# Vehicle & Pedestrian collision prediction

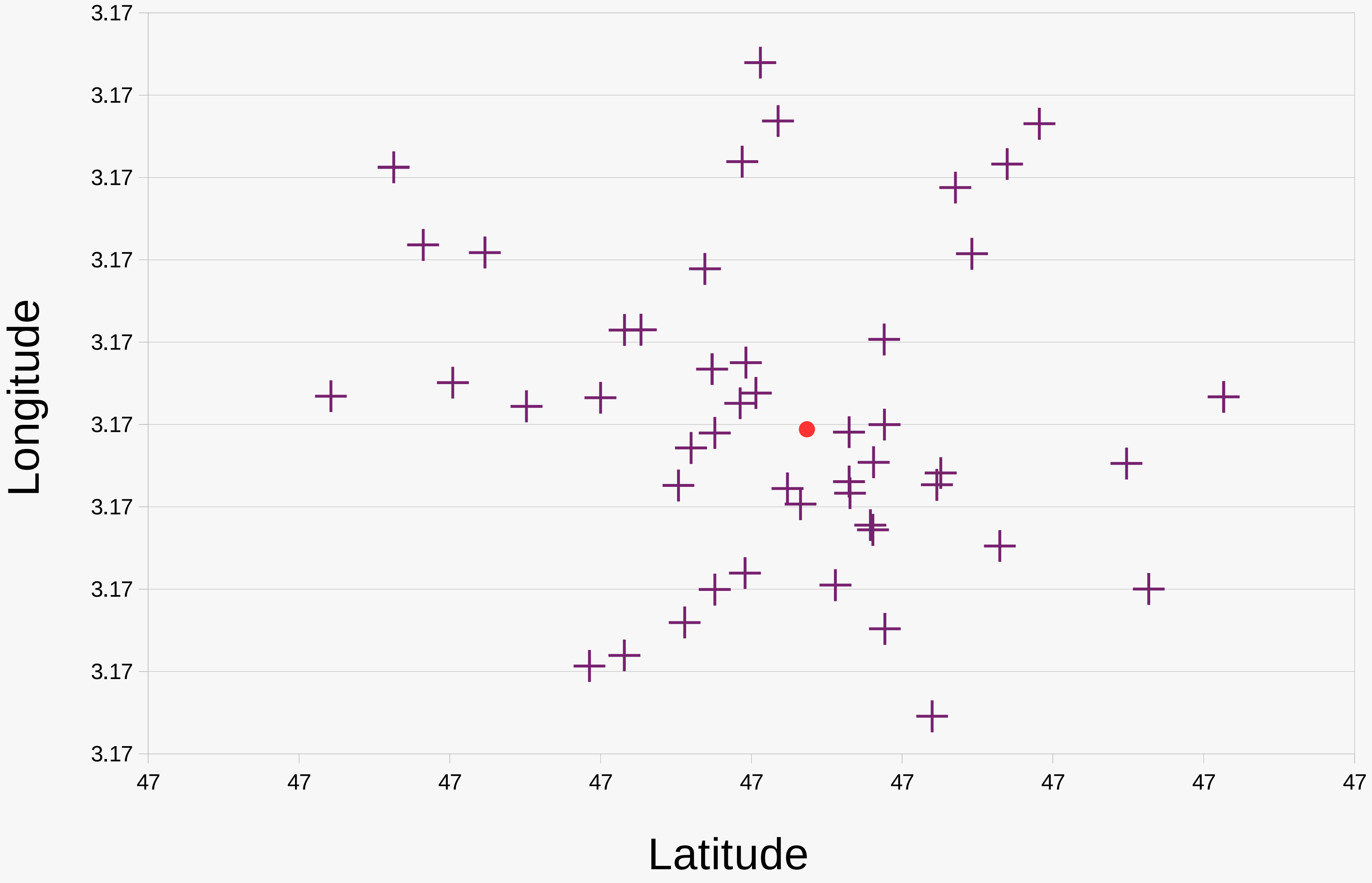
**06/06/2017**

Presented by Aghiles DJOUDI
University Pierre and Marie Curie

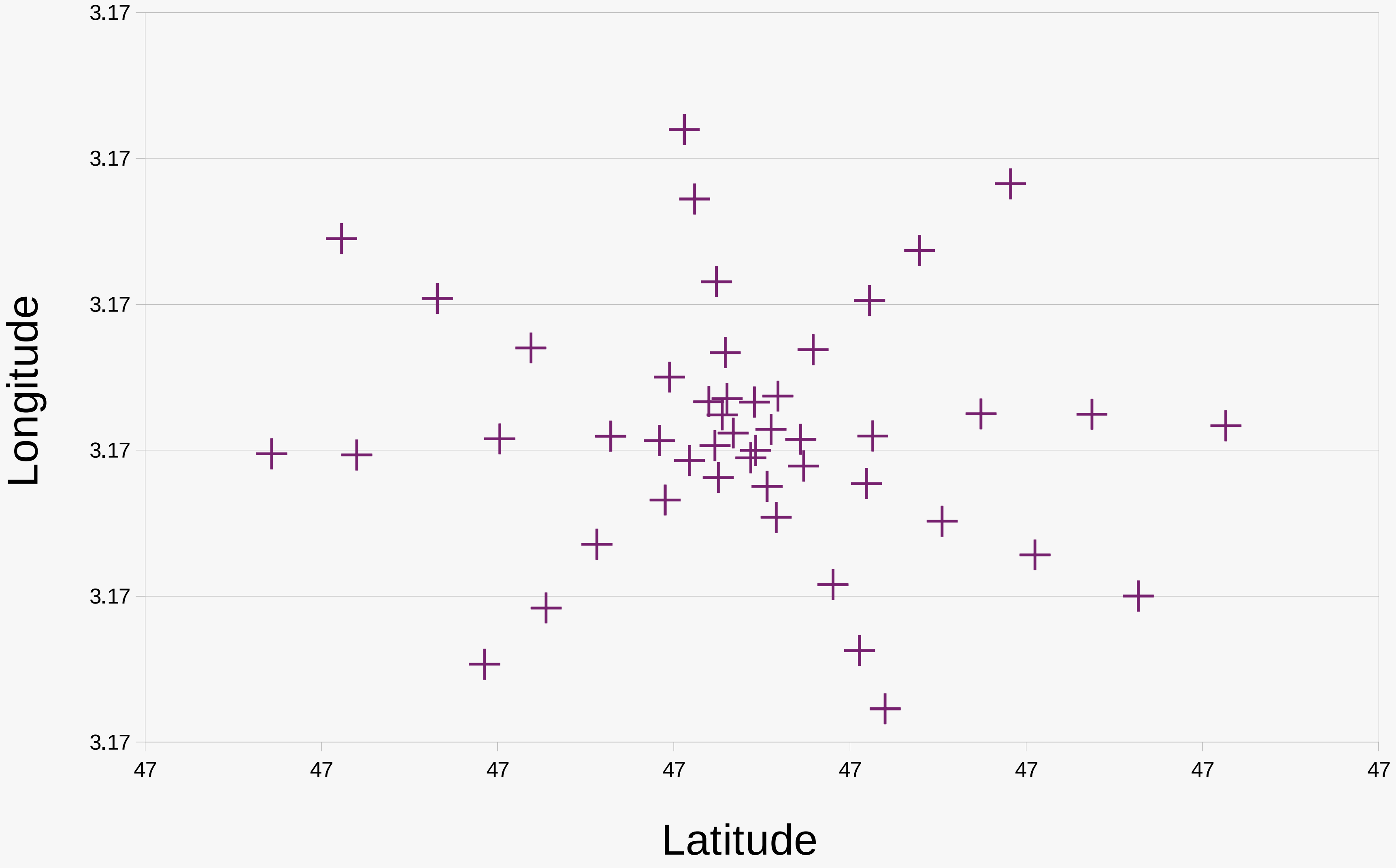
# Urban area

*Location*

Smartphone Location



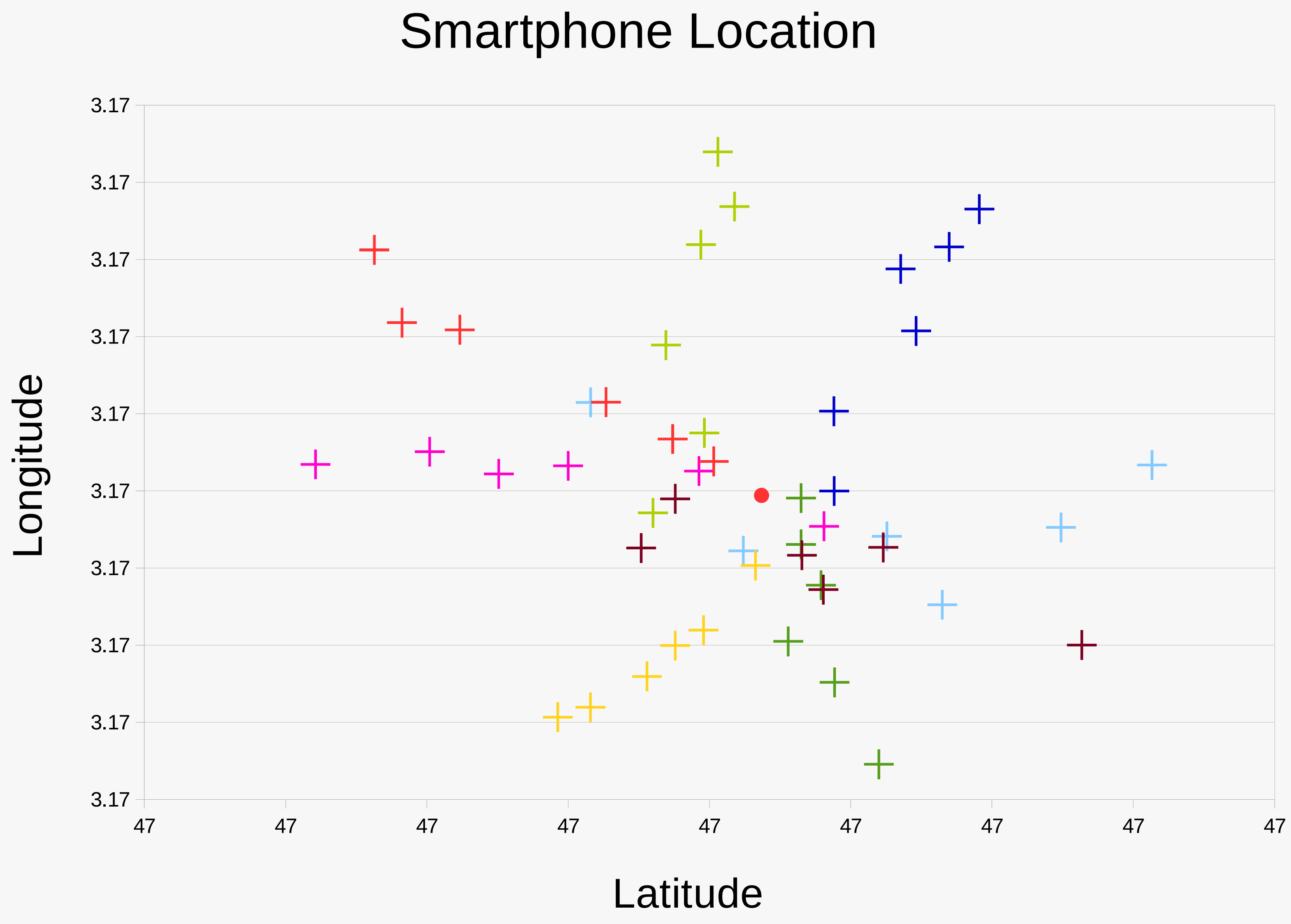
RTK Location

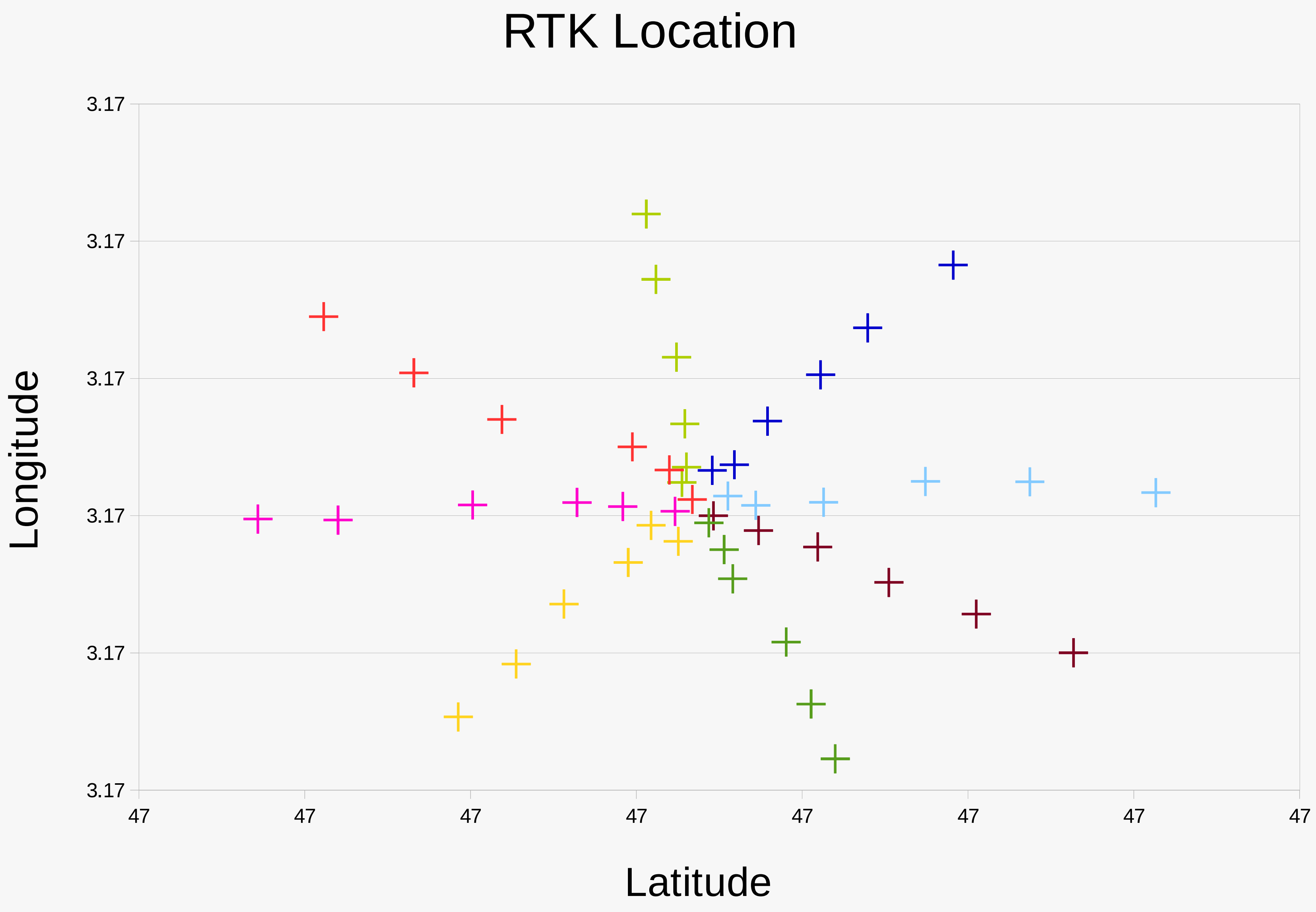


# Urban area

*Location*

*Sunny day*

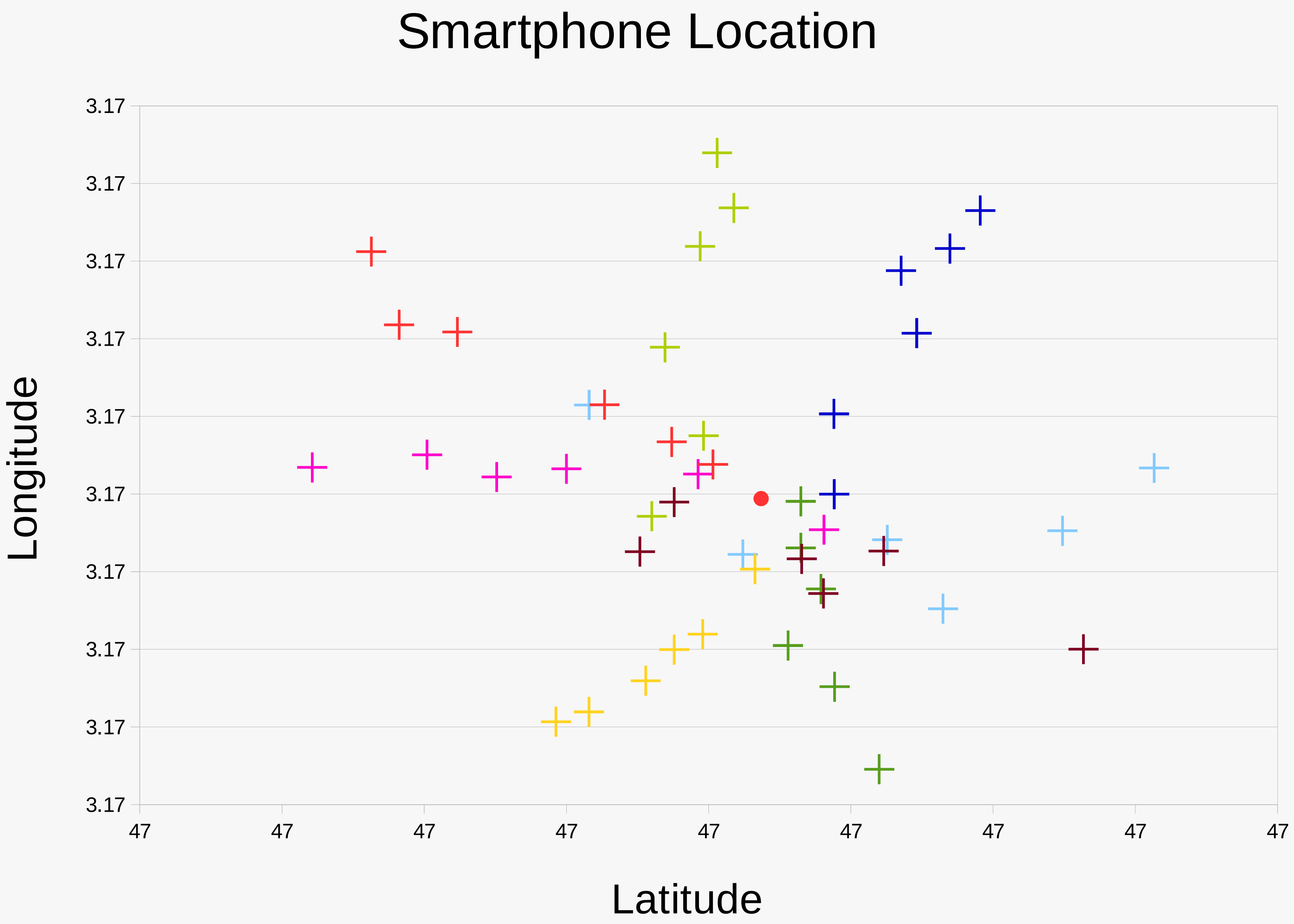




# Urban area

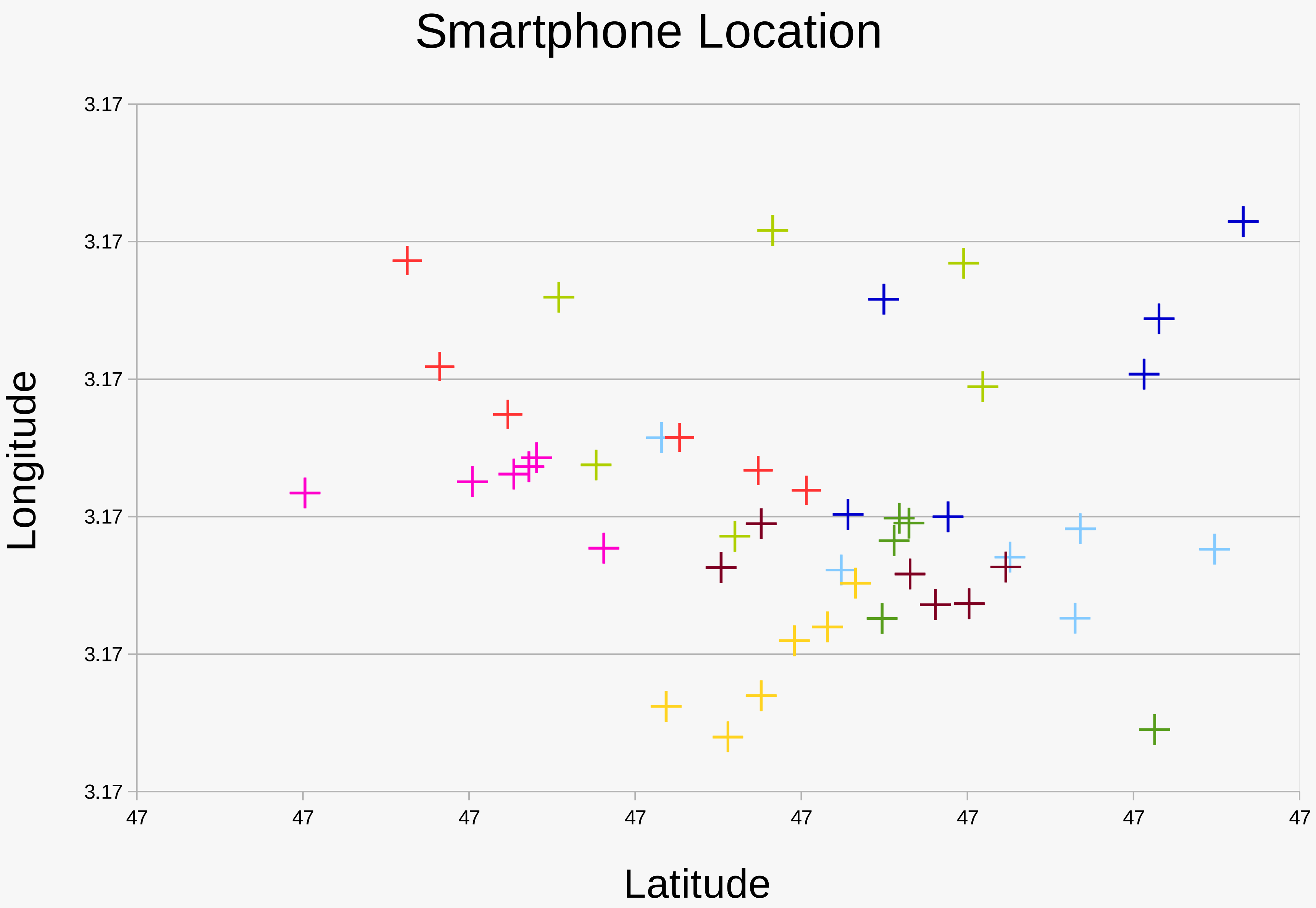
*Location*

*Sunny day*



*Errors: 2 to 7 meters*

*Cloudy day*

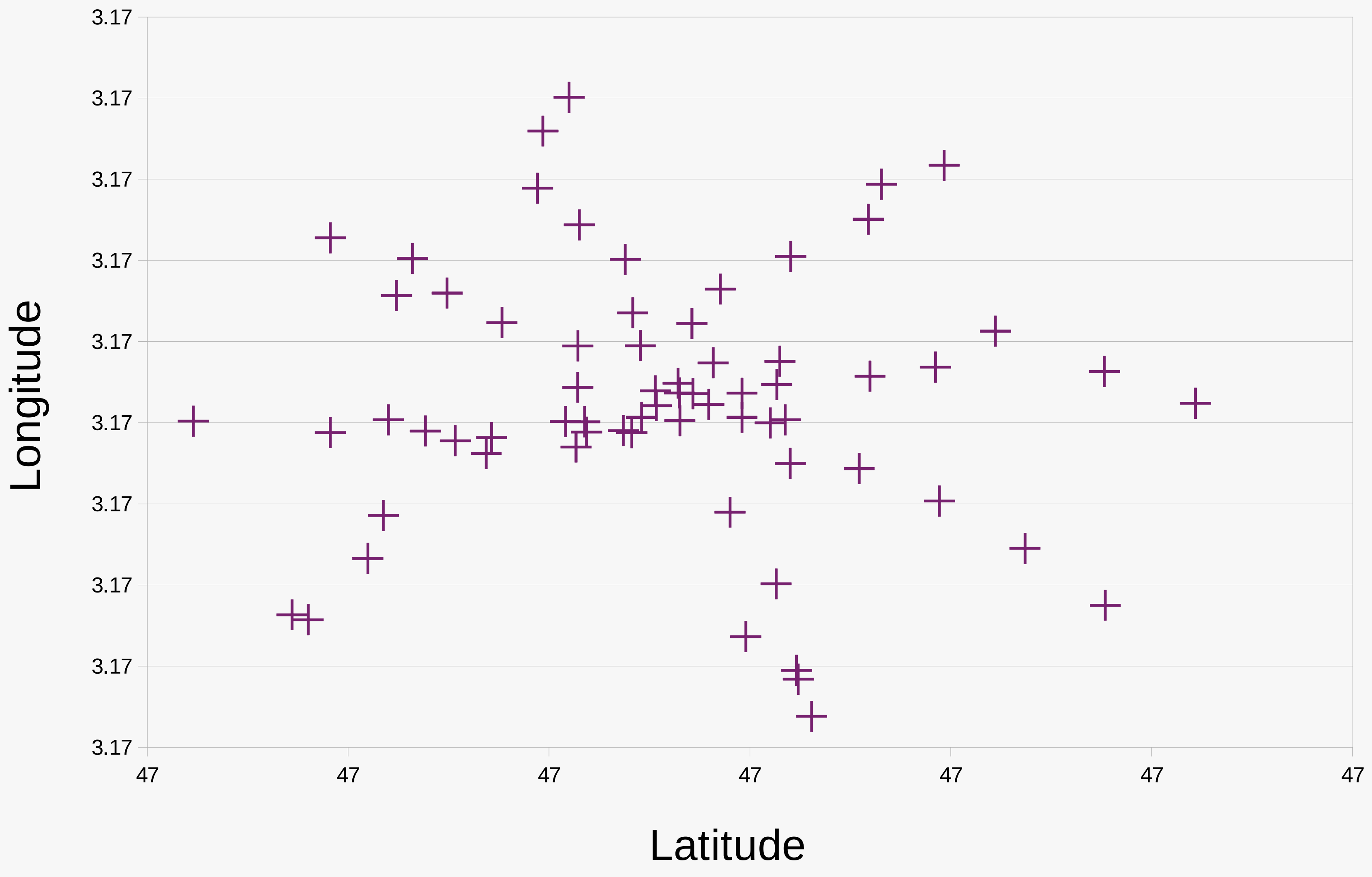


*Errors: 3 to 9 meters*

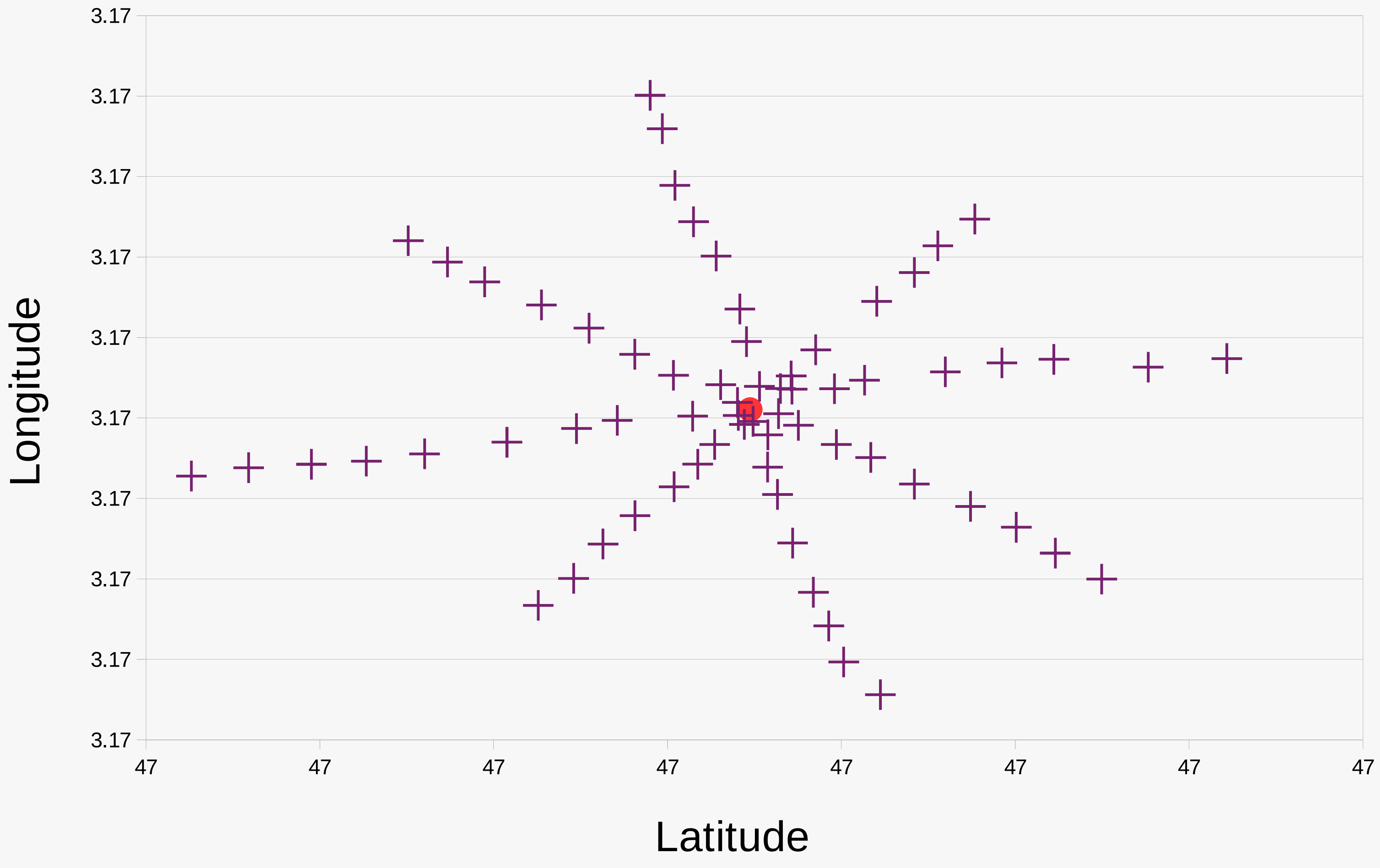
# Plan area

*Location*

Smartphone Location



RTK Location

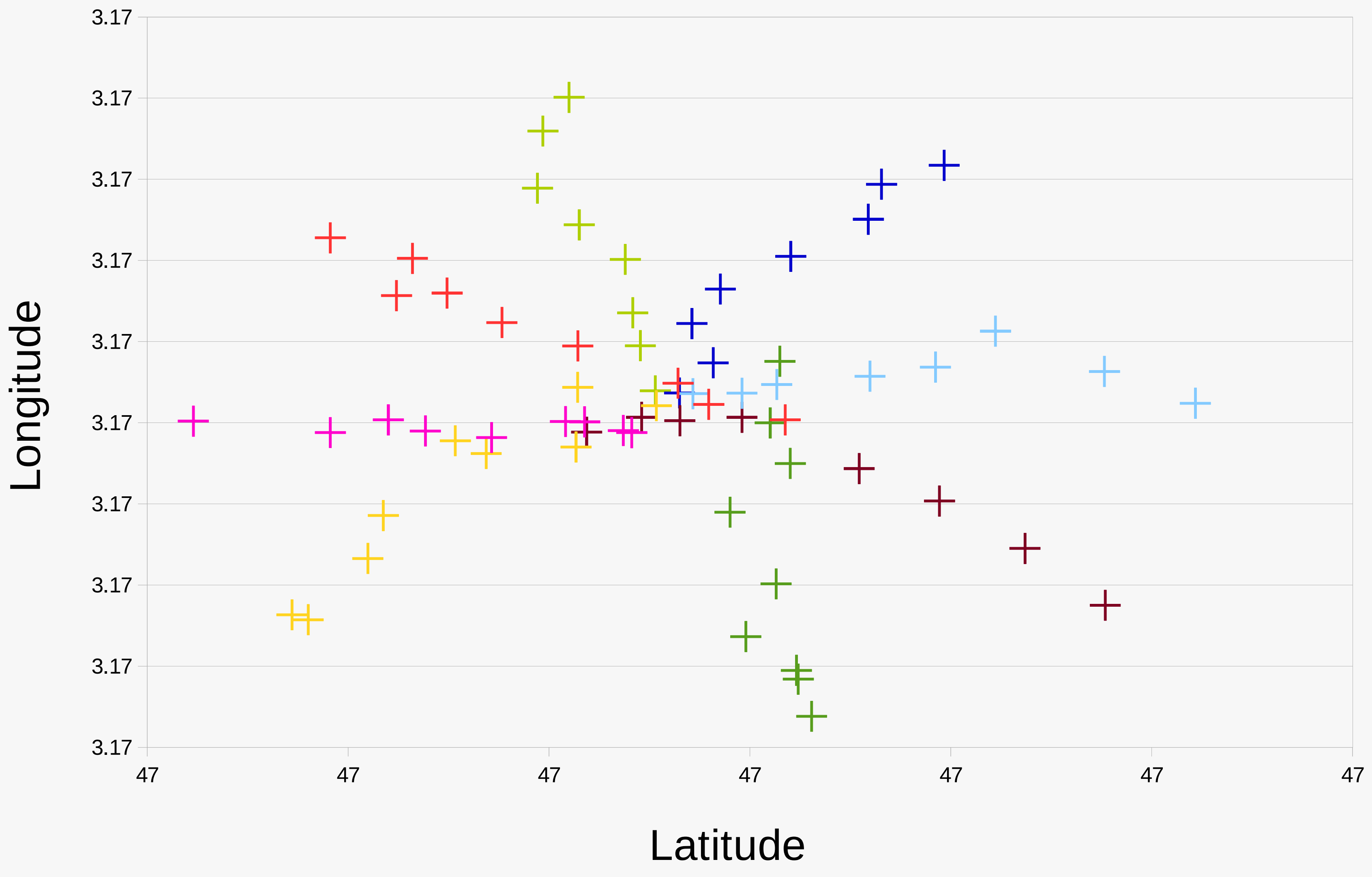


# Plan area

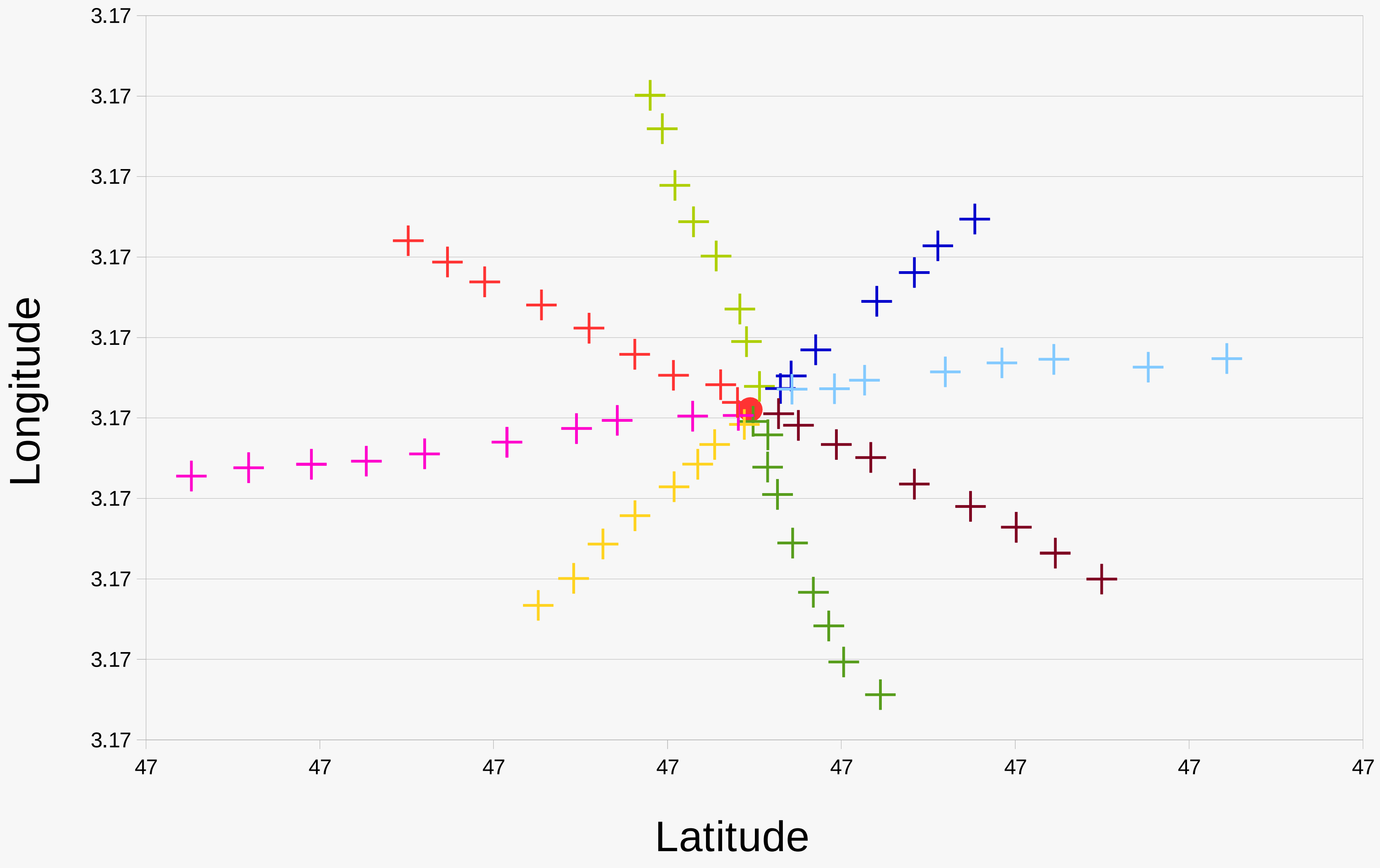
*Location*

*Sunny day*

Smartphone Location



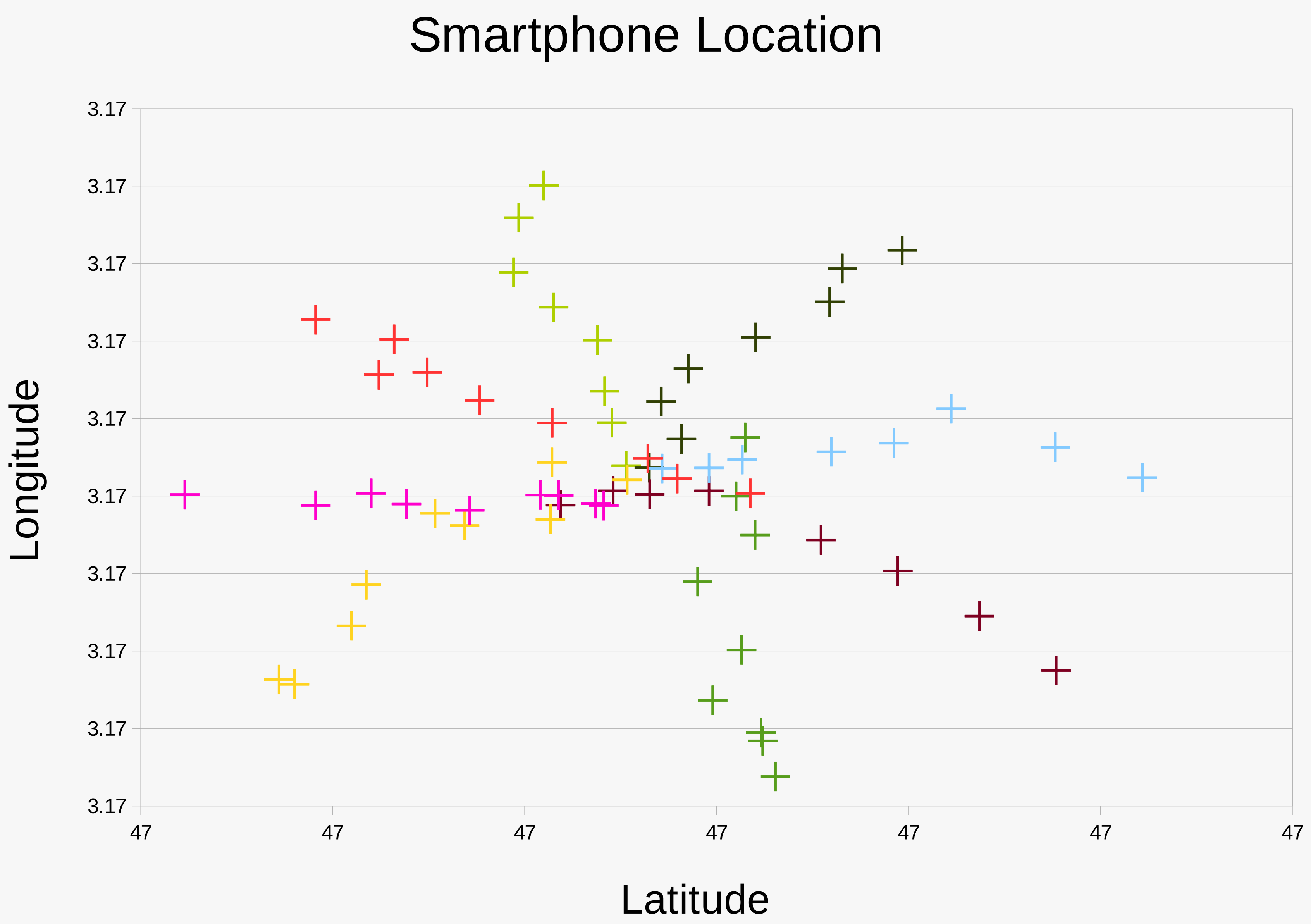
RTK Location



# Plan area

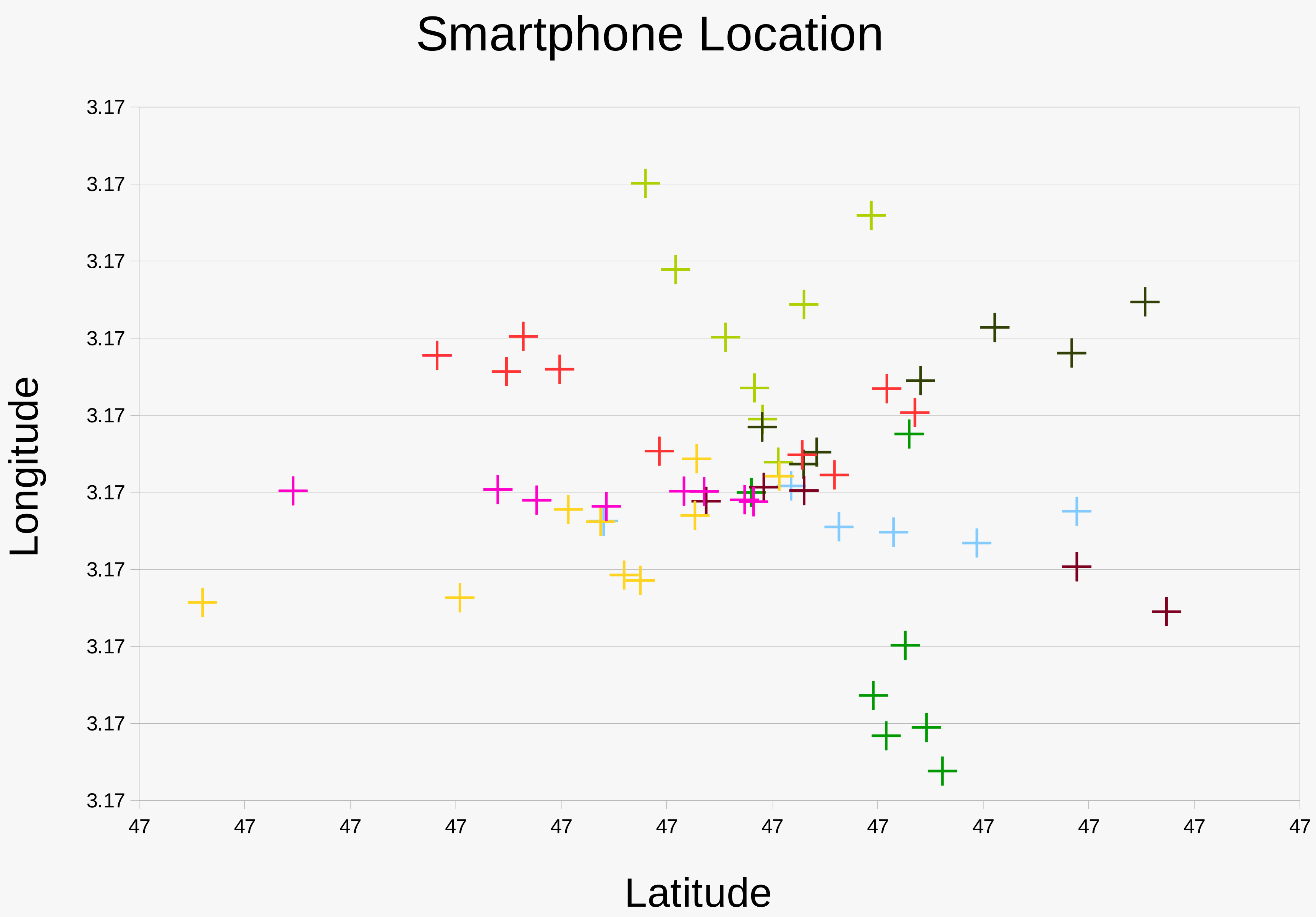
## *Location*

*Sunny day*



*Errors: 2 to 3 meters*

*Cloudy day*



*Errors: 2 to 6 meters*

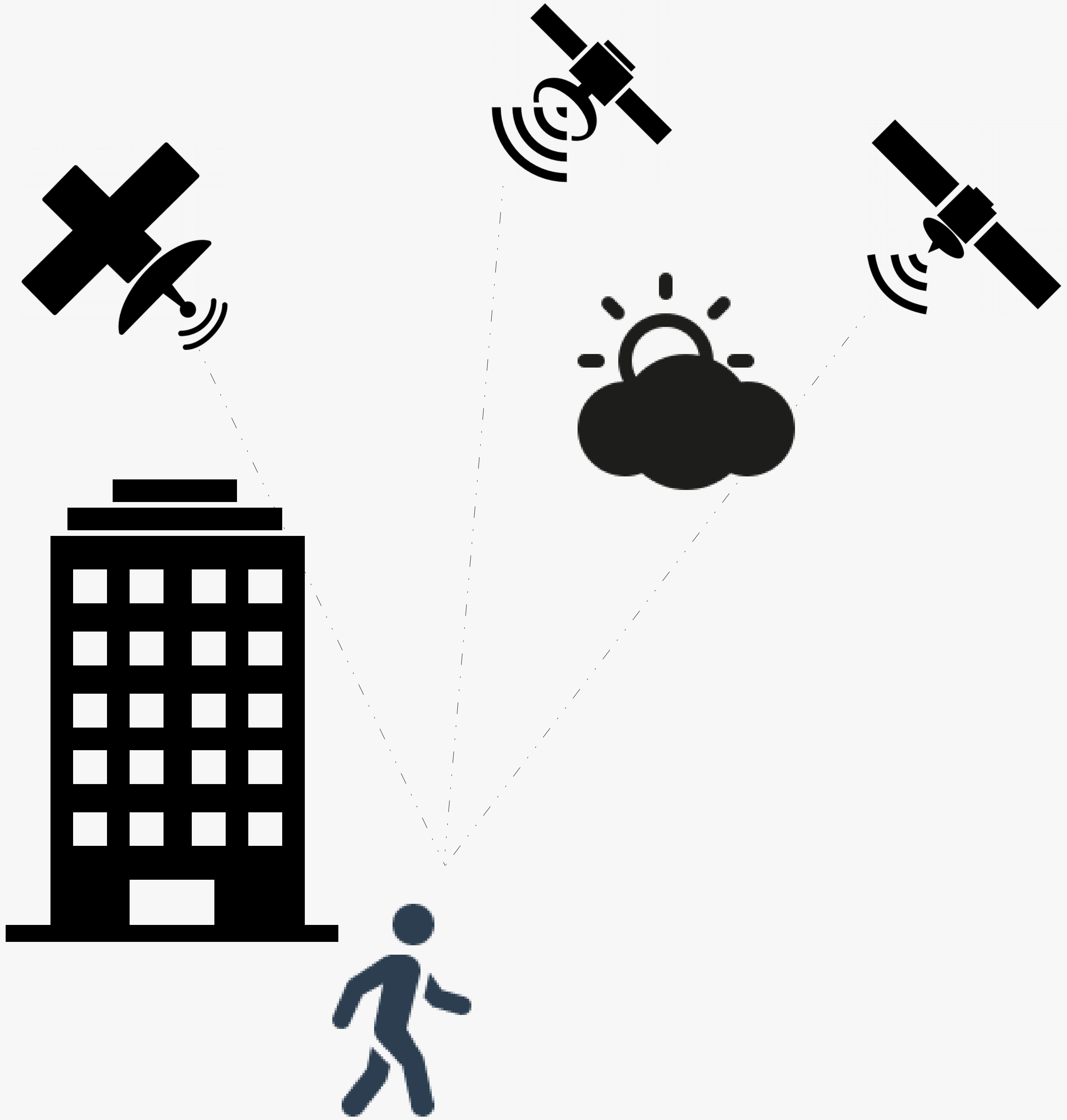
# GPS errors

*Area and weather condition*

| GPS Errors | Urban area | Plan area |
|---|---|---|
| Sunny day | 2 to 7 meters | 2 to 3 meters |
| Cloudy day | 3 to 9 meters | 2 to 6 meters |

# GPS accuracy

*Area and weather condition*



*GPS accuracy depends on:*

*- Signal strength*

*- Weather condition*

*- Building obstacle*

*- Noise and interference*

*How can we increase GPS accuracy ?*

*- Hardware-based approach:*

*- Eliminate noise and interference (signal processing)*

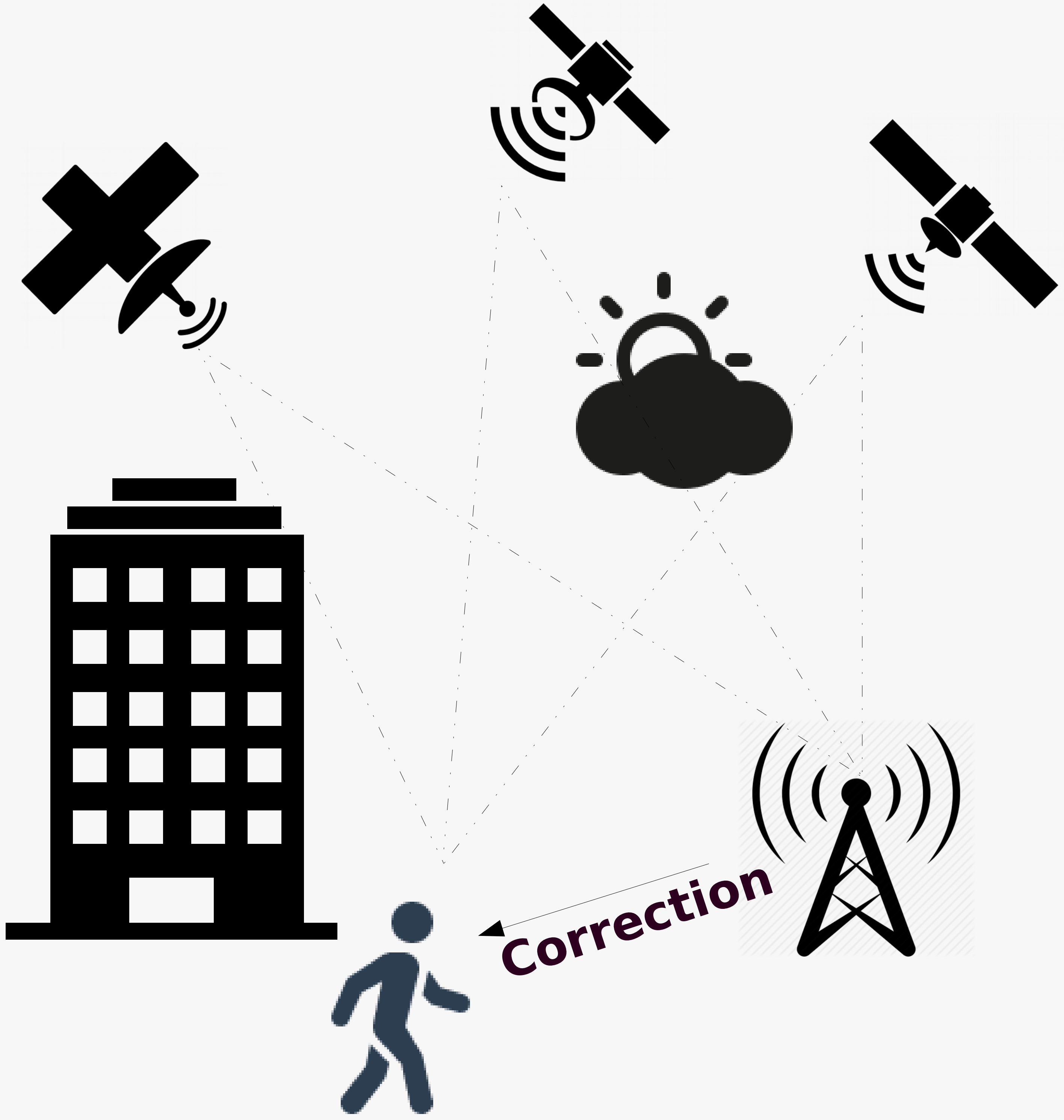
*- Deferential GPS (Base station approach)*

*- Software-based approach:*

*- Use map information to correct the location*

# Hardware-based approach

*Deferential GPS*



*GPS accuracy depends on:*

*- Signal strength*

*- Weather condition*

*- Building obstacle*

*- Noise and interference*

*How can we increase GPS accuracy ?*

*- Hardware-based approach:*

*- Eliminate noise and interference (signal processing)*

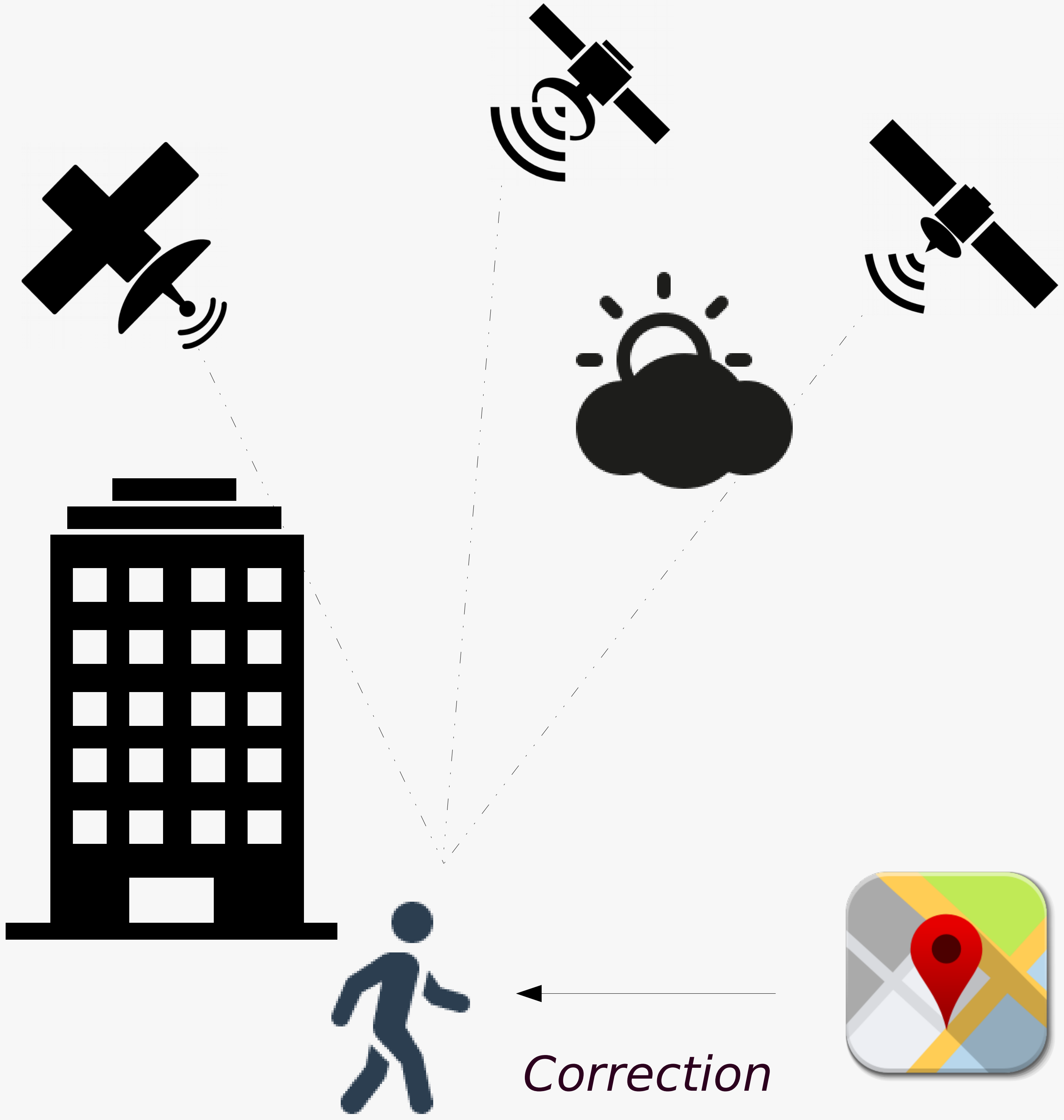
*- Deferential GPS (Base station approach)*

*- Software-based approach:*

*- Use map information to correct the location*

# Software-based approach

*Map-matching approach*



*GPS accuracy depends on:*

*- Signal strength*

*- Weather condition*

*- Building obstacle*

*- Noise and interference*

*How can we increase GPS accuracy ?*

*- Hardware-based approach:*

*- Eliminate noise and interference (signal processing)*

*- Deferential GPS (Base station approach)*

*- Software-based approach:*

*- Use map information to correct the location*

# Idea

Increase horizontal accuracy

# How to increase horizontal accuracy

*2 steps*

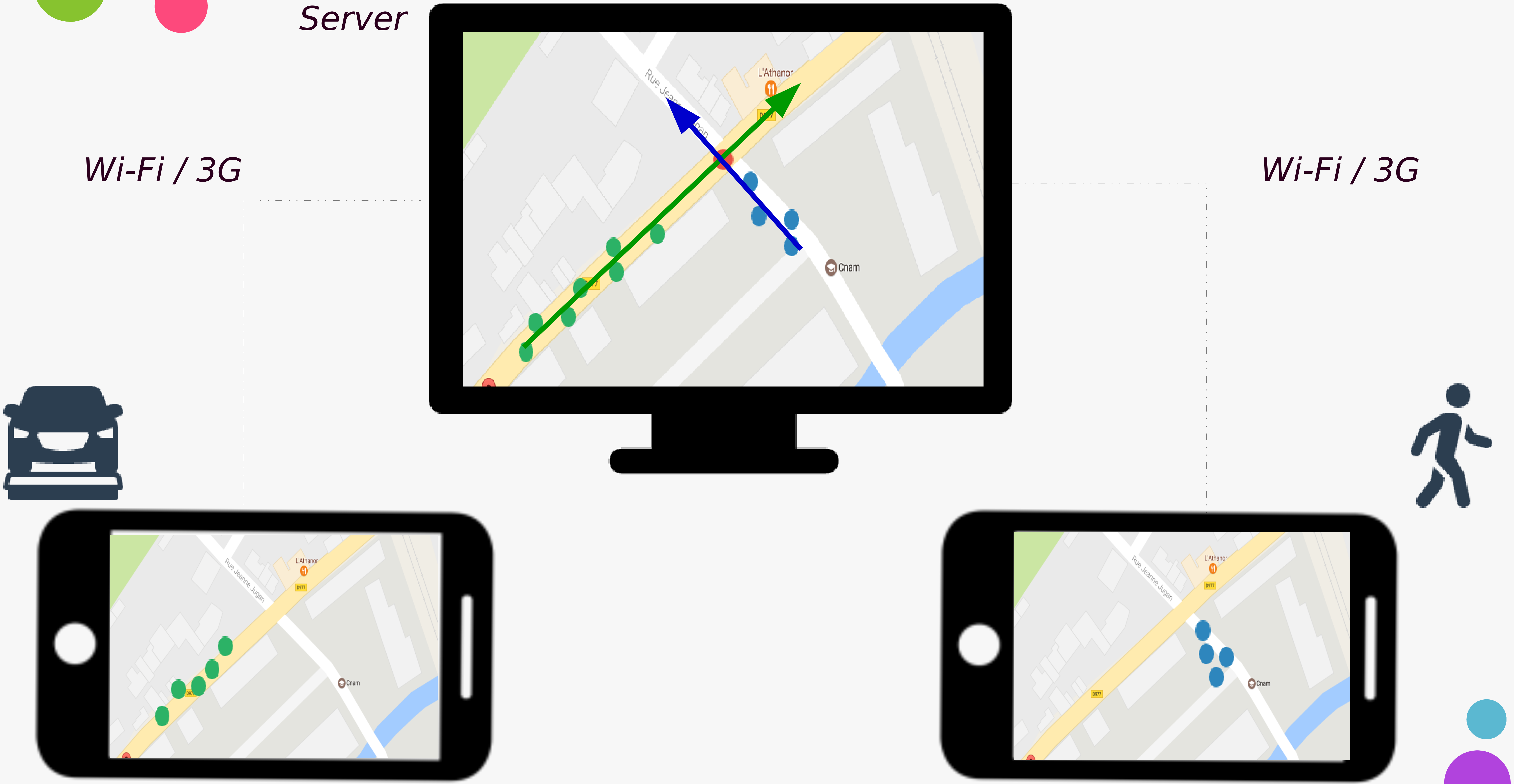
*Step 1:*

*- Identify the direction of the pedestrian*

*Step 2:*

*- Reduce location prediction error based on this direction*

# Step 1: Direction errors

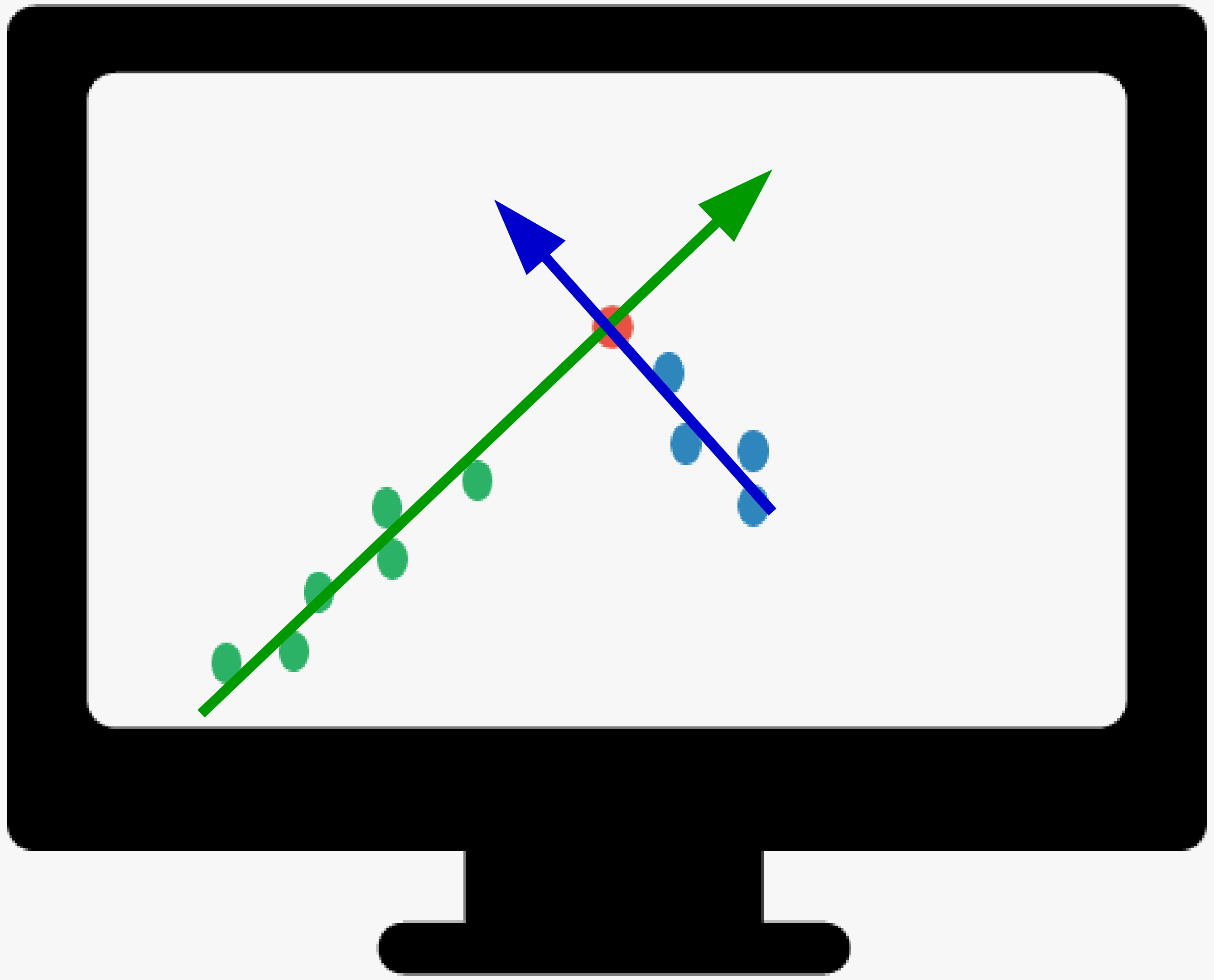


Server

Wi-Fi / 3G

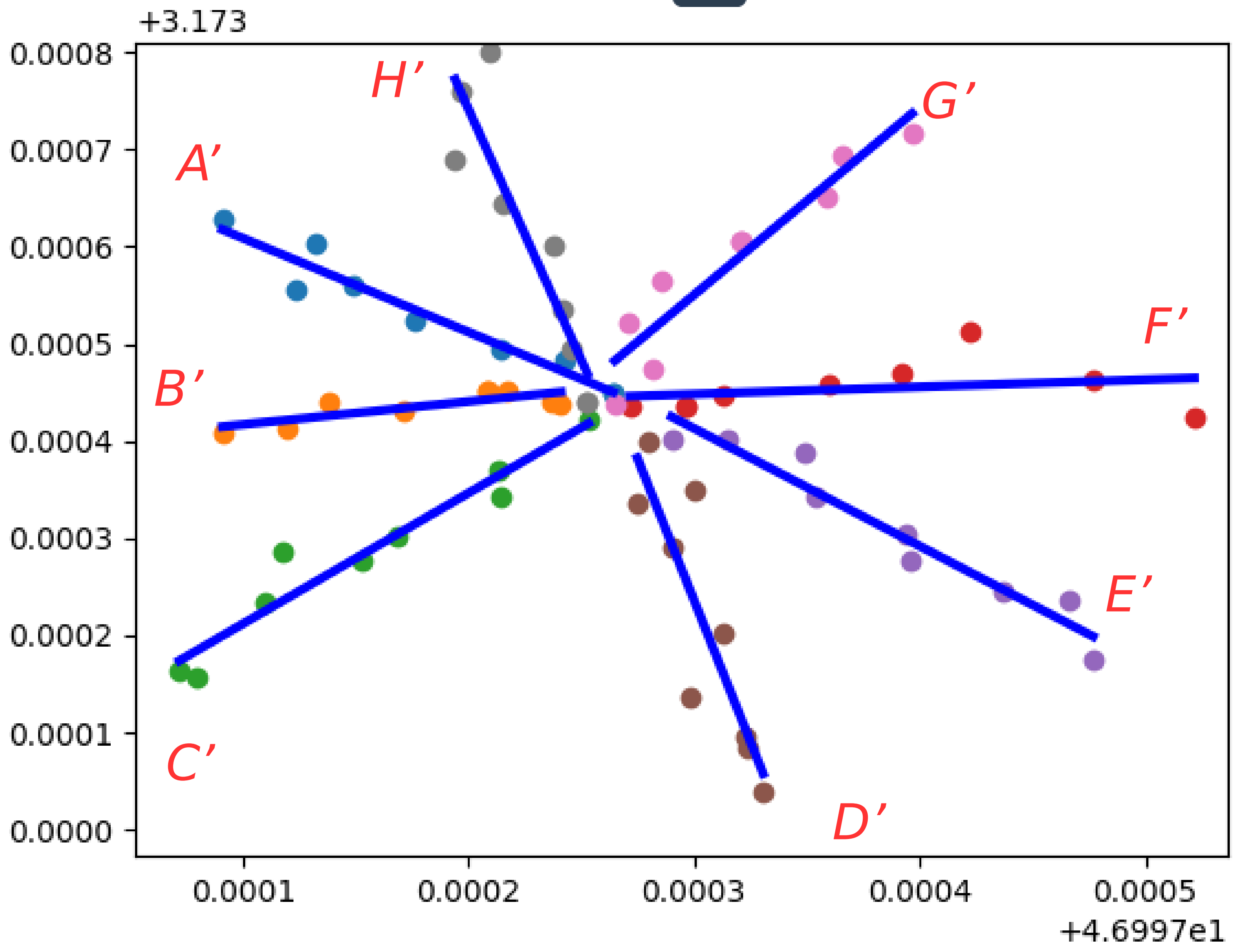
Wi-Fi / 3G

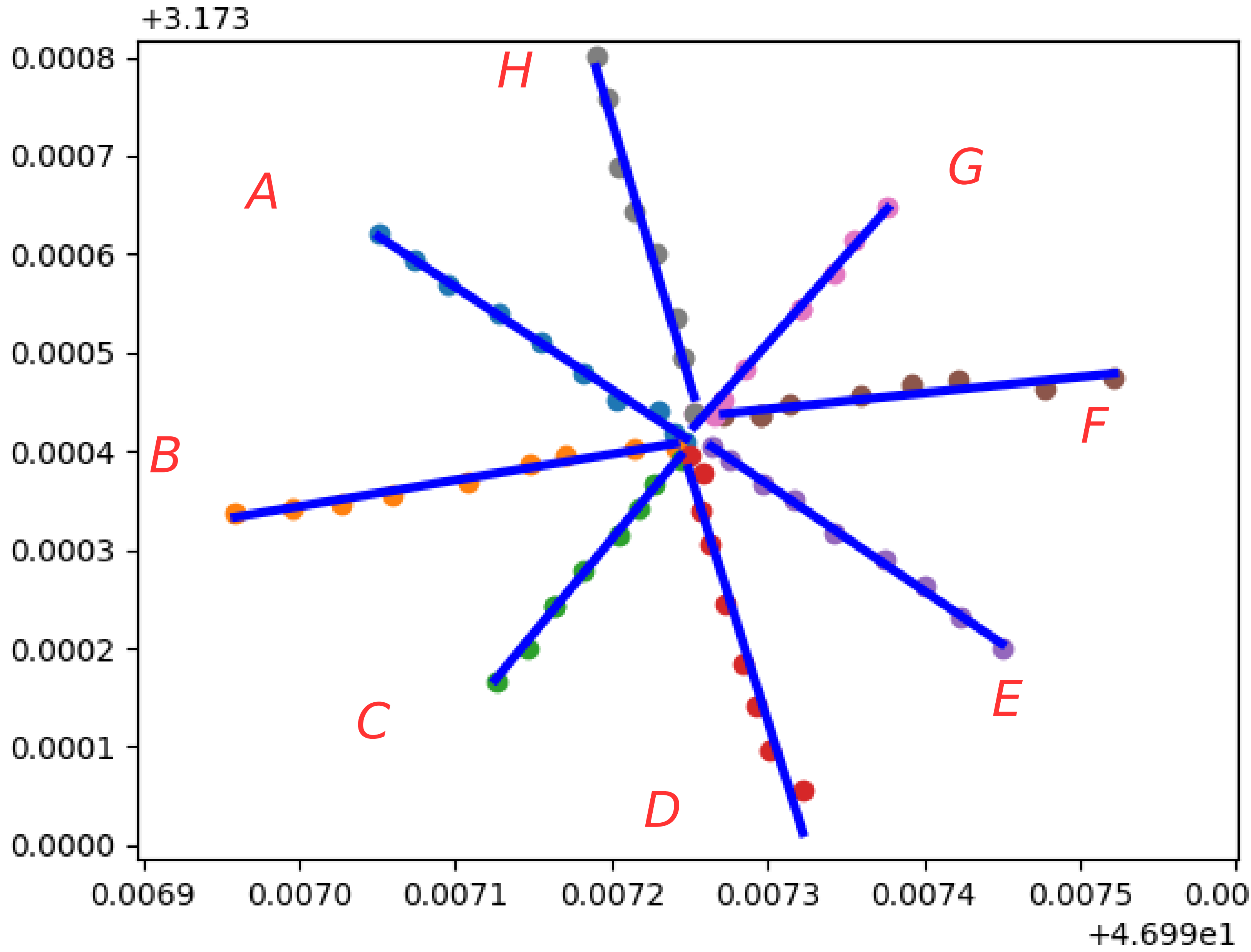
# Linear regression algorithm



# Step 1: Direction errors in plane area

*Linear regression algorithm*





# Step 1: Direction errors in plane area

*RTK*

| Direction | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -1.0350107 | 0.26566971 | 1.91619518 | 0.16367673 | -1.0783095 | -5.0405544 | 1.78695061 | -5.3157862 |

# Step 1: Direction errors in plane area

*RTK*

| Direction | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -1.0350107 | 0.26566971 | 1.91619518 | 0.16367673 | -1.0783095 | -5.0405544 | 1.78695061 | -5.3157862 |

*Smart-phone*

| Direction | A’ | B’ | C’ | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -0.9666001 | 0.23728145 | 1.34641352 | 0.07461613 | -1.2108362 | -5.8262457 | 1.93176513 | -5.1664880 |

# Step 1: Direction errors in plane area

*RTK*

| Direction | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -1.0350107 | 0.26566971 | 1.91619518 | 0.16367673 | -1.0783095 | -5.0405544 | 1.78695061 | -5.3157862 |

*Smart-phone*

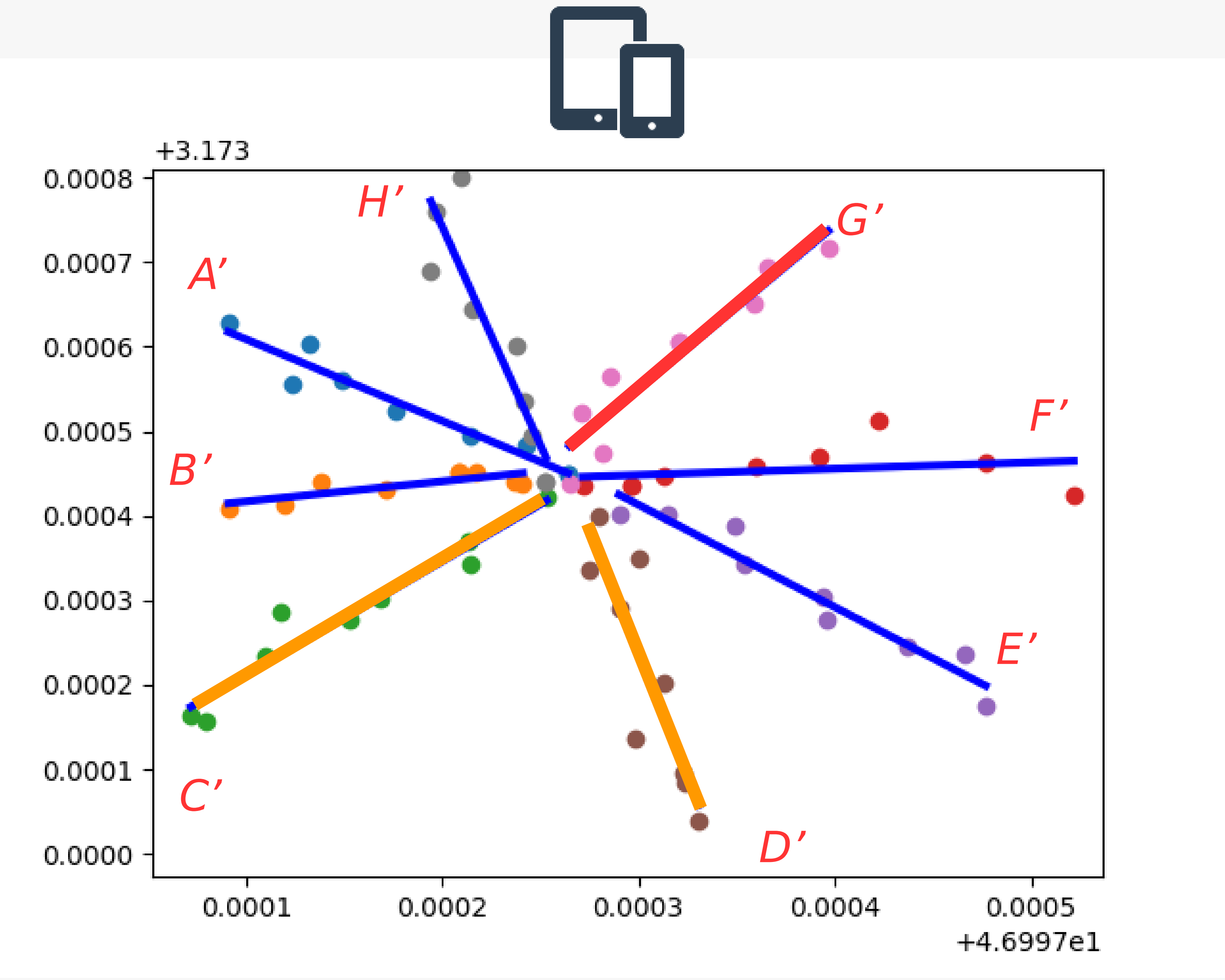
| Direction | A’ | B’ | C’ | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -0.9666001 | 0.23728145 | 1.34641352 | 0.07461613 | -1.2108362 | -5.8262457 | 1.93176513 | -5.1664880 |

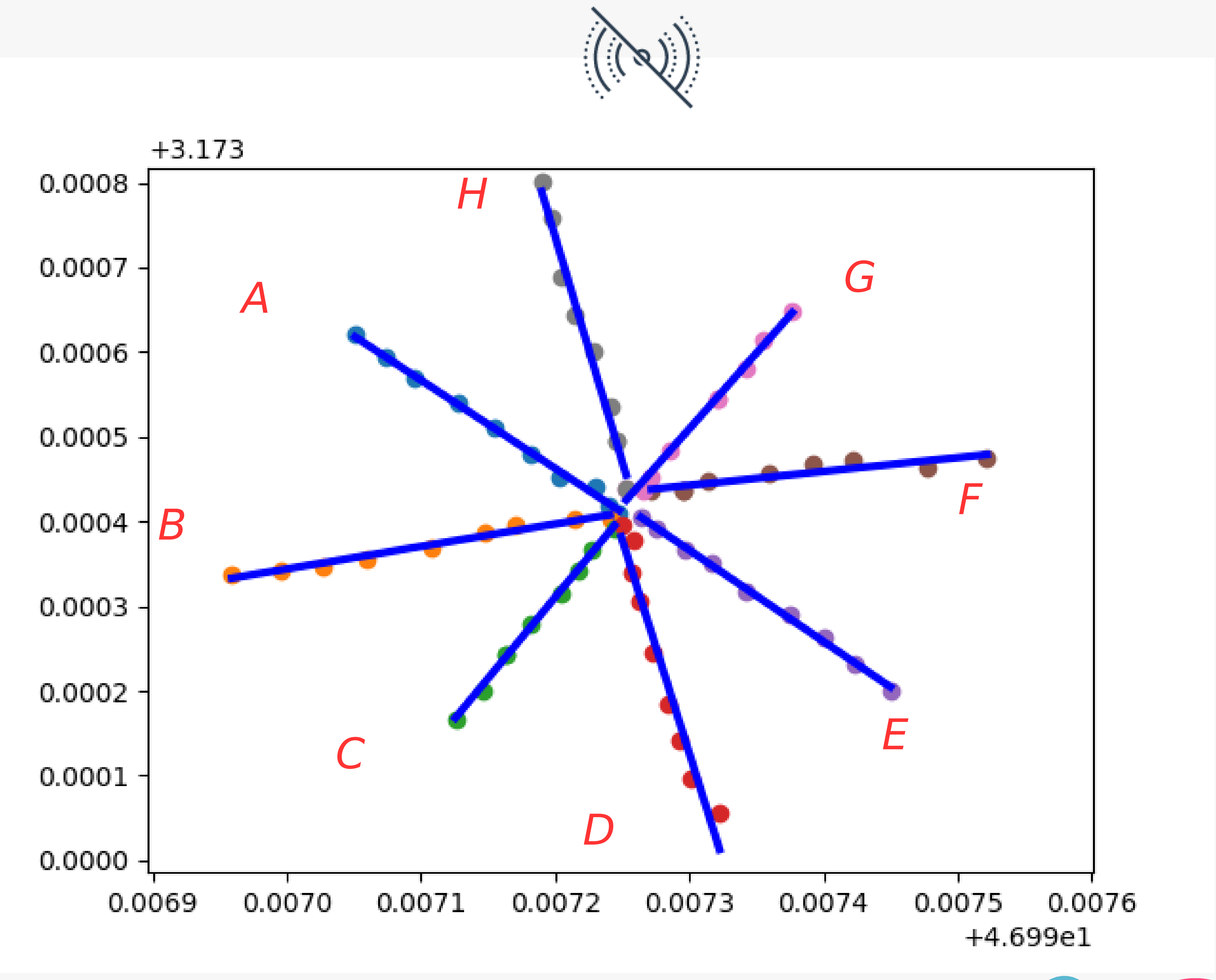
*Difference*

| Direction | A - A’ | B - B’ | C - C’ | D - D’ | E - E’ | F - F’ | G - G’ | H - H’ |
|---|---|---|---|---|---|---|---|---|
| Coefficient | 0,06841052 | 0,0283882 | 0,5697816 | 0,7856913 | 0,1325266 | 0,0890606 | 1,7869506 | 0,14929826 |

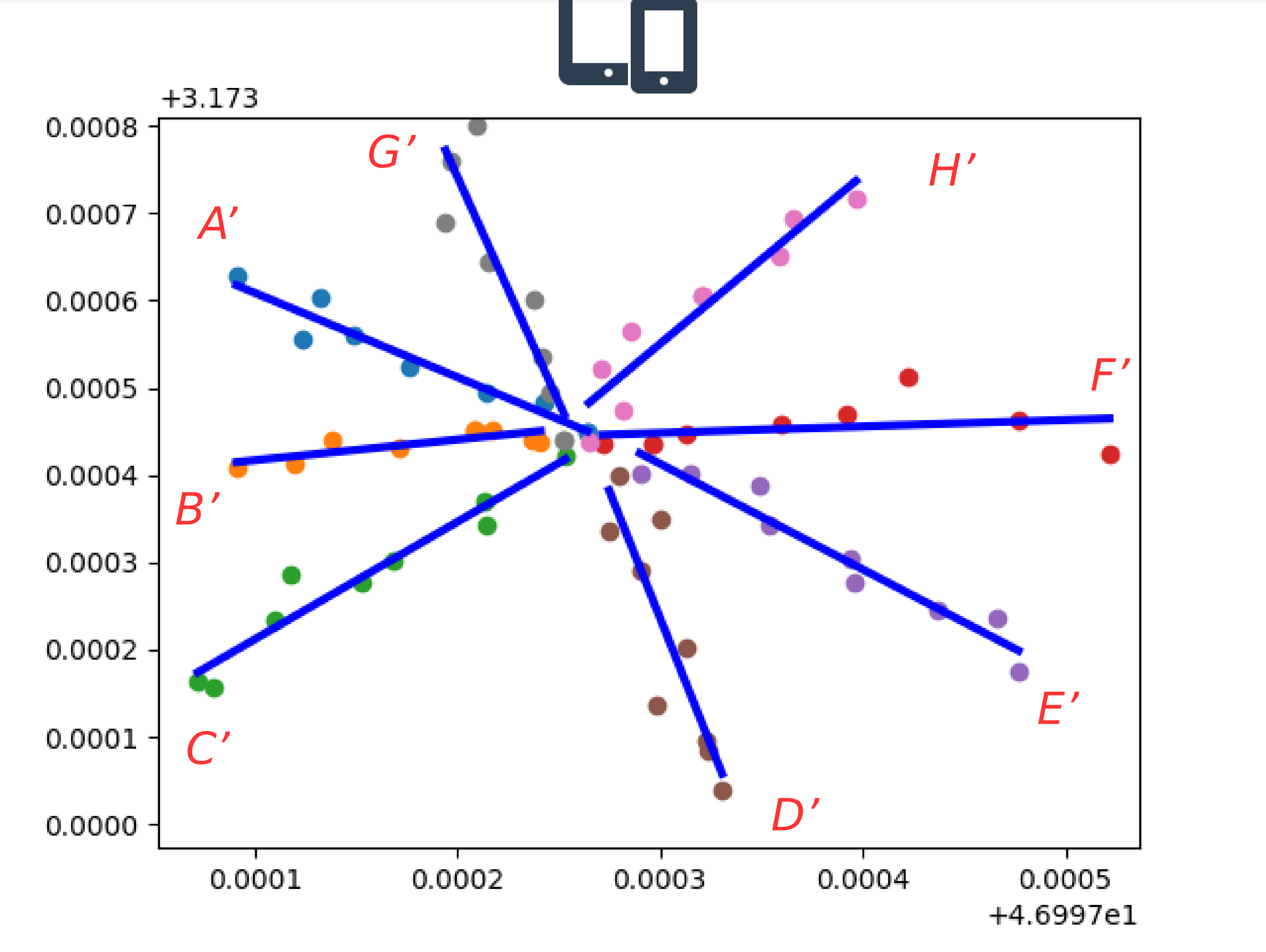
# Step 1: Direction errors in plane area

*Linear regression algorithm*





# Step 2: Location errors in plane area



# Step 2: Location errors in plane area

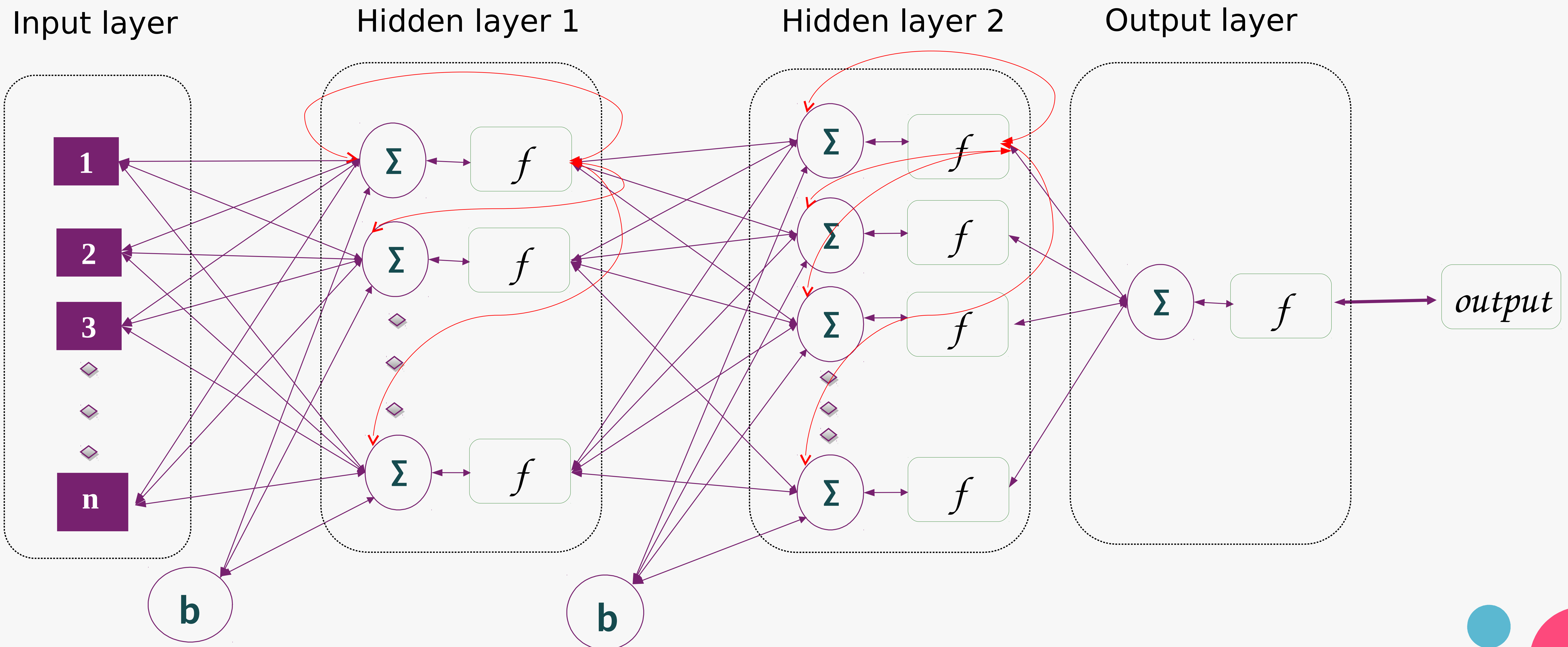
*RTK*

| Error | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| $R^2$ | 0.99550775 | 0.96944446 | 0.99719662 | 0.95419267 | 0.99730340 | 0.89937049 | 0.99230581 | 0.98588831 |

*Smart-phone*

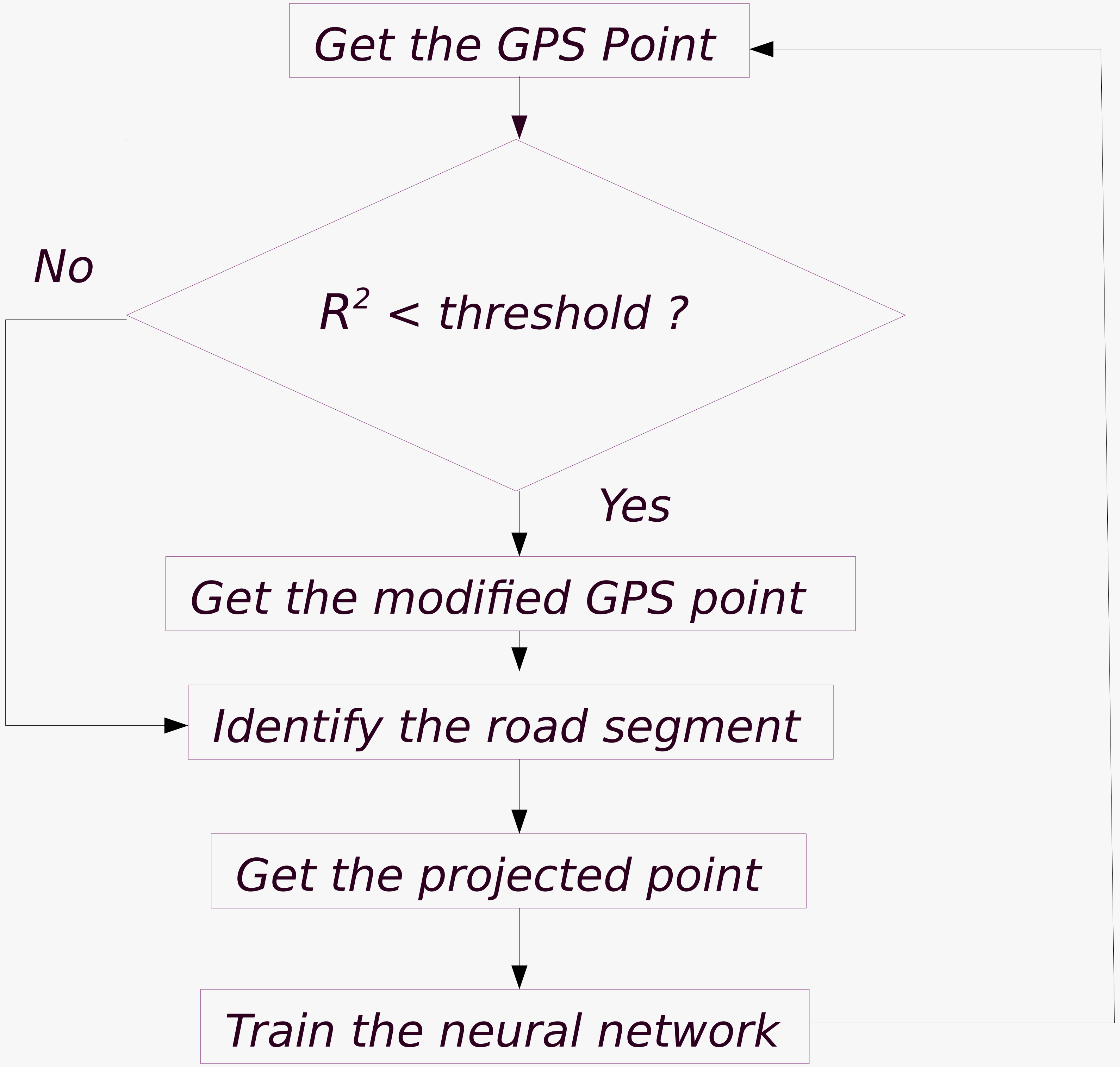
| Error | A’ | B’ | C' | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| $R^2$ | 0.94396797 | 0.63719028 | 0.93629317 | 0.06020403 | 0.93676206 | 0.77280729 | 0.91409588 | 0.82847928 |

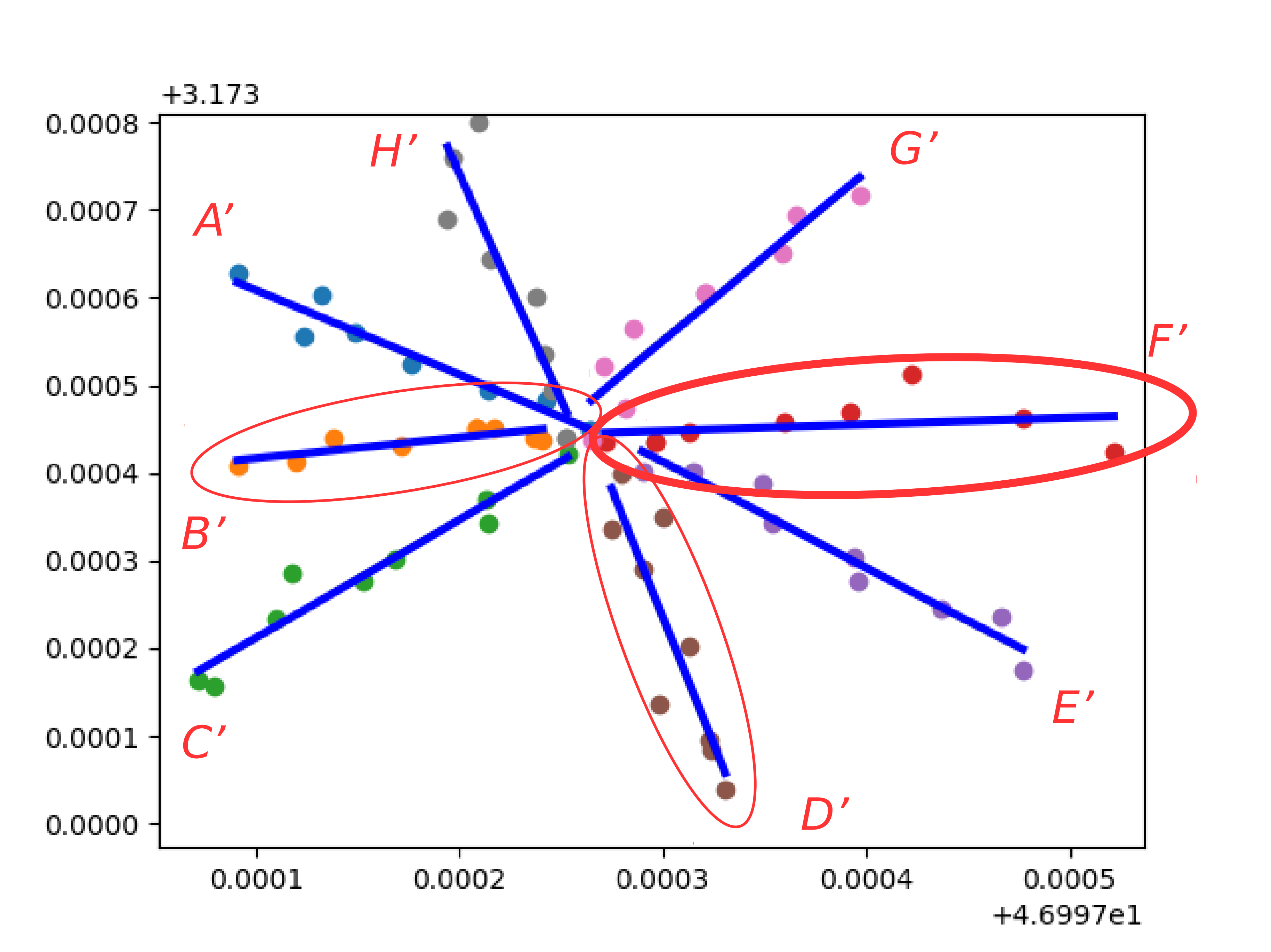
# Artificial neural network



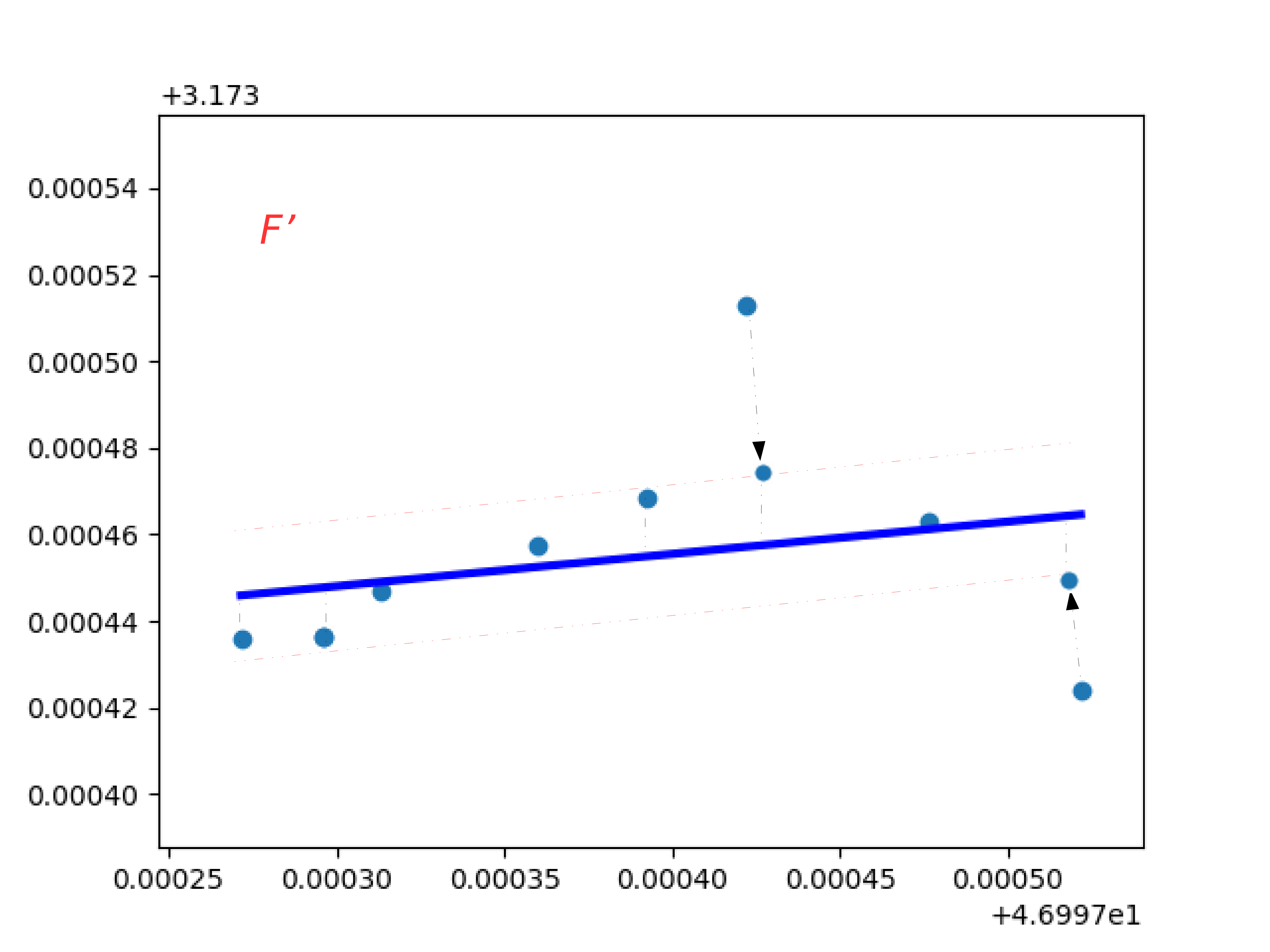
# Step 2: Location errors in plane area

*Map-matching approach*





# Step 2: Location correction in plane area



# Step 2: Location correction in plane area

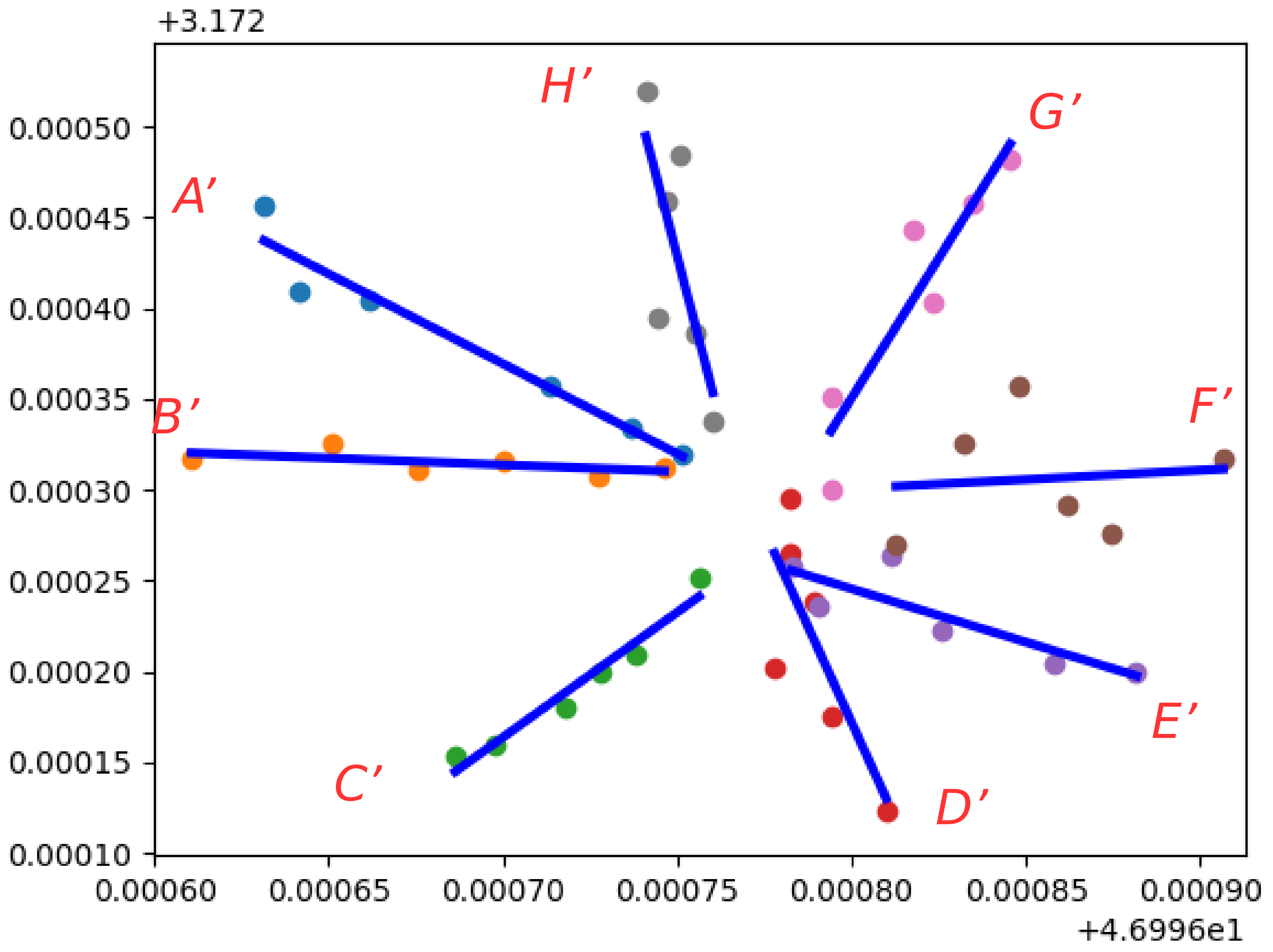
| Error | A’ | B’ | C' | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| $R^2$ | 0.94396797 | 0.63719028 | 0.93629317 | 0.06020403 | 0.93676206 | 0.77280729 | 0.91409588 | 0.82847928 |

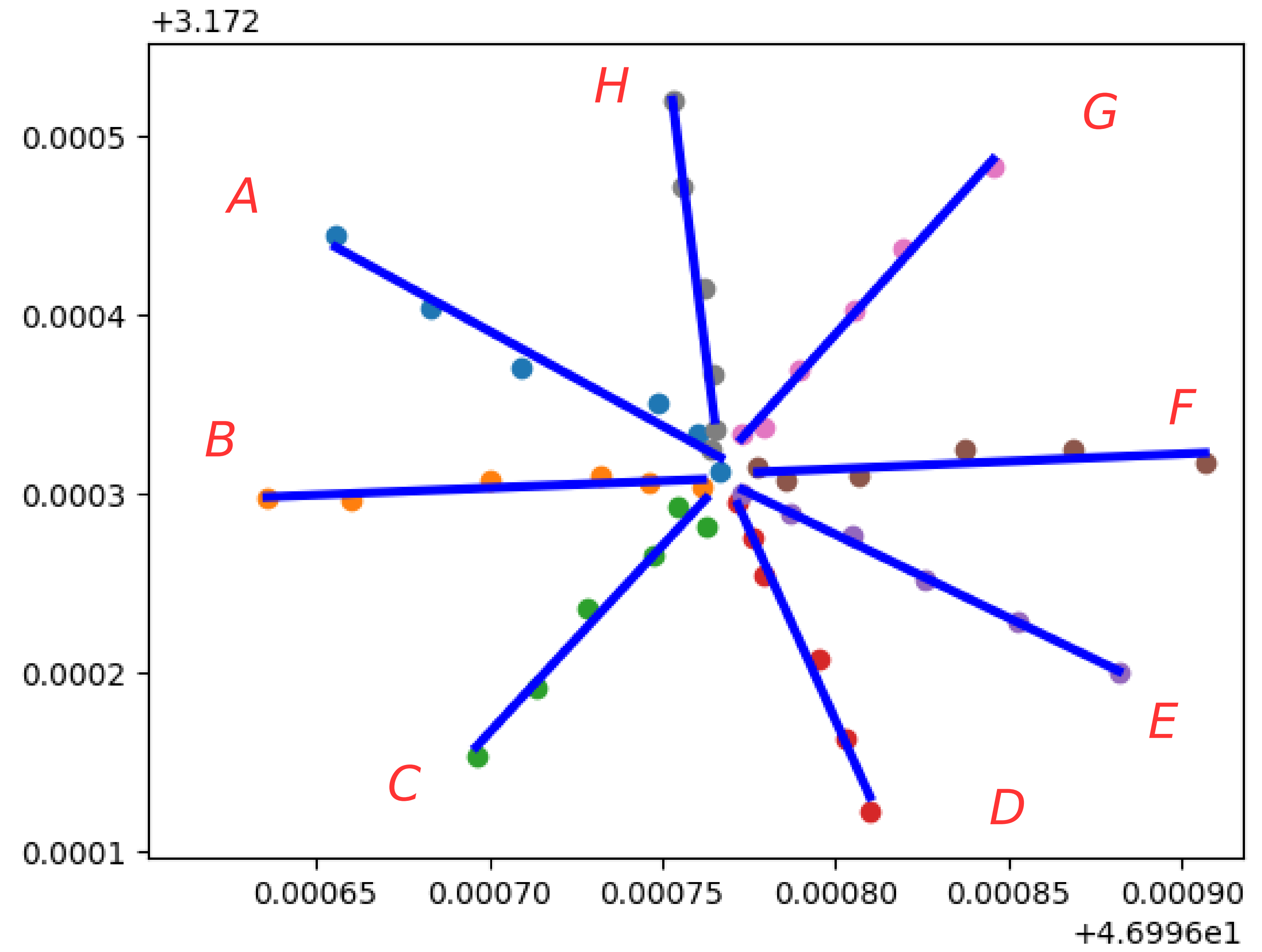
# Step 2: Location correction in plane area

| Error | A’ | B’ | C' | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| $R^2$ | 0.94396797 | 0.63719028 | 0.93629317 | 0.06020403 | 0.93676206 | 0.88961241 | 0.91409588 | 0.82847928 |

# Direction errors in urban area

*Linear regression algorithm*





# Step 1: Direction errors in urban area

*RTK*

| Direction | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -1.0598147 | 0.0781908 | 2.09278067 | -4.2876946 | -0.9337665 | 0.08456471 | 2.14901023 | -14.721285 |

# Step 1: Direction errors in urban area

*RTK*

| Direction | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -1.0598147 | 0.0781908 | 2.09278067 | -4.2876946 | -0.9337665 | 0.08456471 | 2.14901023 | -14.721285 |

*Smart-phone*

| Direction | A’ | B’ | C’ | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -0.9896452 | -0.0737344 | 1.37782101 | -4.2182173 | -0.5864491 | 0.10219062 | 3.07430095 | -7.3396637 |

# Step 1: Direction errors in urban area

*RTK*

| Direction | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -1.0598147 | 0.0781908 | 2.09278067 | -4.2876946 | -0.9337665 | 0.08456471 | 2.14901023 | -14.721285 |

*Smart-phone*

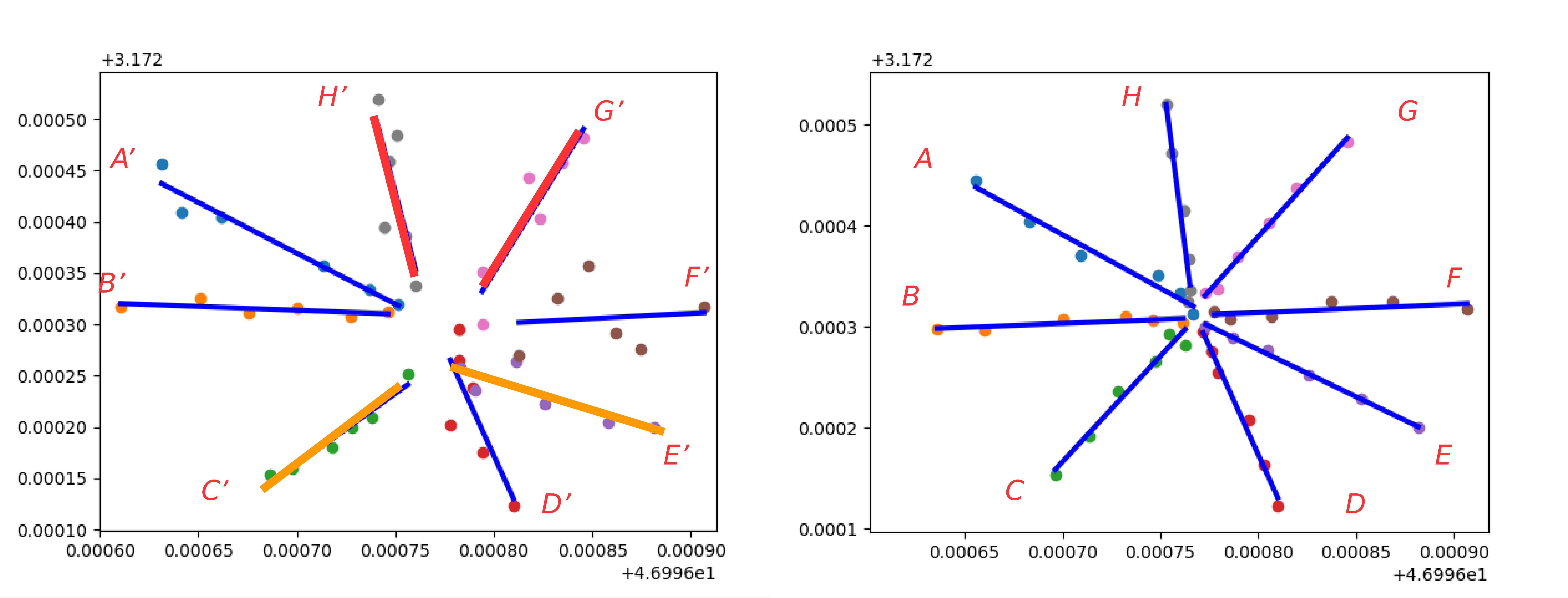
| Direction | A’ | B’ | C’ | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| Coefficient | -0.9896452 | -0.0737344 | 1.37782101 | -4.2182173 | -0.5864491 | 0.10219062 | 3.07430095 | -14.339663 |

*Difference*

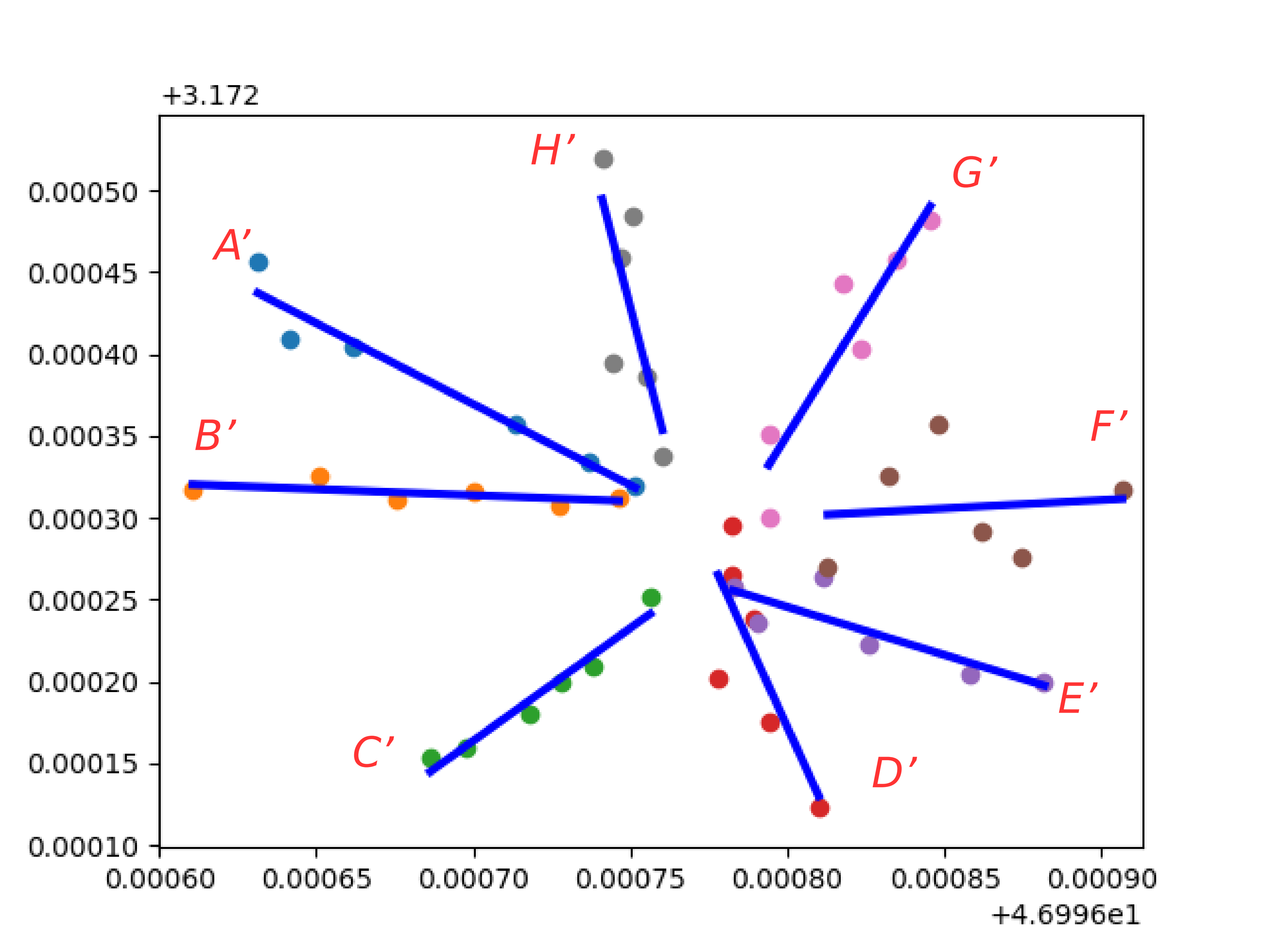
| Direction | A - A’ | B - B’ | C - C’ | D - D’ | E - E’ | F - F’ | G - G’ | H - H’ |
|---|---|---|---|---|---|---|---|---|
| Coefficient | 0,0701695 | 0,15192529 | 0,71495966 | 0,06947737 | 0,3473174 | 0,01762591 | 0,92529072 | 1,38162199 |

# Direction errors in urban area

*Linear regression algorithm*



# Location errors in urban area



# Step 2: Location errors in urban area

*RTK*

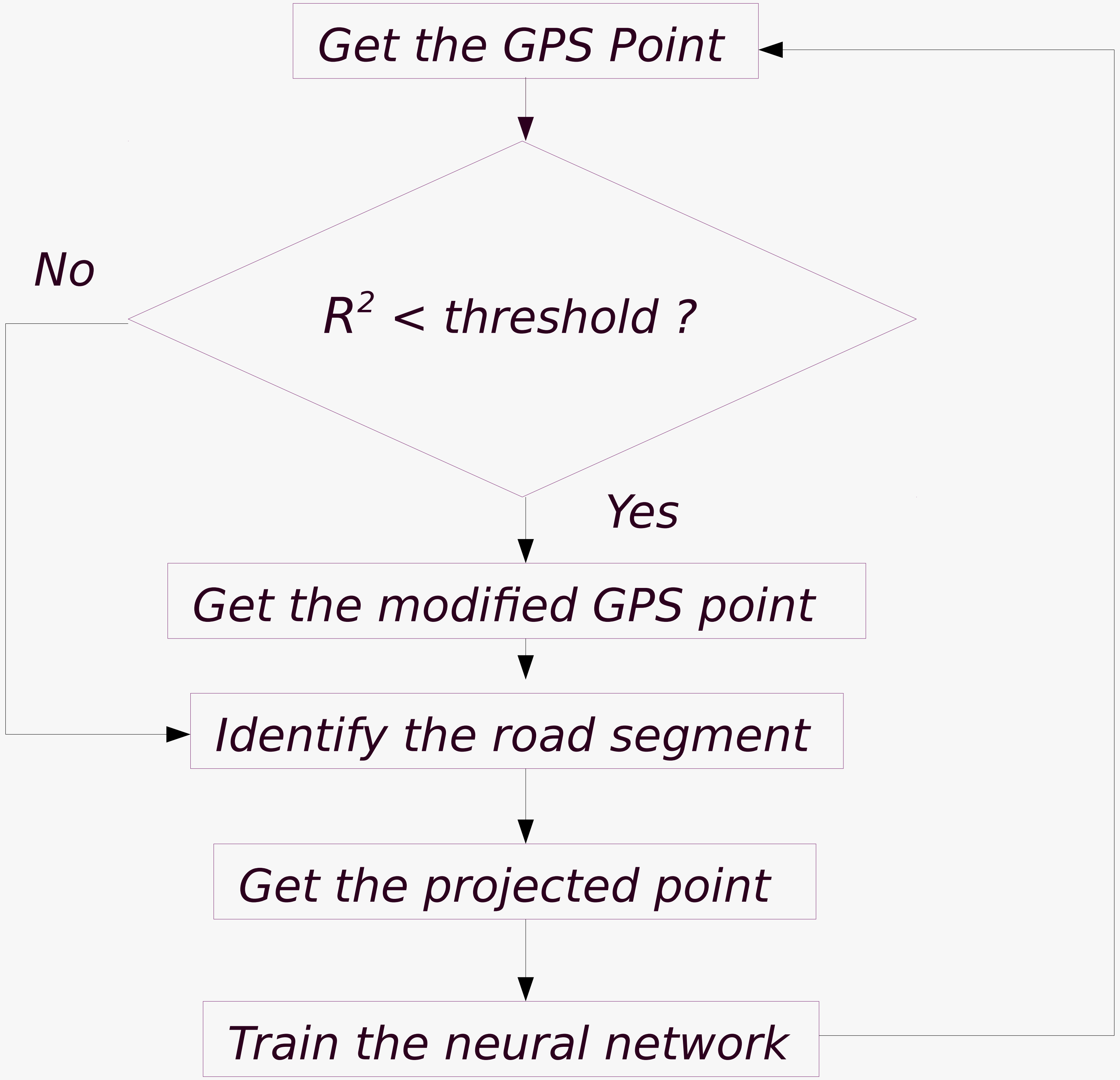
| Error | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| $R^2$ | 0.96518147 | 0.91822714 | 0.96216233 | 0.98503134 | 0.99633253 | 0.93645041 | 0.99287674 | 0.91222112 |

*Smart-phone*

| Error | A’ | B’ | C' | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| $R^2$ | 0.94761938 | 0.84504314 | 0.95482338 | 0.61086726 | 0.73050575 | 0.1025010 | 0.86147849 | 0.56749197 |

# Step 2: Location errors in urban area

*Map-matching approach*



*Get the GPS Point*

*$R^2$ < threshold ?*

*No*

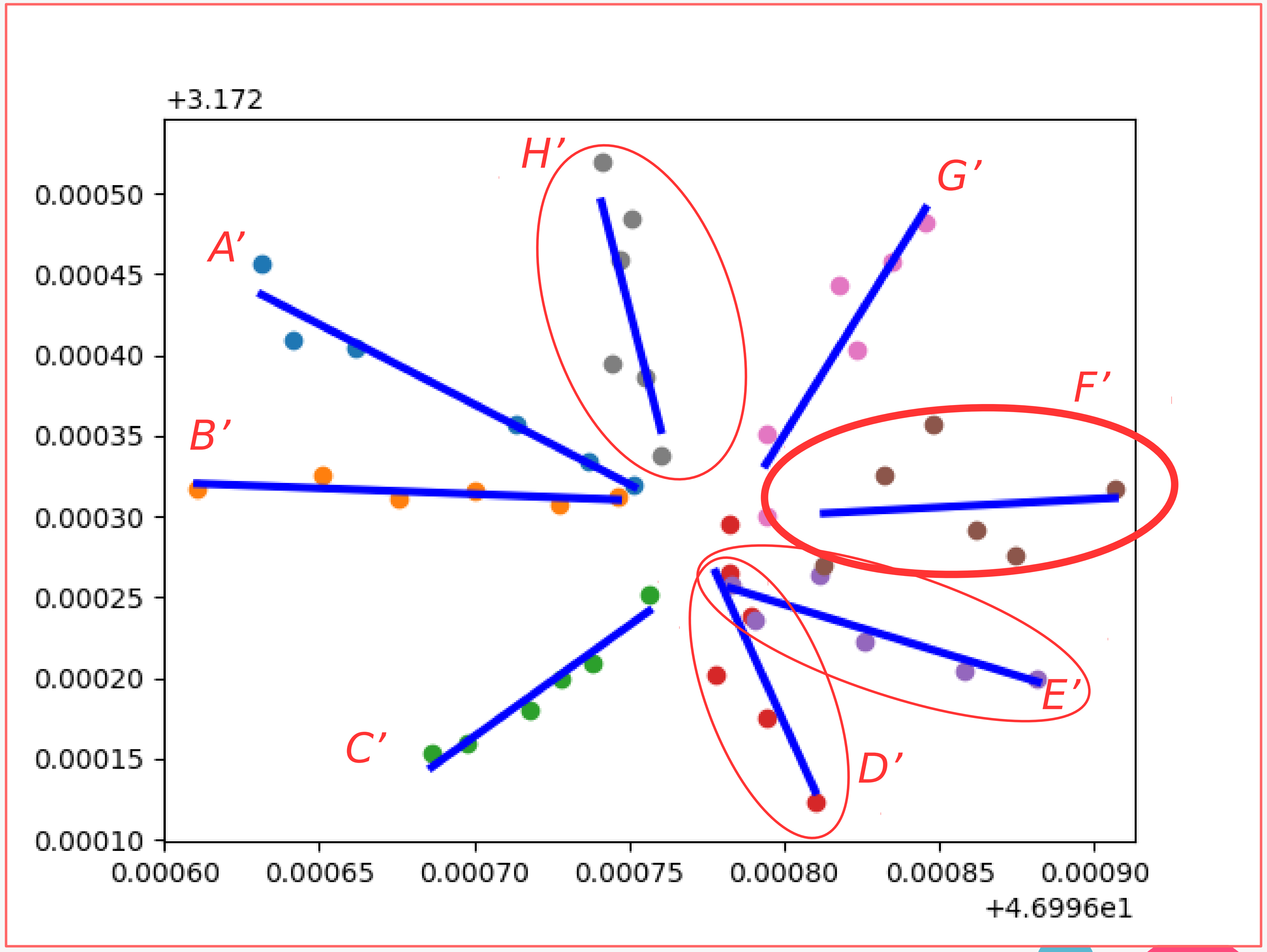
*Yes*

*Get the modified GPS point*

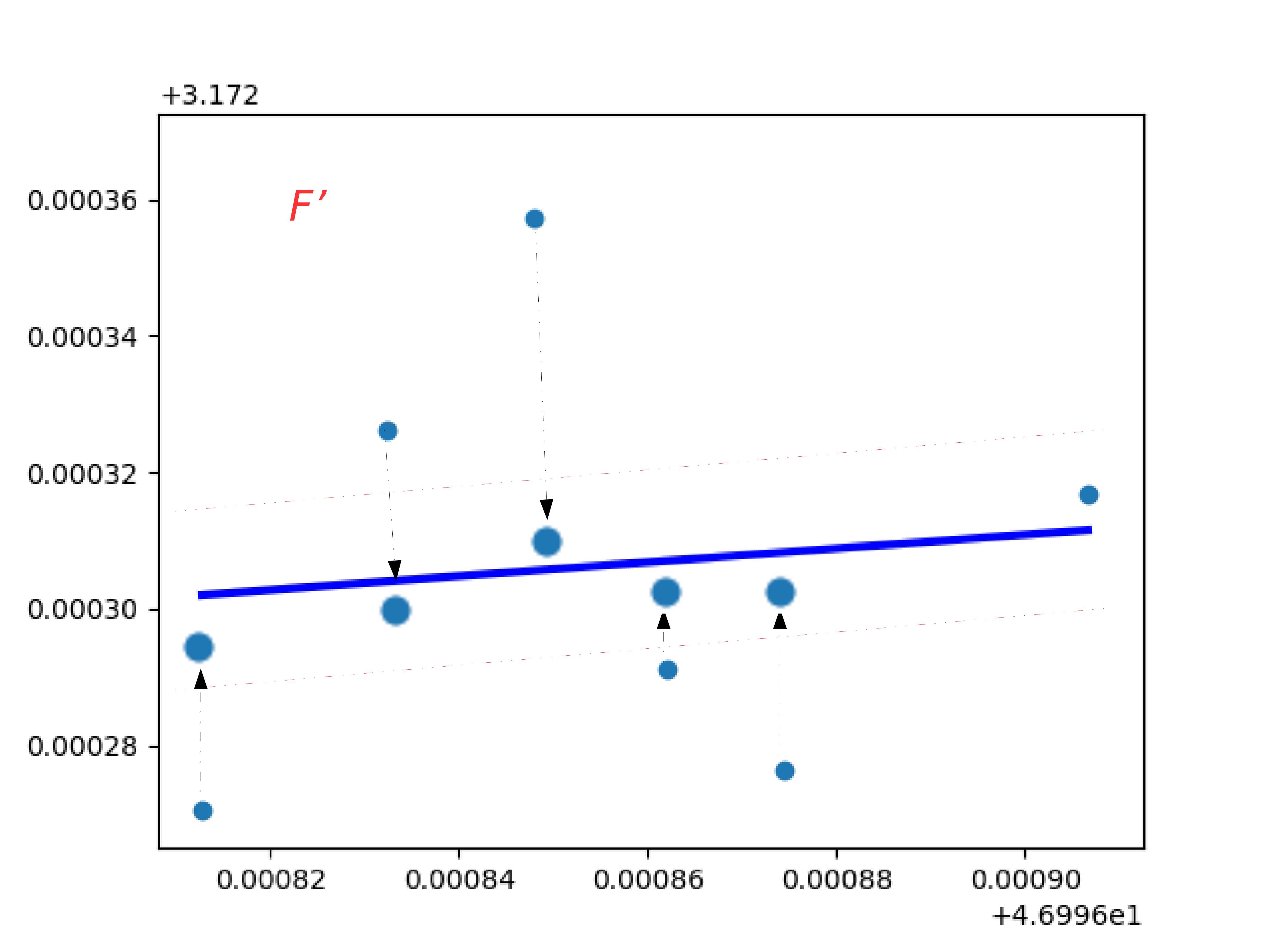
*Identify the road segment*

*Get the projected point*

*Train the neural network*



# Step 2: Location errors in urban area



# Step 2: Location errors in urban area

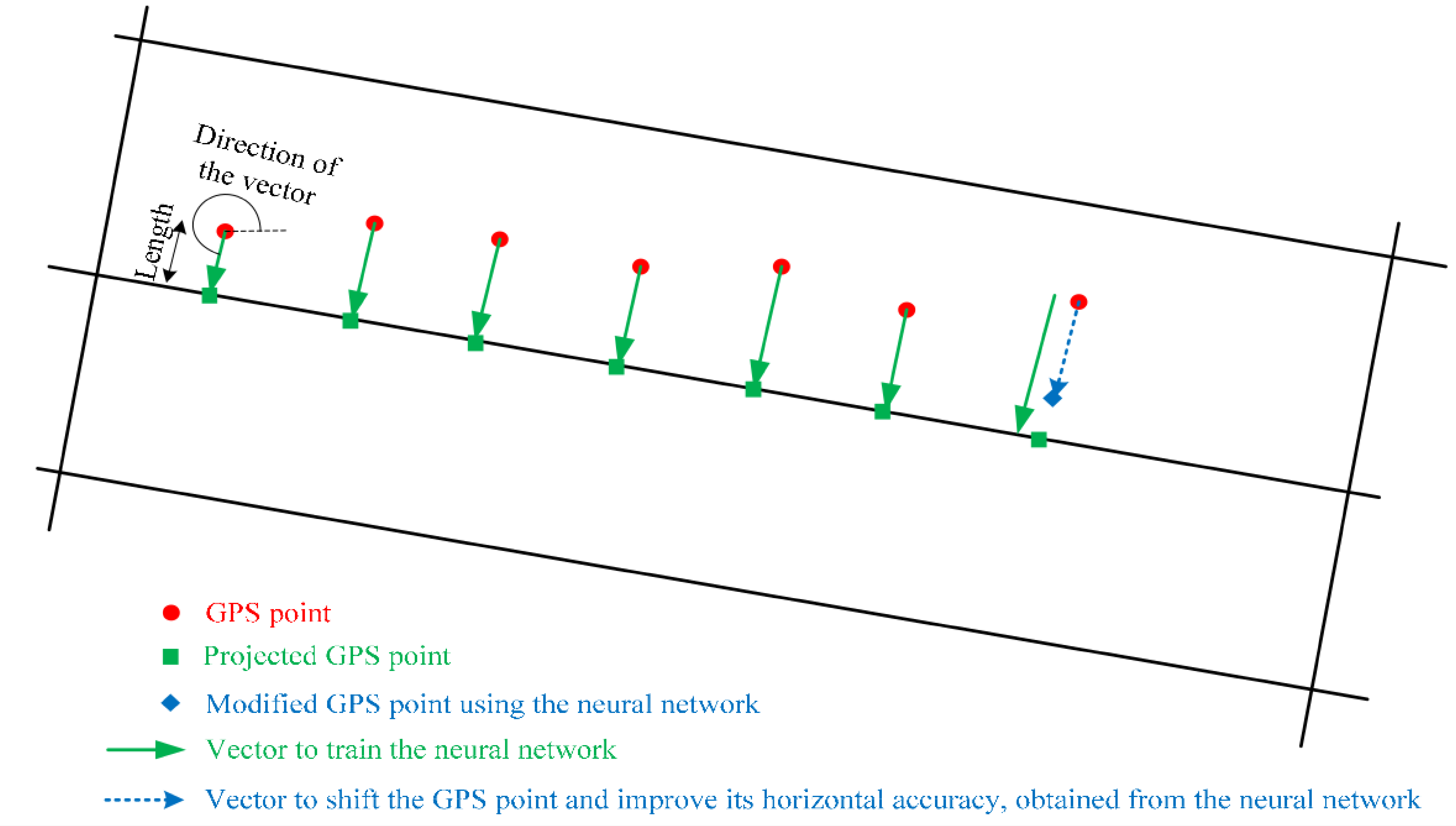
| Error | A’ | B’ | C' | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| $R^2$ | 0.94761938 | 0.84504314 | 0.95482338 | 0.61086726 | 0.73050575 | 0.1025010 | 0.86147849 | 0.56749197 |

# Step 2: Location correction

| Error | A’ | B’ | C' | D’ | E’ | F’ | G’ | H’ |
|---|---|---|---|---|---|---|---|---|
| $R^2$ | 0.94761938 | 0.84504314 | 0.95482338 | 0.61086726 | 0.73050575 | 0.67524985 | 0.86147849 | 0.56749197 |

# Methodology

*Map-matching approach*


Direction of the vector
Length
● GPS point
■ Projected GPS point
◆ Modified GPS point using the neural network
Vector to train the neural network
Vector to shift the GPS point and improve its horizontal accuracy, obtained from the neural network

# Artificial neural network

*Training algorithm*

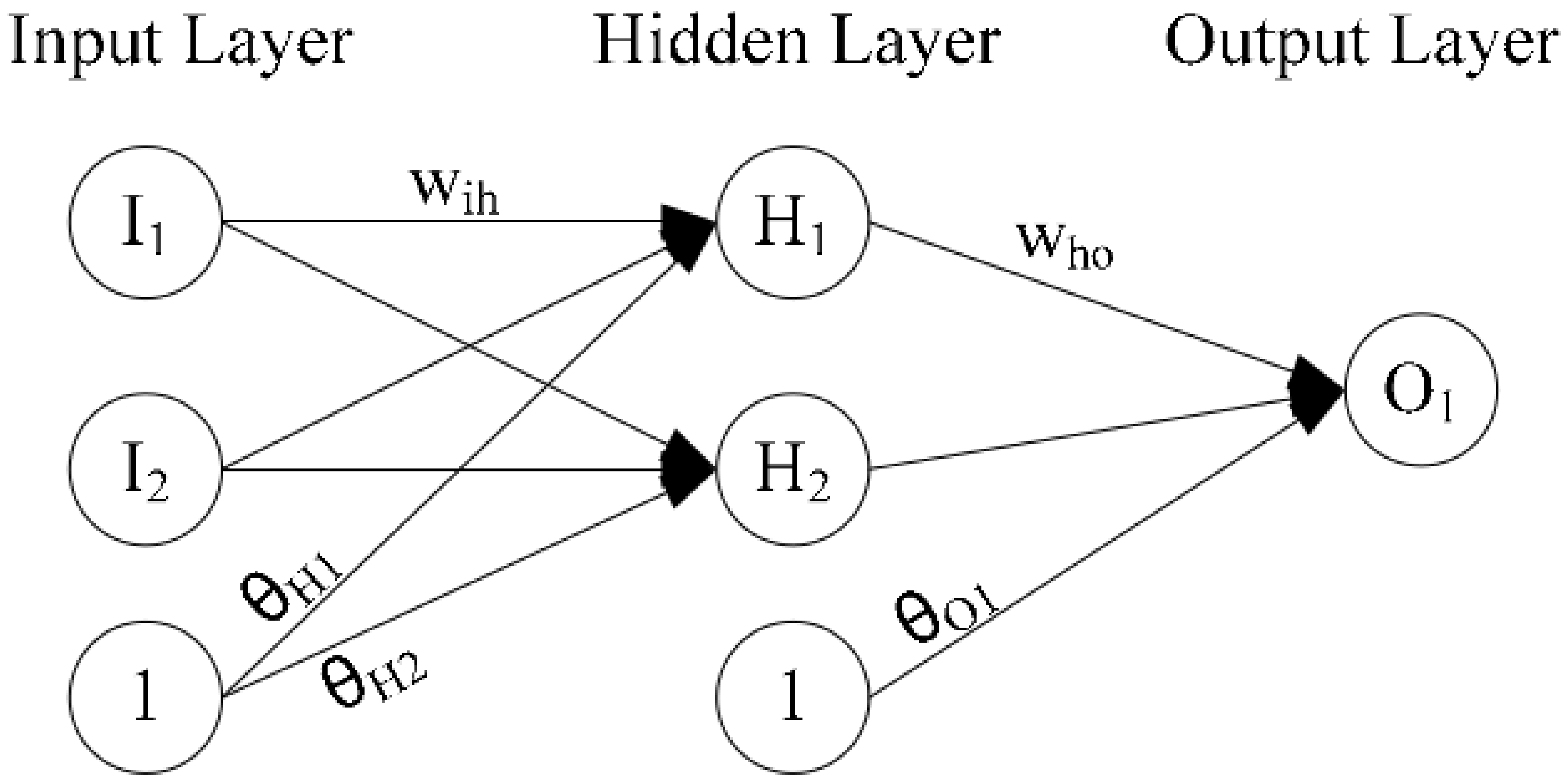
$$\begin{cases} H_h = \left(\sum_i w_{ih} \times I_i\right) + \theta_h & (1) \\ O_o = \left(\sum_h w_{ho} \times H_h\right) + \theta_o & (2) \end{cases}$$

*Error estimation*

$$\begin{cases} \delta_o = desired\ output - actual\ output & (3) \\ \delta_h = \sum_o \delta_o \times w_{ho} & (4) \end{cases}$$

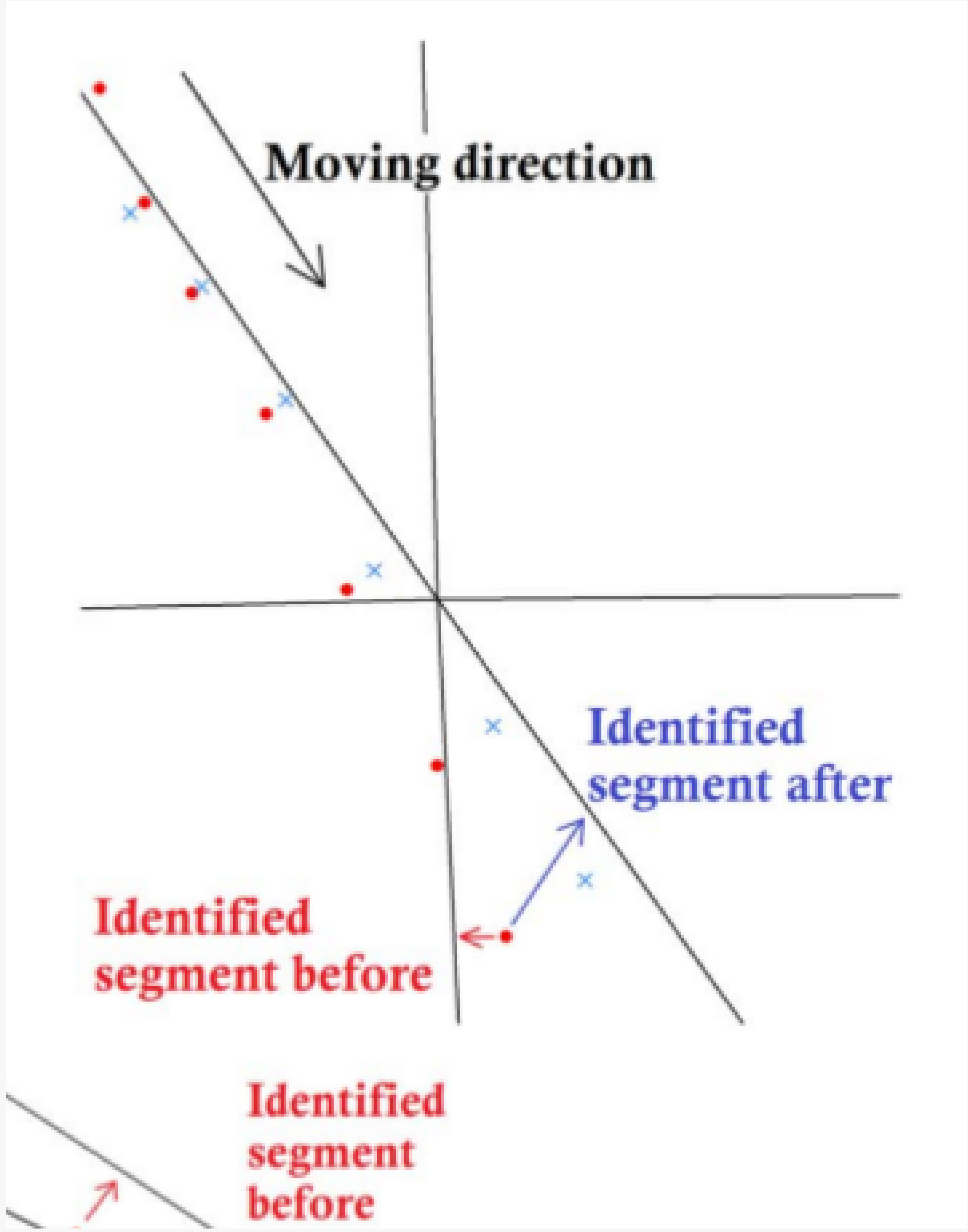
*Update weight*

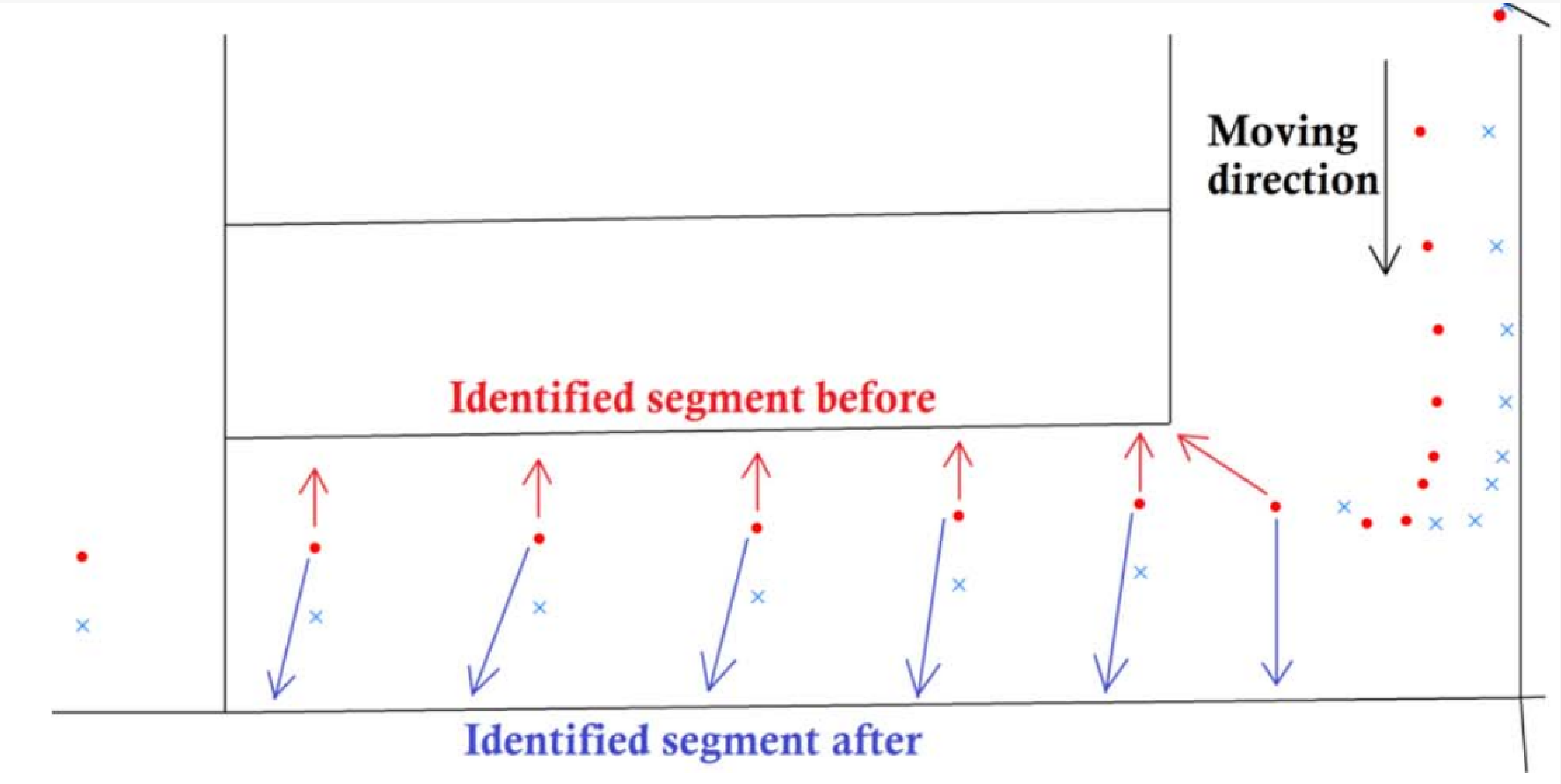
$$\begin{cases} \Delta w_{ho} = \gamma \times \delta_o \times H_h & (5) \\ \Delta w_{ih} = \gamma \times \delta_h \times I_i & (6) \end{cases}$$



# Results

*Horizontal trajectory*

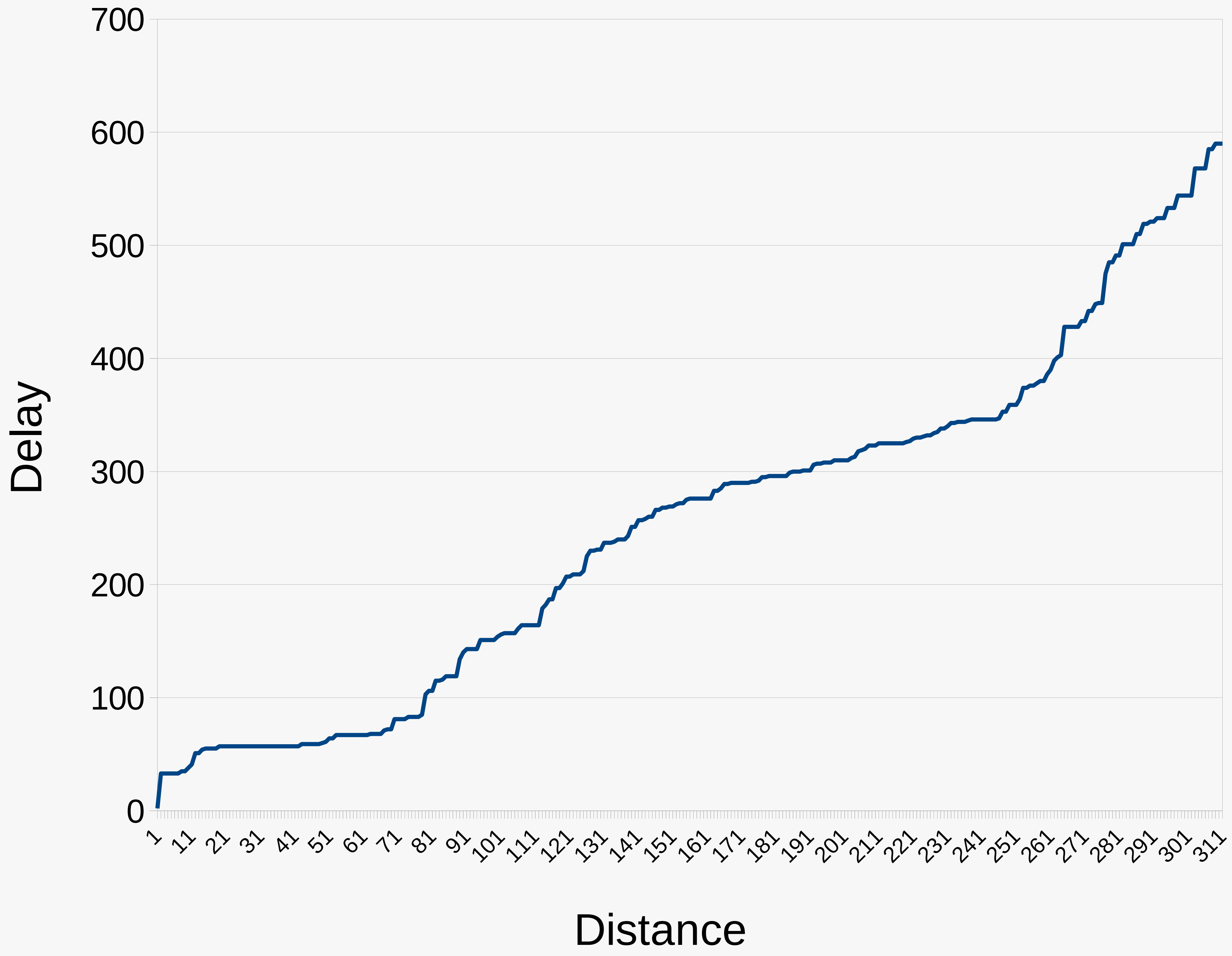




# QoS of the network

*Delay transmission over distance*

*WIFI*

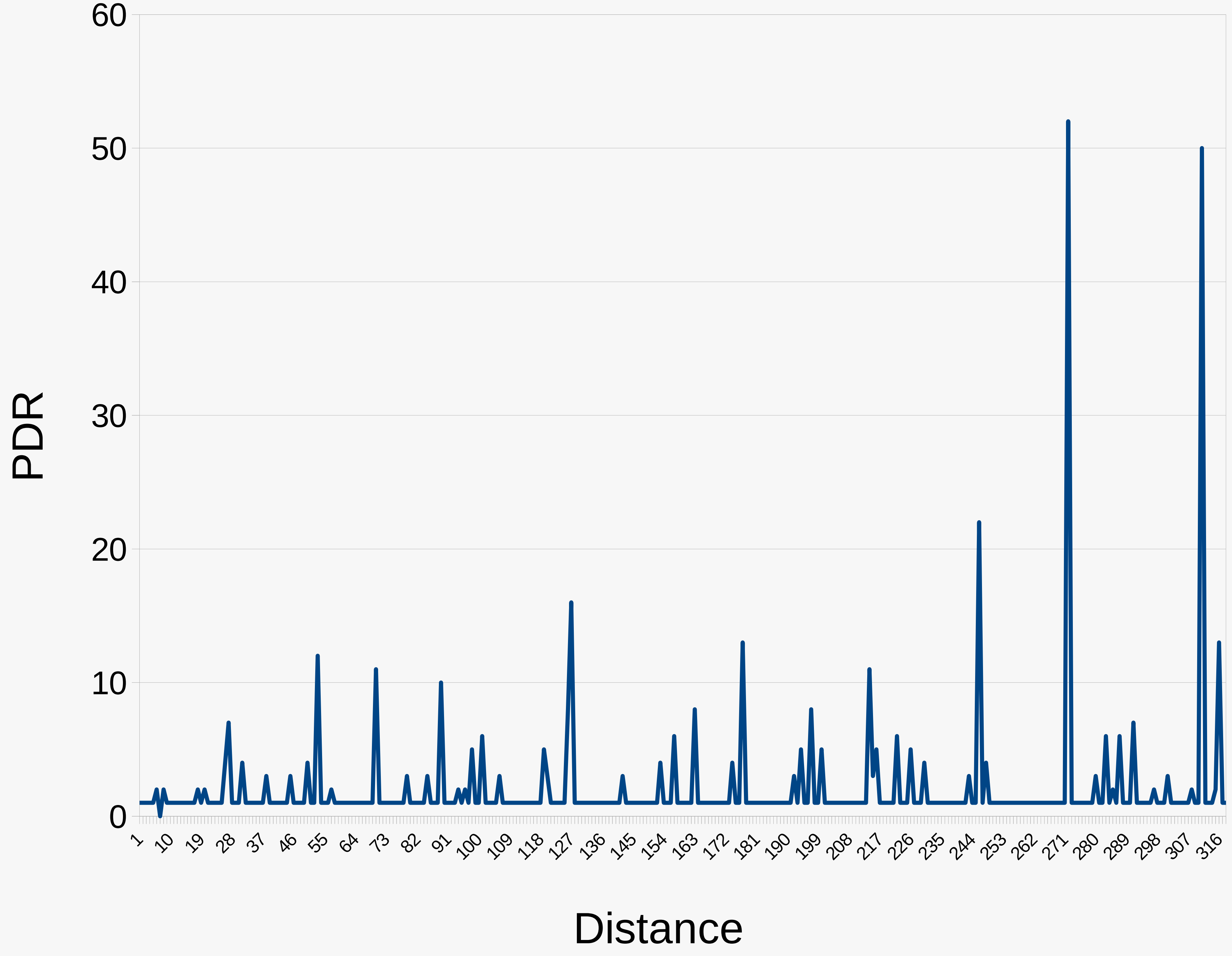


*3G*

# QoS of the network

*Packet delivery ratio over distance*

WIFI



3G

# Algorithm

*Collision detection system*

*Algorithm:*

*Dmin1 = V_veh * (T_perception+T_reaction+T_transmission+T_computation) + GPS_err-veh + GPS_err-ped*

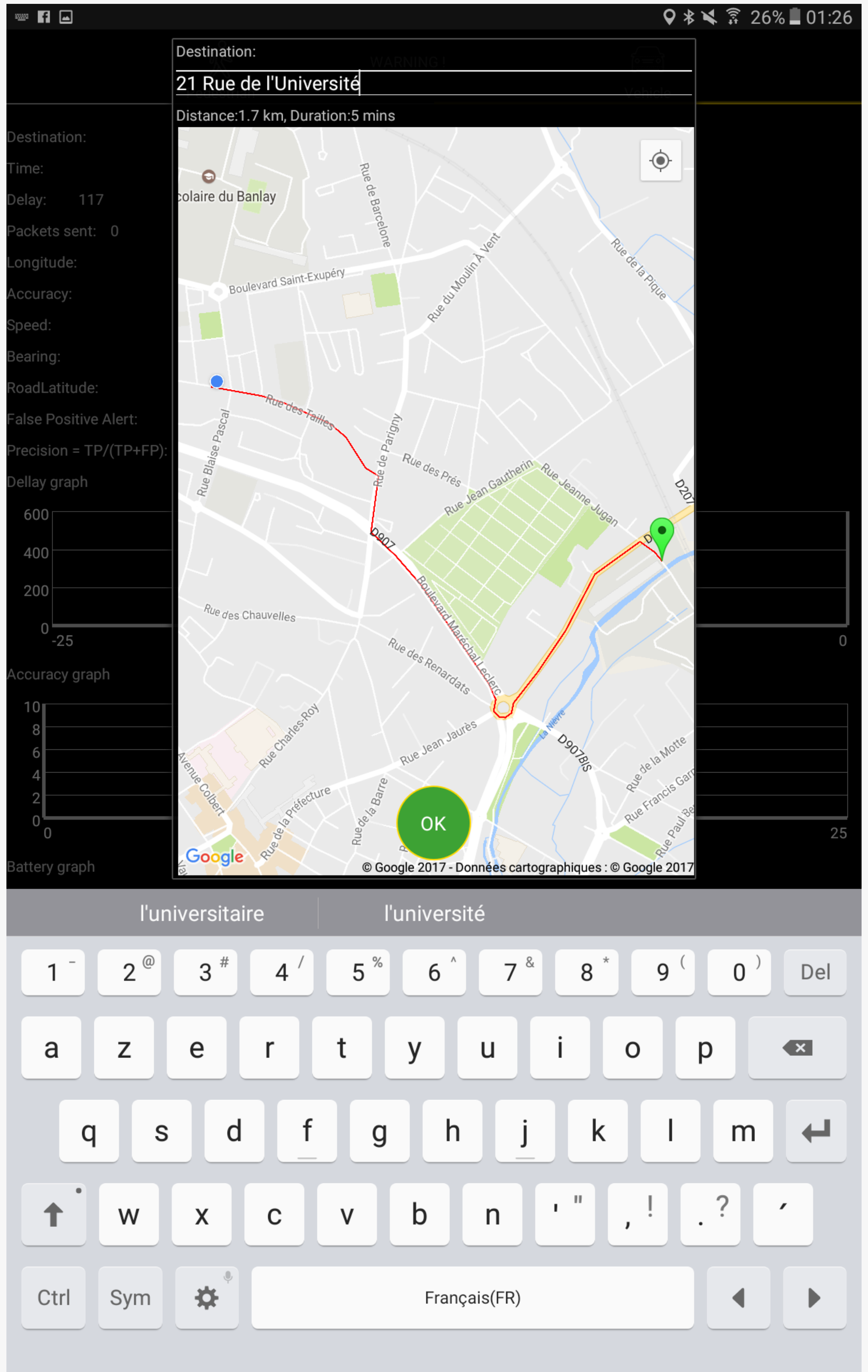
*Evaluation:*

*Dmin2 = V_veh * (830 ms+170 ms+150 ms+200 ms) + 1m + 1m*

```
If ( Dmin2 < Dact and Dmin1 < Dact ) then True_negative_alert++
If ( Dmin2 < Dact and Dmin1 > Dact ) then False_positive_alert++
If ( Dmin2 > Dact and Dmin1 < Dact ) then False_negative_alert++
If ( Dmin2 > Dact and Dmin1 > Dact ) then True_positive_alert++
```

# Application

*Real environment*



Destination:

21 Rue de l'Université

Distance:1.7 km, Duration:5 mins

Destination:

Time:

Delay: 117

Packets sent: 0

Longitude:

Accuracy:

Speed:

Bearing:

RoadLatitude:

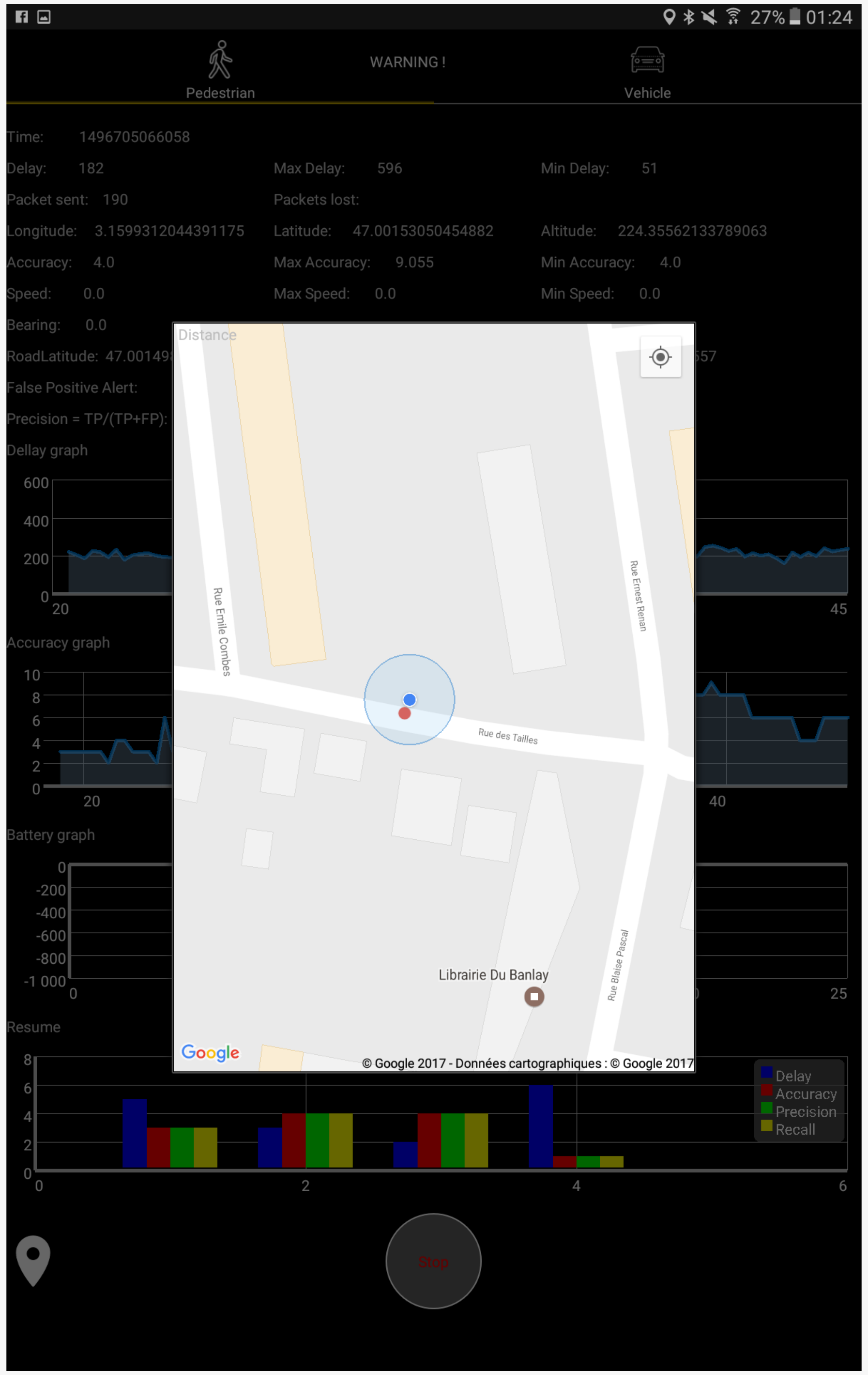
False Positive Alert:

Precision = TP/(TP+FP):

Dellay graph

Accuracy graph

Battery graph



Time: 1496705066058

Delay: 182 Max Delay: 596 Min Delay: 51

Packet sent: 190 Packets lost:

Longitude: 3.1599312044391175 Latitude: 47.00153050454882 Altitude: 224.35562133789063

Accuracy: 4.0 Max Accuracy: 9.055 Min Accuracy: 4.0

Speed: 0.0 Max Speed: 0.0 Min Speed: 0.0

Bearing: 0.0

RoadLatitude: 47.00149 557

False Positive Alert:

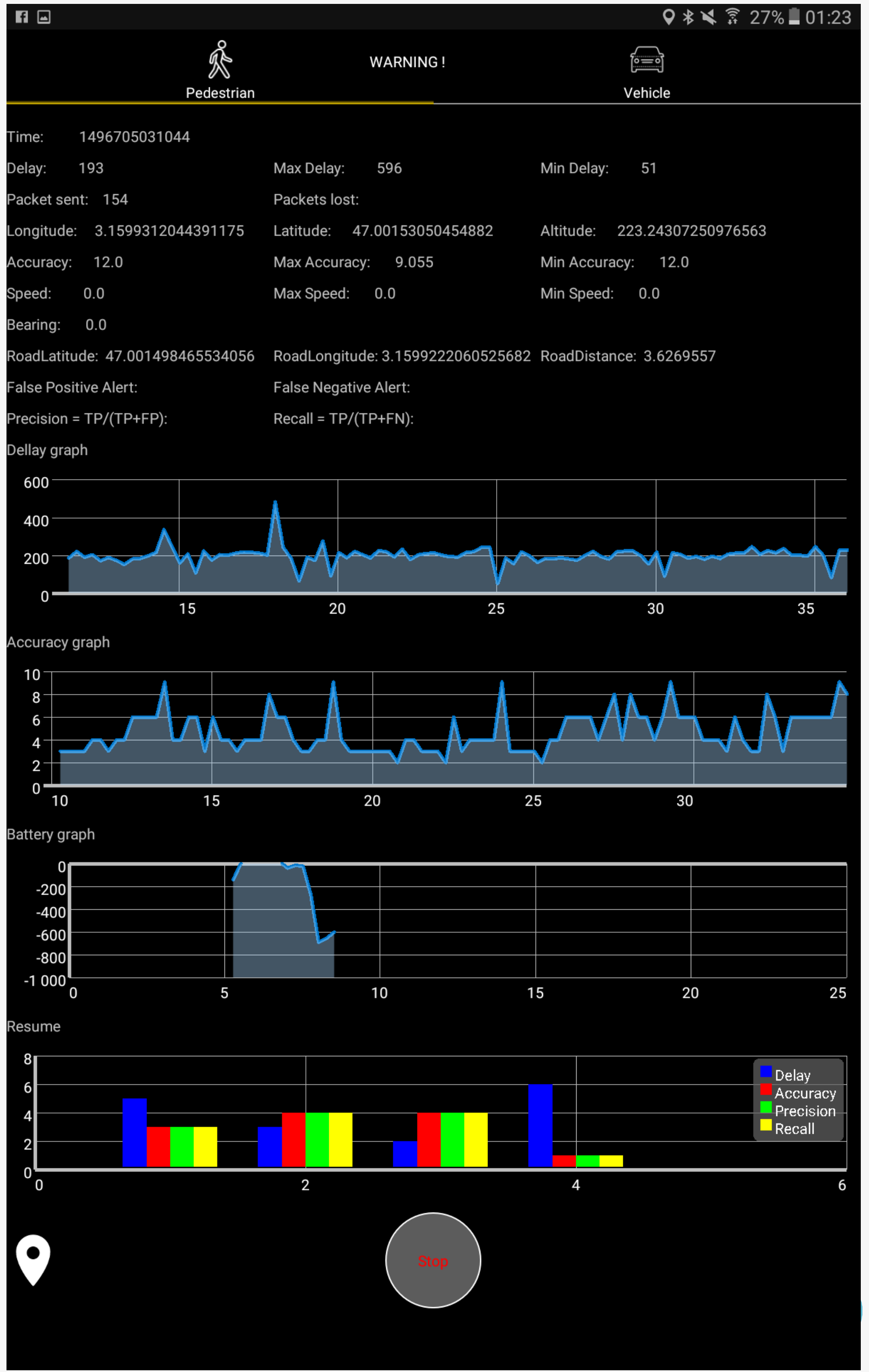
Precision = TP/(TP+FP):

Dellay graph

Accuracy graph

Battery graph

Resume



Time: 1496705031044

Delay: 193 Max Delay: 596 Min Delay: 51

Packet sent: 154 Packets lost:

Longitude: 3.1599312044391175 Latitude: 47.00153050454882 Altitude: 223.24307250976563

Accuracy: 12.0 Max Accuracy: 9.055 Min Accuracy: 12.0

Speed: 0.0 Max Speed: 0.0 Min Speed: 0.0

Bearing: 0.0

RoadLatitude: 47.001498465534056 RoadLongitude: 3.1599222060525682 RoadDistance: 3.6269557

False Positive Alert: False Negative Alert:

Precision = TP/(TP+FP): Recall = TP/(TP+FN):

Dellay graph

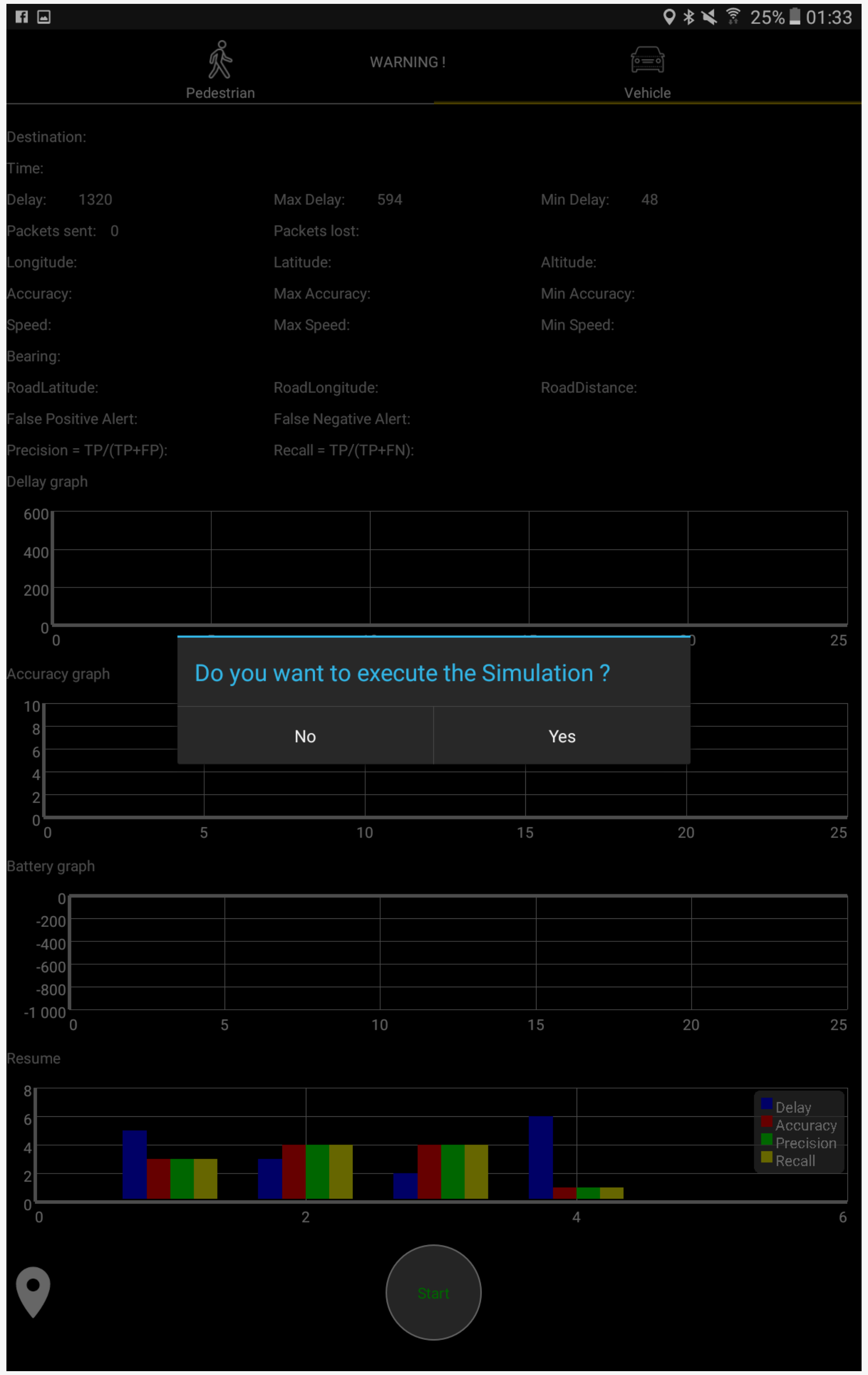
Accuracy graph

Battery graph

Resume

# Application

## *Simulation*



WARNING !

Destination:

Time:

Delay: 1320 Max Delay: 594 Min Delay: 48

Packets sent: 0 Packets lost:

Longitude: Latitude: Altitude:

Accuracy: Max Accuracy: Min Accuracy:

Speed: Max Speed: Min Speed:

Bearing:

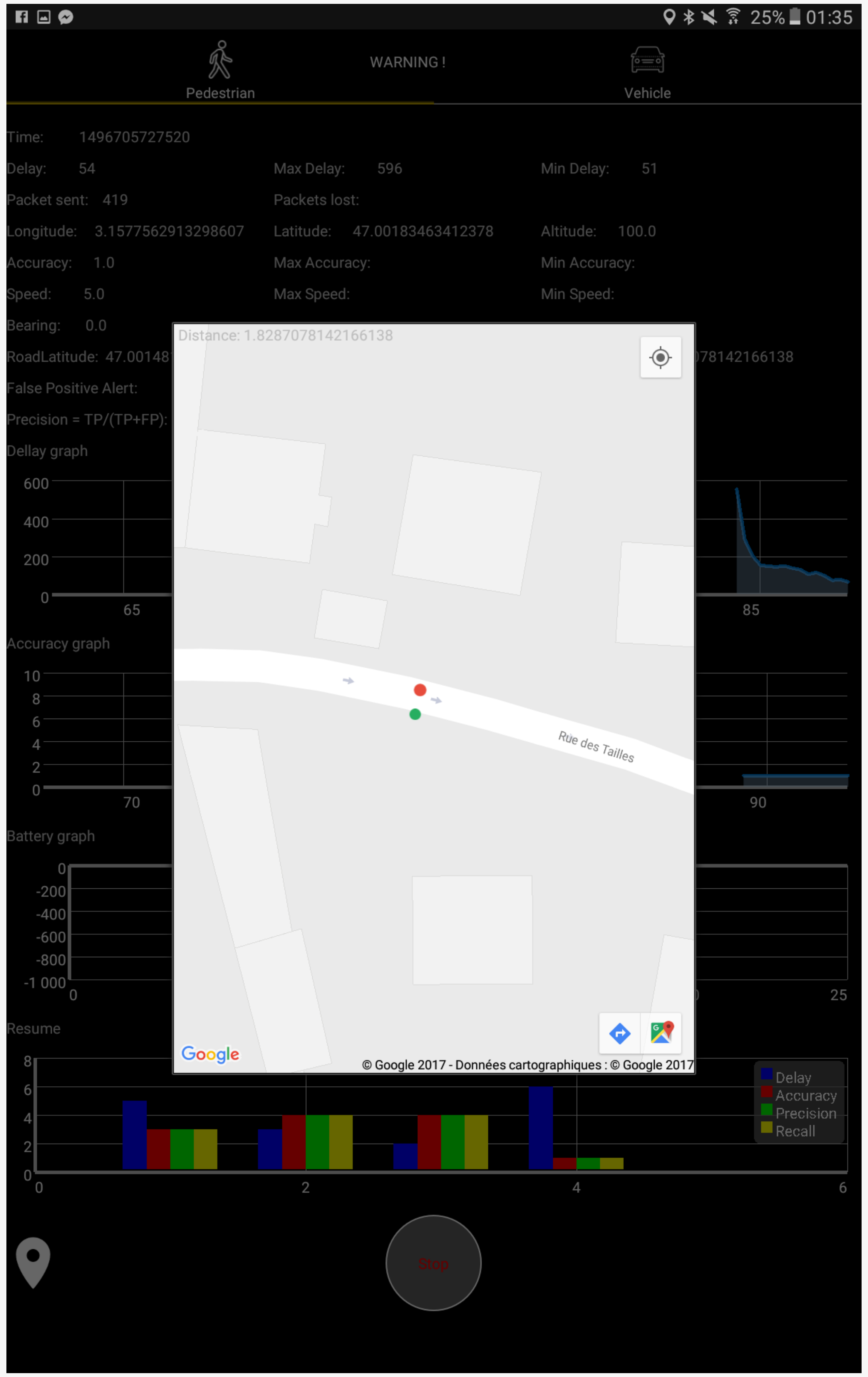
RoadLatitude: RoadLongitude: RoadDistance:

False Positive Alert: False Negative Alert:

Precision = TP/(TP+FP): Recall = TP/(TP+FN):

Do you want to execute the Simulation ?

No Yes



WARNING !

Time: 1496705727520

Delay: 54 Max Delay: 596 Min Delay: 51

Packet sent: 419 Packets lost:

Longitude: 3.1577562913298607 Latitude: 47.00183463412378 Altitude: 100.0

Accuracy: 1.0 Max Accuracy: Min Accuracy:

Speed: 5.0 Max Speed: Min Speed:

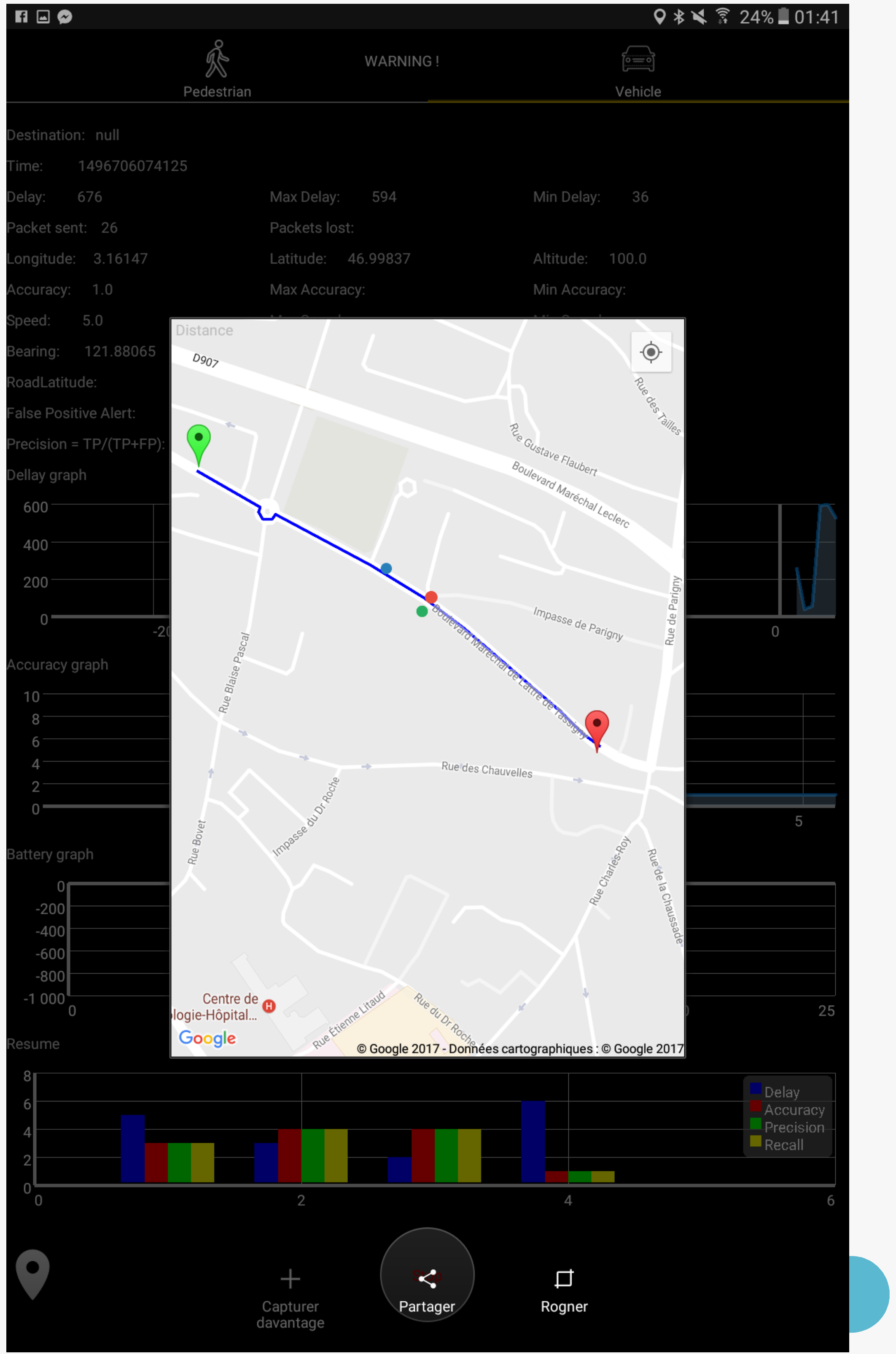
Bearing: 0.0

RoadLatitude: 47.00148

False Positive Alert:

Precision = TP/(TP+FP):

Distance: 1.8287078142166138



WARNING !

Destination: null

Time: 1496706074125

Delay: 676 Max Delay: 594 Min Delay: 36

Packet sent: 26 Packets lost:

Longitude: 3.16147 Latitude: 46.99837 Altitude: 100.0

Accuracy: 1.0 Max Accuracy: Min Accuracy:

Speed: 5.0

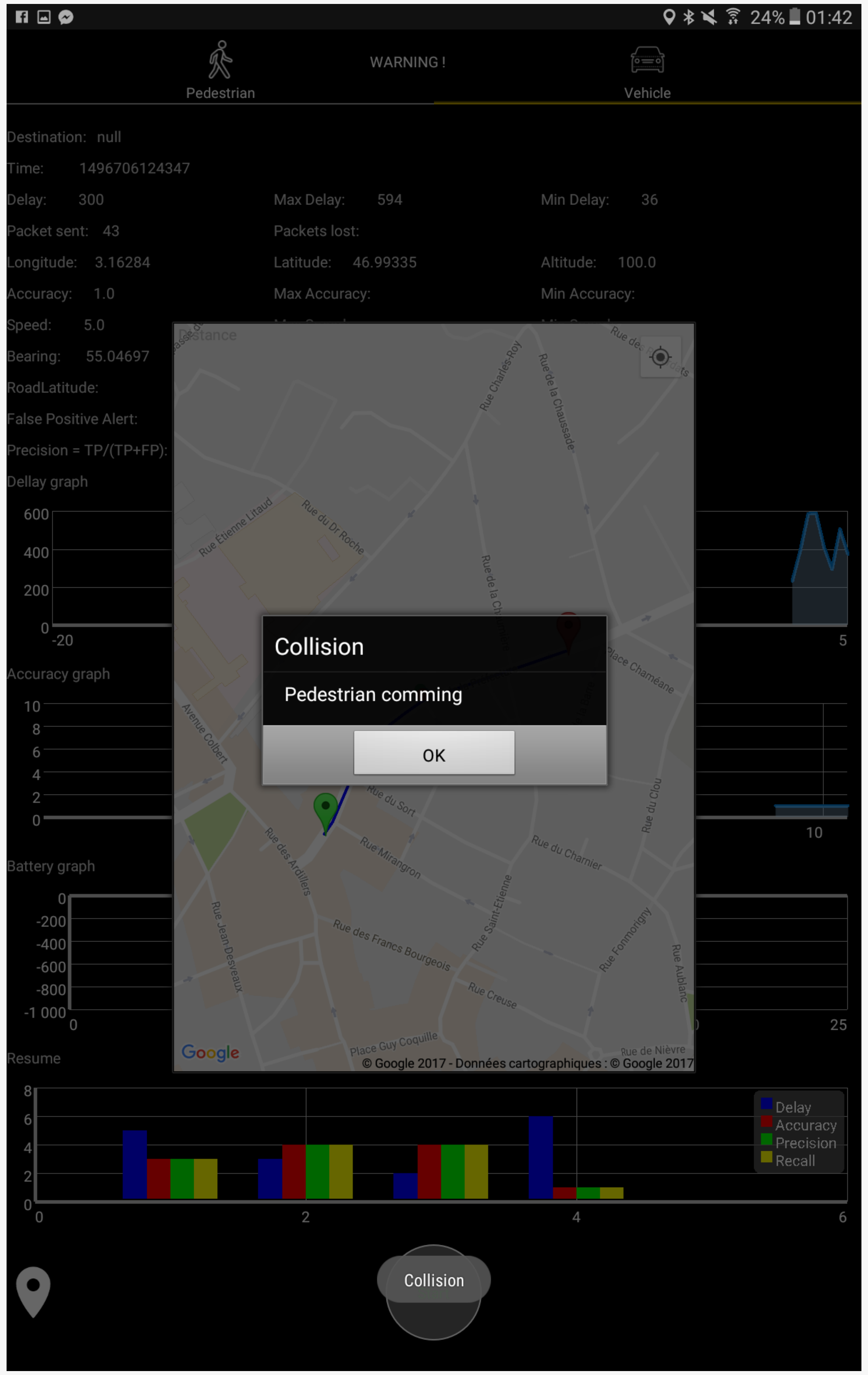
Bearing: 121.88065

RoadLatitude:

False Positive Alert:

Precision = TP/(TP+FP):

Distance



Destination: null

Time: 1496706124347

Delay: 300 Max Delay: 594 Min Delay: 36

Packet sent: 43 Packets lost:

Longitude: 3.16284 Latitude: 46.99335 Altitude: 100.0

Accuracy: 1.0 Max Accuracy: Min Accuracy:

Speed: 5.0

Bearing: 55.04697

RoadLatitude:

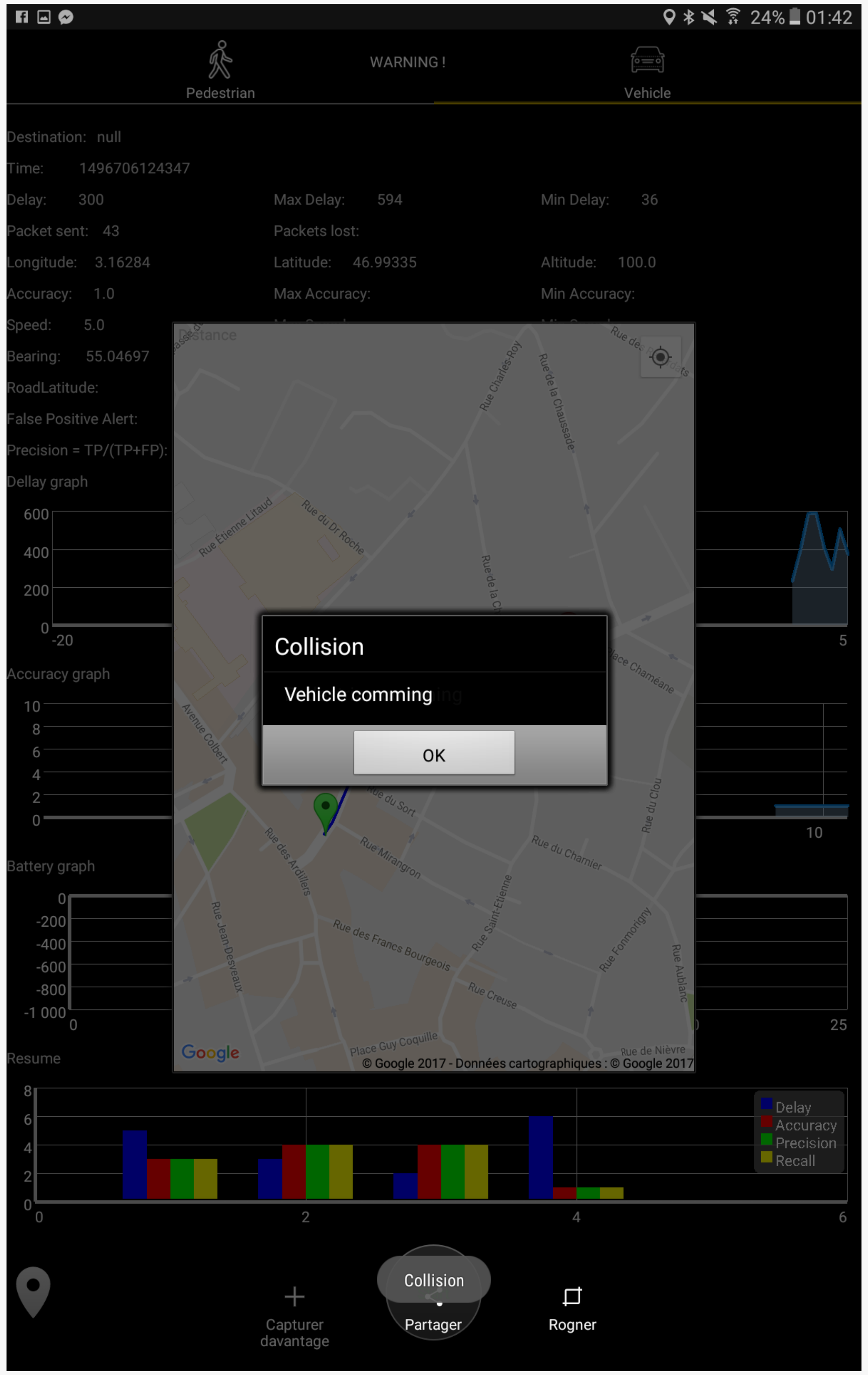
False Positive Alert:

Precision = TP/(TP+FP):

**Collision**

Pedestrian comming

OK



Destination: null

Time: 1496706124347

Delay: 300 Max Delay: 594 Min Delay: 36

Packet sent: 43 Packets lost:

Longitude: 3.16284 Latitude: 46.99335 Altitude: 100.0

Accuracy: 1.0 Max Accuracy: Min Accuracy:

Speed: 5.0

Bearing: 55.04697

RoadLatitude:

False Positive Alert:

Precision = TP/(TP+FP):

**Collision**

Vehicle comming

OK

# Thank You for Watching!

Any Questions?