RISK ZONE PREDICTION

Capstone Project 2020-2021

Multiprotexion
Hannes Wiedenhofer 14411

1 Introduction

Multiprotexion is a company that provides satellite, data, and active video surveillance security for the transportation and logistics business, focused on preventing events such as cargo and warehouse thefts and robberies. It is a partner of many international distribution, transportation, and logistics companies.
Multiprotexion offers a range of devices which aim at protecting drivers, goods, and vehicles. These devices create a lot of data. It is essential to analyse and evaluate the collected data. This data can be used to develop new devices and sensors, as well as providing a base for developing predictive models.
The goal of this project is to analyse the collected data and determine the potentially dangerous regions for a truck. This could be useful for deciding whether to equip a truck with additional security devices when it needs to travel through a risk zone, or maybe to decide whether to avoid driving through certain zones.

2 Dataset

Multiprotexion provided two databases containing information about events triggered by sensors or other security devices, and the associated location, driver, reasons the event was triggered, which action was taken to remedy the event, and many more.
Figure 1 shows the tables present in the database “dbcentrale” and their relations. Attributes have been omitted for space reasons. The complete ER-diagrams for both databases can be found in the Appendix.
The data was provided via two databases. We downloaded the relevant information and saved it in csv files so as to not have to access the database each time we need to access the data.

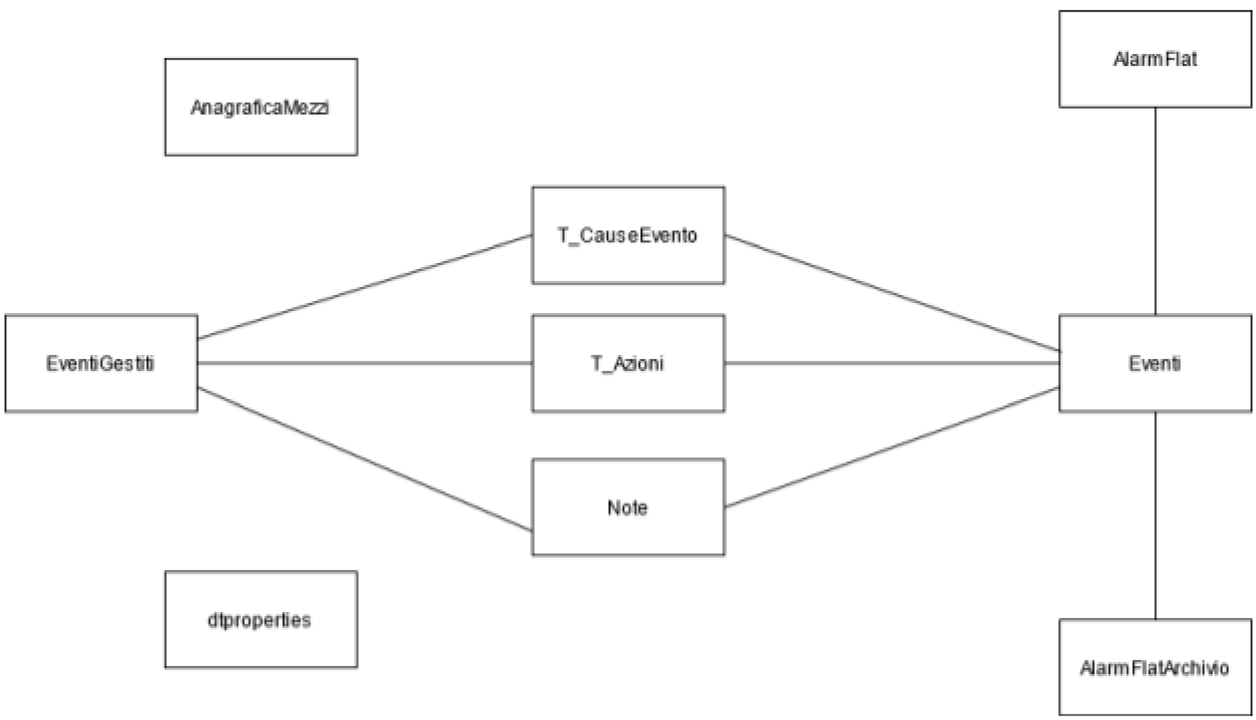
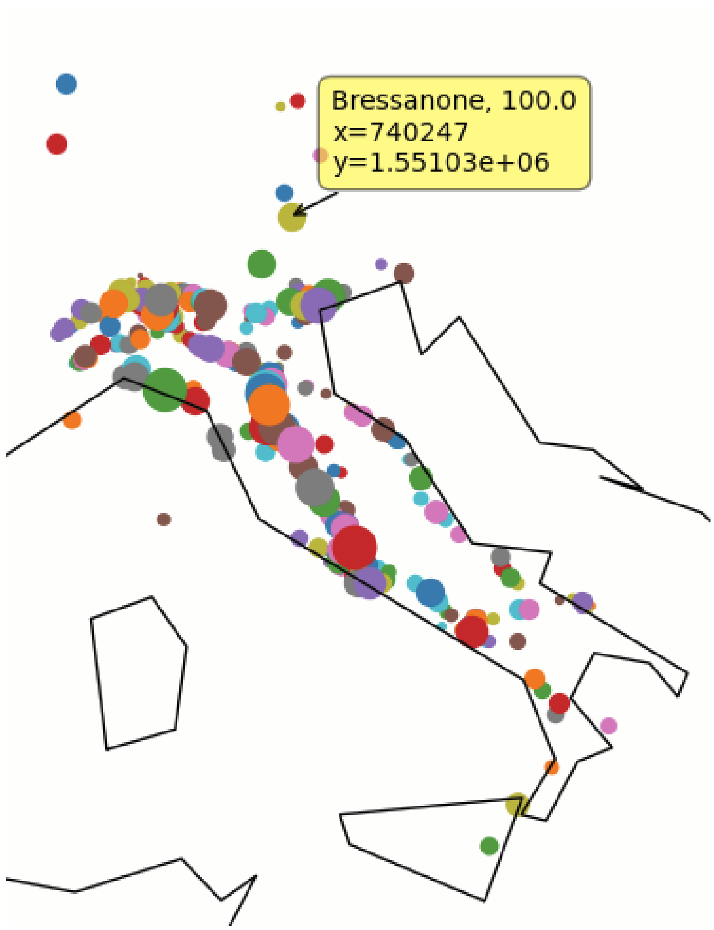


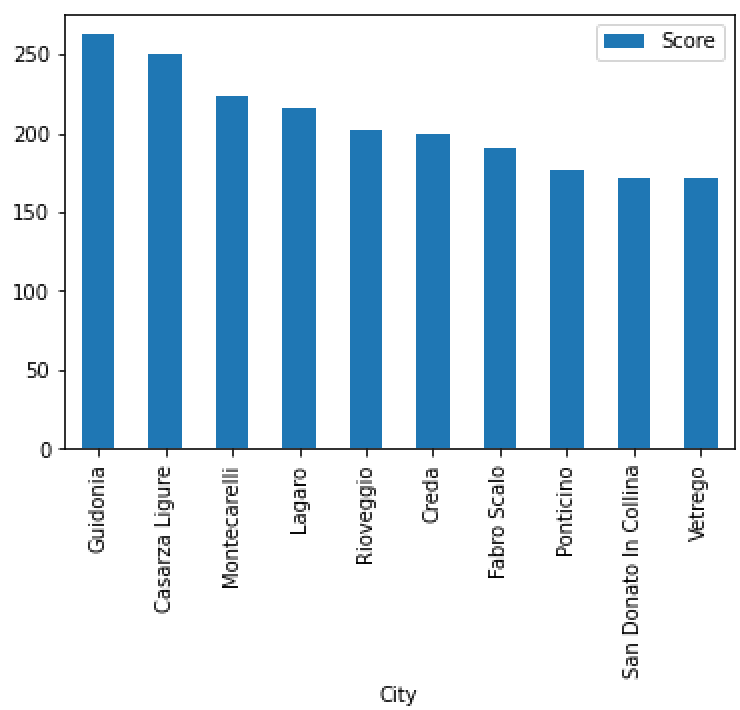
Figure 1: Representation of dbcentrale

3 Most Dangerous Cities

For analyzing the most dangerous cities we looked at the relation between the number of total alarms triggered by the sensors and the number of real alarms. Real alarms are all the alarms that are not triggered due to an anomaly of the sensors, test alarms, or other types of checks.
The data does not contain explicit information about the closest city for each triggered alarm, but it does contain the geographical coordinates for the location where the alarm was triggered.
We downloaded a dataset containing the geographical coordinates for European cities[1] and mapped each alarm to the closest city. Then we divided the number of real alarms by the number of total alarms and created a map where each dot represents a city. The size of the dot represents the risk score. For visualization purposes the score was multiplied by 3. Therefore the risk score ranges from 0 to 300. As we can see in the image below, Bressanone has a score of 100. This means that one third of the alarms near Bressanone were real alarms. The biggest dots and therefore most dangerous cities seem to be along the A1. However, this could also be explained by the fact that the biggest amount of traffic is located on the A1. In the second figure, we see the cities with the highest risk score. Guidonia, Casarza Ligure, and Montecarelli have the three highest risk scores.

[1] https://www.clearlyandsimply.com/clearly_and_simply/2010/10/geocoding-databases-for-europe.html

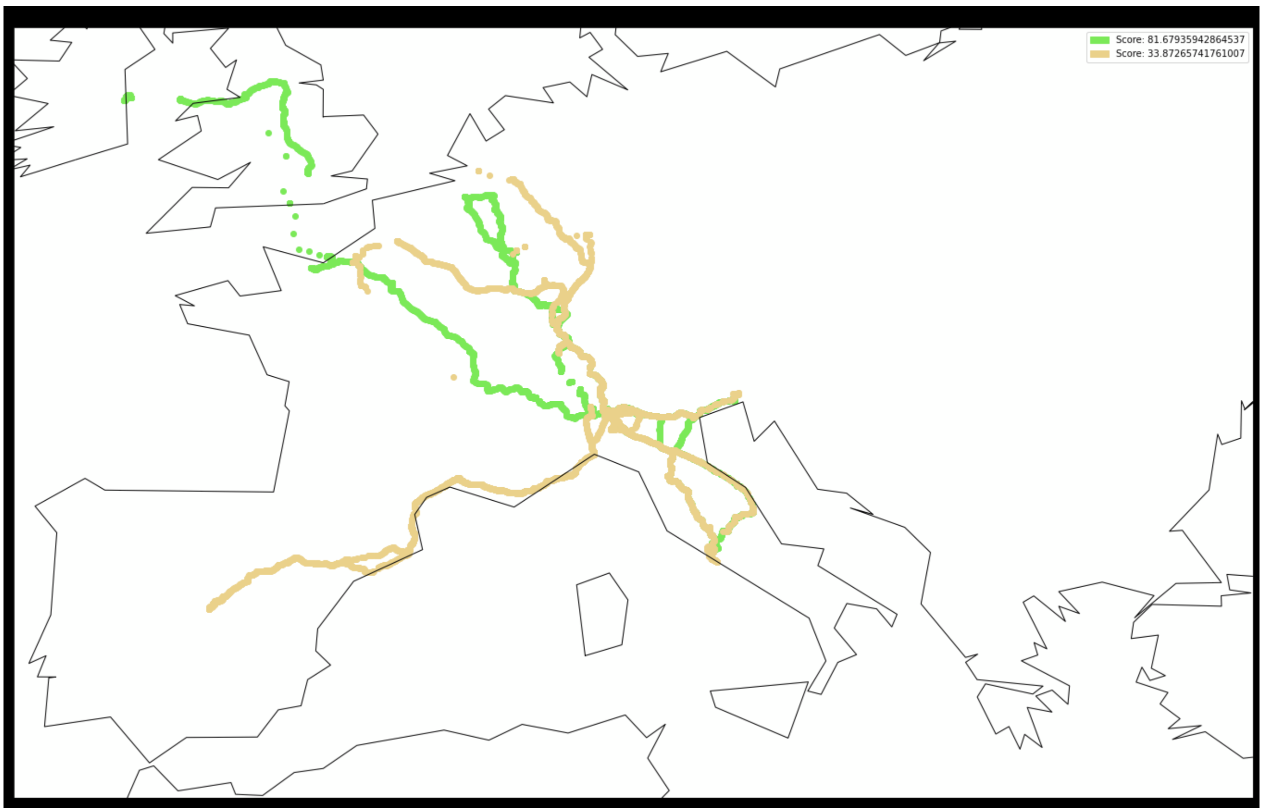

Bressanone, 100.0
x=740247
y=1.55103e+06

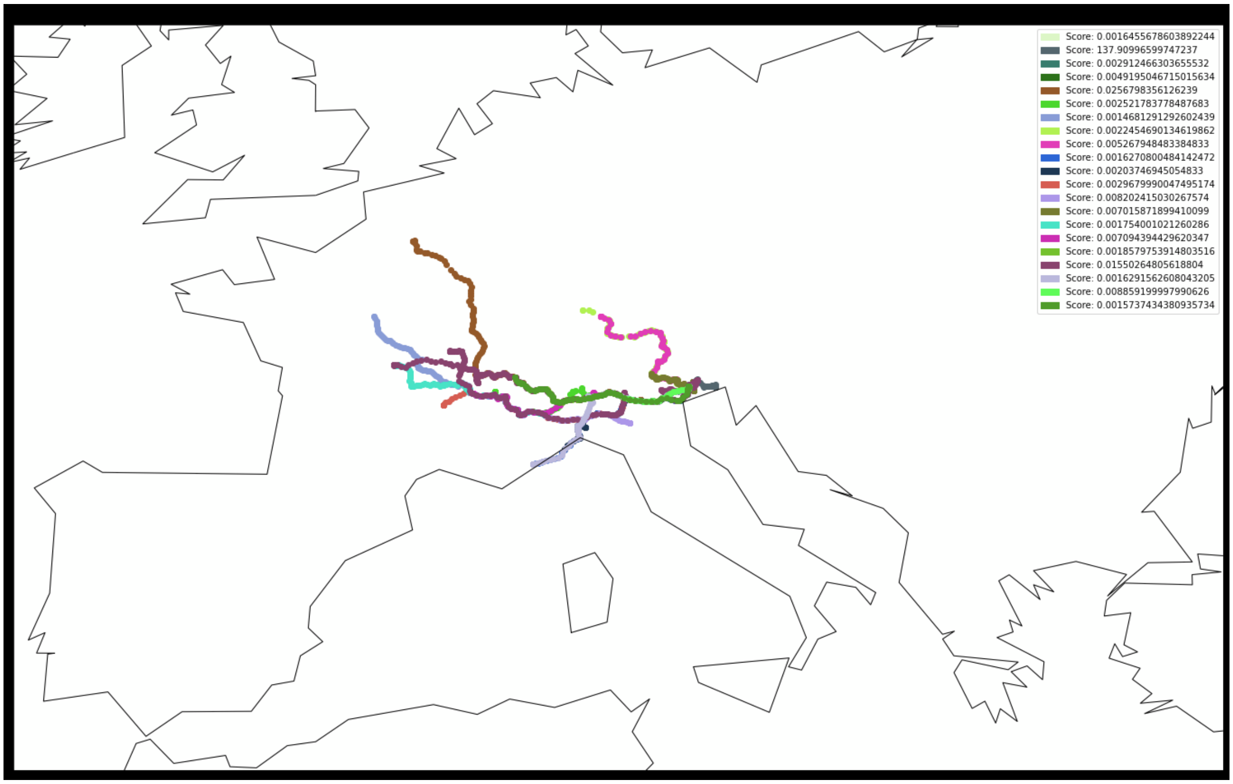

Score
250
200
150
100
50
0
Guidonia
Casarza Ligure
Montecarelli
Lagaro
Rioveggio
Creda
Fabro Scalo
Ponticino
San Donato In Collina
Vetrego
City

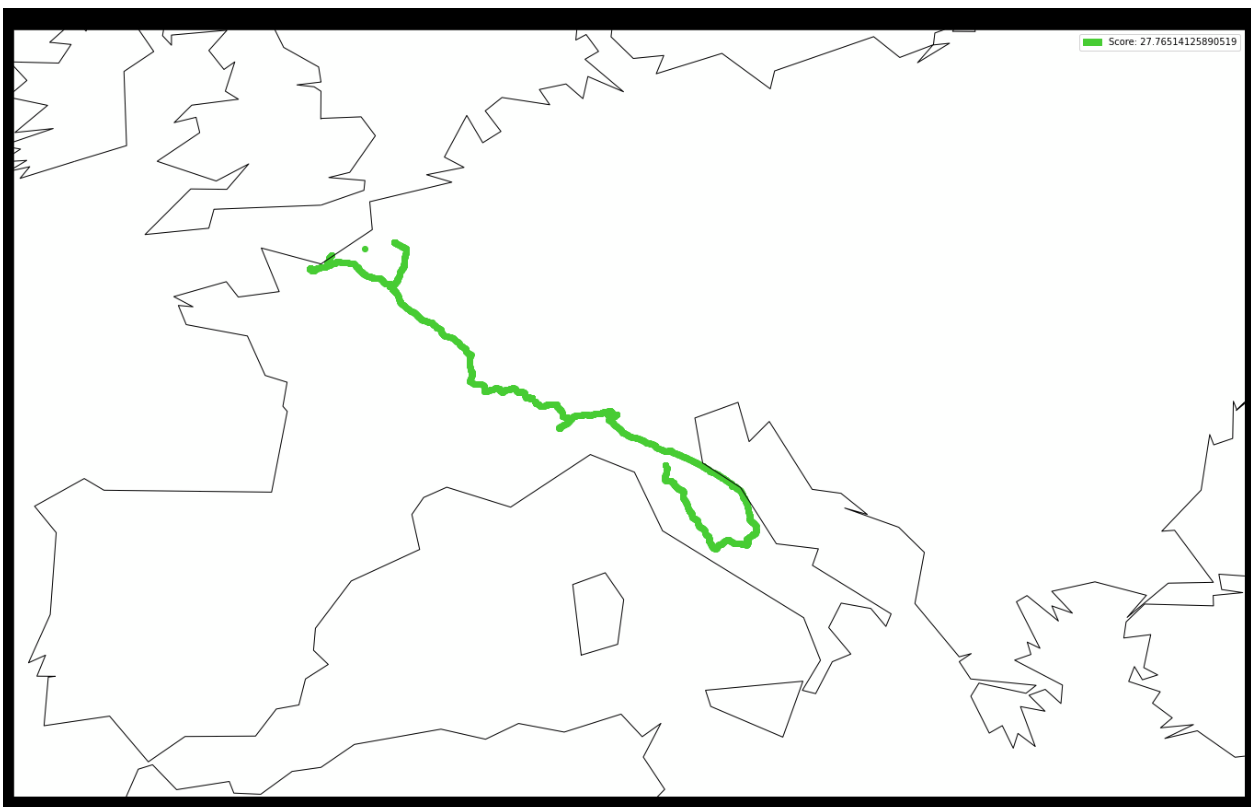
4 Route Risk Score

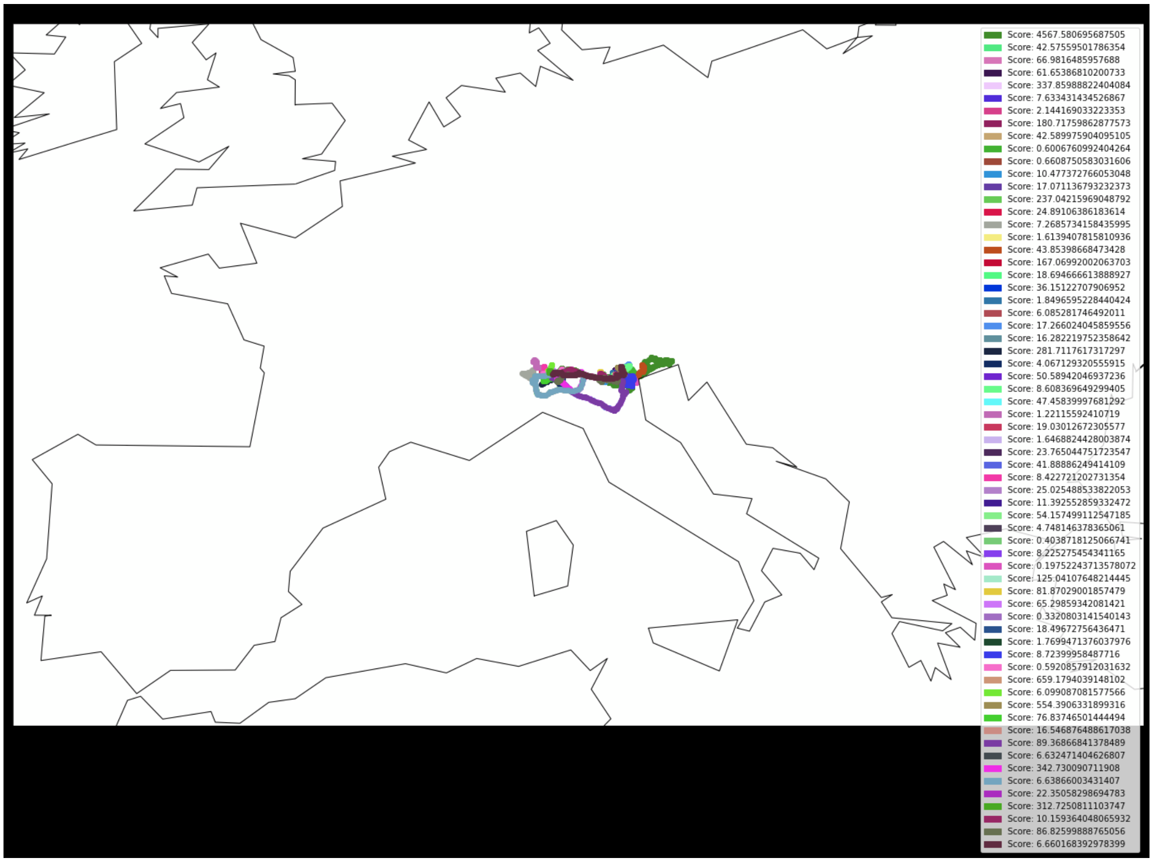
The previous approach only focused on the alarms and did not take into consideration the total number of routes passing through a certain point. Intuitively, the risk score of a route should be represented by the number of real alarms triggered along a route divided by the total length of the route. We wanted to use this concept for assigning risk scores to each route.
For achieving this goal we first had to look at how to extract the information about routes from the data. The dataset did not contain explicit information about routes. However, it did contain geographical points for every communication between the sensors on the truck and the central server. The average temporal difference between two successive communications is about 16 minutes. This means that we have information about each truck's route where each point is on average 16 minutes apart from the previous point. This gives us a detailed route for each truck. Most of the trucks did not just drive one route, but multiple routes. Given that there was no explicit information about when a route starts or ends, we had to develop our own means of determining when one route finishes and another route starts. We decided to declare a route finished and a new route started whenever the difference between two points was greater than 4 hours. Now we had information about all the routes for all the trucks.
The next step was to assign a risk score to each route. We already extracted the data for the alarms in the previous approach. Therefore, we only had to count the number of real alarms that happened on each route. To achieve this, we used the Python library shapely. Using its function *buffer* we created a buffer of 0.001 for each point of a route, so the area for a possible intersection with alarms was larger than just a single point. Then, using the function *intersects* we checked for each alarm point if it intersected the previously created buffer. This way, we counted the number of alarms that occurred on each route and used this as a score. However, now the longer the route the more likely it is to have a large amount of alarms along it. Therefore, we decided to divide the length of the route by the previously calculated score. This was done using the first and last point of each route and calculating the Python library geopy's function *distance.distance* and use

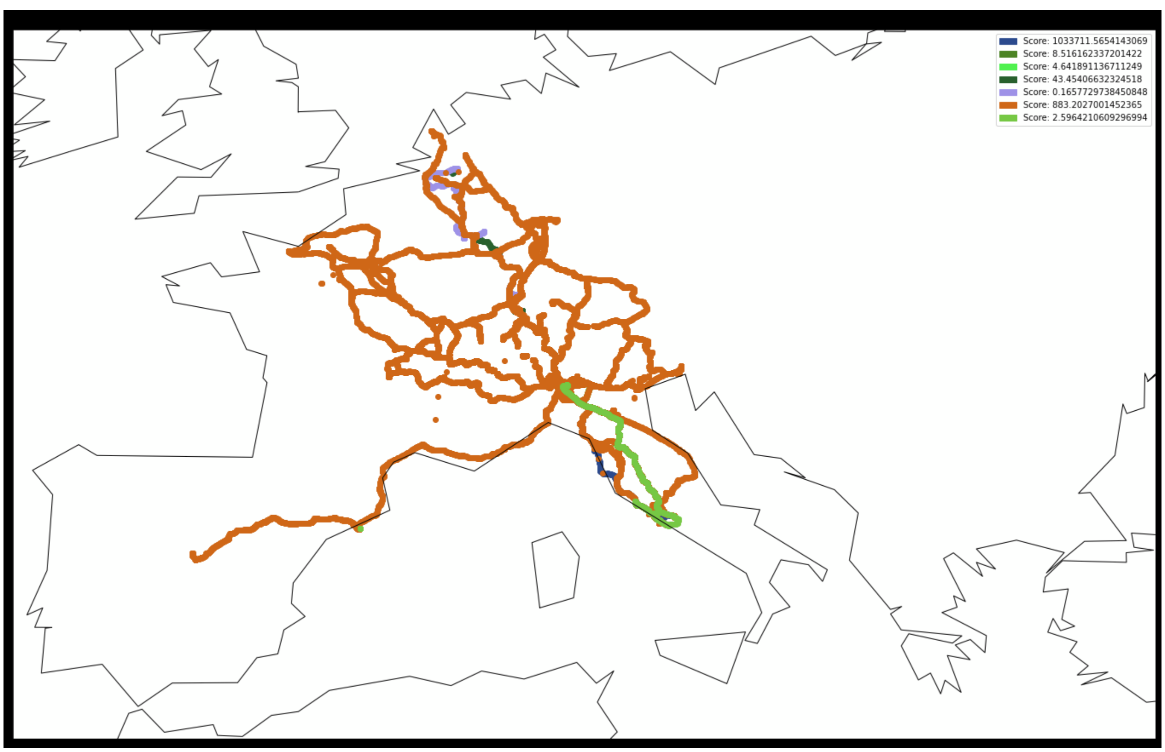
the result as risk score. Intuitively, this should yield values between 0 and 1. However, using the previously mentioned function we did not calculate the actual length of the route, but just the distance between the start and end point as the crow flies (bee-line). Therefore, some scores consist of larger numbers. In the following some examples. Some particularly dangerous routes according to the calculated risk score are Monfalcone-Treviso, Grosseto-Livorno, and Venice-Milan:




Score: 0.0016455678603892244
Score: 137.90996599747237
Score: 0.002912466303655532
Score: 0.0049195046715015634
Score: 0.0256798356126239
Score: 0.002521783778487683
Score: 0.0014681291292602439
Score: 0.0022454690134619862
Score: 0.005267948483384833
Score: 0.0016270800484142472
Score: 0.00203746945054833
Score: 0.0029679990047495174
Score: 0.008202415030267574
Score: 0.007015871899410099
Score: 0.001754001021260286
Score: 0.007094394429620347
Score: 0.0018579753914803516
Score: 0.01550264805618804
Score: 0.0016291562608043205
Score: 0.008859199997990626
Score: 0.0015737434380935734


Score: 27.76514125890519


Score: 4567.580695687505
Score: 42.57559501786354
Score: 66.9816485957688
Score: 61.65386810200733
Score: 337.85988822404084
Score: 7.633431434526867
Score: 2.144169033223353
Score: 180.71759862877573
Score: 42.589975904095105
Score: 0.6006760992404264
Score: 0.6608750583031606
Score: 10.477372766053048
Score: 17.071136793232373
Score: 237.04215969048792
Score: 24.89106386183614
Score: 7.2685734158435995
Score: 1.6139407815810936
Score: 43.85398668473428
Score: 167.06992002063703
Score: 18.694666613888927
Score: 36.15122707906952
Score: 1.8496595228440424
Score: 6.085281746492011
Score: 17.266024045859556
Score: 16.282219752358642
Score: 281.7117617317297
Score: 4.067129320555915
Score: 50.58942046937236
Score: 8.608369649299405
Score: 47.45839997681292
Score: 1.22115592410719
Score: 19.03012672305577
Score: 1.6468824428003874
Score: 23.765044751723547
Score: 41.88886249414109
Score: 8.422721202731354
Score: 25.025488533822053
Score: 11.392552859332472
Score: 54.157499112547185
Score: 4.748146378365061
Score: 0.4038718125066741
Score: 8.225275454341165
Score: 0.19752243713578072
Score: 125.04107648214445
Score: 81.87029001857479
Score: 65.29859342081421
Score: 0.3320803141540143
Score: 18.49672756436471
Score: 1.7699471376037976
Score: 8.72399958487716
Score: 0.5920857912031632
Score: 659.1794039148102
Score: 6.099087081577566
Score: 554.3906331899316
Score: 76.83746501444494
Score: 16.546876488617038
Score: 89.36866841378489
Score: 6.632471404626807
Score: 342.730090711908
Score: 6.63866003431407
Score: 22.35058298694783
Score: 312.7250811103747
Score: 10.159364048065932
Score: 86.82599888765056
Score: 6.660168392978399


Score: 1033711.5654143069
Score: 8.516162337201422
Score: 4.641891136711249
Score: 43.45406632324518
Score: 0.1657729738450848
Score: 883.2027001452365
Score: 2.5964210609296994

5 Conclusions and Possible Next Steps

We presented two different approaches for risk zone prediction. We assigned a risk score to cities depending on the relation between all alarms and real alarms. The second approach was to assign a risk score to each route based on the number of real alarms that occurred along the route. For this approach we used the distance between the start and end points as the crow flies. Therefore, we obtained some very high scores. A possible next step could be to calculate the length of a route in a different way, so the actual length of the route can be used when calculating the risk score. This will lead to a more reliable risk score.

6 Appendix