Data Mining

INFO-H423

Hack my ride

Yahya BAKKALI

Amirmohammad FALLAHI

Maxime HAUWAERT

Damien NIZERY

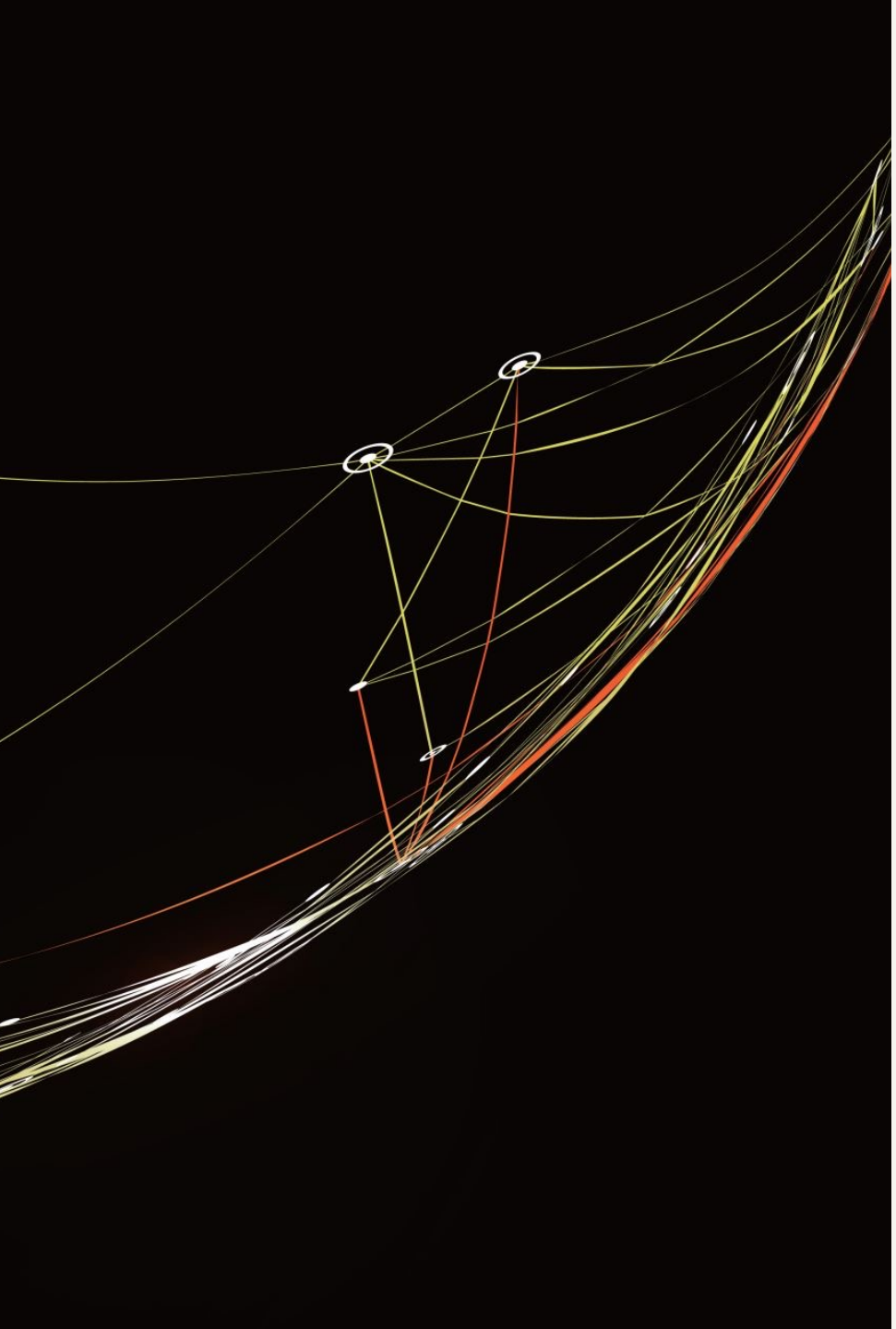
December 2021

Introduction

- Project goal
  - Analysis of STIB open-source data
- Study and analyses of data
  - From 03/09/2021 to 23/09/2021
- Presentation plan
  - Data preparation
  - Speed analysis
  - Delays analysis
  - Prediction on arrival time
  - GPS track inference
  - Reachability analysis

Data preparation

- Extracting
- Computing
- Cleaning



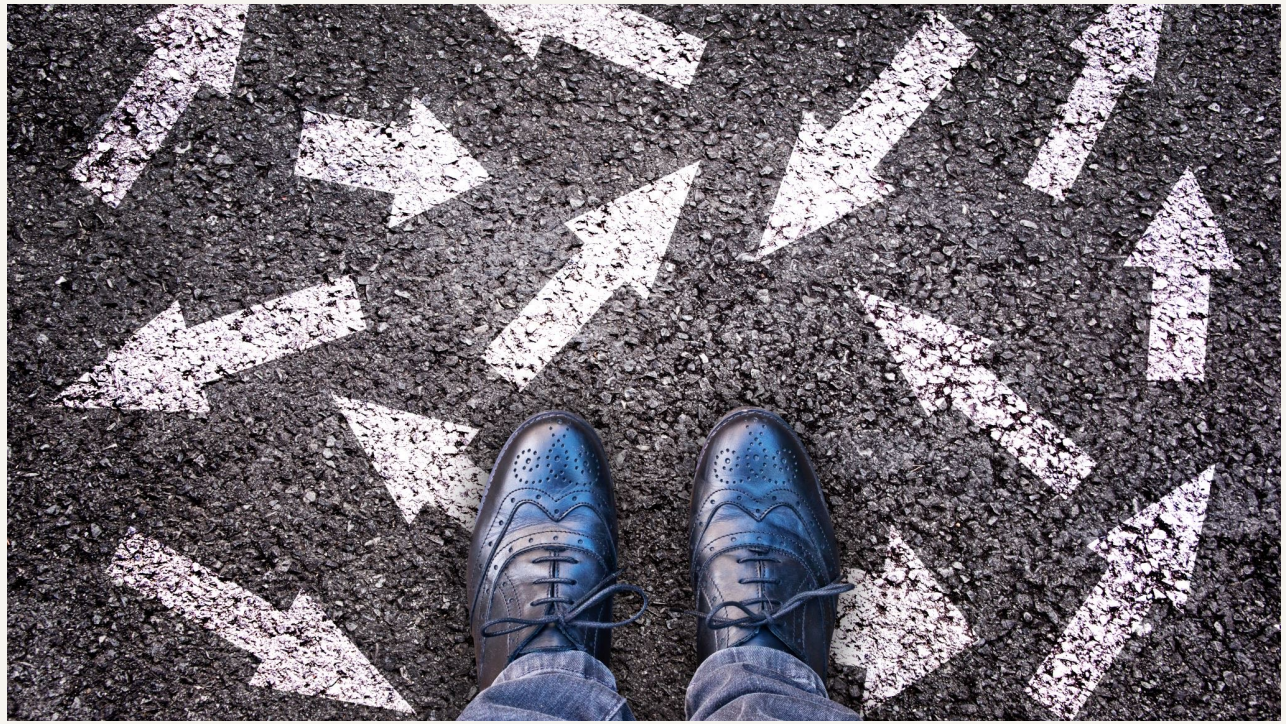
Lines information

- Lines information
- Stops
- Order
- Distance

Incoherent stop IDs

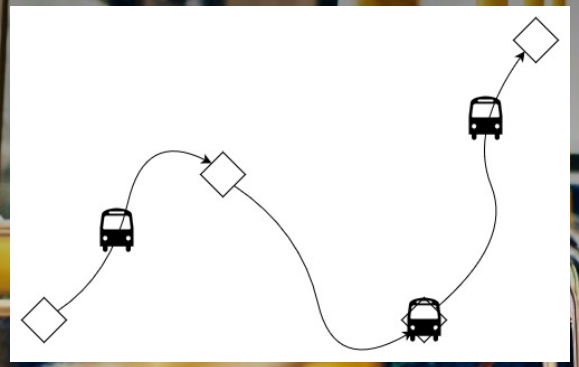
- Direct 1082 ←→ 1082
- Leading zeroes 89 ←→ 0089
- Letter 1131 ←→ 1131B

Variance

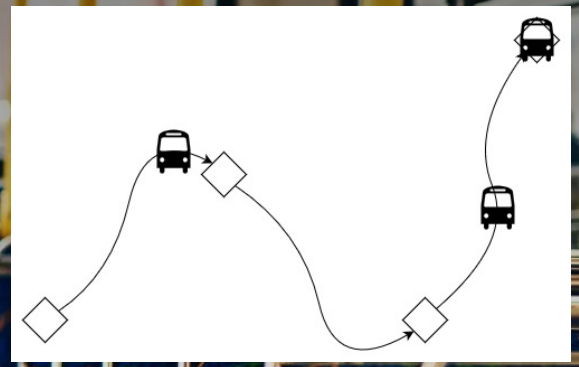


- Search which variance contains the direction ID
- $V_1 \cap V_2 = \emptyset$
- <!> Sometimes terminus of one variance = first stop of the other variance

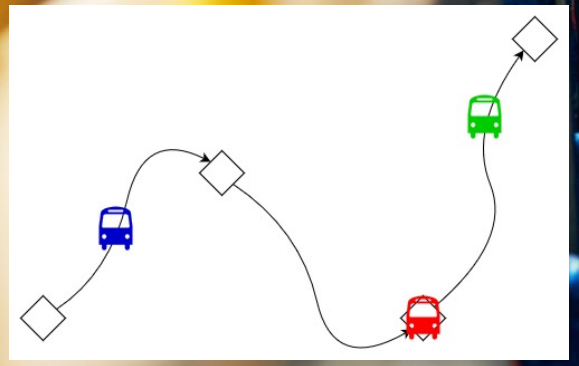
Vehicle ID

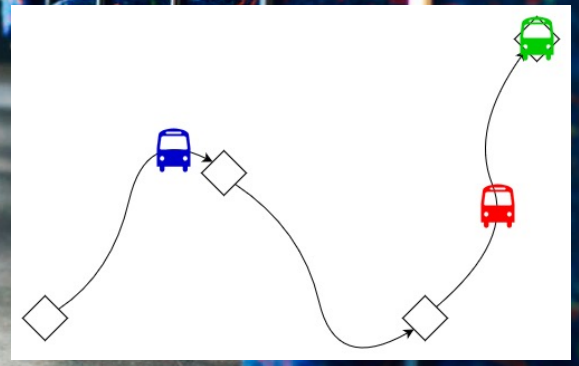


T_1



T_2





Cleansing

- Direction ID not present in line
- Invalid distance
- Technical stops

Task 1: Analysis of vehicle speed

- $V_{avg} = \Delta X / \Delta t \rightarrow$ Speed depends on distance & (inverse of) time
- Used files
  - ✓ vehiclePositionID.csv
  - ✓ LinesInformation.csv
- Distance calculation
  - ✓ Standardize the origin of calculation
    - ❖ Set up all distances according to the first point of the line
- Time calculation

Line	Variance	Stop	DistanceFromPoint	...
71	1	ULB	243	...
71	1	ULB	402	...
71	1	Cimetière d'Ixelles	0	...
71	1	Cimetière d'Ixelles	34	...
...	...	...	...	...

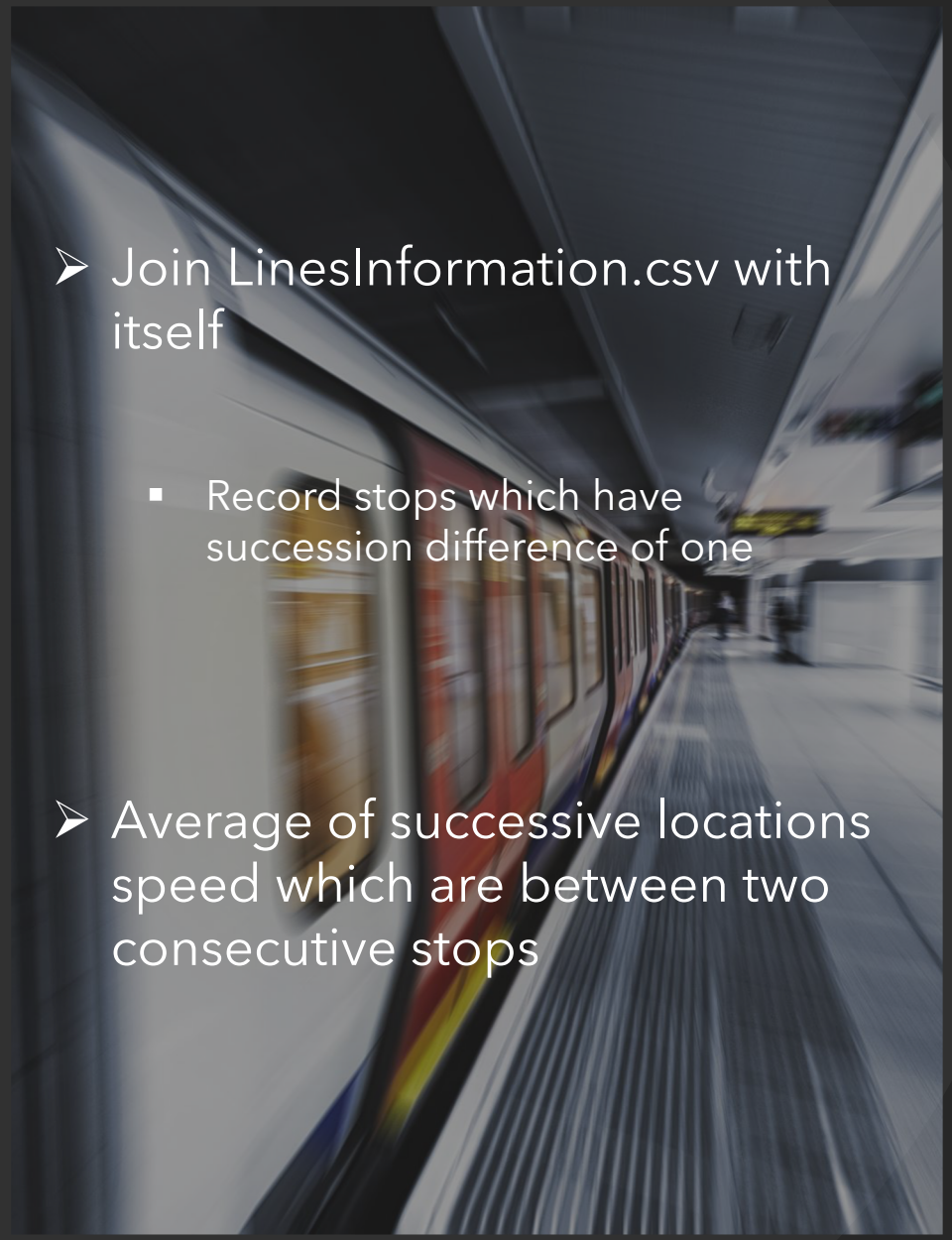
LinesInformation.csv

Line	Variance	Stop	Distance	...
71	1	ULB	5420.399	...
71	1	Cimetière d'Ixelles	5988.051	...
...	...	...	...	...

Line	Variance	Stop	Distance	...
71	1	ULB	5663.399	...
71	1	ULB	5822.399	...
71	1	Cimetière d'Ixelles	5988.051	...
71	1	Cimetière d'Ixelles	6022.051	...
...	...	...	...	...

That gives us the speed between two consecutive pair of location

Task 1: Analysis of vehicle speed - How obtain the speed between two consecutive stops ?



- Join LinesInformation.csv with itself
  - Record stops which have succession difference of one
- Average of successive locations speed which are between two consecutive stops

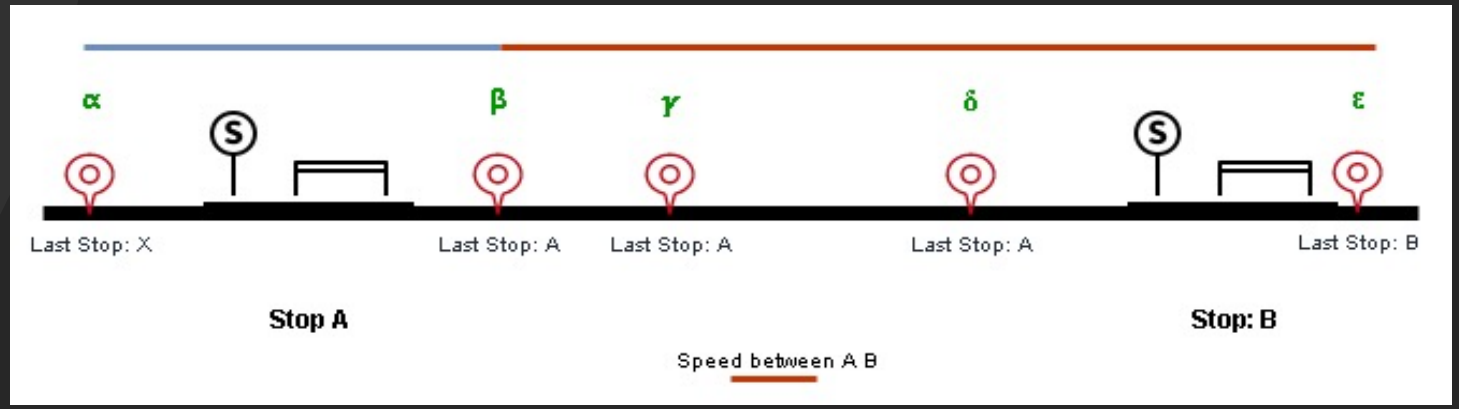
Line	Variance	Stop	Succession	...
2	2	Elisabeth	1	...
2	2	Ribaucourt	2	...
2	2	Yser	3	...
2	2	Rogier	4	...
...	...	...	...	...

+

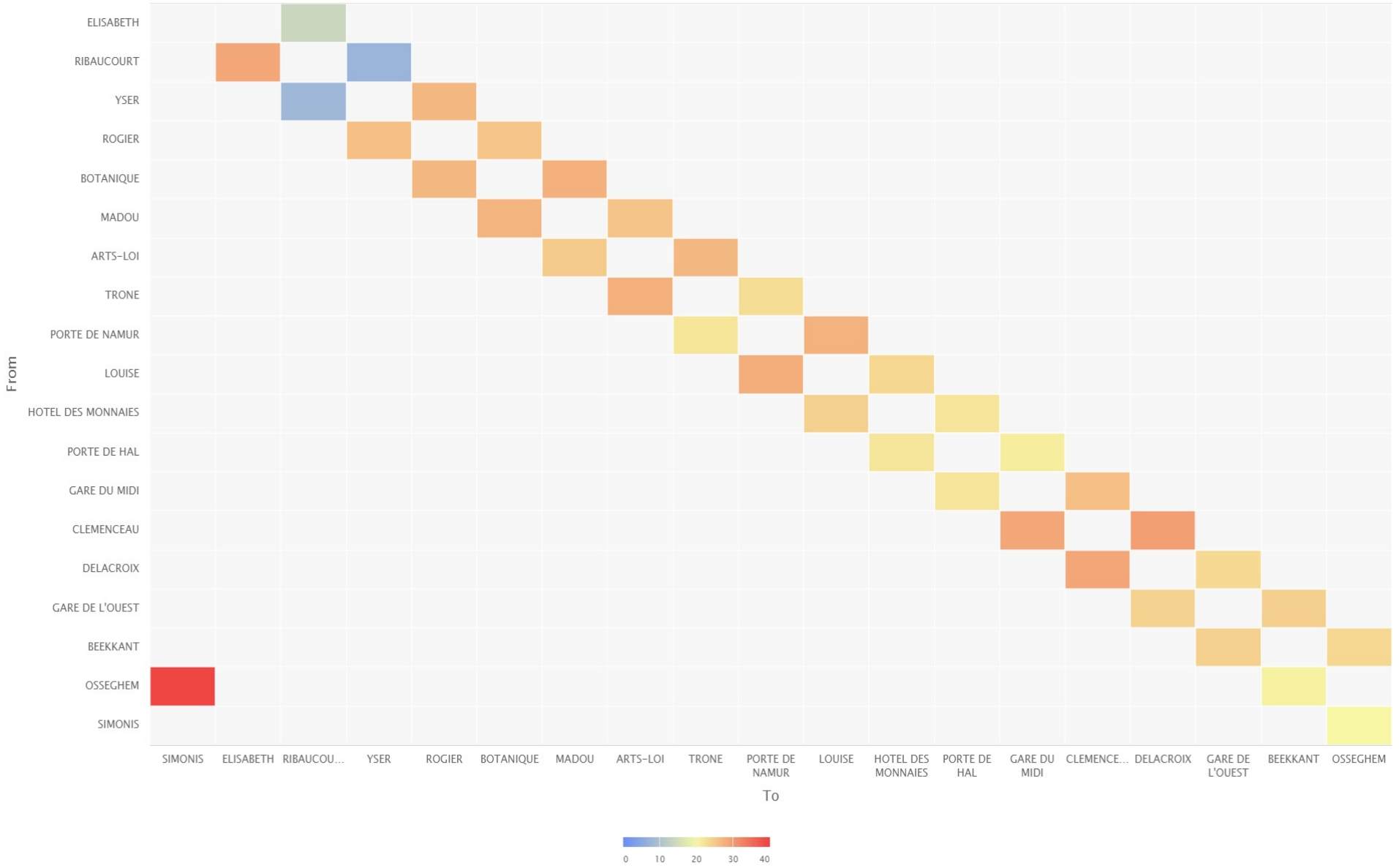
Line	Variance	Stop	Succession	...
2	2	Elisabeth	1	...
2	2	Ribaucourt	2	...
2	2	Yser	3	...
2	2	Rogier	4	...
...	...	...	...	...

=

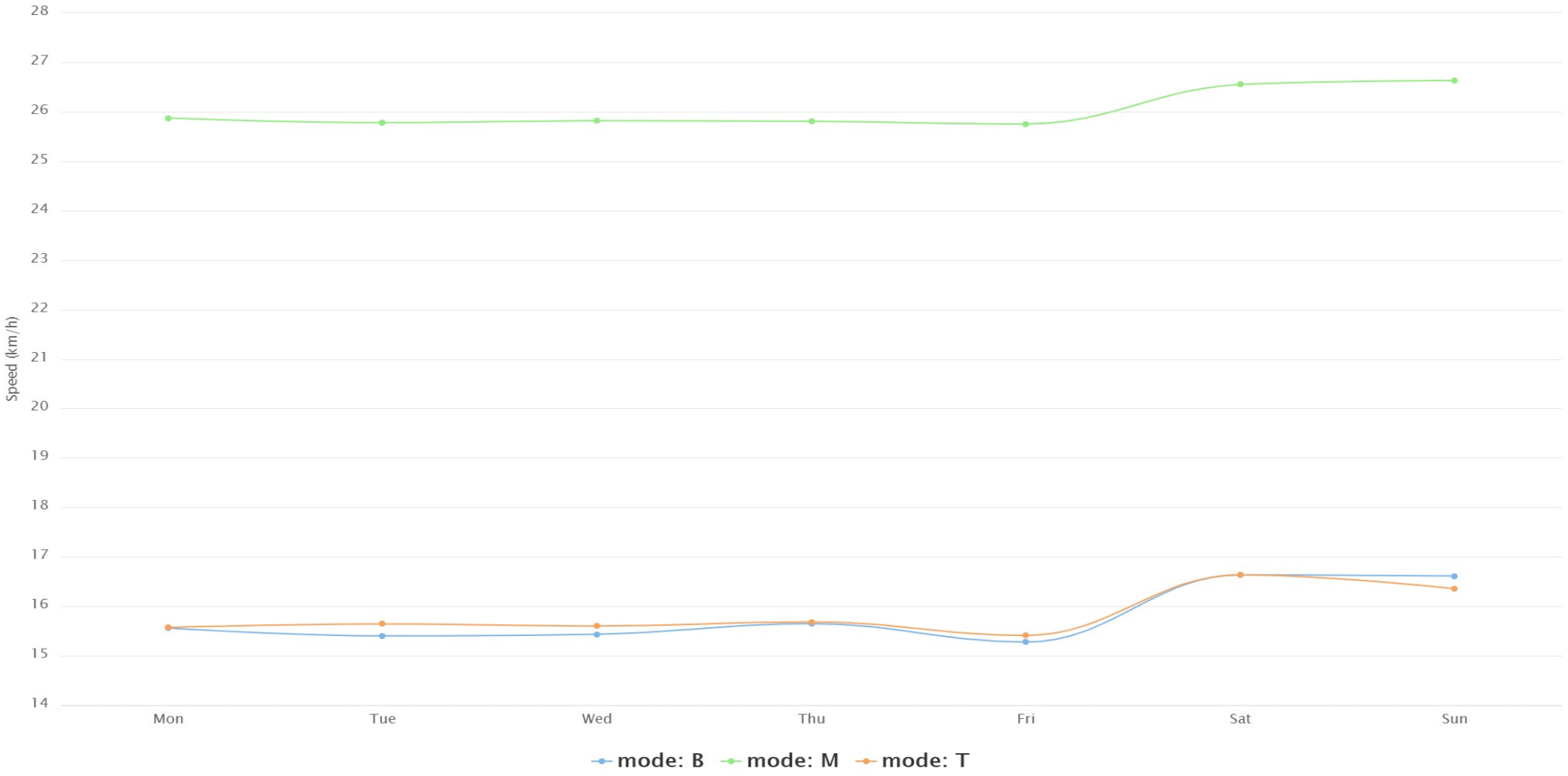
Line	Variance	From	To	...
2	2	Elisabeth	Ribaucourt	...
2	2	Ribaucourt	Yser	...
2	2	Yser	Rogier	...
...	...	...	...	...



Task 1: Analysis of vehicle speed - Line 2 average speed in km/h over the different segments

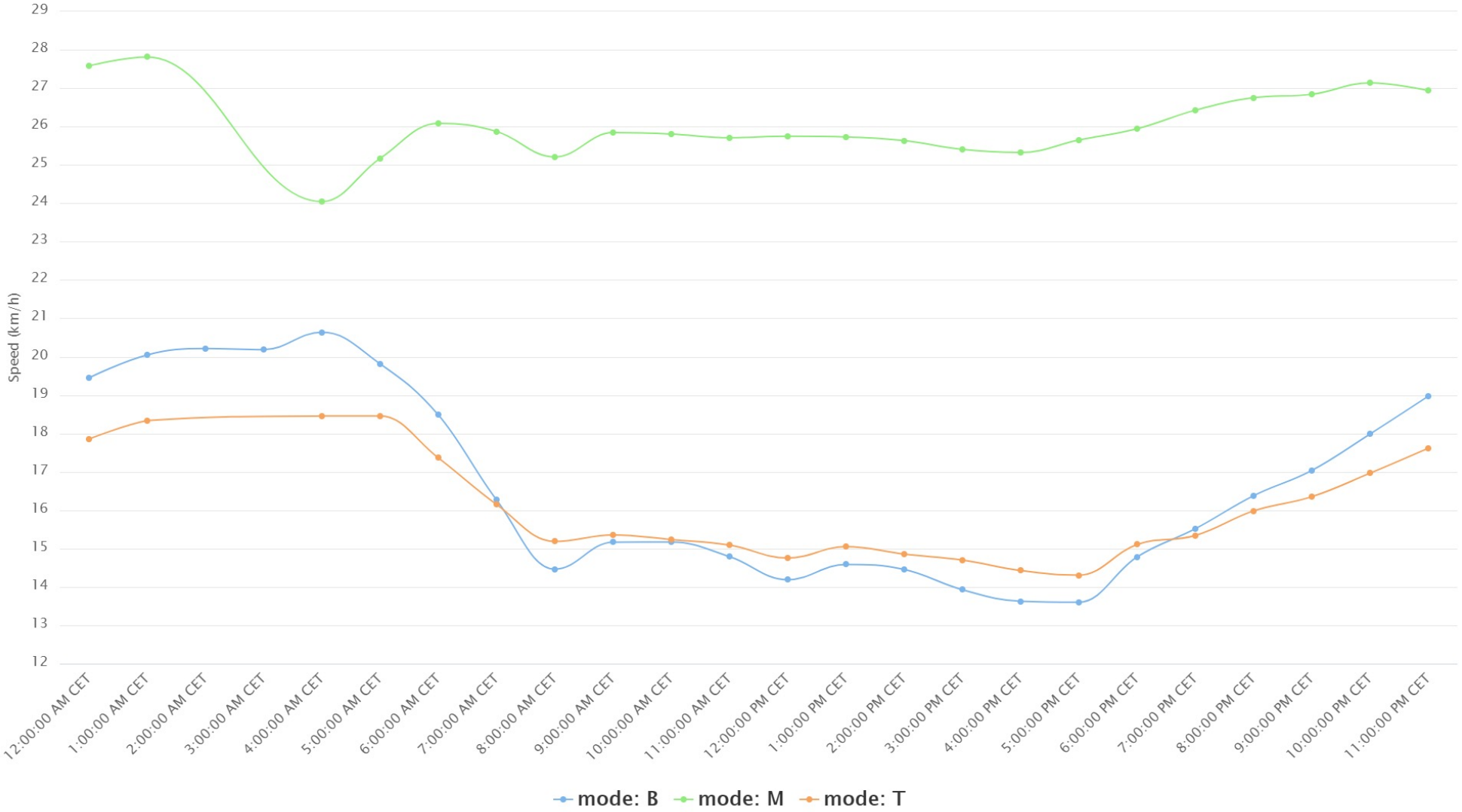


Task 1: Analysis of vehicle speed - Transport modes speed according to each day of week



WEEKLY AVERAGE		
Metro	27.9	27.9
Tram	16.2	16.1
Bus (without Noctis)	15.5	15.7

Task 1: Analysis of vehicle speed - Transport modes speed according to day hours



Task 2: Analysis of vehicle delays

- Vehicle's filtering
  1. *Positions of the stops*
  2. *First arrival time*
- GTFS files (3 & 23 Sept)
  - *Routes.txt*
  - *Trips.txt*
  - *Stop_times.txt*
  - *Calendar.txt*
  - *Calendar_dates.txt*

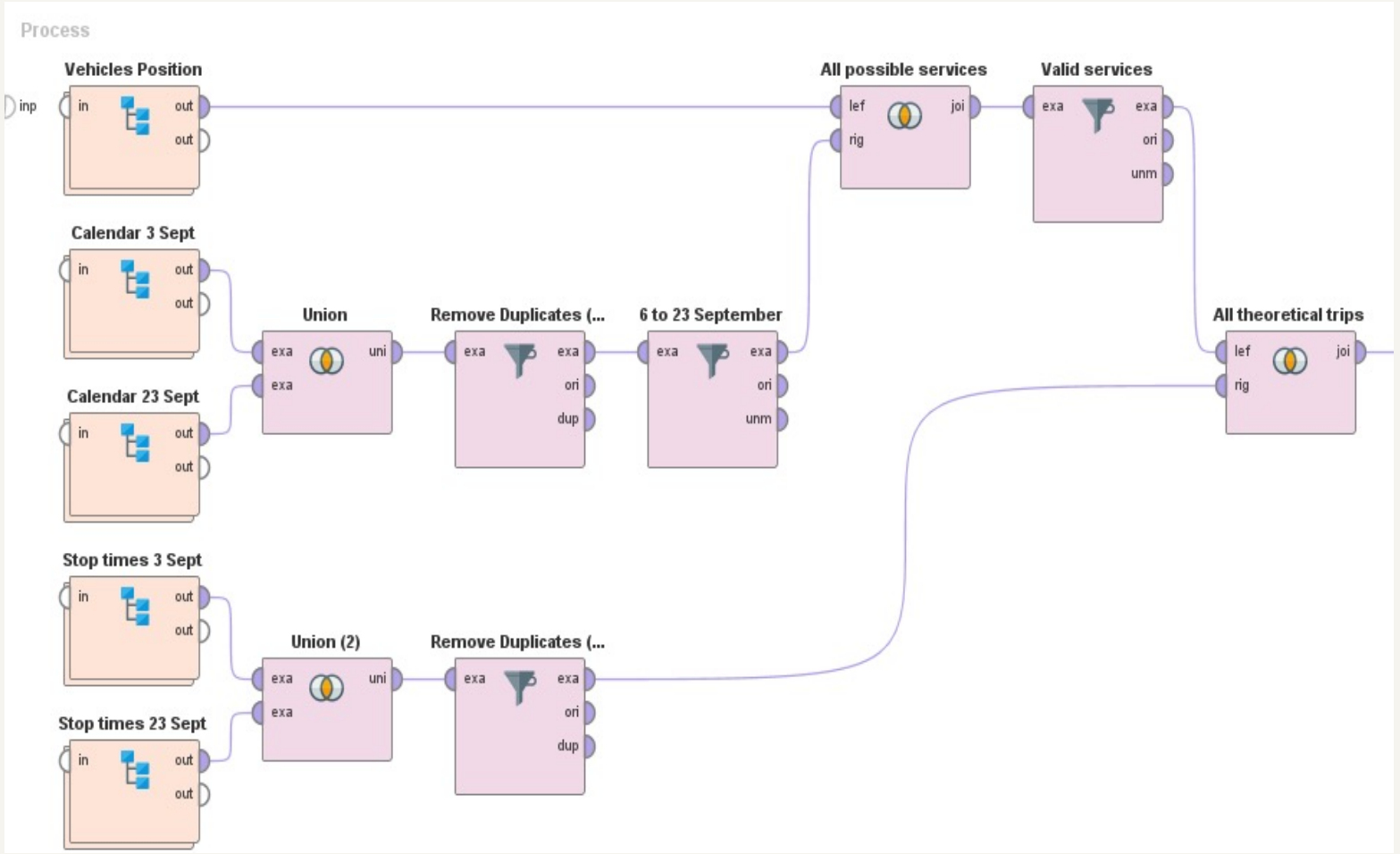
Task 2: Analysis of vehicle delays - *Exceptions*

- Exceptions for the services in the calendar.txt

Type 1 : new service added
(238162502,20210920,1)
→
238162502,0,0,0,0,0,1,0,20210920,20210920

Type 2 : service removed
(238074051,20210927,2)
238074051,1,1,1,1,1,0,0,20210924,20211003
→
238074051,1,1,1,1,1,0,0,20210924,20210926
238074051,1,1,1,1,1,0,0,20210928,20211003

Task 2: Analysis of vehicle delays - Joins



- Vehicles = {vehiclePositionID.csv} : Time, VehicleID, LineID, Variance, DirectionID, pointID and distanceFromPoint.
- Calendar = {calendar.txt, routes.txt, trips.txt} : service id, “Days of week”, start date, end date, LineID, and Variance.
- Stop Times = {stop times.txt, routes.txt, trips.txt} : trip id, stop id and arrival time, service id, LineID, and Variance.

Task 2: Analysis of vehicle delays - *Trips*

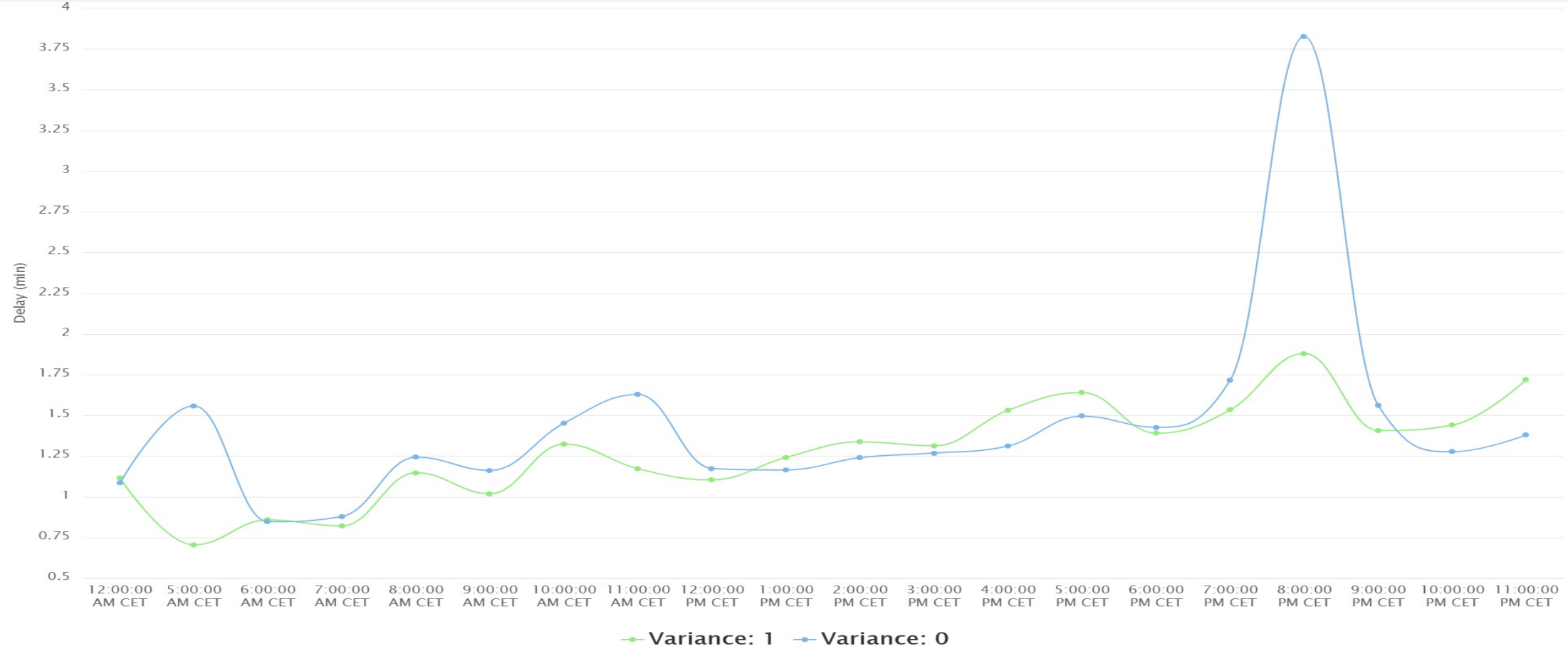
❑ Delay status

- *On time*
- *Late*
- *Early*

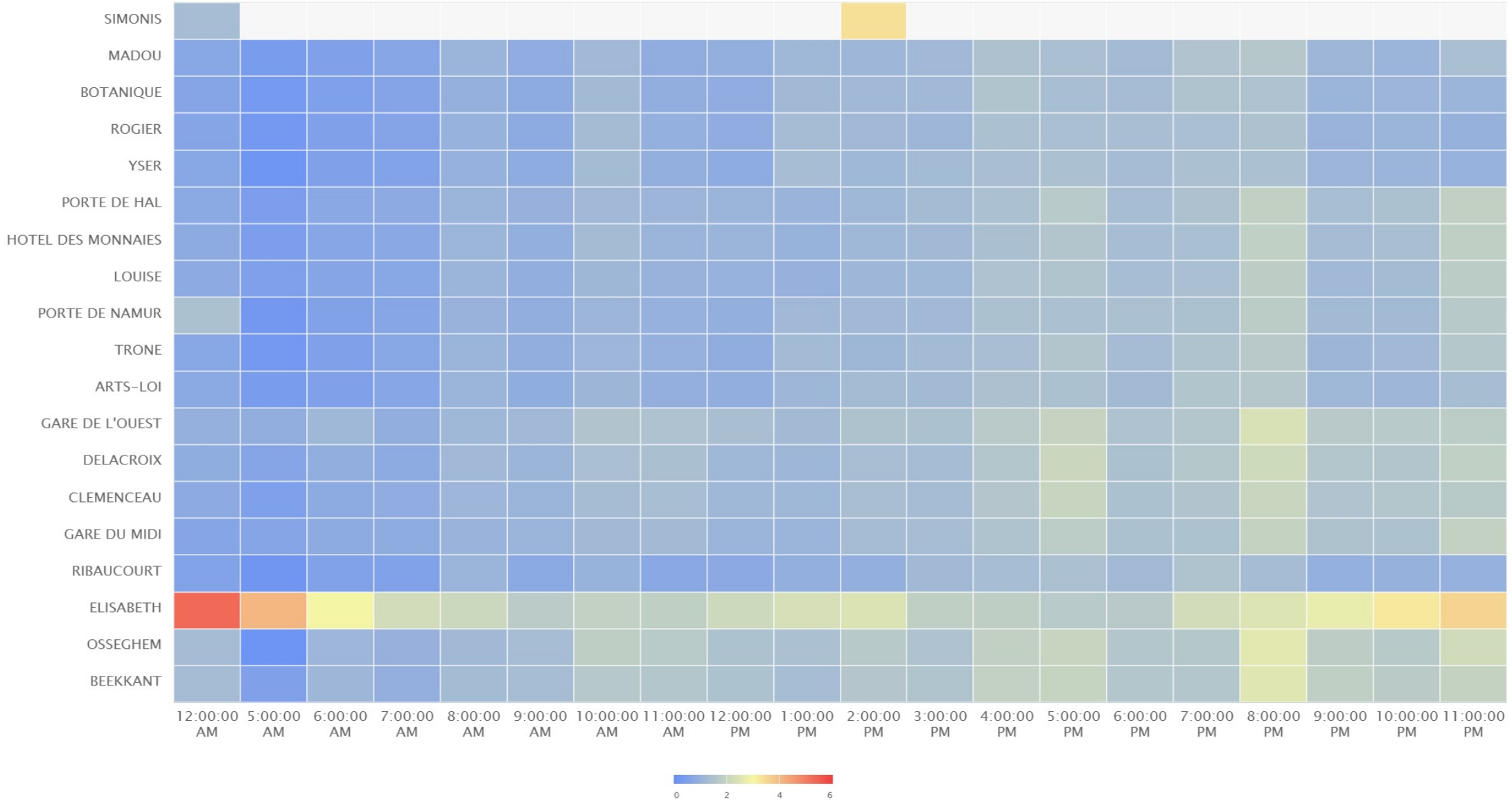
❑ Compute delay

1. *Keep only complete theoretical trips (covering all stops)*
2. *Compare the real time trip (VehicleID) to each remaining theoretical trip*
    - 2.1 Compute the delay for each stop of the trip
    - 2.2 Sum all the delays
3. *Choose the trip with the lowest sum of delays*

Task 2: Analysis of vehicle delays - Average delay in minutes of line 2 per hour



Task 2: Analysis of vehicle delays - Line 2 variance 0 average stops delay over hours



Task 3: Arrival time forecasting

- INPUT :
  - *Line ID*
  - *Variant*
  - *Start stop*
  - *End stop*

- OUTPUT :
  - *Arrival time forecasting*

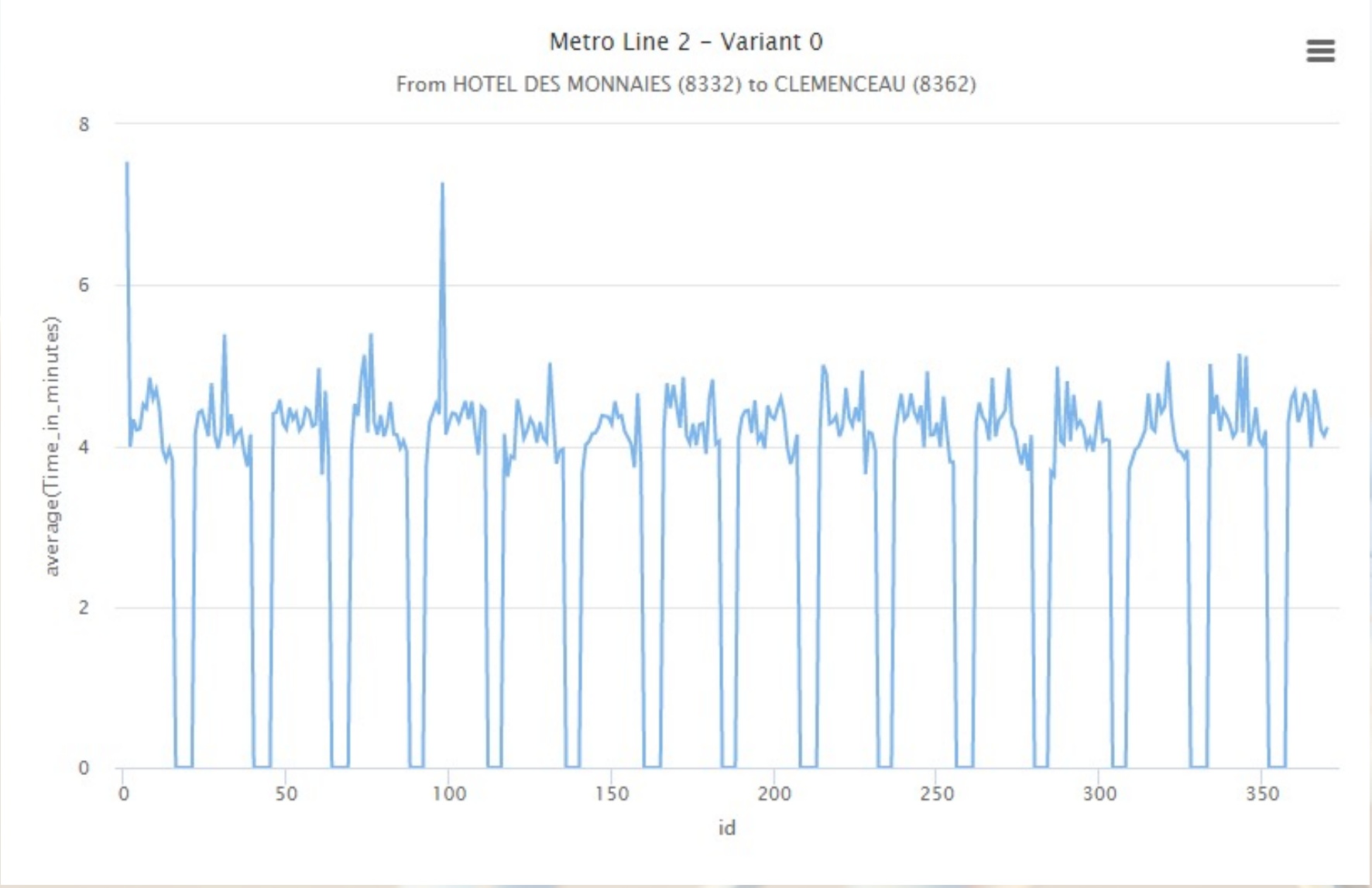
Task 3: Arrival time forecasting - METHOD

- METHODOLOGY :

1. *Keep only the positions = 0 (vehicle positions = stops positions)*
2. *Keep only arrival times at stops*
3. *Compute traveling time between start stop and end stop for each vehicle*
4. *Compute average traveling time per hour*
5. *Fill in missing values with zeroes*
6. *Generate time series of traveling time*
7. *Explore time series*
8. *Validate and apply traveling time forecasting*

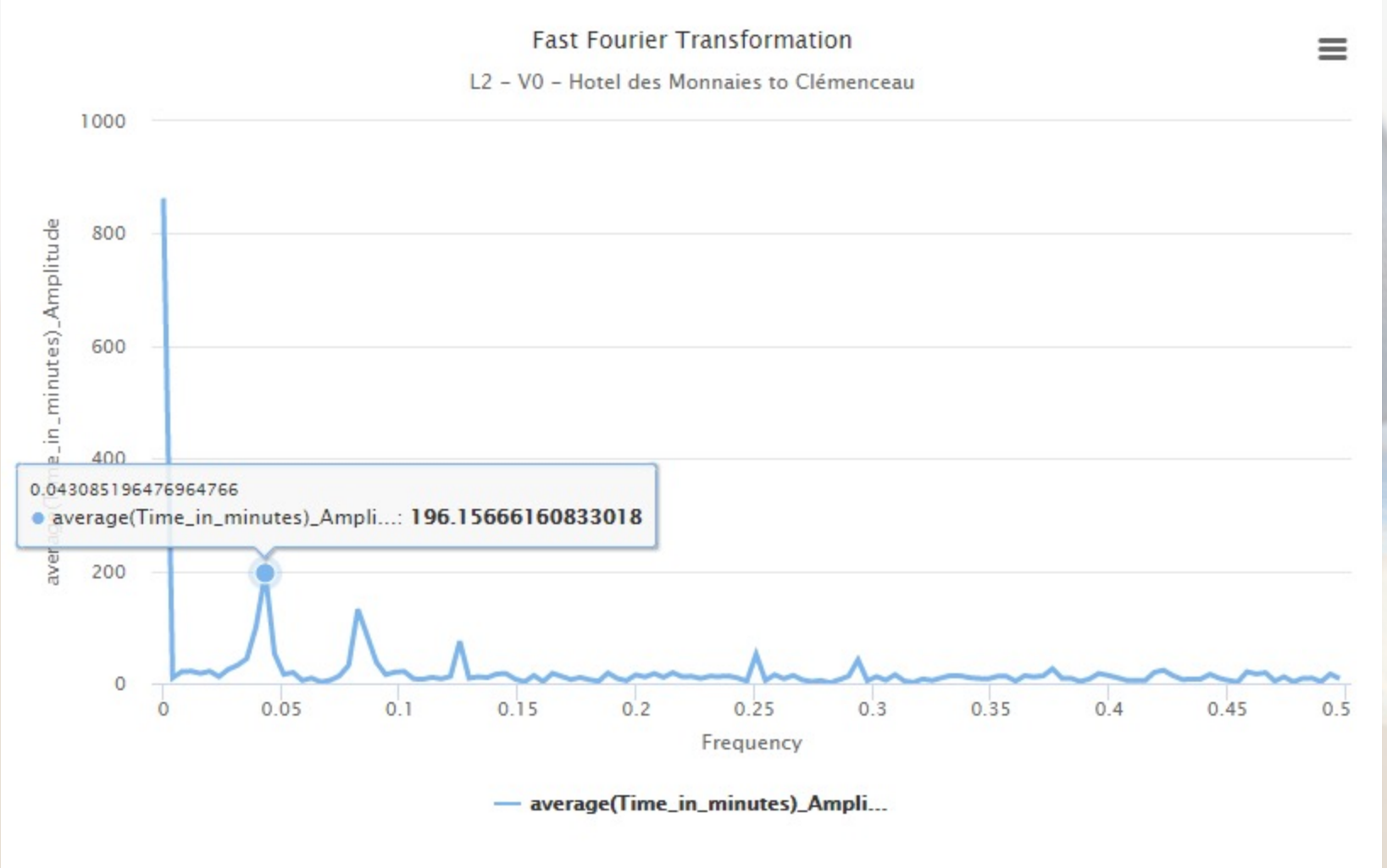
Task 3: Arrival time forecasting - EXPLORATION

1. Time series



Task 3: Arrival time forecasting - EXPLORATION

2. Fast Fourier Transformation

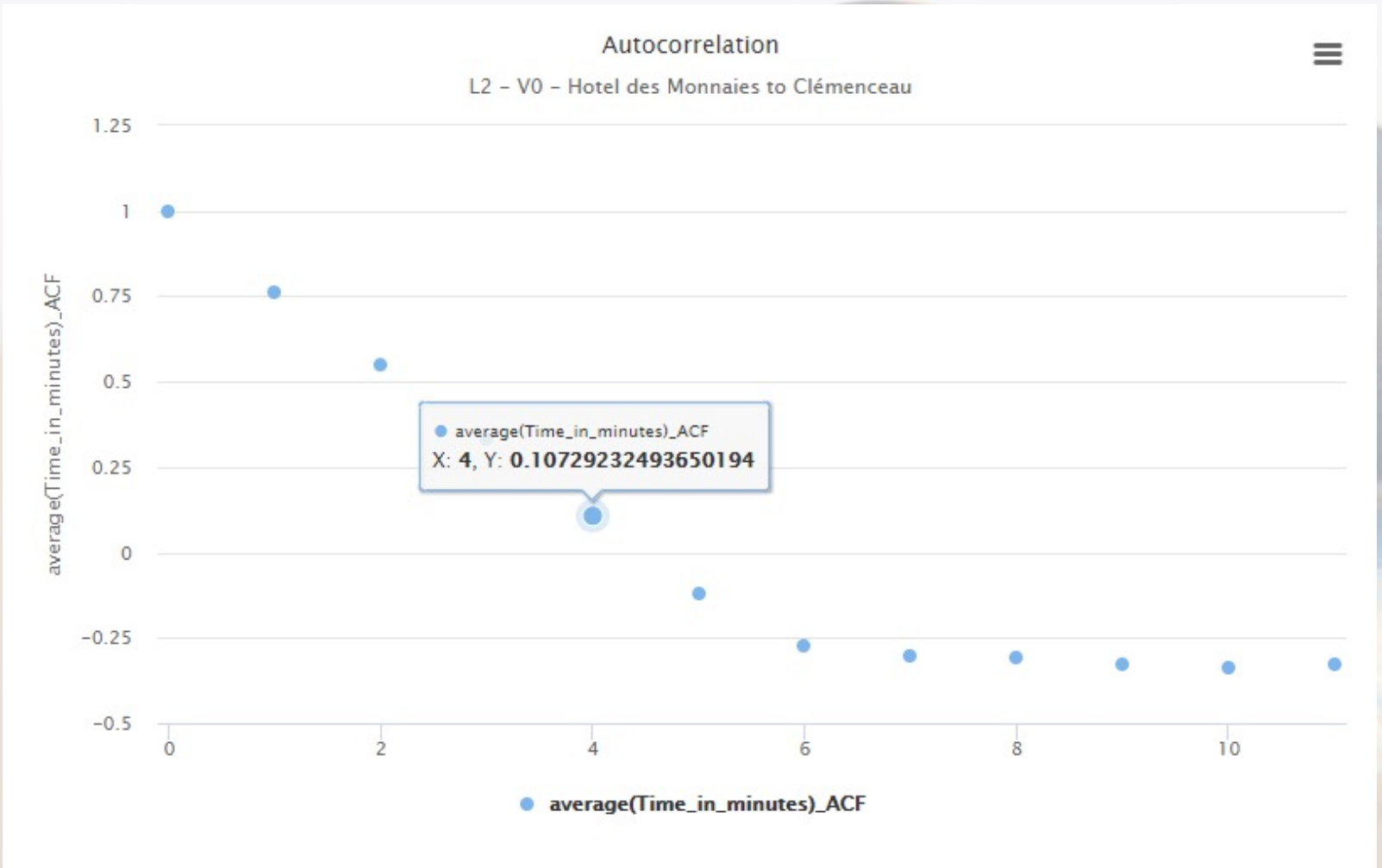


Task 3: Arrival time forecasting - EXPLORATION

3. STL decomposition

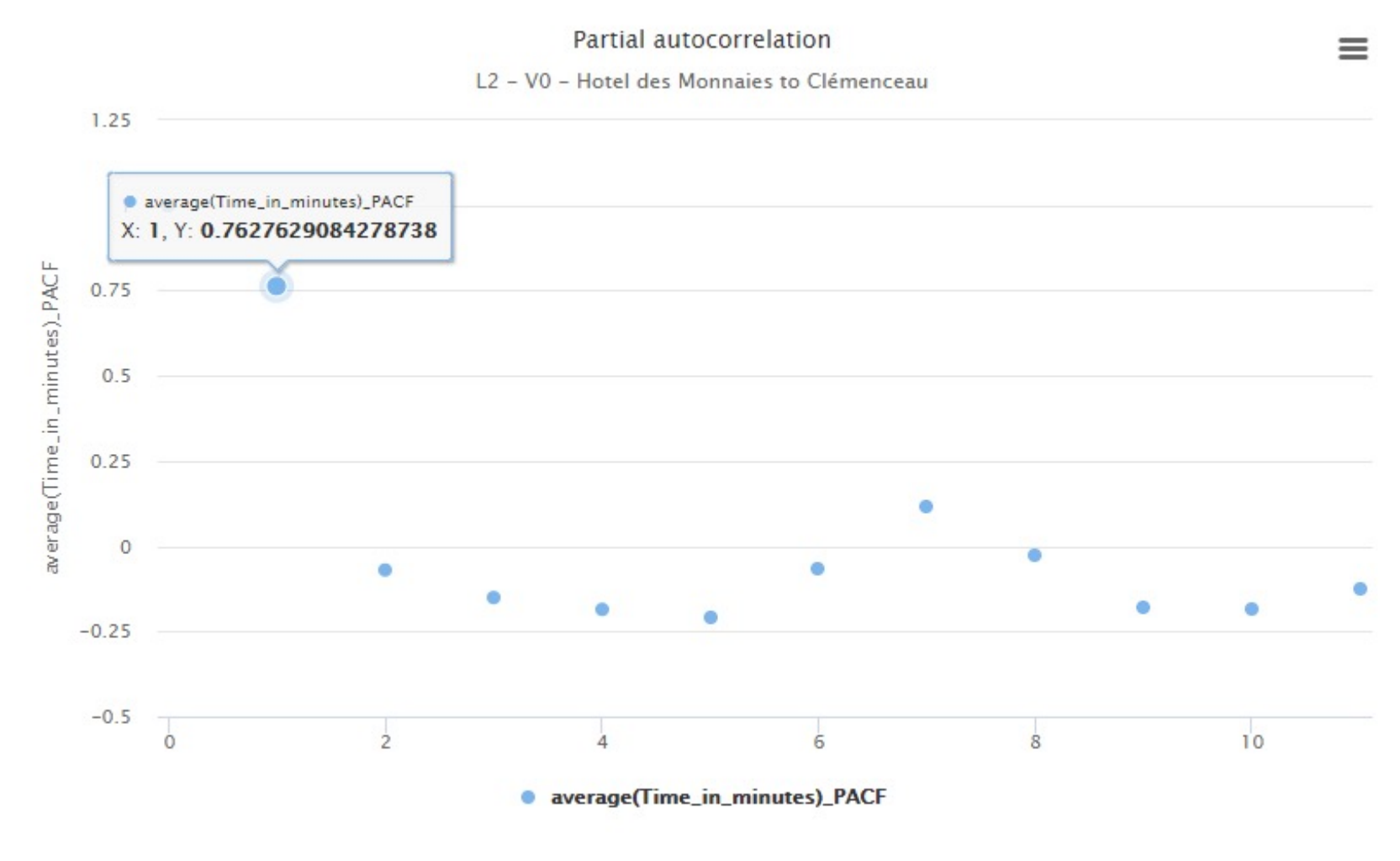
Task 3: Arrival time forecasting - EXPLORATION

4. Autocorrelation



Task 3: Arrival time forecasting - EXPLORATION

5. Partial autocorrelation



Task 3: Arrival time forecasting - FORECASTING

- FORECAST VALIDATION :

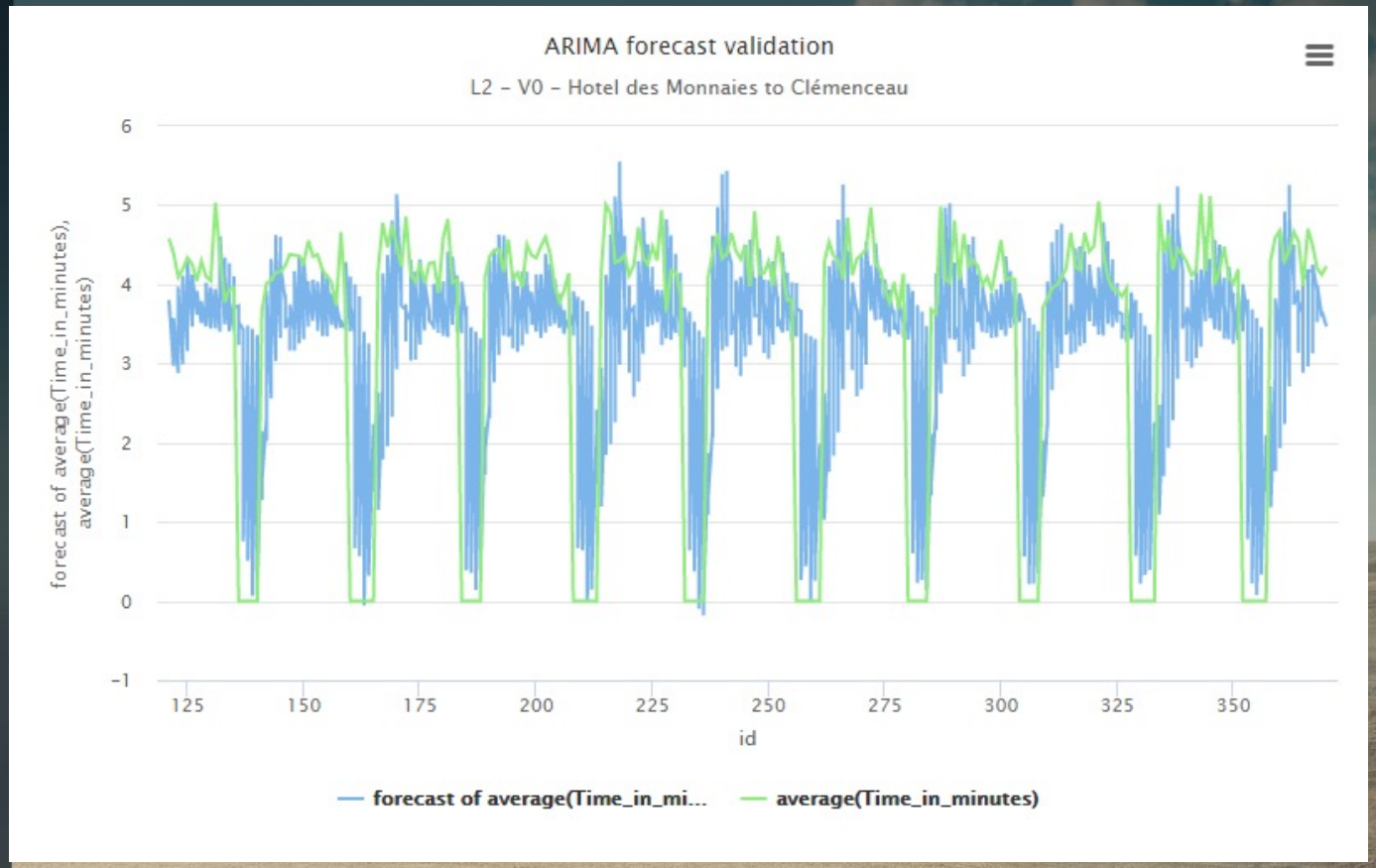
Parameters :

- ✓ *From exploration, we select ARIMA parameters : p = 1, d = 0, q = 4*
- ✓ *Windows size (training set) = 120 hours*
- ✓ *Step size : 1 hour*
- ✓ *Horizon size (test set) : 5 hours*
- ✓ *Main criterion : AIC*

THESE PARAMETERS NEED TO BE ADAPTED FOR EVERY FORECAST

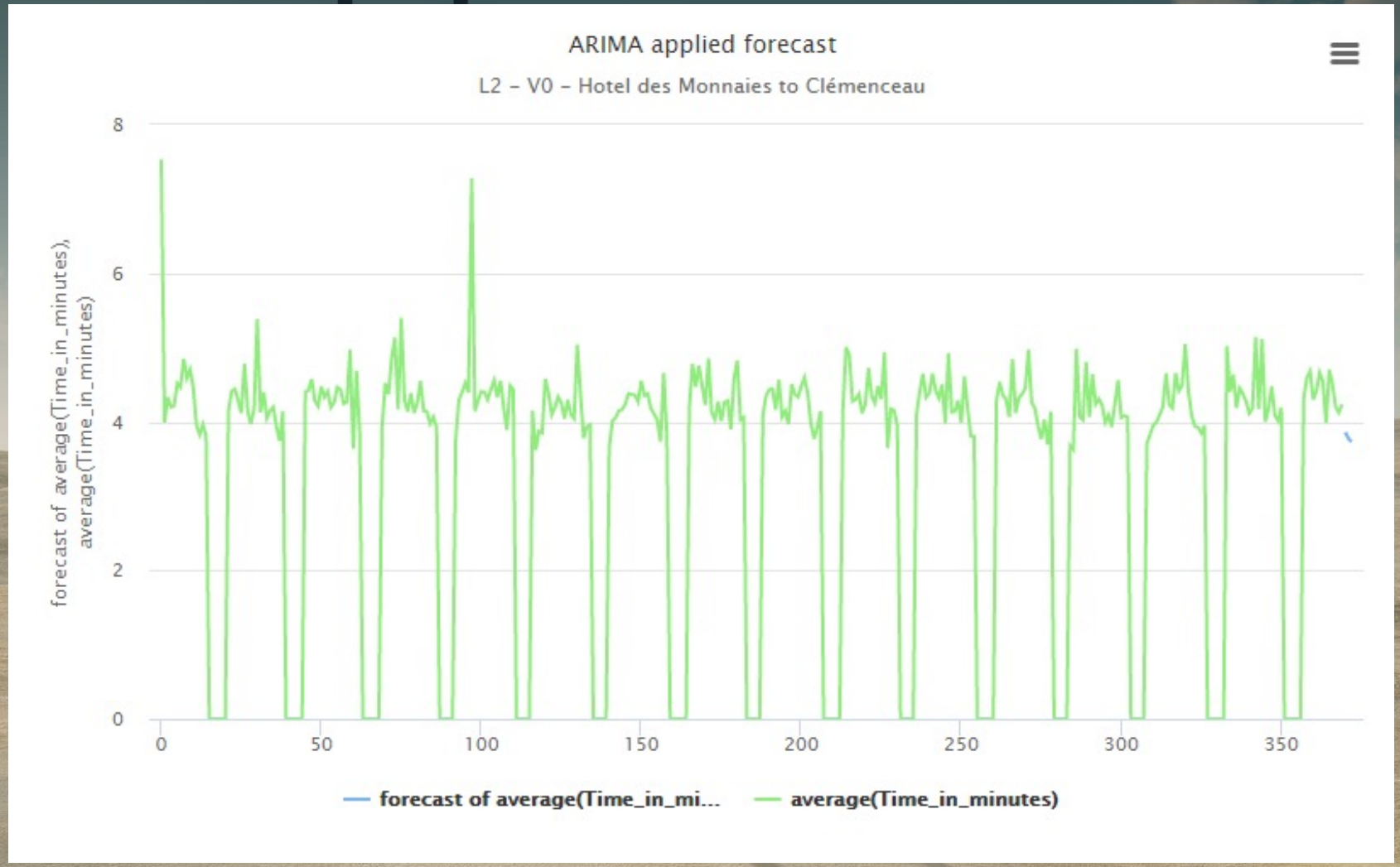
Task 3: Arrival time forecasting - FORECASTING

- FORECAST VALIDATION :
  - ✓ *Root Mean squared error : 1,27 min*
  - ✓ *Absolute error : 1,11 min*
  - ✓ *Relative error : 17,9 %*



Task 3: Arrival time forecasting - FORECASTING

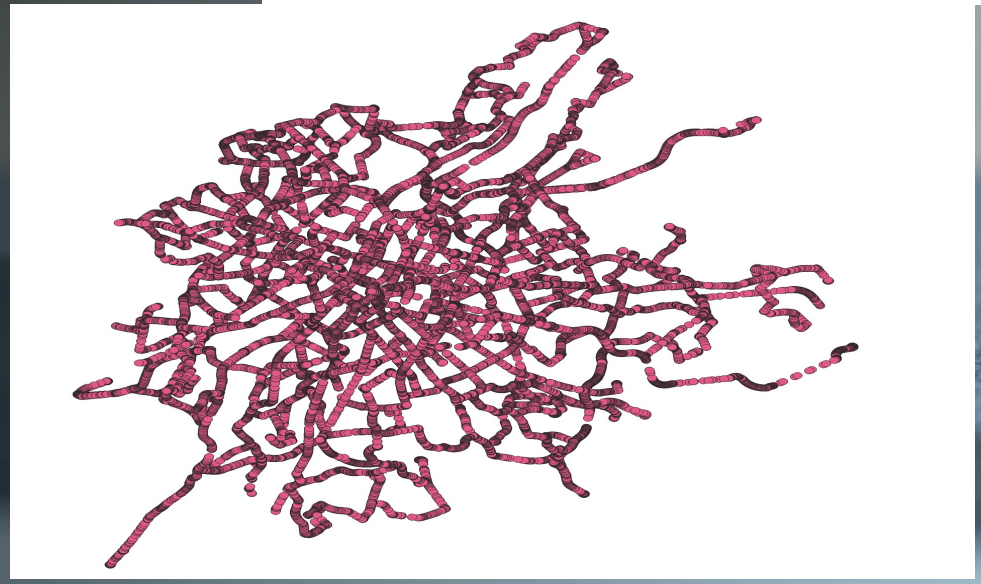
- FORECAST (forecast horizon = 3 hours) :



Task 4: GPS Track Inference

- Data

- STIB Shape files
  - GeoPandas : Points of each line



- Position analysis
- Speed analysis

Task4: GPS Track Inference - Position analysis

1. Track point (P_t) vs. Line point (P_l)
   - ❖ Different CRS
     - ➢ WGS 84 (World Geodetic System) → EPSG: 4326
     - ➢ Belgian Lambert 72 → EPSG: 31370

2. $|P_t - P_l| \leq 10 \rightarrow l$ is candidate

3. If l was candidate for 75% of $P_t \rightarrow l$ will be candidate for t

4. All candidate lines store in a list

Task 4: GPS Track Inference - Speed analysis

1. For each candidate line l :

   - $P_s \rightarrow$ Closest stop to the first point of the track
   - $P_e \rightarrow$ Closest stop to the last point of the track

2. S_l : Speed between P_s and $P_e \rightarrow$ STIB data

3. S_t : Speed of the track

   - Scale of 30 seconds

4. $|S_t - S_l| \geq 15 \rightarrow$ Ignore l

Task 4: GPS Track Inference - Results of GPStracks.csv file

Track ID: 7
.....
Position analysis:

Candidate lines: [('007t', 1), ('007t', 2), ('025t', 1), ('025t', 2)]
Votes: [166, 189, 152, 175] / 195
Percentages: [85.128, 96.923, 77.949, 89.744]
.....
Speed analysis:

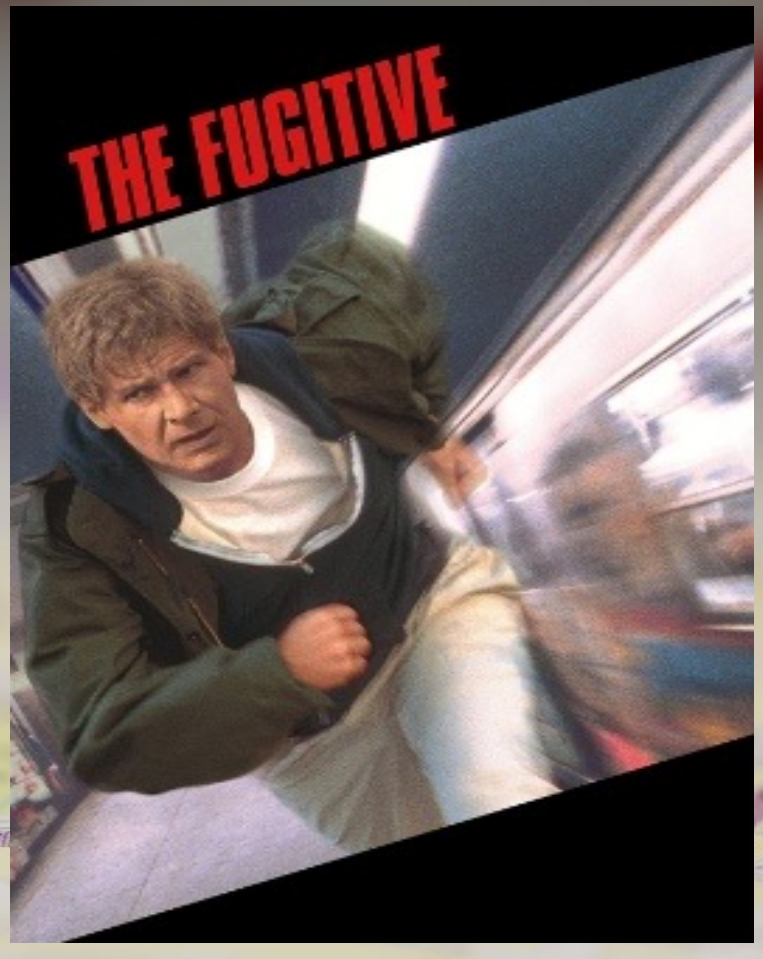
Line and variance ('7', '1')
Stops (from --> to): 0906 --> 5258
Average line speed: -inf
Average track speed: 19.948385691111582
.....
Line and variance ('7', '2')
Stops (from --> to): 0901 --> 3481
Average line speed: 13.851716761672703
Average track speed: 19.948385691111582
.....
Line and variance ('25', '1')
Stops (from --> to): 0906 --> 3480
Average line speed: -inf
Average track speed: 19.948385691111582
.....
Line and variance ('25', '2')
Stops (from --> to): 0901 --> 2397F
Average line speed: 12.965666453178818
Average track speed: 19.948385691111582
.....
Result:

Track: 7 --> transport mode t (Probably the line is: 7)

Tracks ID/ Information	Transport mode	Possible line
1	Other (o)	—
3	Tram (t)	82
4	Bus (b)	50
5	Tram (t)	82
6	Other (o)	—
7	Tram (t)	7
8	Other (o)	—
10	Other (o)	—
11	Tram (t)	25

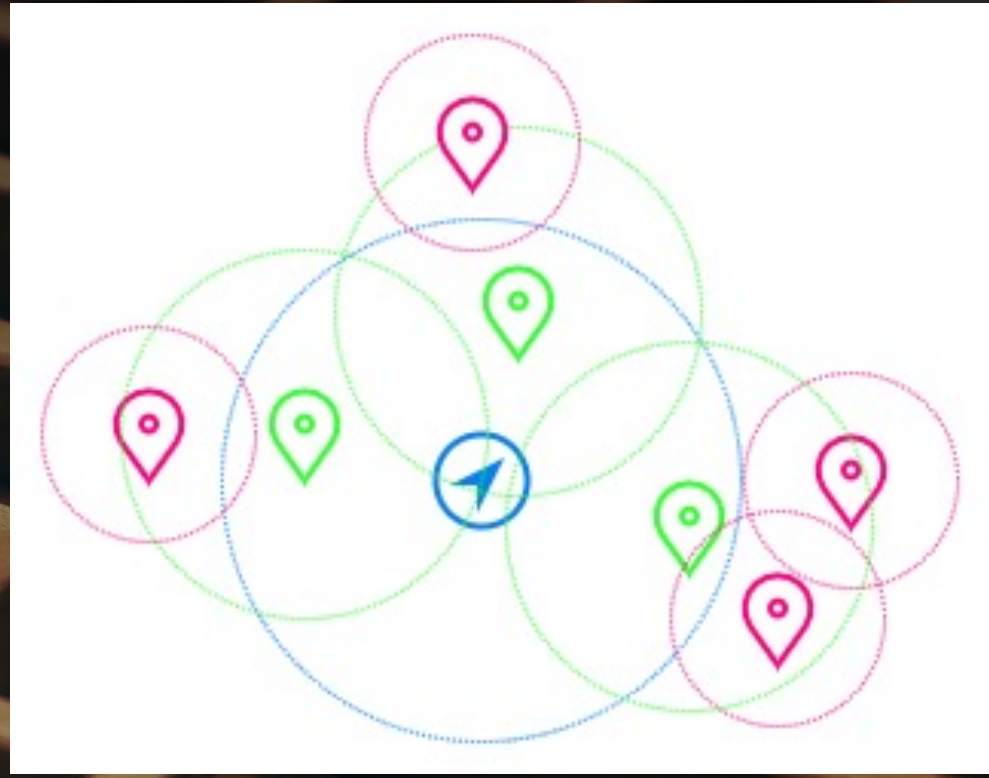
Task 5: Reachability analysis

- What is it ? How can it be useful ?
- Supposition
  1. Just STIB network
  2. Walking at most with 5 Km/h
- Using an iterative algorithm

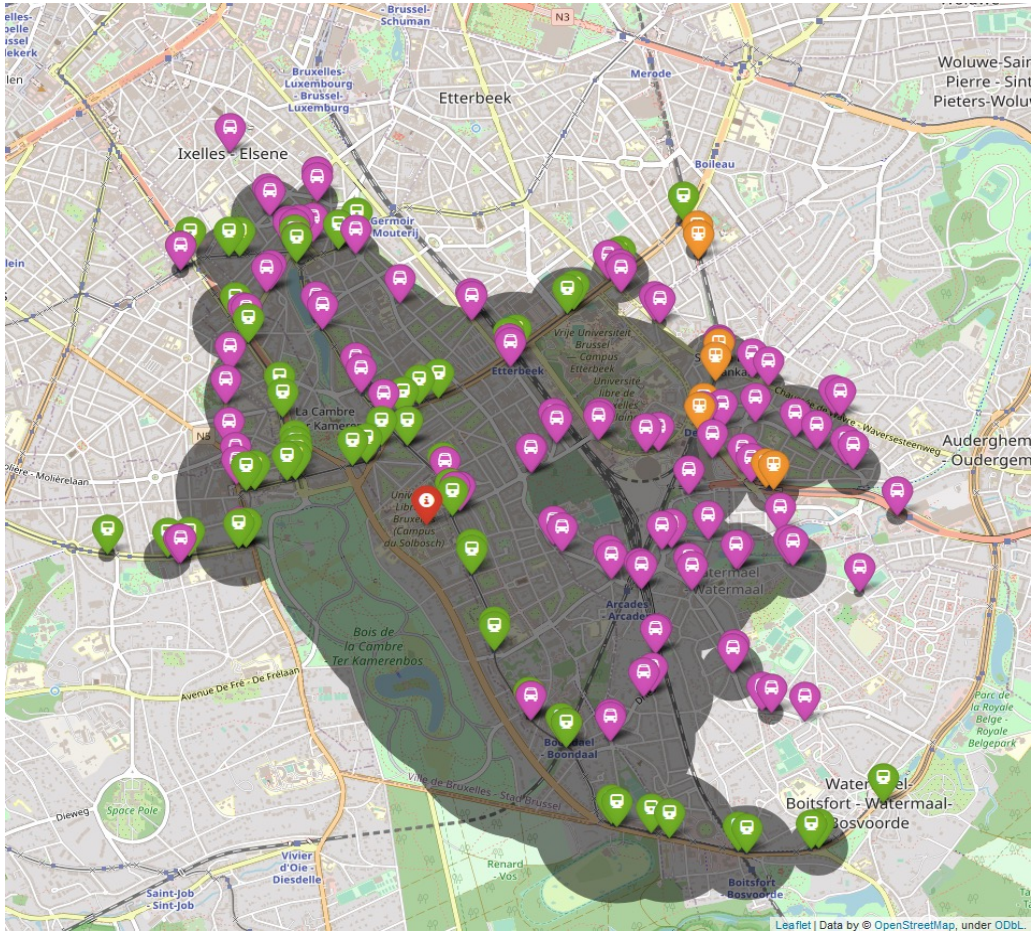


Task 5: Reachability analysis - Algorithm

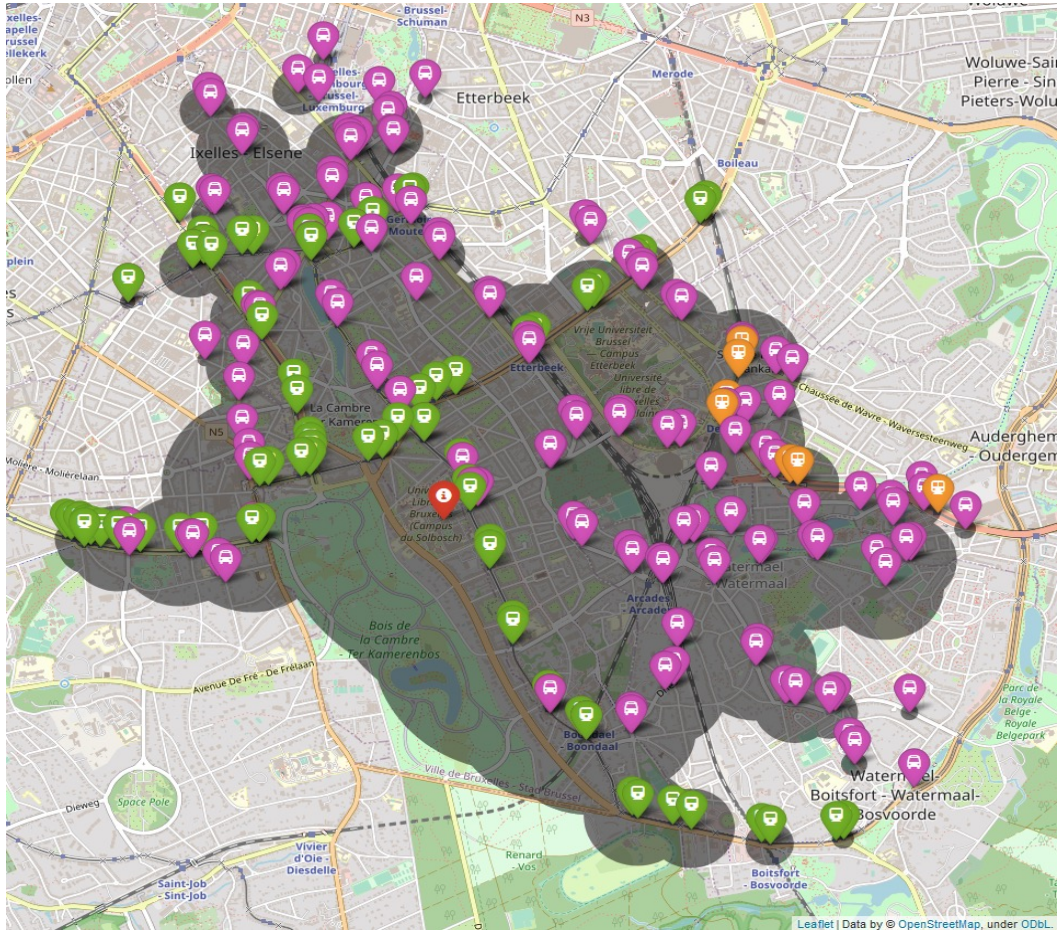
- While there are possibilities
  - Compute all the reachable stops by walking or public transports
  - Calculate arrival time
  - If stop already visited
    - If new arrival time earlier than last one
      - Update
    - Else
      - Dismissed



Task 5: Reachability analysis - Result



23/10/2021 - 8:10



23/10/2021 - 23:10

CONCLUSION

- Suggestions for data improvement
  - ✓ *In the JSON files :*
    - *Adding Vehicle IDs in JSON files*
    - *Adding trip IDs to JSON files*
    - *Adding metros' precise positions to JSON files*
  - ✓ *Harmonising stop IDs between JSON file and GTFS files*

- Tough but interesting project overall !

Thank you for your attention