PROTEZIONE CIVILE
Presidenza del Consiglio dei Ministri
Dipartimento della Protezione Civile

CONSORTIUM FOR SMALL SCALE MODELING
COSMO

cima
RESEARCH
FOUNDATION

COSMO PT-EVOCS: Testing and application of the open-source software package TITAN for a quality control of ground data

Elena Oberto & Umberto Pellegrini

A brief summary

- Quality control on (Italian or other) raingauge network is a strategic topic for many purposes as verification, data assimilation..
- Main goal is to prepare a “clean” precipitation field for model verification, for instance
- Starting from official Italian network, not from personal weather stations (PWS)
- Problems arise from the presence of different regional raingauges networks, different metadata, and lack of information on metadata
- Automatic data quality control procedures are needed to support human-based quality control
- Automatic data quality control is needed due to exponential increase of number of available observations

TITANLIB: automatic spatial data quality control

- TITANLIB is an open source software developed at the Norwegian meteorological institute (MET Norway, aka METNO). It is a library of automatic quality control routines for weather observations
- Spatial checks for use with dense observation networks
- Written in C++, bindings for python and R
- Multiple set of functions for performing tests on data
- Possibility to use multiple source of data (in case of raingauges, radar, HSAF, any gridded/irregular data)
- Possibility to set the level of trust for each source
- Sequential tests are possible for quality control assessment (serial/parallel)

https://github.com/metno/titanlib

TITANLIB tests

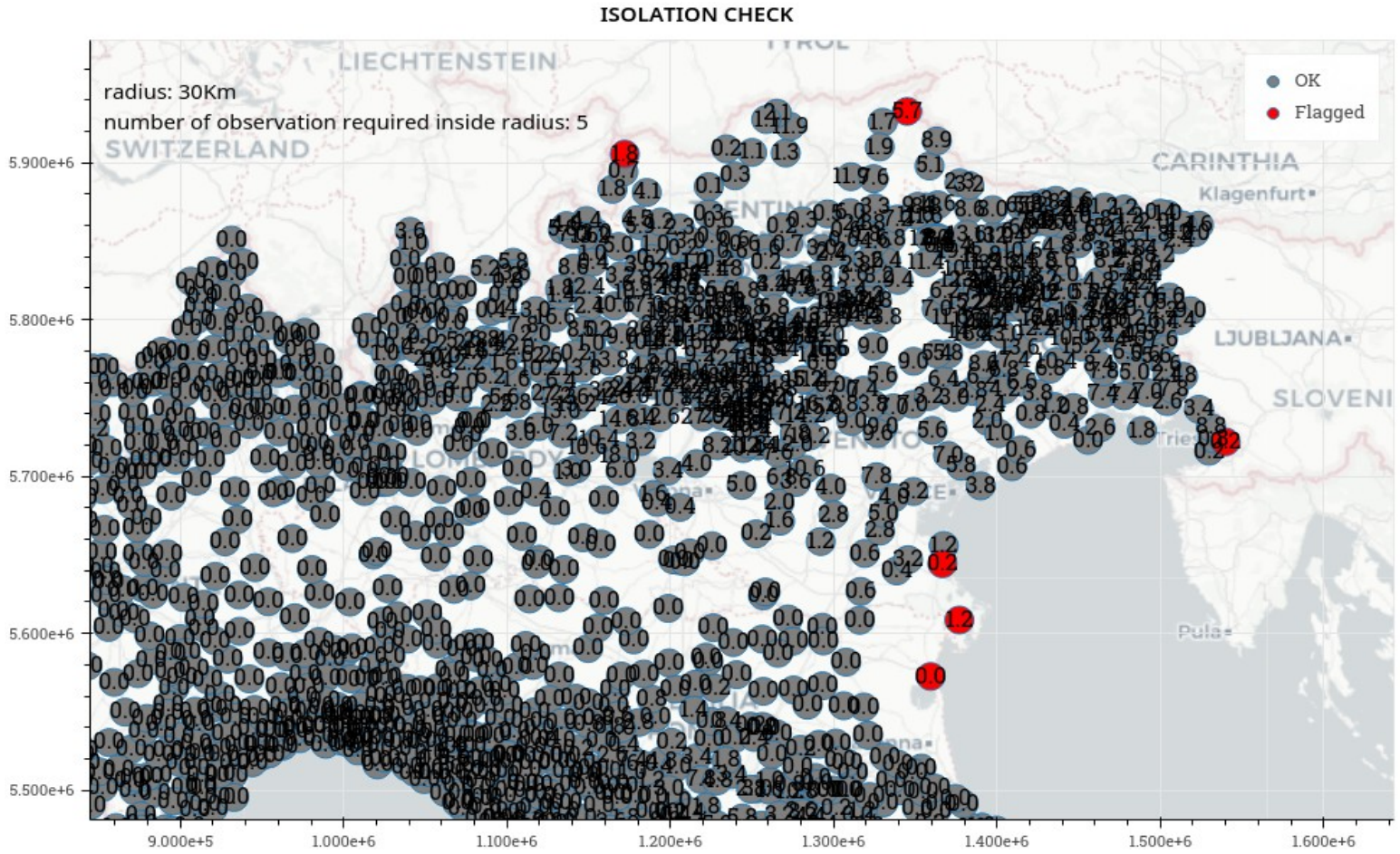
- **Buddy event check**
- **Buddy check**
- Climatological range check
- First-guess check
- **Isolation check**
- Metadata check
- **Range check**
- SCT spatial consistency check
- SCTR spatial consistency resistant check
- SCTD spatial consistency check dual

Quality control process scheme: our first try on raingauge data

- Titanlib range check (<0mm; >600mm) on hourly data
- Completeness of daily values (rejected daily values if number of hours < 24)
- Titanlib range check on daily data (>1000mm)
- Titanlib isolation check (pre-operational fine tuning)
- Titanlib sequential tests on daily data: **buddy check** and **buddy event check** with different configurations (5 runs)
- Titanlib sequential test on seasonal data: **buddy check** and **buddy event check** (1 run)
  - ➔ Possible correction on daily data based on seasonal qc check, typically station removal

Isolation check

The isolation check flags stations that have fewer than **num_min** buddies within a specified **radius** (m).



Buddy check

The buddy check compares an observation against its neighbours (i.e. buddies) and flags outliers in a radius specified by the user.

The buddy check flags observations if the (absolute value of the) difference between the observations and the average of the neighbours normalized by the standard deviation in the circle is greater than a predefined **threshold**.

If the standard deviation of values in the neighbourhood is less than a chosen value, **min_std**, then a value of **min_std** is used instead.

min_std should be roughly equal to the standard deviation of the error of a typical observation. If it is too low, then too many observations will be flagged in areas where the variability is low.

Buddy check equation

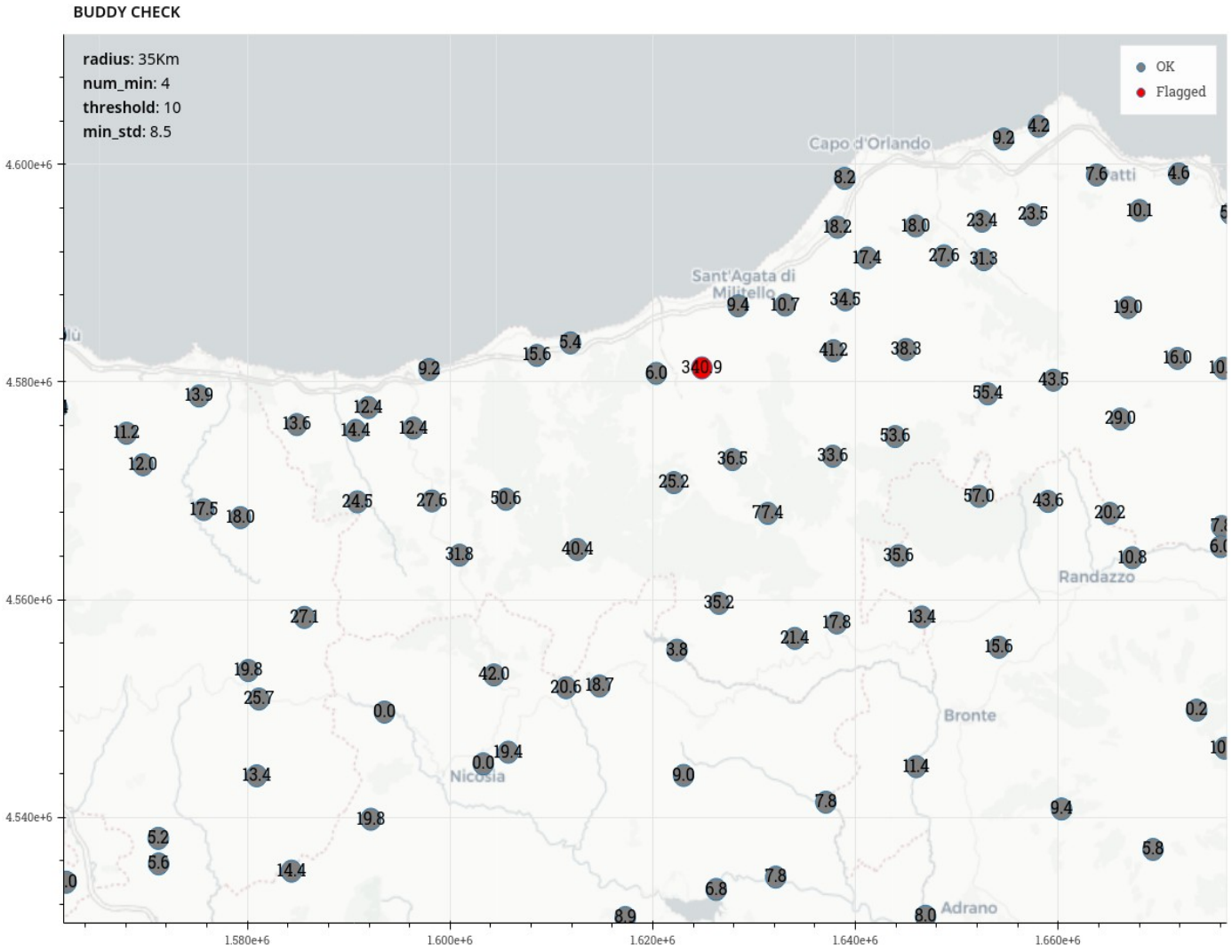
pog=|(buddy[i]- media_buddies)|/std_adjusted

std_adjusted = sqrt(variance + variance / n_buddies)
buddy[i] = i-buddy value inside the circle of fixed radius
n_buddies= data number inside the circle
variance = buddies variance inside the circle

IF pog> threshold THEN data rejected
IF std_adjusted < min_std, I use min_std

The rainfall has not statistics, so that is the problem: many tests have to be perform to catch the evident outliers, at the moment big threshold and big std_adjusted values have been used

BUDDY CHECK on daily data



Buddy event check

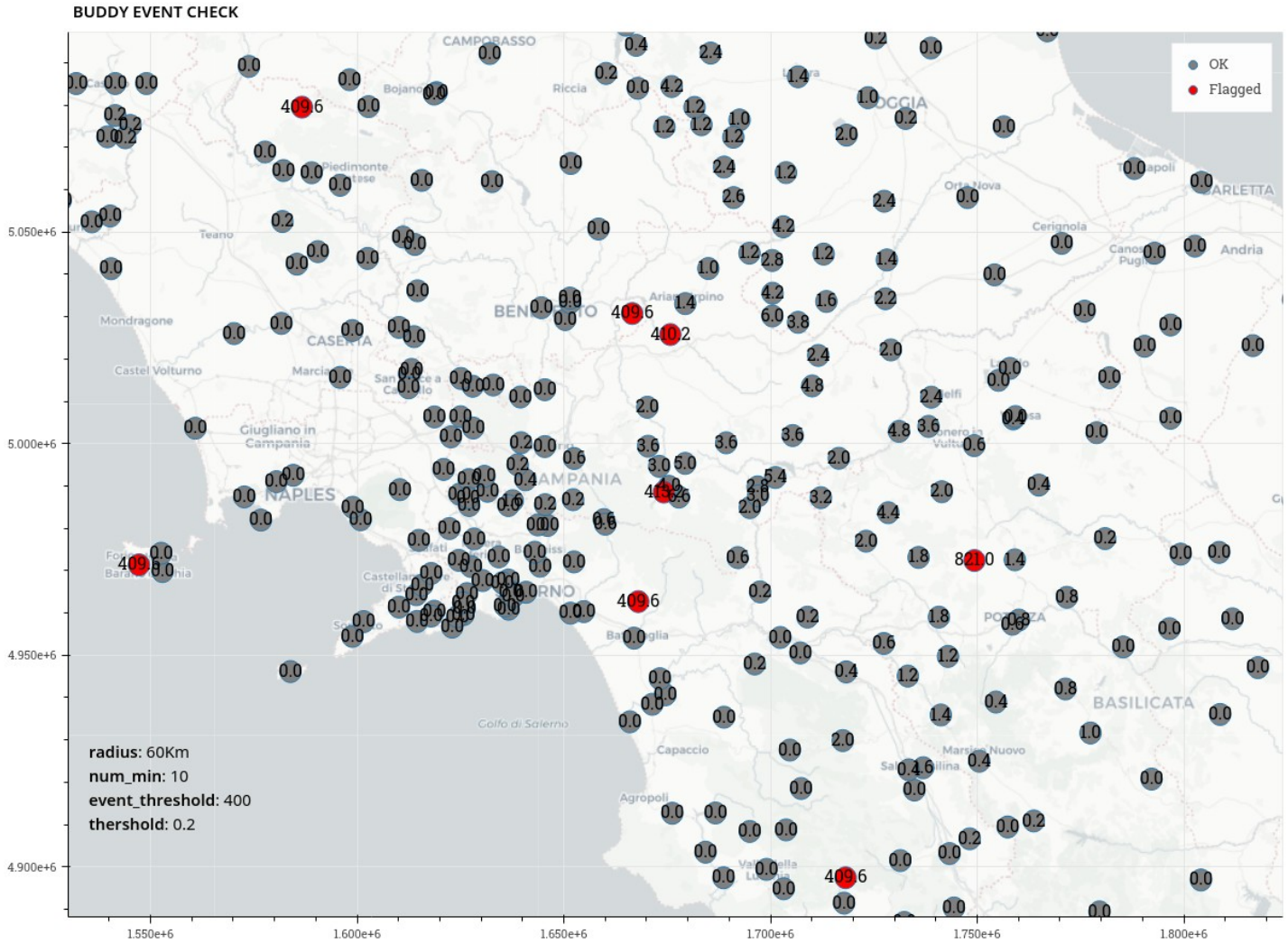
This test is similar to the buddy check, except that observations are converted into yes/no values of exceeding a specified threshold (**event_threshold**).

In addition to some of the arguments in the buddy check, the test requires an **event_threshold**, which is the threshold that converts the observations into a yes/no event.

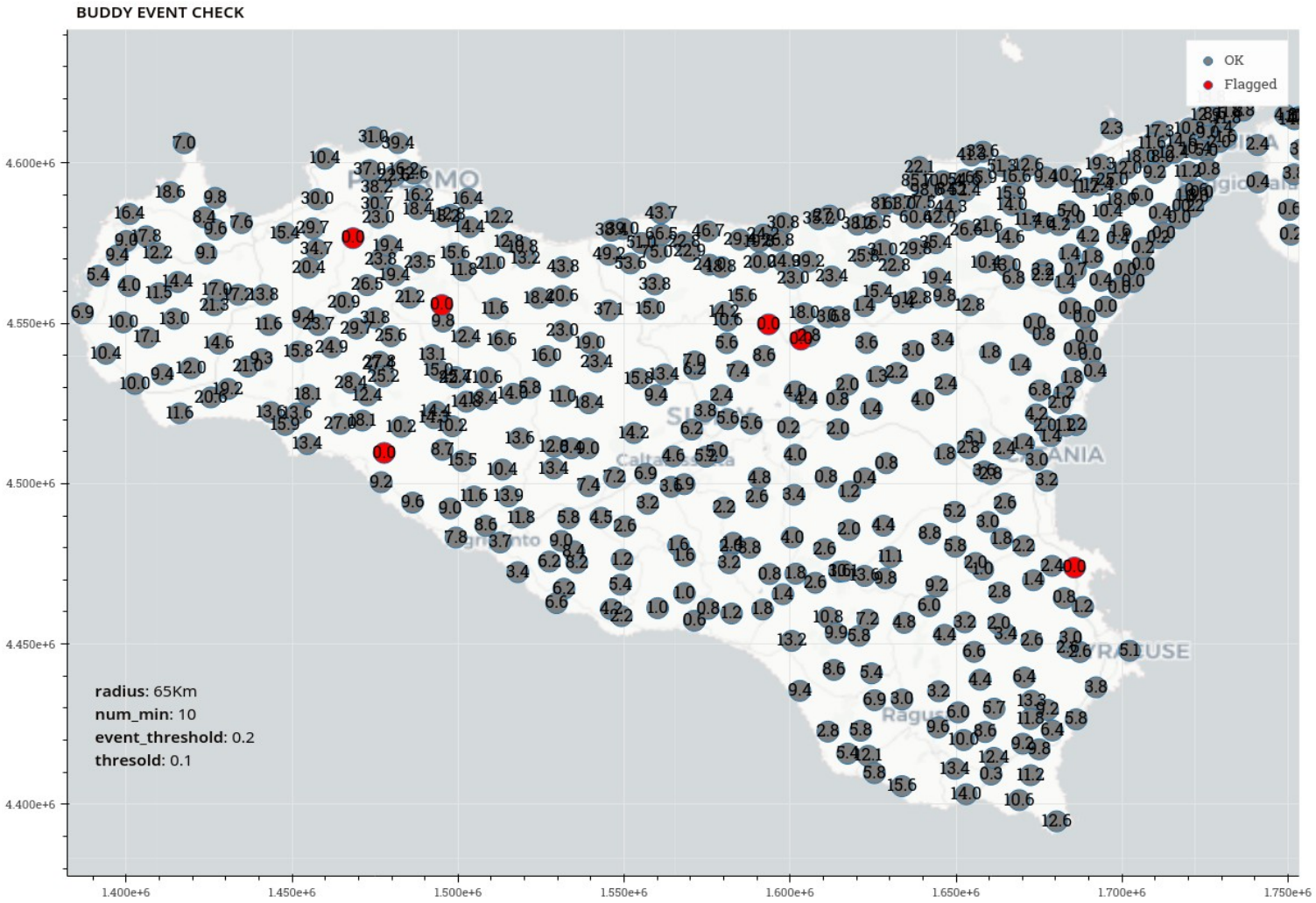
The **threshold** argument in this test is minimum fraction of other observations in the neighbourhood that must agree with the observation being inspected.

If **event_threshold**=0.2 and **threshold**=0.25, then an observation exceeding 0.2 requires that at least 25% of the other observations within the radius also exceed 0.2. If an observation is less than 0.2, then 25% must also be below 0.2.

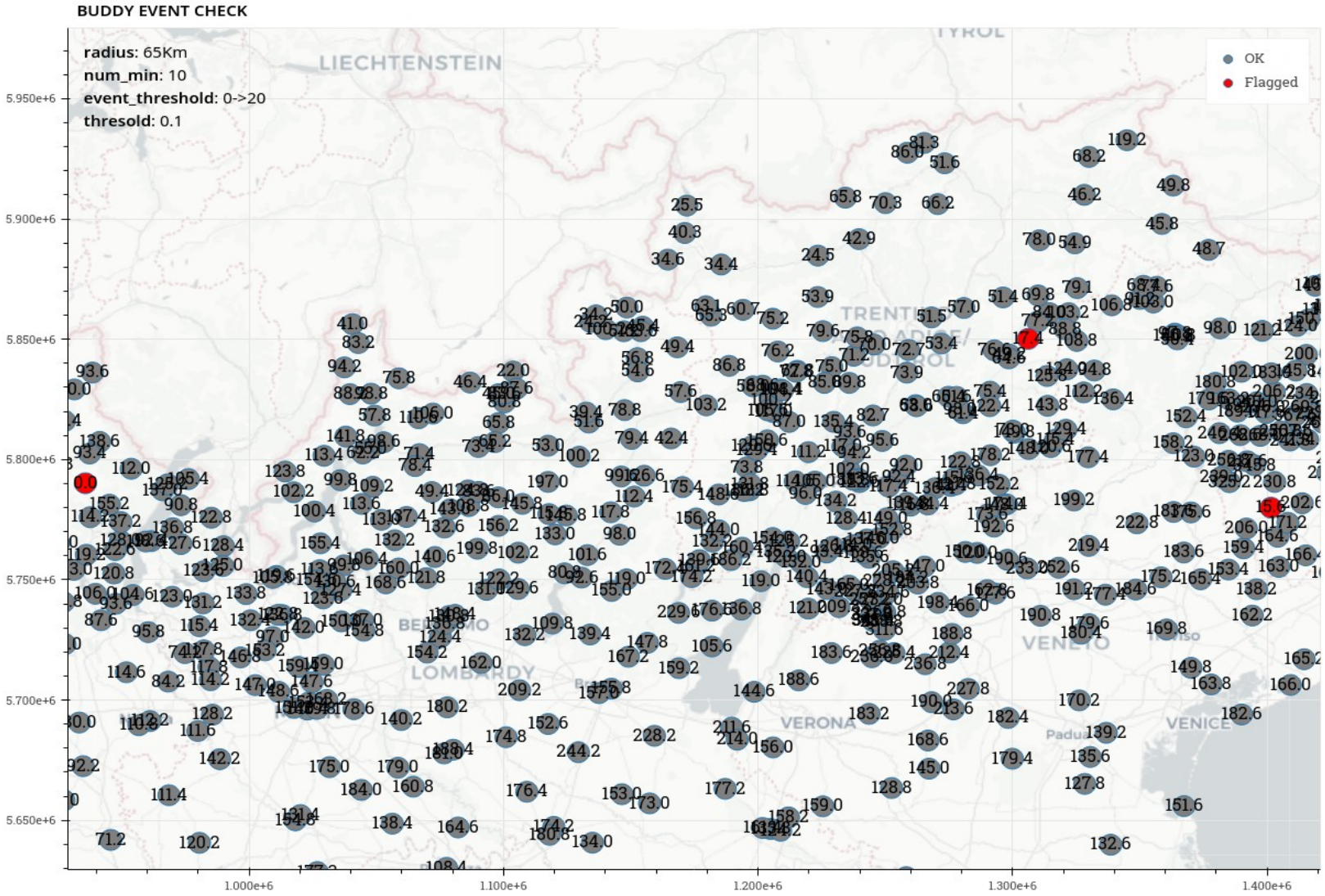
BUDDY EVENT CHECK on daily data



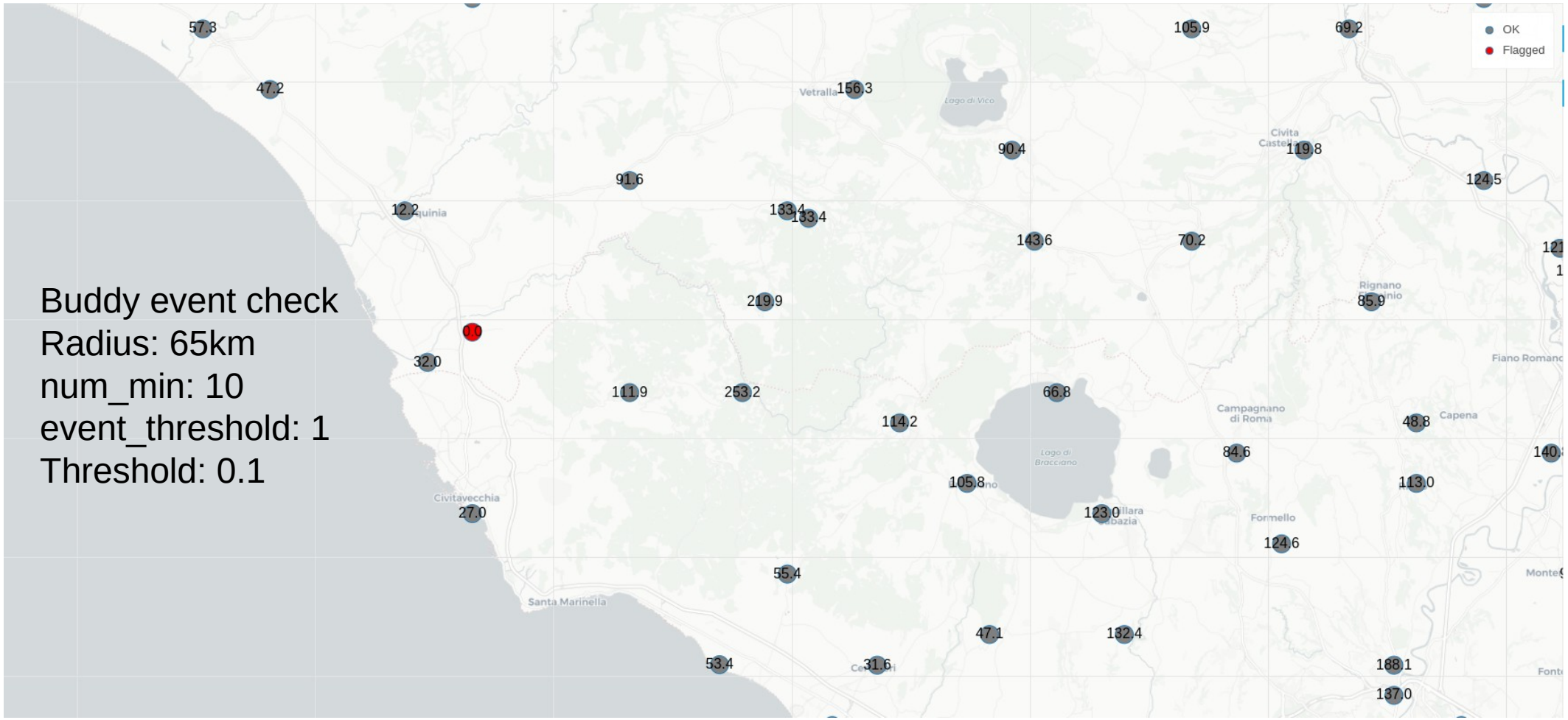
BUDDY EVENT CHECK on daily data



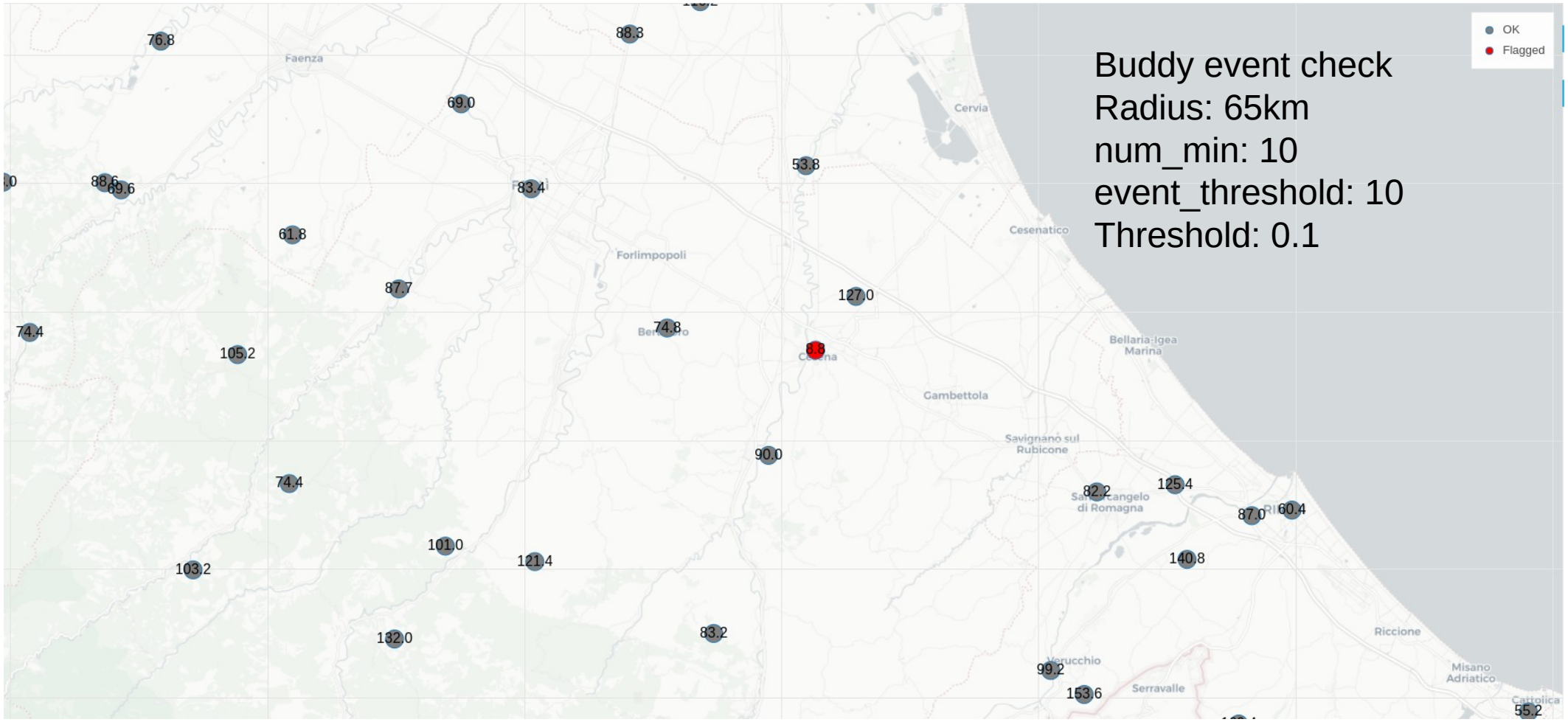
BUDDY EVENT CHECK on seasonal data



BUDDY EVENT CHECK on seasonal data (JJA 2023) Mignone station (Lazio)

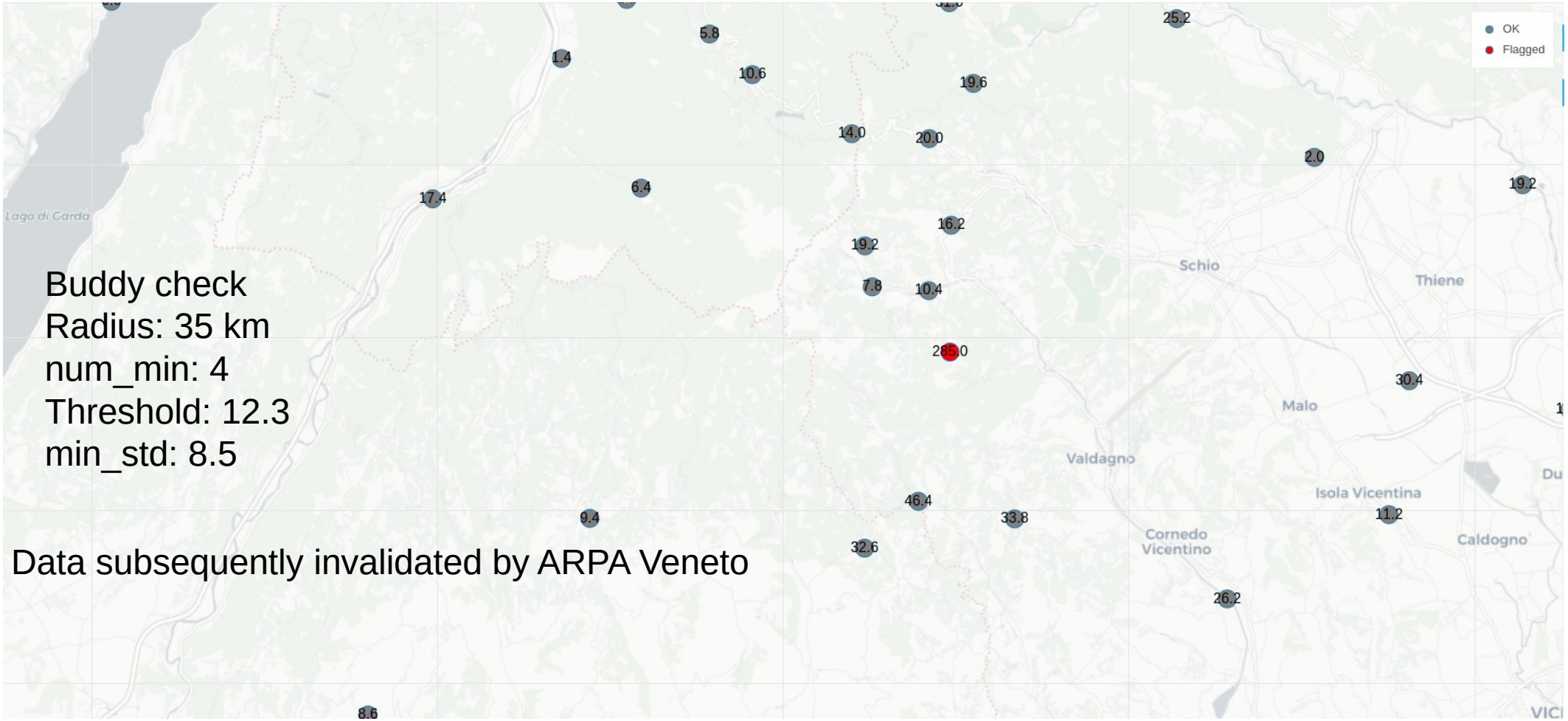


BUDDY EVENT CHECK on seasonal data (JJA 2023) Cesena station (Emilia Romagna)



BUDDY CHECK on daily data

20230603 Recoaro station (Veneto)

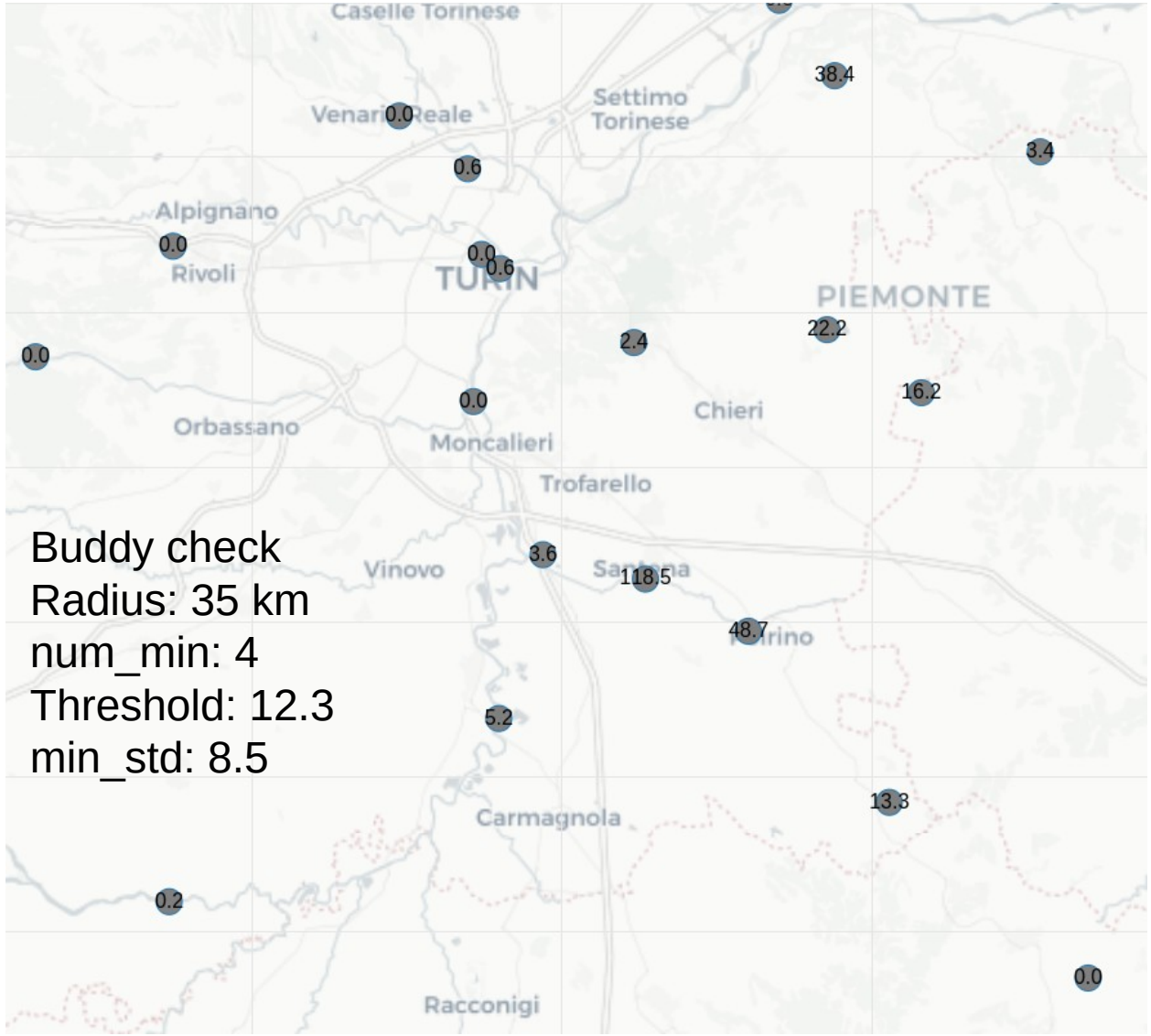


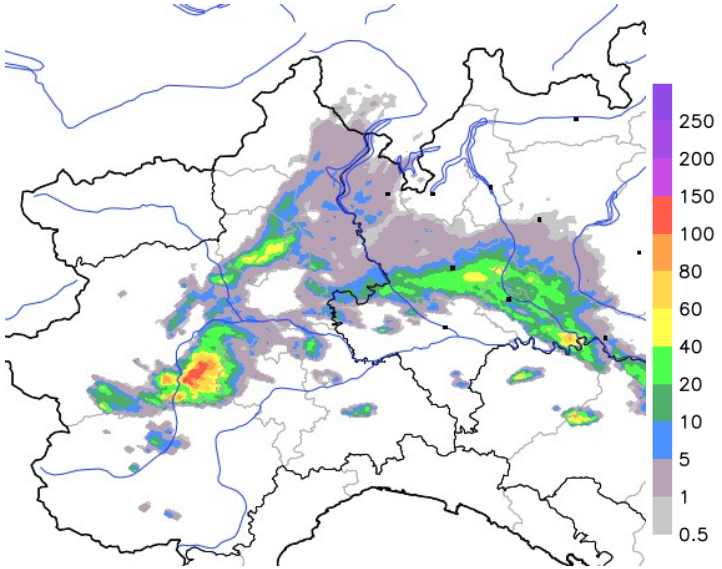
Buddy check
Radius: 35 km
num_min: 4
Threshold: 12.3
min_std: 8.5

Data subsequently invalidated by ARPA Veneto

BUDDY CHECK on daily data

20230714 Santena station (Piemonte)

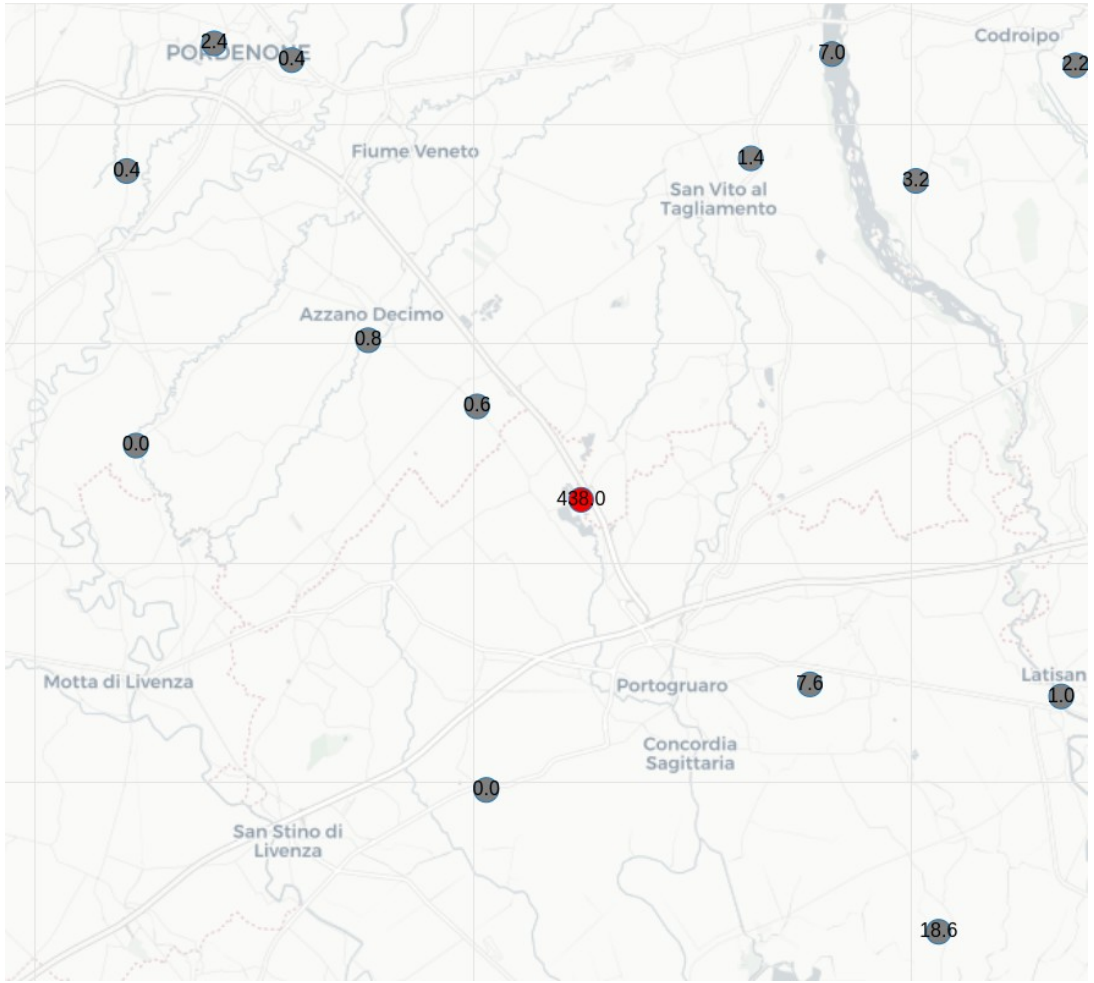




BUDDY EVENT CHECK on daily data

20230606 Mure station (Friuli)

Buddy event check
Radius: 60 km
num_min: 10
event_thresold: 400
Thresold: 0.2



“EXPERIMENTAL” CONFIGURATION IN CASE OF CONVECTION

During summer time with convection events it is not possible to use the same tests as winter time since the radius and the density of the stations would lead to invalidate also correct data

Buddy event check daily R:60000 B:10 ET:400 T:0.2 → to catch very big outliers

Buddy event check daily R:60000 B:10 ET:0.1 T:1 → to catch “0” values

Buddy check daily R:35000 B:4 T:12.3 STD:8.5 → to catch suspicious data (big value inside small, small value inside big)

Buddy event check daily R:15000 B:10 ET:0.1 T:0.01 → to catch “0” isolated or “not 0” isolated

Buddy event check seasonal R:60000 B:10 ET:0.1/1/5/10 T:0.2 → to catch seasonally very small and isolated value

SUMMARY

- The tool is powerful and flexible
- There are degrees of freedom in the choice of thresholds/radius of influence
- These values depend on the characteristics of the network and must be found empirically
- These values depend on the season considered and they must be tuned empirically. Particular attention must be paid to the convective cases in order to not eliminate right value due to thunderstorm events
- It could be used for other variables (temperature for instance)