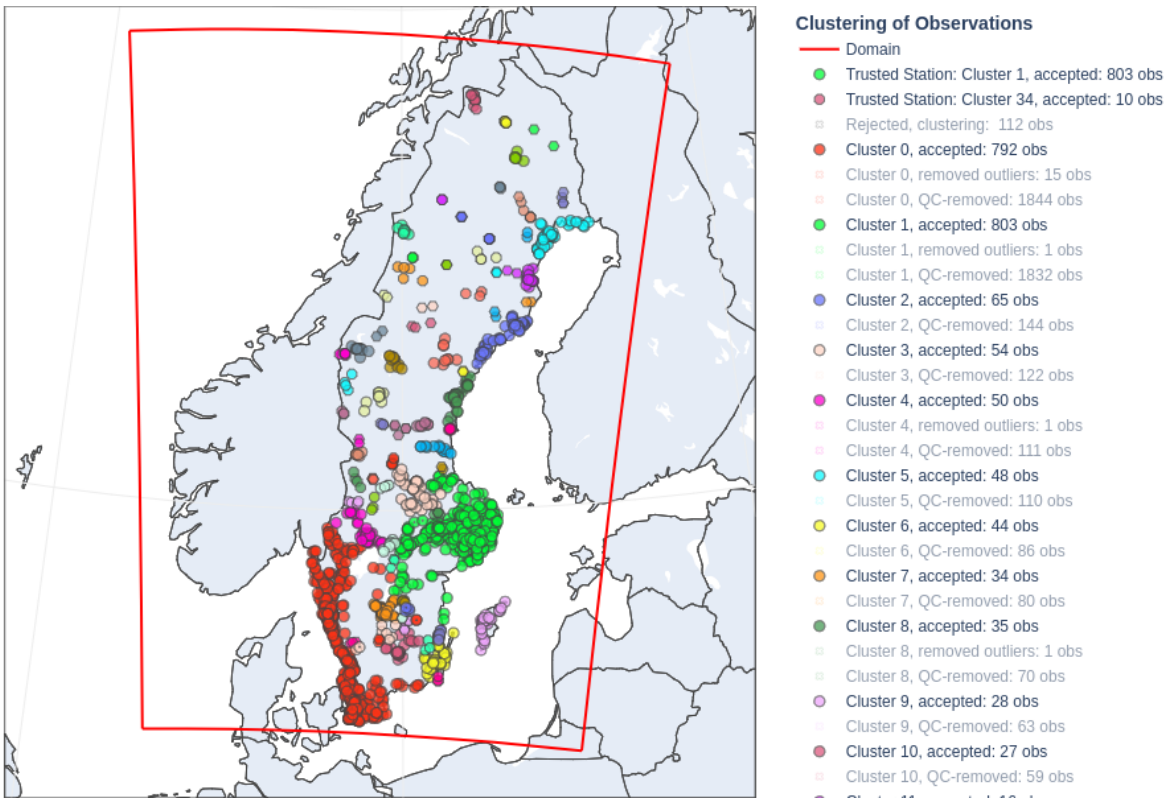
EUMETNET R&D Study A1.05 - Quality Control Tools for Observations from Personal Weather Stations

Final Report

NetAtmoQC Tool, Paulo Medeiros, SMHI



Introduction

Personal weather stations (PWSs) are an important source of meteorological data. Across Europe, for instance, they now deliver millions of meteorological observation data entries every minute [1]. The sheer amount of data, however, brings a series of difficulties when trying to establish the quality of the provided observations, including difficulties in the standardisation of the reported measurements as well as the technical challenges in parsing the data in an efficient way.

In this report, part of EUMETNET R&D Study A1.05 - Quality Control Tools for Observations from Personal Weather Stations, we at the Swedish Meteorological and Hydrological Institute (SMHI) use the NetAtmoQC package, developed in-house, to investigate the use of unsupervised machine learning clustering methods for performing the quality control (QC) of such types of observations. More specifically, we investigated the following questions: (i) To which extent the comparison against neighbouring stations is effective in detecting gross measurement errors? (ii) Where do such methods return the most satisfactory results? Addressing these questions is the aim of this report.

The NetAtmoQC Code

NetAtmoQC is a python package that uses Machine Learning Clustering methods to perform QC of observations collected from personal weather stations. Its development started at SMHI as part of the iObs project [2], originally aimed at dealing with NetAtmo [3] observations – thus the package name.

Development, however, continued during this study, and now the code is also able to deal with WOW data [4] as provided in the EUMETNET Sandbox [1]. Admittedly, this renders the NetAtmoQC name somewhat less-than-ideal, which may cause the name of the package to be changed in the future. For simplicity, however, we decided to continue using it during the course of this work.

The code is documented in its project's Wiki [5], which is updated continuously as needed. NetAtmoQC also has numerous options for providing usage information. These can be accessed from the command line by using the “-h” options of the main command and its subcommands. Moreover, the code contains tools to visualise input and output data, and these tools were used to produce the figures included in this report. Relevant details about the methods and strategies implemented in NetAtmoQC are discussed in the next sessions.

Graphical User Interface

NetAtmoQC has an integrated graphical user interface (GUI) aimed at helping users visualise the results of applying the implemented QC methods on the input data, as well as to make it possible to quickly investigate how the various input parameters affect the quality control process. More complex tasks, however, are only available via command line.

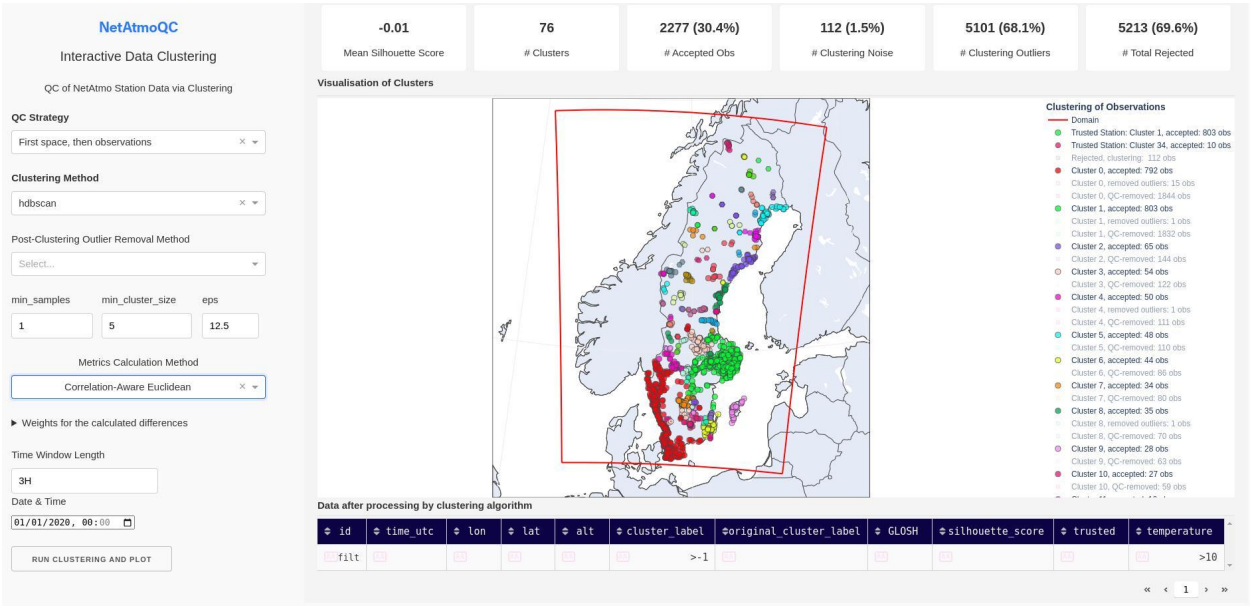


Figure 1: Graphical user interface (GUI) provided with NetAtmoQC.

Data Preparation

Raw Data Retrieval and Processing

NetAtmo and WOW data for Sweden for the period of January 2020 were downloaded from the EUMETNET sandbox and stored locally. Naturally, the format of the raw data was not the one expected by NetAtmoQC, and some work on the pre-processing of the raw data was thus needed.

Although it is in principle possible to add code into NetAtmoQC to ingest the raw data directly *as is*, such a strategy has proven inefficient due to the way the raw data is structured. For instance, if one wishes to parse the raw data directly and select measurements for a given station within a given time period, then various files need to be parsed multiple times each, which makes the whole process very slow. We therefore adopted the second most obvious strategy in this case: converting the sandbox raw data into the format that NetAtmoQC already supports.

The data format supported by NetAtmoQC is described in the code's online Wiki [5]. The code has been extended to contain tools to perform the conversion of the provided sandbox data. This can be done using the **convert** command, implemented specifically for this project. The command is self-documented, with help options accessible directly from the command line as illustrated below:

```
netatmoqc convert -h # For help about the command options
netatmoqc convert sandbox-netatmo  -h # For Netatmo data conversion help
netatmoqc convert sandbox-wow  -h # For WOW data conversion help
```

The conversion process consists mostly of reformatting of the data structure, incorporating metadata, removing unneeded data fields, standardising station IDs, renaming data columns (e.g., "Latitude" becomes "lat"), etc. No quality control is performed at this stage except for the removal of invalid (NaN) data entries.

Used Data Columns

The following parameters were used during the QC process:

- id: Standardised station IDs

- time_utc: Reported observation times, in the UTC timezone
- lon: Station longitudes
- lat: Station latitudes
- alt: Station altitudes
- temperature: Reported temperatures

Except for “temperature”, all the other columns are mandatory. Other variables can also be included in the QC process (e.g., pressure, humidity, etc.). This is supported by the code. However, we have found that the raw data contains a large number of incomplete entries (e.g., entries reporting only temperature at a given time, and only pressure at another time), which would cause a large number of data entries to be excluded due to the presence of invalid (NaN) values. For this reason, we restricted this study to the analysis of the temperature variable only.

The process of station ID standardisation consists in creating unique IDs for the reporting stations by combining the various (potentially non-unique) identifiers found in the raw data for each station.

The QC Method

Assumptions

The premise adopted in NetAtmoQC is: similar measurements reported by a large enough number of neighbouring stations are probably trustworthy.

Such an assumption, of course, is not bullet-proof. Consider NetAtmo stations, for example. These stations are positioned by the users themselves. Eventually, It may happen that a group of users in a given neighbourhood wrongly place the external temperature sensors indoors. This could cause temperatures in excess of 20°C to be reported mid-winter, say at midnight in Sweden, where such an event is highly unlikely. Such measurements would however be considered trustworthy by the implemented method, provided that a large

enough number of them were available in such neighbourhoods. We argue, nonetheless, that the adopted premise is a reasonable one, as such circumstances should in principle not be encountered often. In fact, if they were to occur frequently, then one would need to ponder if even the apparent good-quality measurements reported were to be trusted at all, given the obvious difficulties in getting the stations placed correctly. Fortunately, for the cases analysed in this study, these occurrences seem to be exceptions.

Clustering

Clustering is the process partitioning data into distinct groups such that similar data entries are grouped together (i.e., they get the same labels). The clustering method used throughout this study was HDBSCAN [6].

Adopted QC Method

The method initially adopted in this study was outlined in the submitted proposal. In that method, we defined a generalised metric (measure of distances between observations) and passed it to the adopted clustering routine. During the execution of the study, however, we implemented and tested other strategies and found it more efficient and effective to use a slightly different approach than that outlined when submitting the application. In this section, we describe the method actually used for the QC in this report.

The strategy we originally intended to follow consisted in defining a generalised metric, taking into account both spatial distances as well as weighted differences between the values of the reported measurements. While we could obtain satisfactory results in our preliminary tests using such an strategy – which remains implemented in NetAtmoQC, it led to very high rejection rates (around 95%). The reasons for this are multifold, including difficulties in calibrating the weights associated with each term in the custom metric. This motivated us to look for alternatives.

Among all strategies we attempted (which also remain implemented in NetAtmoQC), the one we adopted is “first spatial clustering and then outlier removal”. By this, we mean that we perform the QC in two steps: First, we disregard non-spatial information and perform a purely spatial clustering of the input data. This allows us to identify clusters of observations that are located close together in space. The second step is then to iterate over all the identified clusters and apply an outlier removal scan [7] over the data only, knowing that observations within each cluster are, by construction, located near each other in space. This method allowed us to reduce rejection rates to about 70% while keeping miss rates at a low level.

Furthermore, at the time we wrote the proposal, we used to select a large time window (e.g. one month), split it into smaller windows (e.g., 3 hours), perform separate QC runs for observations reported within each of these smaller time intervals, and then flag, at the end, those stations that had been rejected in more than a certain percentage of the individual QC runs. While this is a valid approach (which is still implemented in NetAtmoQC), we shifted focus to actually performing QC independently on a selected set of small time windows only, as this has the potential to allow the code to be used in real-time at, for instance, a pre-processing stage before each cycle in a numerical weather prediction simulation.

Before we proceed, it is worth mentioning two of the main limitations of the method used here:

- At the moment, the code performs no correction for biases in the reported measurements
- The code can fail to reject bad observations if a large enough number of similarly bad observations are located close to each other, as explained when we discussed the method’s assumptions.

Results

The Selected Test Case

NetAtmoQC aims to be able to identify outlier measurements without previous knowledge of what a good measurement should look like. It does it by arranging similar measurements into similarity groups - clusters, according to the methods described in the previous sections. This means that it seeks to spot extreme measurements, *i.e.*, measurements that stick out from others.

We have thus chosen to run our tests using reported **temperatures** over Sweden in the period between 2020-01-01T00:00:00 and 2020-01-02T23:59:59. We considered this to be an interesting choice because we have noticed, by manually inspecting some of the NetAtmo data available in the sandbox, that many stations in that region report temperatures over 20°C during this period, including at midnight. This, of course, is not at all expected for the Swedish winter, and these measurements are in fact generally not consistent with more reasonable ones reported by neighbouring stations. This is illustrated in the figure below:

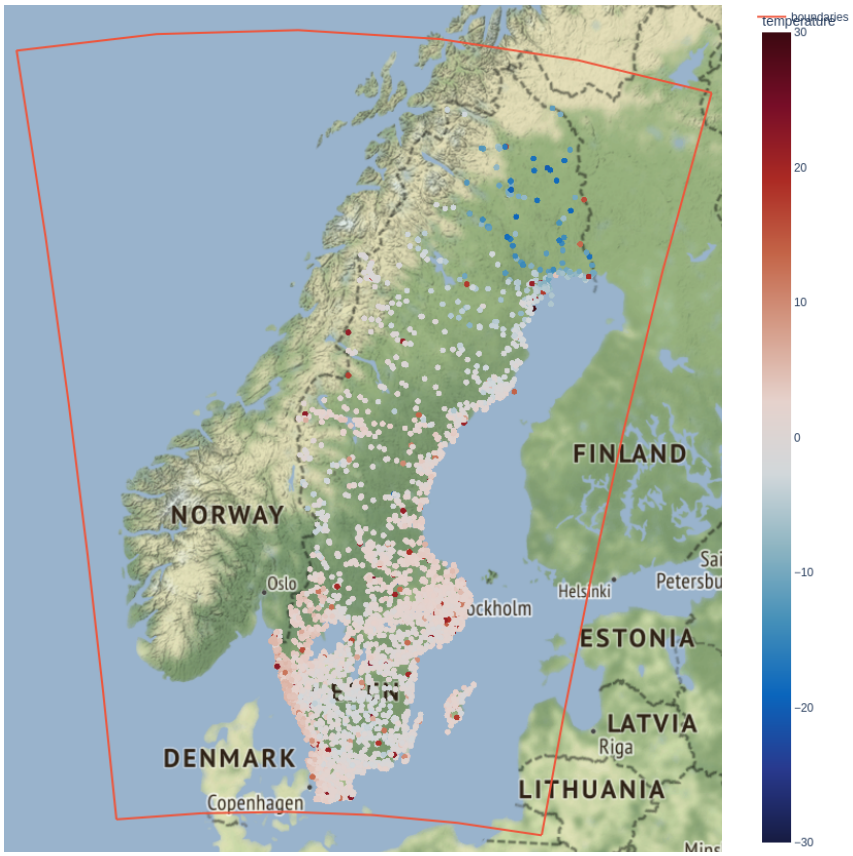


Figure 2: Scatter map of temperatures over Sweden in the period 2020-01-02T00:00 ± 15 minutes as reported by NetAtmo stations. Each dot represents a measurement, and the colours of the dots indicate the reported values according to the scale shown on the right. The data used to produce this figure had not yet been submitted to quality control, and obvious outliers can be observed (red dots).

In the figure, it is easy to identify the outliers: red dots representing reported temperatures that are much higher than those reported by their neighbouring stations. This is the type of gross measurement error that NetAtmoQC aims to spot and remove without human intervention or previously available reference (trusted) data.

In the following, we will focus our analysis on the specific time window 2020-01-02T00:00 ± 15 minutes. The results and conclusions are qualitatively the same using data from other

time windows. Furthermore, the results presented here have been obtained using only NetAtmo data from the sandbox. Although the code is prepared to deal with WOW sandbox data, the discussion and conclusions in this report should not depend on the particular data source.

Removal of Observations with Gross Measurement Errors

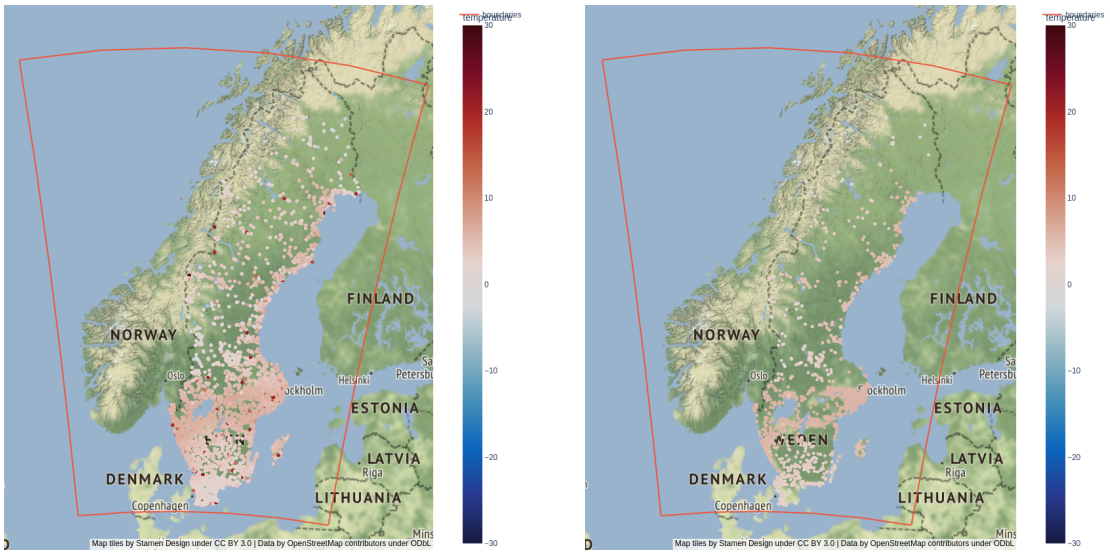


Figure 3: Temperatures reported over Sweden in the time interval 2020-01-02T00:00 ± 15 minutes prior (left-hand side) and after (right-hand side) quality control using NetAtmoQC. Each dot represents a measurement, and the colours of the dots indicate the reported values according to the scales shown. After quality control, all obvious outliers (red dots on the right-hand side subfigure) were removed.

The figure above shows the initial, non quality-controlled, data (left-hand side) as well as the observations accepted after performing QC with NetAtmoQC (right-hand side) for observations in the time interval 2020-01-02T00:00 ± 15 minutes. It is evident, from a quick visual inspection of the figures, that the obvious outlier measurements (red dots in the first figure) are removed in the QC process. In fact, we have verified from the list of accepted measurements produced by NetAtmoQC for this run, that there are no observations at all

with temperatures above 6°C after QC of the data. The code has therefore identified and removed all the observations expected to have been removed, leading to a hit rate of 100% for this particular case. This is illustrated in the figure below.

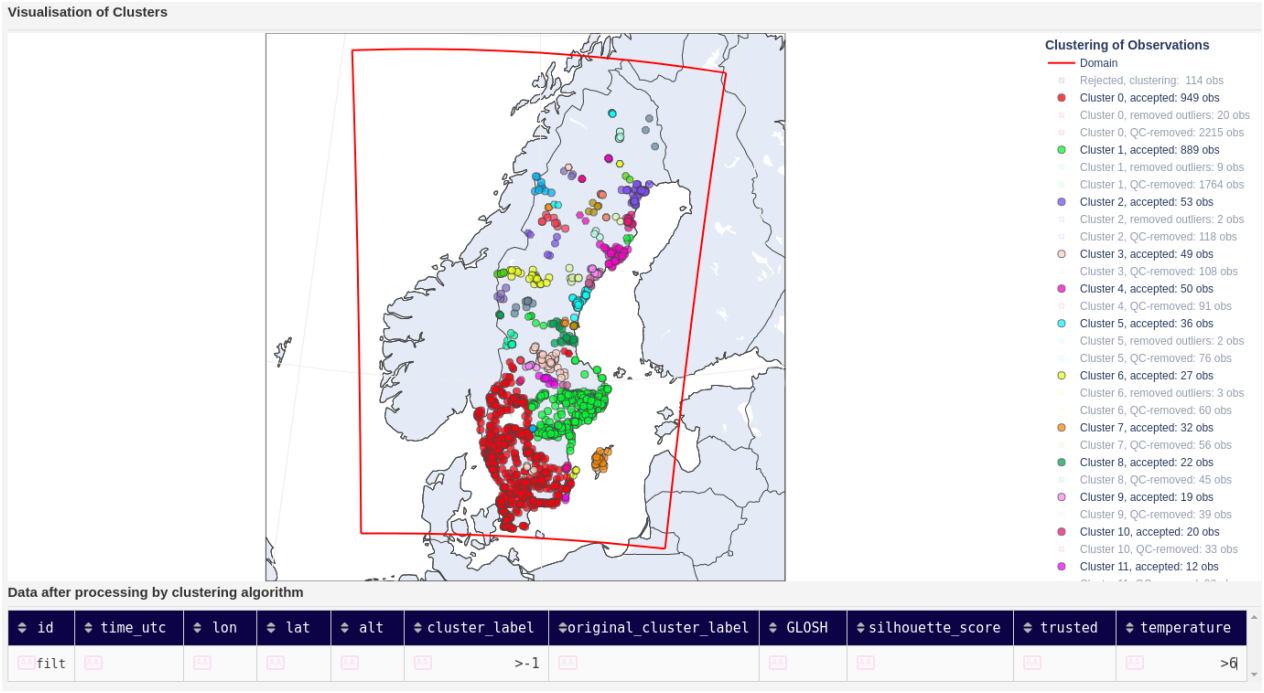


Figure 4: Output from NetAtmoQC's GUI showing QC results for temperatures reported over Sweden in the time interval 2020-01-02T00:00 ± 15 minutes. The lower panel shows a query, on the quality-controlled data, looking at accepted stations (cluster_label > -1) that have temperatures above 6°C. The query returns no results.

A clear downside, on the other hand, is that a very large number of the initial observations are rejected in the QC process. We have observed this trend on multiple occasions using NetAtmoQC, and in this particular case the total rejection rate was 68.6%: 5100 out of the initial 7439 observations were rejected. Although it is difficult to ascertain how many of these were incorrectly rejected (as we don't know which data is correct to start with), it is clear that the NetAtmoQC rejections may contain a large number of false positives.

Comparison Against Trusted Measurements

As part of this study, we have adapted NetAtmoQC to ingest data from trusted stations (e.g. synop). These data are treated in the same way as data from the regular stations, except for receiving a “trusted” flag, which helps us to identify when they have been rejected, thus allowing us to estimate the false positive rates. Note, however, that the “trusted” flag is **not** used in the clustering processes during QC. The used algorithms, therefore, do not learn any more from the trusted data than it does from the PWS data. The method remains completely unsupervised.

When ingesting trusted stations data, we replicate it so that each data point leads to twice the minimum allowed cluster size in NetAtmoQC. For this work, the minimum allowed cluster size was 5, and therefore each trusted observation was transformed into 10. Furthermore, a small random noise is added to the geographical coordinates of the replicas. Each trusted observation thus gives rise to a small cluster that has the possibility to be identified by the clustering algorithms used in the code.

The trusted data used for this work was obtained from the MORA [8] database at SMHI, and only data flagged as approved by the internal QC routines have been used.

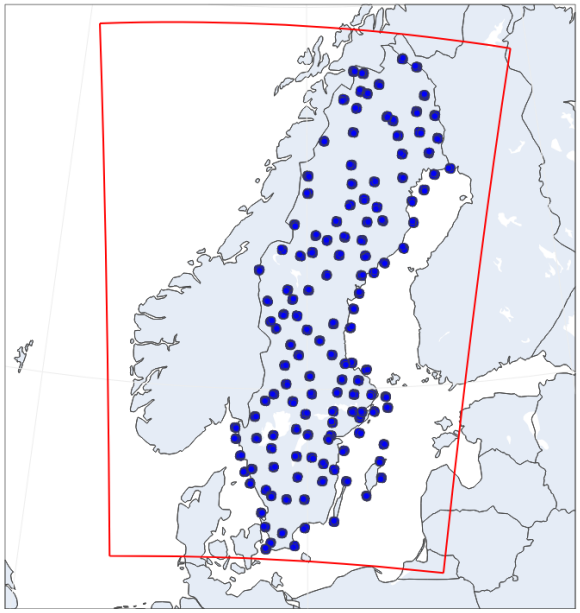


Figure 5: Quality-controlled data retrieved from the MORA database at SMHI for the period 2020-01-02T00:00 ± 15 minutes.

Including Trusted Observations Together with Regular PWS Data

Performing QC in the combined trusted+PWS data serves, in principle, two main purposes:

1. It creates clusters of good quality data around the PWS data to be analysed, which indirectly gives the algorithm a better reference of what a good observation might look like (although, as explained above, the trusted data is not training data; the method remains unsupervised);
2. It provides a way to estimate false positive rates. As we assume that the trusted data has good quality, the false positive rates can be estimated from the number of trusted observations that end up being rejected.

We tested this using the same data input as above, *i.e.*, Sweden, 2020-01-02T00:00 ± 15 minutes. We have found that 523 out of the 1350 trusted observations were rejected by the

NetAtmoQC quality control process, from which we estimate a false positive rate of about 39% for this case.

After the addition of good-quality observations, we found that 5078 out of the 7439 PWS observations were rejected, giving a rejection rate of 68.3%. A direct comparison with the QC without trusted measurements discussed earlier (where the rejection rate was 68.6%) indicates that adding trusted data seems to affect NetAtmoQC very little. We point out, however, that more noticeable changes may be observed if using a larger number of trusted data points.

Using only Trusted Observations with Added Gross Errors

It is also instructive to examine the behaviour of the code when used to perform QC on data that has already been sanitised and is known to be trustworthy. We therefore tested NetAtmoQC having as input the trusted measurements used in the previous test instead of PWS data.

Before we can use the trusted data in a meaningful way, however, we need to replicate each data point as, although the data provides a good coverage over Sweden, the stations are only a few and far between. We replicated each station in the reference trusted data by allowing the horizontal coordinates to deviate from the original ones by about 15 Km in every direction following a random uniform distribution. The altitude was allowed to vary uniformly between 0 and 10 metres. This way, we get a more realistic picture of stations spread in separate neighbourhoods, instead of having clusters of stations located at the same point.

We also simulated the common problem of incorrectly placing indoors the outdoor sensor in a PWS. We did this by randomly selecting 15% of the observations in each cluster (group

of replicas created from the same origin) and setting their temperatures to values between 19°C and 25°C picked according to an uniform distribution.

In this test, we found a rejection rate of 70%, which is similar to those in previously discussed cases. Still similarly, we have found that all observations with the introduced errors in the temperature parameter were successfully removed after QC.

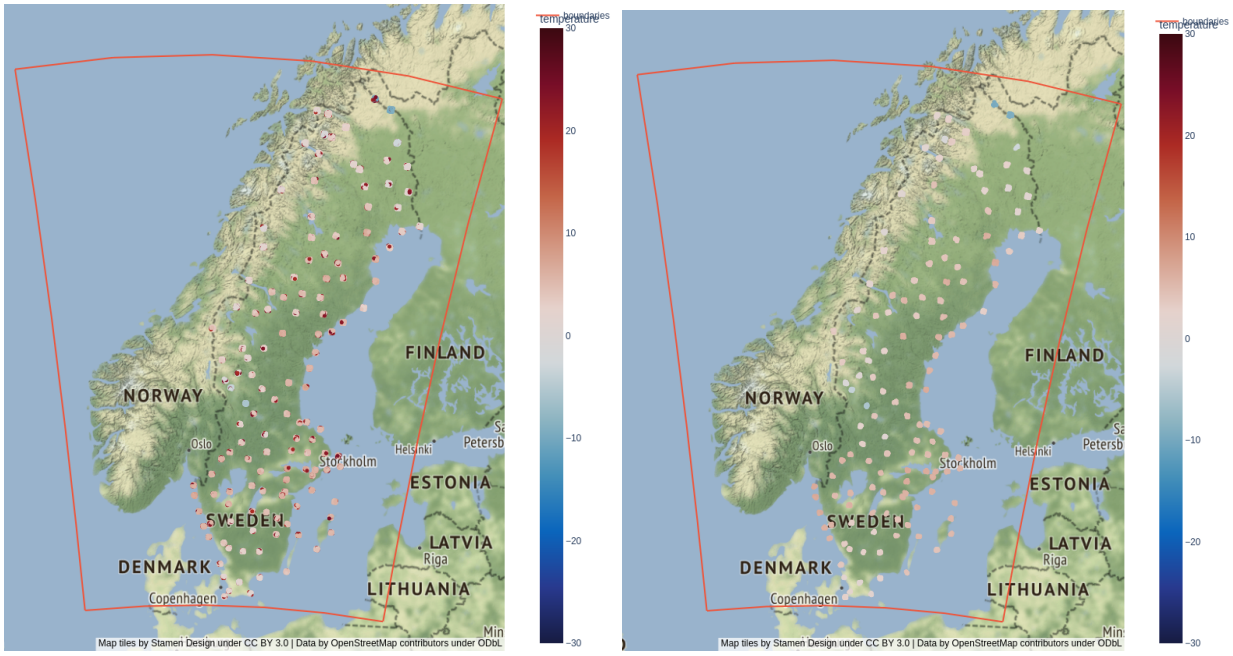


Figure 6: Temperatures over Sweden in the time interval 2020-01-02T00:00 ± 15 minutes prior (left-hand side) and after (right-hand side) quality control using NetAtmoQC. Each dot represents a measurement, and the colours of the dots indicate the reported values according to the scales shown. After quality control, all obvious outliers (red dots on the right-hand side subfigure) were removed. The data containing outliers was artificially created by adding noise to quality-controlled data retrieved from SMHI's MORA database.

Conclusions

We have shown that machine-learning clustering methods provide a viable strategy to perform quality control of personal weather station (PWS) measurements. By comparing each station/measurement with their neighbouring counterparts, one can detect and remove data entries that contain gross measurement errors.

The method used in this study has shown great effectiveness, in the adopted study case, in the unassisted identification of problematic data entries provided by PWS measurements. More specifically, it identified indoors-like temperatures provided by outdoors temperature sensors with a miss rate of 1, which was also achieved when the QC method was run on a purposely polluted dataset derived from a set of trusted measurements.

On the other hand, the method adopted here results in very large rejection rates of around 70%, and our tests with mixed PWS and trusted data indicate that it may have a false positive rate of about 40%, i.e., it may reject about 40% of those observations that could be good.

We thus believe that the NetAtmoQC code and the methods therein implemented could be a good alternative in cases where it is acceptable to discard some potentially high-quality observations in order to remove as many low-quality measurements as possible.

References

1. Met Office; Netatmo (2021): EUMETNET Sandbox: surface observations from Met Office WOW and Netatmo networks.. NERC EDS Centre for Environmental Data Analysis. http://catalogue.ceda.ac.uk/uuid/37d6ea7956a74af0bef827b94e0fb602
2. iObs Project. Please visit <https://wiki.neic.no/wiki/IOBS> for more information.
3. NetAtmo: <https://www.netatmo.com/sv-se>
4. WOW data. Please visit <https://weatherobservationswebsite.blogspot.com/> for more details.
5. Documentation for NetAtmoQC is available at <https://source.coderefinery.org/iOBS/wp2/task-2-3/netatmoqc/-/wikis/home>. The Wiki will continue to be updated as the work in this project progresses.
6. McInnes et al, (2017), hdbscan: Hierarchical density based clustering, Journal of Open Source Software, 2(11), 205, doi:10.21105/joss.00205. Please also visit <https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html>.
7. LOF Outlier Detection Method. For more information, please visit <https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html>.
8. MORA Database, SMHI. Please visit <https://www.smhi.se/data/meteorologi/ladda-ner-meteorologiska-observationer#param=airtemperatureInstant,stations=core>.