filtered and considered as duplicates.

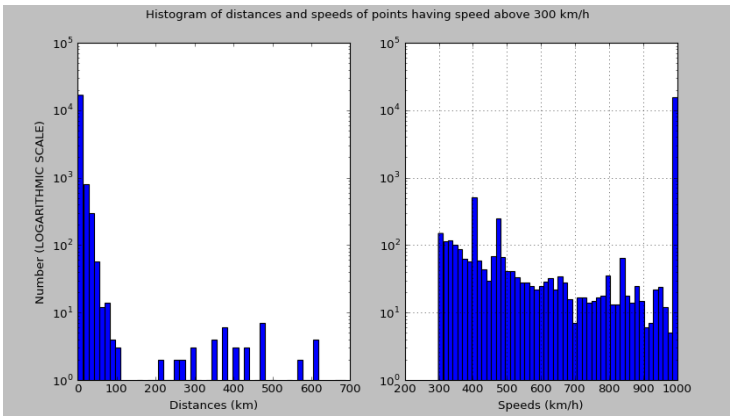

Figure 5.6: Speeds and distances histogram of points having speeds above certain threshold

Thus, the following predicates for filtered values have been set:

Jumps Points with speed over 83 m/s [300 km/h] and at the same time distance below 100 km/h

Duplicates Points with distance below 100m and at the same time duration below 100s

5.3 Evaluation of stop detection algorithms

5.3.1 Mobility Index approach

Mobility Index analysis is performed in order to detect single movements or series of movements to be classified as stop or slow movement over the restricted area.

By the nature of mobility index, more values in the window, higher the mobility index is. Thus, if only one value is in the window, one can apply "stop duration" threshold, however if windows contains more values, we require higher threshold to be able to fit sum of inverses of the movements.

As an example, having window of 20 minutes and threshold of 1/(10 minutes), if we fit into window single point with duration of 15 minutes, this value will be classified as a stop. However, if we obtain 2 values, 5 and 10 minutes respectively (1/5 + 1/10), our threshold requires to be higher in order to include slow movements in the window and classify this 2 values as stop in restricted area.

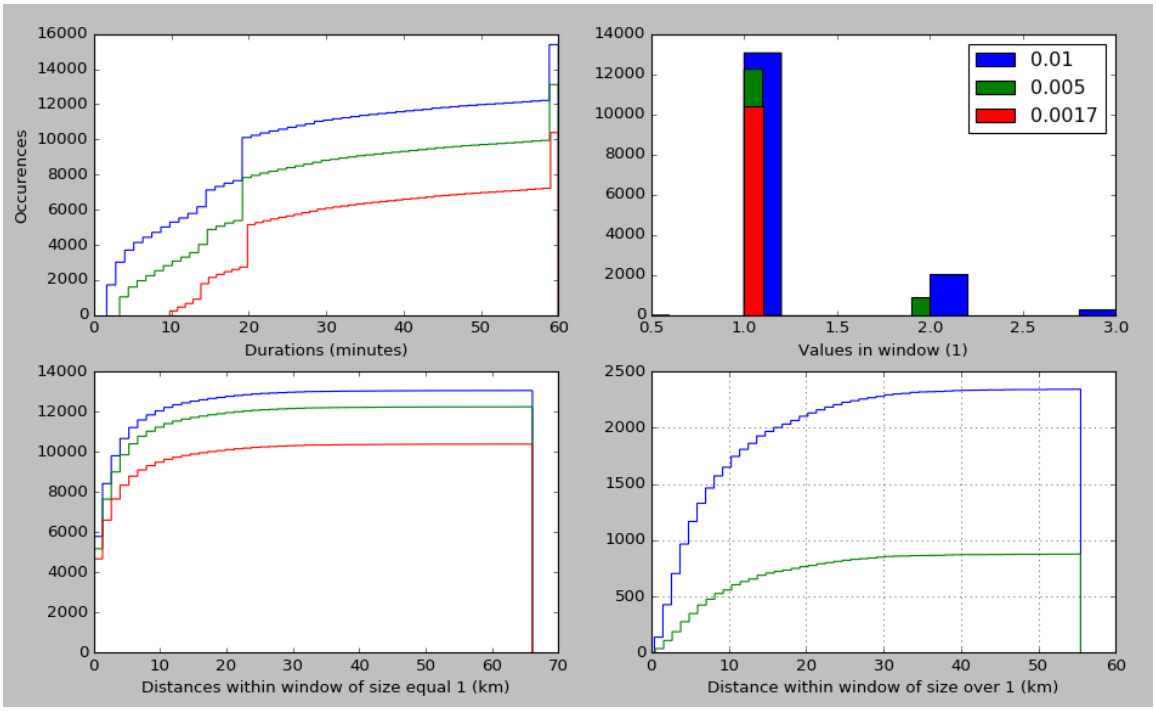

Figure 5.7: Analysis of detected stops for varying MobilityIndexThreshold for time window of 20 minutes. Cumulative distribution of inverted mobility index (top left). Histogram of window sizes (top right). Cumulative distributions of total distances between points in the window of size 1 (bottom left) and above 1 (bottom right).

Figure 5.7 shows the analysis of the stops detected using mobility index, using sliding time window approach.

Cumulative distribution of inverted mobility index (top left) shows that for window of 20 minutes, increasing MobilityIndexThreshold from 0.0017 (1/10 minutes) to 0.01 (1/2minutes), sum of inversions of durations (MobilityIndex) for points within the window, increases total number of stops from 10000 to 15000, where the increase (5000 points) is caused by points in the range 2-10 minutes, which is expected result increasing the threshold. The rapid jump at 20 minutes is caused by the fact that at that threshold some inverted mobility indexes were result of summing window of more then one value.

Histogram of sizes of the window (number of points within a 20 minutes window - top right) shows, that with parameter 0.0017, all the points are contained within the windows having size of 1 point. Increasing the threshold to 0.005 (1/4 minutes), it resulted in 95 percent of points being in windows of size 1 and 5 percent in windows of size 2 (2 points in the window - re-

stricted area of stop). Doubling the threshold to 0.01 resulted in 75 percent being in single value window, and 25 percent in over 2 points per window.

The graphs in the bottom of Figure 5.7 show what are the distances covered in the windows having 1 or more points. It occurs, that in case of 1 point in the window (which means that this point is a stop), distances between points range from 0 to 60km, and this is expected. However, in case of 2 or more points per window (thus stops within a restricted area bound by points), it occurs that only small fraction of values has total distance covered less then 2km, and major fraction of points represent long distance covered.

The above analysis shows, that mobility index cannot be effectively used to detect group of points which might be considered as a stop and only very small fraction of windows having more points then 1 is considered as a stop within a small restricted area. The rest is spanned across bigger distances. Thus, small values of mobility index window and threshold might start detecting trips over longer distance as a stop and providing erroneous results.

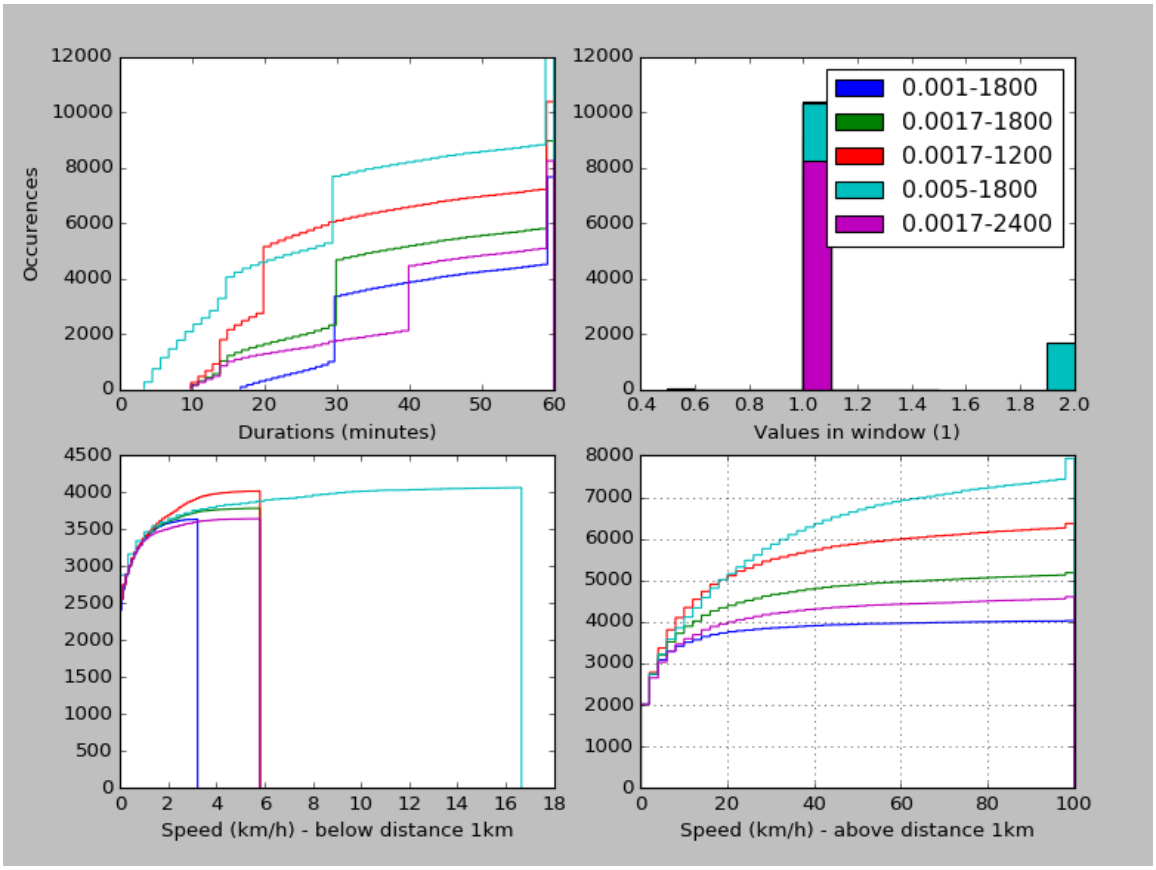

Figure 5.8: Analysis of detected stops for varying MobilityIndexThreshold-MobilityIndexWindow(seconds), correspondingly as in the label of measurement. Cumulative distribution of duration for detected stops (top left). Histogram of window sizes (top right). Cumulative distributions of speed between point for distance below 1km (bottom left) and above 1km (bottom right).

Figure 5.8 shows the analysis of the stops detected using mobility index threshold, using

sliding time window approach, however in this case it is used to identify mobility of a user considering only last recorded point in the window to decide if the movement can be a stop considering its past mobility.

This approach tries to tackle the problem discussed in subsection 3.1.3, where it has been said that over longer distances e.g. 2-10 km between recorded points, detection using speed is difficult since stop for a short time and movement with high speed will produce very high average speed in that time period.

Figure 5.8 gives cumulative histograms considering different pair of parameters (MobilityIndexThreshold-MobilityIndexWindow) considered in producing stop locations. Cumulative distribution of duration at detected stops (top left) shows that considering the same MobilityIndexThreshold (0.0017-1200s, 0.0017-1800s, 0.0017-2400s) and increasing MobilityIndexWindow, causes less stops to be detected and higher average duration between points detected. Considering the same MobilityIndexWindow (0.001-1800s, 0.0017-1800s, 0.005-1800s) and increasing MobilityIndexThreshold, number of stops increases, and this increase is bound only to values below MobilityIndexWindow threshold. High MobilityIndexThreshold produces more stops with shorter duration of stop.

Cumulative distribution of speed for detected points show that algorithm with all tested parameters has been able to identify correctly stops over distance less then 1km and with very small speeds (long stay duration at the point). However, MobilityIndexThreshold or decreasing MobilityIndexWindow cause that average speed of detected points had higher speed over distances > 1km and these were not able to filter trips over long distances correctly. For thresholds 0.001-1800s, 0.0017-1800s and 0.0017-2400s higher speeds are being filtered and only longer stays are left, and the difference between performance of these parameters is a trade-off between an accuracy and number of detected stops.

Histogram of values in the window (top right) also confirms, that for 0.001-1800s, 0.0017-1800s and 0.0017-2400s thresholds, detected stops had window of 1 value, thus none of the windows having 2 or more values were not qualified for being a stop and thus being filtered out as high mobility trip. This behavior of algorithm allows to identify stays with simple duration threshold and at the same time filter out all multi-point trips using mobility index window.

5.3.2 Human Mobility approach for long/short distances

Speed/Distance analysis is performed in order to detect single movements as stop or slow movement. If the average speed between the points is below the certain threshold, it is assumed to be a stop.

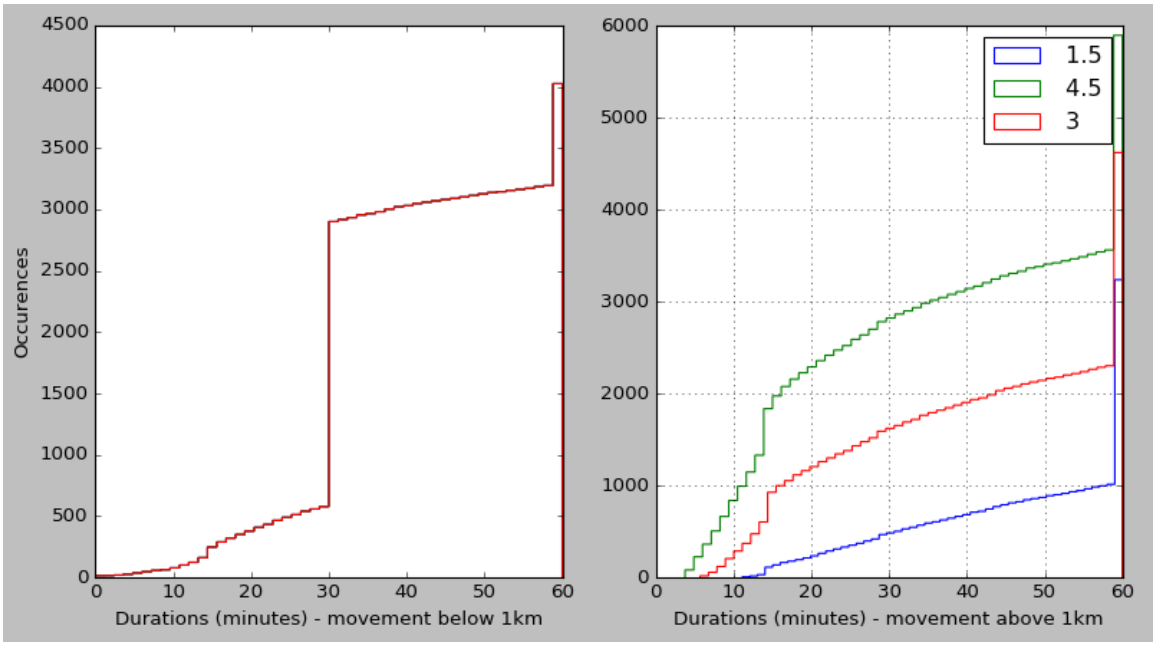

Figure 5.9: Speed/Distance detection with StopCertaintyMaxDistance, StopCertaintyMaxSpeed and TravelCertaintyMinSpeed thresholds. Cumulative distribution of duration for detected stops below and above distance StopCertaintyMaxDistance of 1km for different TravelCertaintyMinSpeed thresholds.

Figure 5.9 shows analysis of detected stops having TravelCertaintyMinSpeed varying and StopCertaintyMaxSpeed fixed. Increasing the speed threshold for points above 1km, we observe that number of points in interval 15 to 60+ minutes stays fixed for all parameters, however increasing threshold increases number of stops detected in interval 5-15 minutes diracticaly. It means, that with these thresholds, much more movements which has very low duration over the distance are classified as a stop and identification might be erroneous to mislead long distance trip with a stop.

Furthermore, having TravelCertaintyMinSpeed threshold of 1.5 m/s, we can see that speed threshold correctly identifies the stops, however the number of detected stops in the movements above 1km compared to mobility index approach is low (Figure 5.8)

On the other hand, for distances below 1km, speed detection has not only been able to identify longer stays at location, but also a short distance, very slow movements over the area, successfully.

5.3.3 Conclusions

Table 5.1: Comparison of two analyzed stop detection algorithms for proactive localization

Case	Mobility Index	Speed/Distance
Slow movements over small area or distance	Possible, but adjusted parameters will cause results to be erroneous for different detections	Very good
Stays at short distance	Good, but only for longer stays	Very good
Stays at long distance	Very good, but only in case of adjusted parameters which might be erroneous at short distances	Possible, but might be erroneous and stops maybe mislead with travel due to high speed averages in these cases

Table 5.1 shows comparison of two approaches to stop detection. It occurs, that for short distances it is more efficient to use speed/distance algorithm to be able not only capture longer stays at short distance, but also very slow movements in the limited area. For longer distances it is preferable to use mobility index to be able distinguishing travels from stops better using mobility index, which preserves mobility history for the certain past time window. MobilityIndexWindow becomes a duration threshold at which mobility window with multiple points are being accepted, and MobilityIndexThreshold is defining which multiple points or single points cannot be classified as a stop and be a series of movements/movement.

5.4 Clustering algorithm

In this section we will describe how we evaluated our dbscan algorithm and the choice of the *epsilon* and *minimum points per cluster* parameter.

5.4.1 DBSCAN

We visualised our results in QGIS, a powerful graph visualisation tool. Running the dbscan with different parameters resulted in different cluster sizes and number of clusters. We started off with the target area of Berlin. The *epsilon* parameter (radius of the cluster) was chosen by looking at the graininess (accuracy) of our given data set. The accuracy between points was 110 meters, this distance is refered to as *accuracy distance*. We took Alexanderplatz in Berlin as an example cluster (a popular train stop area in Berlin). We interpret one cluster as being one stop point of interest, and for this application want Alexanderplatz to be represented as one cluster. Setting the *epsilon* parameter to the *accuracy distance*, 110 meters, gave us good looking clusters whilst higher value of the parameter resulted in clusters being unnaturally large and distance below the *accuracy distance* resulted in only single-point clusters (points are stacked at the same location). Note that the *minPts*, minimum points per cluster, parameter is not taken into *consideration* here (it is set arbitary and only determines the number of clusters, while we here are interested in the *size* of the clusters). See figures below.

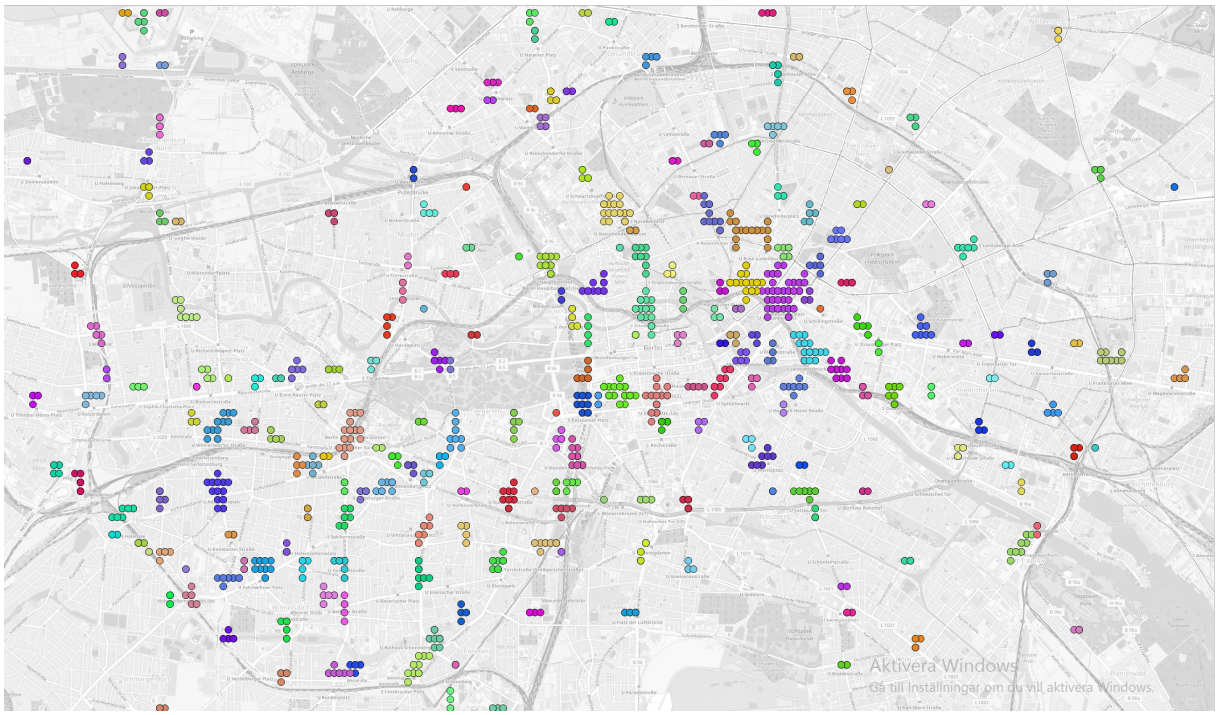

Figure 5.10: Clusters with good parameters *epsilon* = 110 meter, *minPts* = 5. Alexanderplatz (pink) has 85 data points in it.

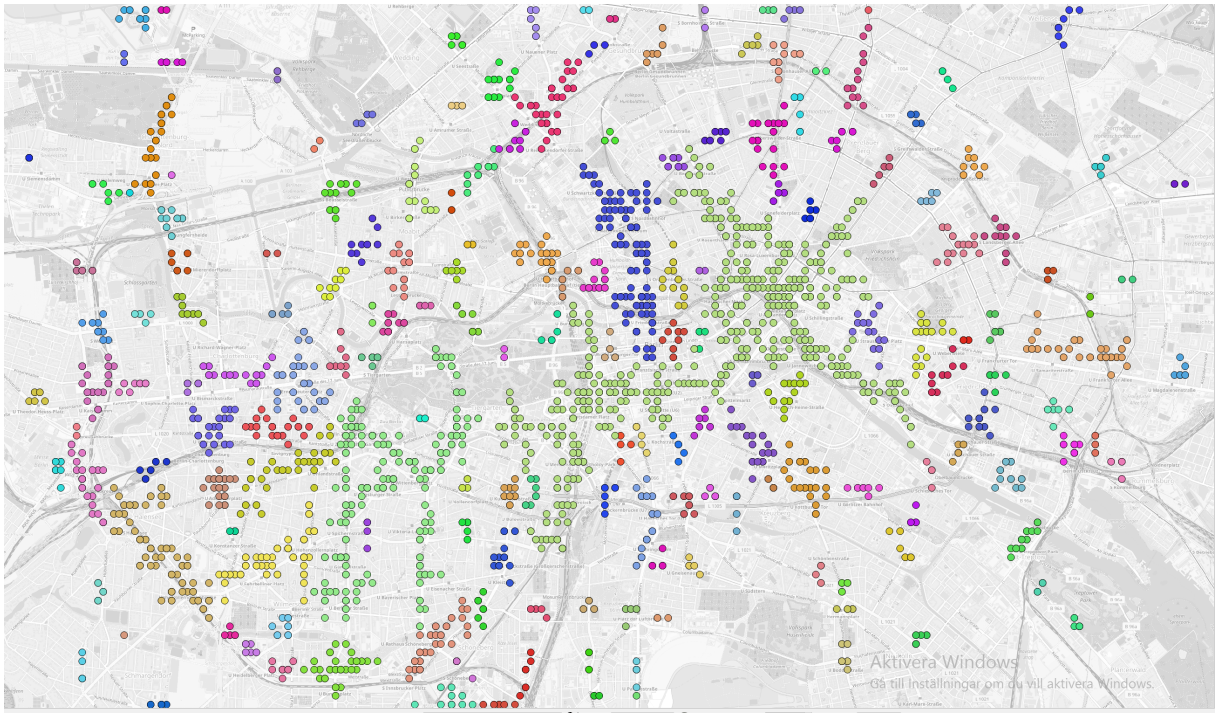

Figure 5.11: Example of a clusters with bad parameters, here *epsilon* = 220 meter is too large, *minPts* = 5. "Alexanderplatz" (light green) has 748 data points in it and stretches over the whole centre (Mitte) of Berlin.

Appendices

Appendix 1

Figure 1: Visualization of distances, speed and mobility indexes for example from Figure 2 with identified stop candidates

Figure 2: Movement example for one user in the period of 24h, with recorded locations due to the movement and annotated timestamps for detected stop candidates