A machine learning perspective towards fully automated QC in daily weather time series

Michel Alioscha-Perez and Hichem Sahli

Electronics and Informatics Dept. (ETRO)
Vrije Universiteit Brussel (VUB)

Outline

Introduction

An overall idea of our QC approach

Selecting the right Neural Network model for daily temperatures QC

Preliminary results in Belgian climatological stations

Final remarks and practical aspects

Conclusions and Next steps

Introduction

Importance of QC

Quality control (QC) leads to more reliable data, and is therefore fundamental to increase robustness and accuracy on futher analyses.

Current solutions

Several outliers detection methods are based on specific tests, for example:

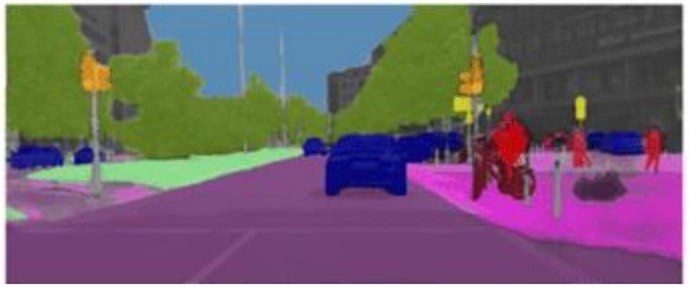
- Internal consistency (Tmax > Tmin)
- Variability tests
- Spatial consistency (i.e. using spatial interpolation from neigboring stations)
- Limit checks

* These methods are widely used and have been largely successfull in QC tasks

Motivation

However, once a possible outlier has been flagged, they require intervention of well-trained operators to assess and correct the suggested outliers. As consequence, it can be time consuming and expensive.

Introduction



Source: nvidia.com

Machine learning, Deep Learning, Recurrent Neural Networks

- Allows the computer "to learn" based on examples, and trials/errors
- Provides the ability of generalizing in unseen data
- Strong foundations based on statistics and optimization
- Outstanding results with full automation, exceeding human capabilities in many cases

* *Question:* Is it possible to do something similar in weather timeseries ?

Outline

Introduction

An overall idea of our QC approach

Selecting the right Neural Network model for daily temperatures QC

Preliminary results in Belgian climatological stations

Final remarks and practical aspects

Conclusions and Next steps

Our QC approach

An overall idea on our approach

A very simplistic example (real values/scalars): $f \cdot x = y$ intuitively one can (try) to solve f by having x and y

x : Raw data

y : Correction

we just find a "model" f that relates x and y

For example: a simple one (3 , 14, 5) => (3, 4, 5)
(17, ?, 19)

Our QC approach

An overall idea on our approach

A very simplistic example (real values/scalars): $f \cdot x = y$ intuitively one can (try) to solve f by having x and y

x : Raw data

y : Correction

we just find a “model” f that relates x and y

For example: a simple one (3 , 14, 5) => (3, 4, 5)
(17, ?, 19)

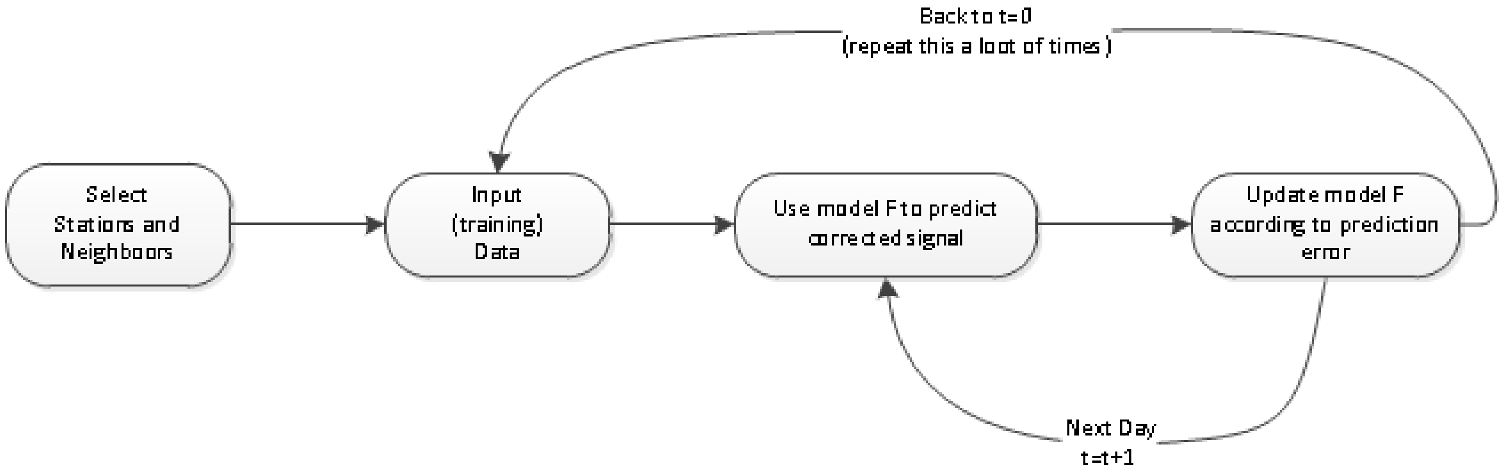
* However, it is very difficult to distinguish natural variations from errors introduced in the signal in the real problem

The real problem

To find F involves solving a highly non-linear non-convex optimization problem, commonly solved using SGD.
No assumptions on the distribution of x nor y are necessary.
Considerable progress has been recently made on how to solve these type of problems.

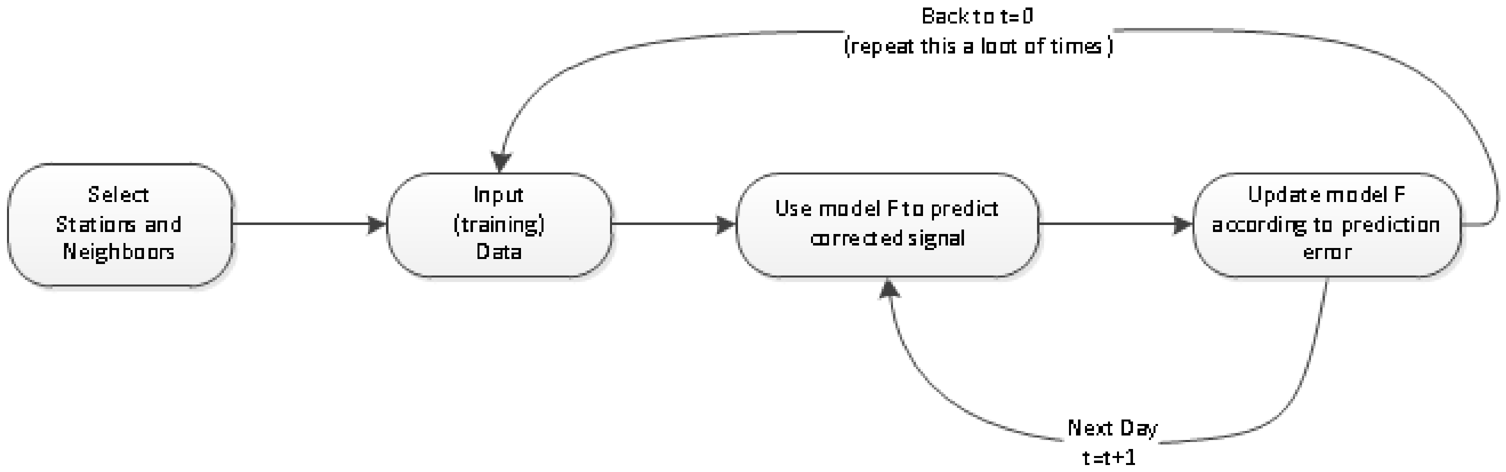
Our QC approach

Learning the model



Our QC approach

Learning the model



Back to t=0
(repeat this a loot of times)
Select Stations and Neighboors
Input (training) Data
Use model F to predict corrected signal
Update model F according to prediction error
Next Day
t=t+1

Some remarks

- This learning strategy is general and works for any model (even unknown or black-box models) as long as one can access the gradient/subgradient of the objective function[1]

- If the model can be defined as a weighted combination of different distributions, then under some (relatively mild) assumptions we can know how many iterations we have to perform to reach a desired estimation error[2]

* This means that we can indeed learn sequences to some extent, as long as we have enough samples and we perform at least certain amount of iterations; if we don’t know how many, we simply do a looooot of iterations

* Temperature values to the left and to the right of some specific date, and perhaps extra information (stations) is what’s needed to correct

[1] Reddi S.J et. al: *Stochastic Variance Reduction for Nonconvex Optimization*, Proceedings of 33th International Conference in Machine Learning, 2016.
[2] Alioscha-Perez et. al: *MKL via multi-epochs SVRG*, 9th NIPS Workshop on Optimization for Machine Learning, Dec 2016.

Outline

Introduction

An overall idea of our QC approach

Selecting the right Neural Network model for daily temperatures QC

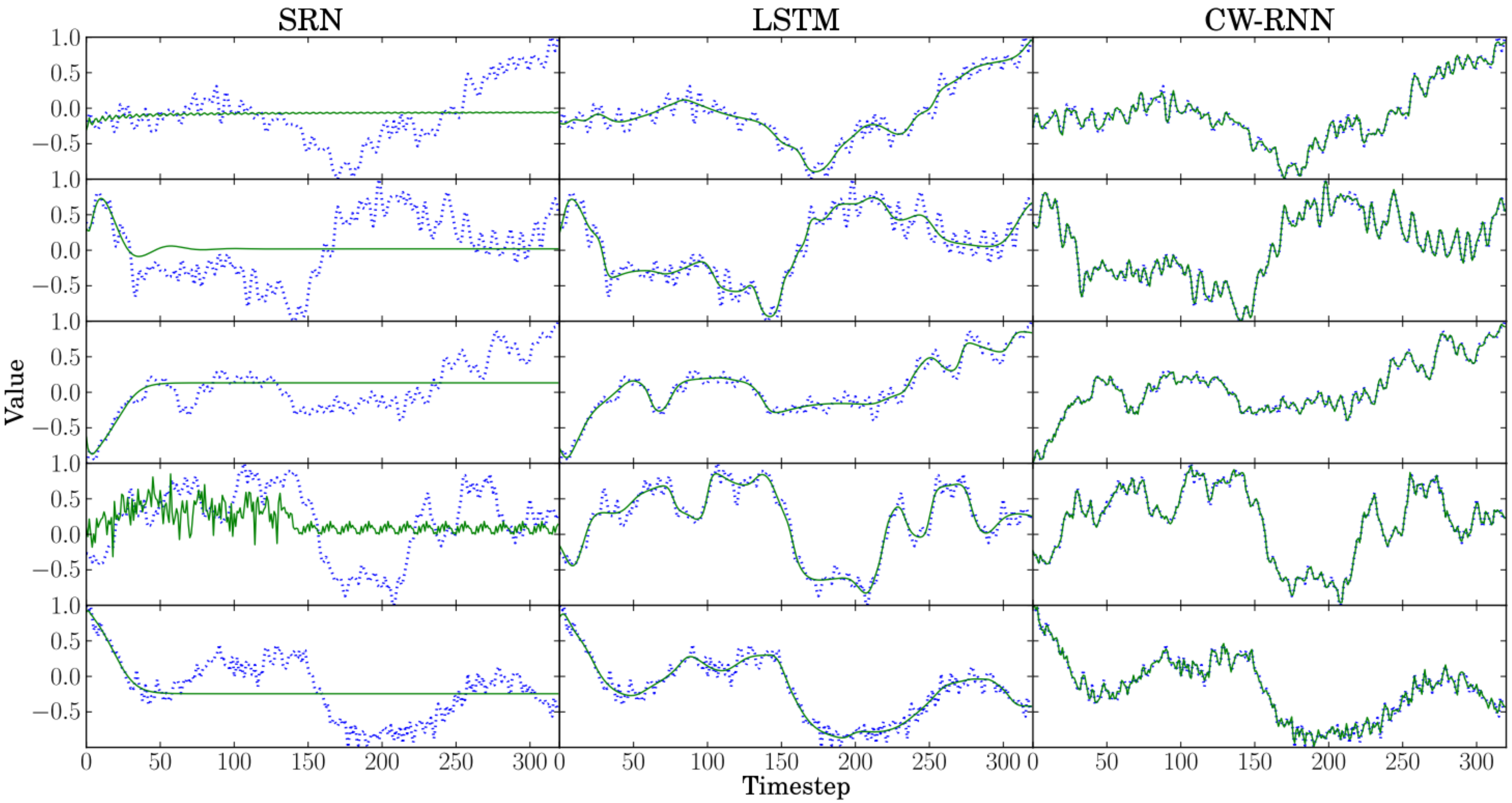
Preliminary results in Belgian climatological stations

Final remarks and practical aspects

Conclusions and Next steps

Selecting the right RNN model

Performance of different RNN in toy data



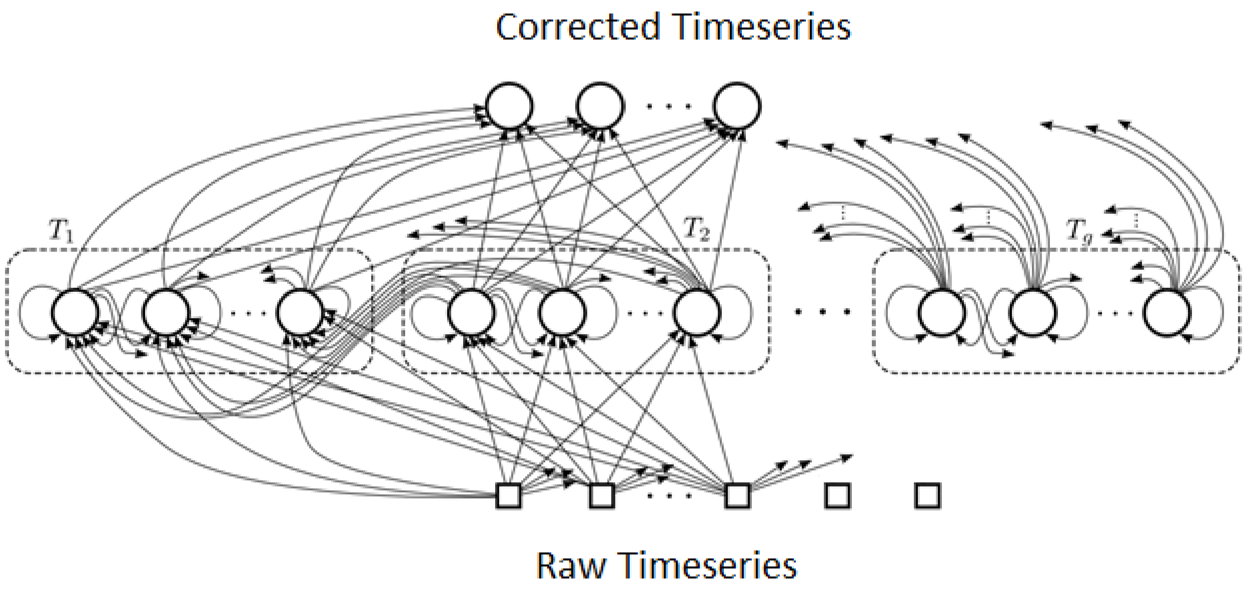
SRN: - unable to learn long-term dependencies (i.e. > 10days)

LSTM: - much better, yet high-frequency components are not well captured
- Probably a good candidate for monthly data analysis

CW-RNN: it can learn quite well long-term variations including the peaks

A closer look to the CW-RNN model

Architecture of the CW-RNN:



Outline

Introduction

An overall idea of our QC approach

Selecting the right Neural Network model for daily temperatures QC

Preliminary results in Belgian climatological stations

Final remarks and practical aspects

Conclusions and Next steps

Experiments & Discussions

Selected stations

- 10 years time periods
- Stations with at most 1% of missing temperatue values
- Missing data was filled with random values

Reference/neighbouring stations

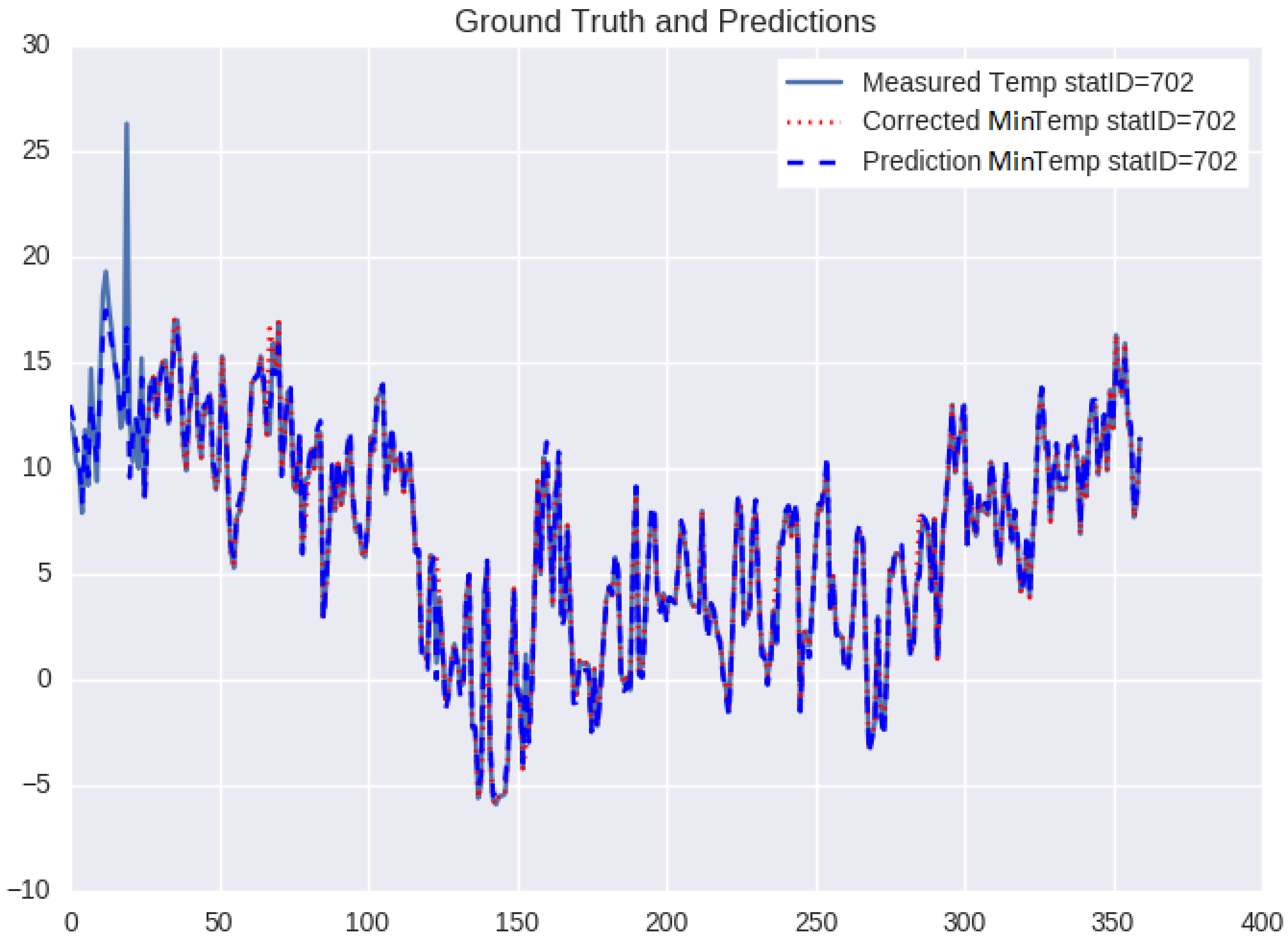
- For each station, the N closest neighbours were selected as reference

Training and testing

- The model was trained using data from Apr-2005 until Apr-2015
- Then it was used to predict data from Apr-1989 until Apr-2000

Experiments & Discussions

Preliminary results in station 702: predicting correct MIN temp values from raw MIN values and no reference

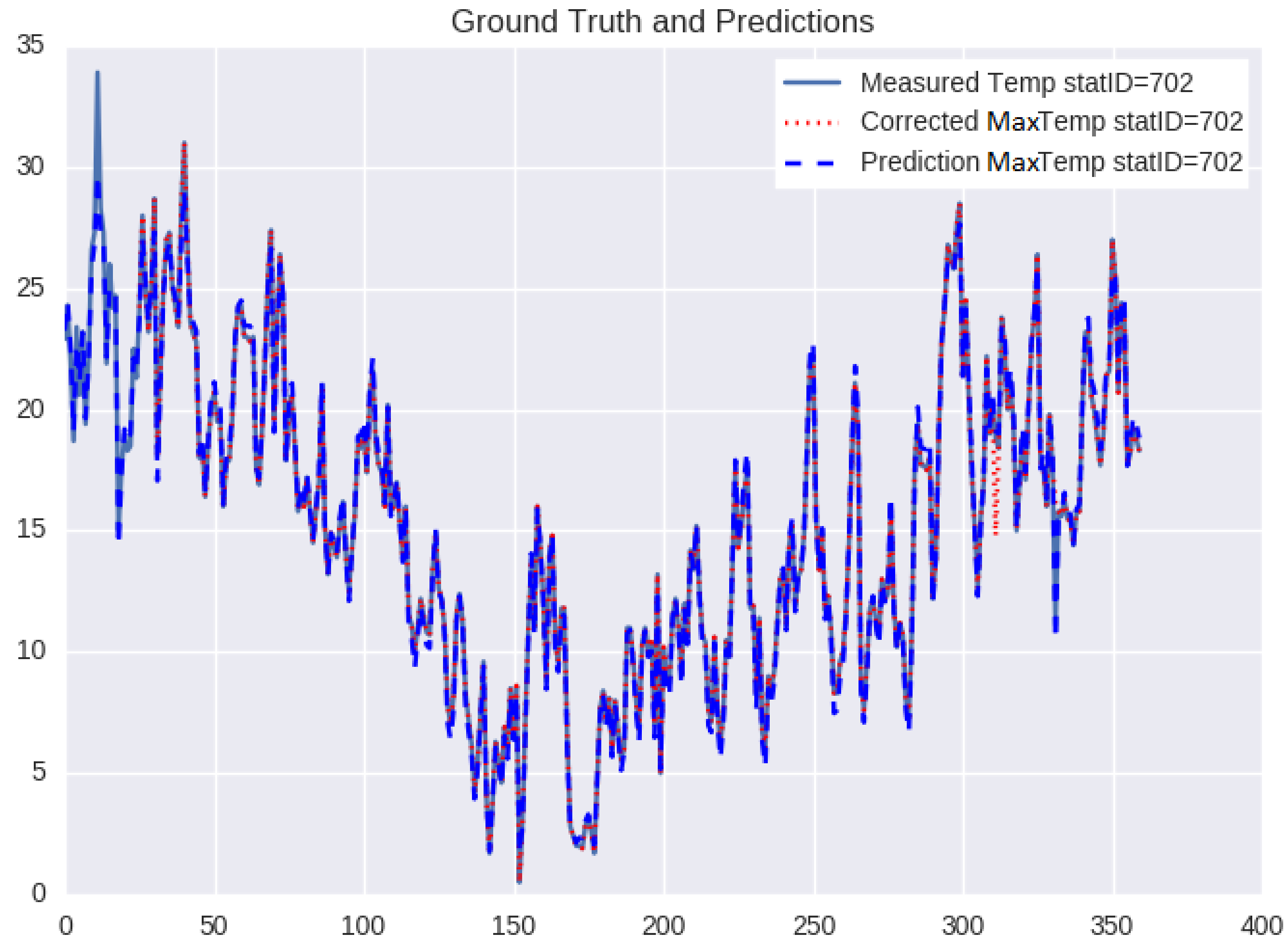


A typo from “16” to “26” is detected and a value suggested.

No neigbouring station is used as reference

Experiments & Discussions

Preliminary results in station 702: predicting correct MAX temp values from raw MAX values and no reference

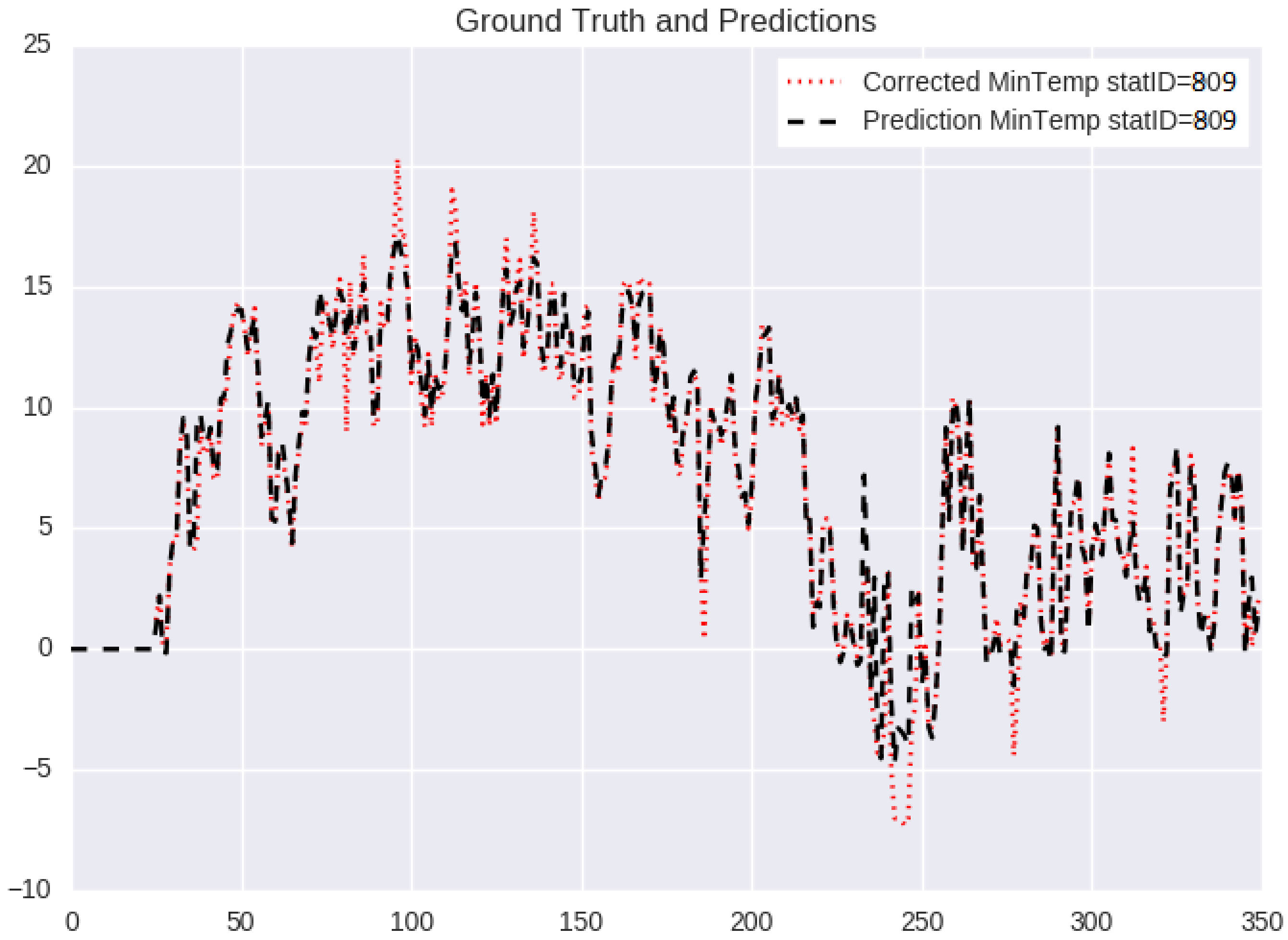


Most of the signal is correctly predicted, yet one correction is apparently missed.

Once more, no reference station is included

Experiments & Discussions

Preliminary results in station 809: predicting correct MIN temp values from raw MIN/MAX values and 1 reference

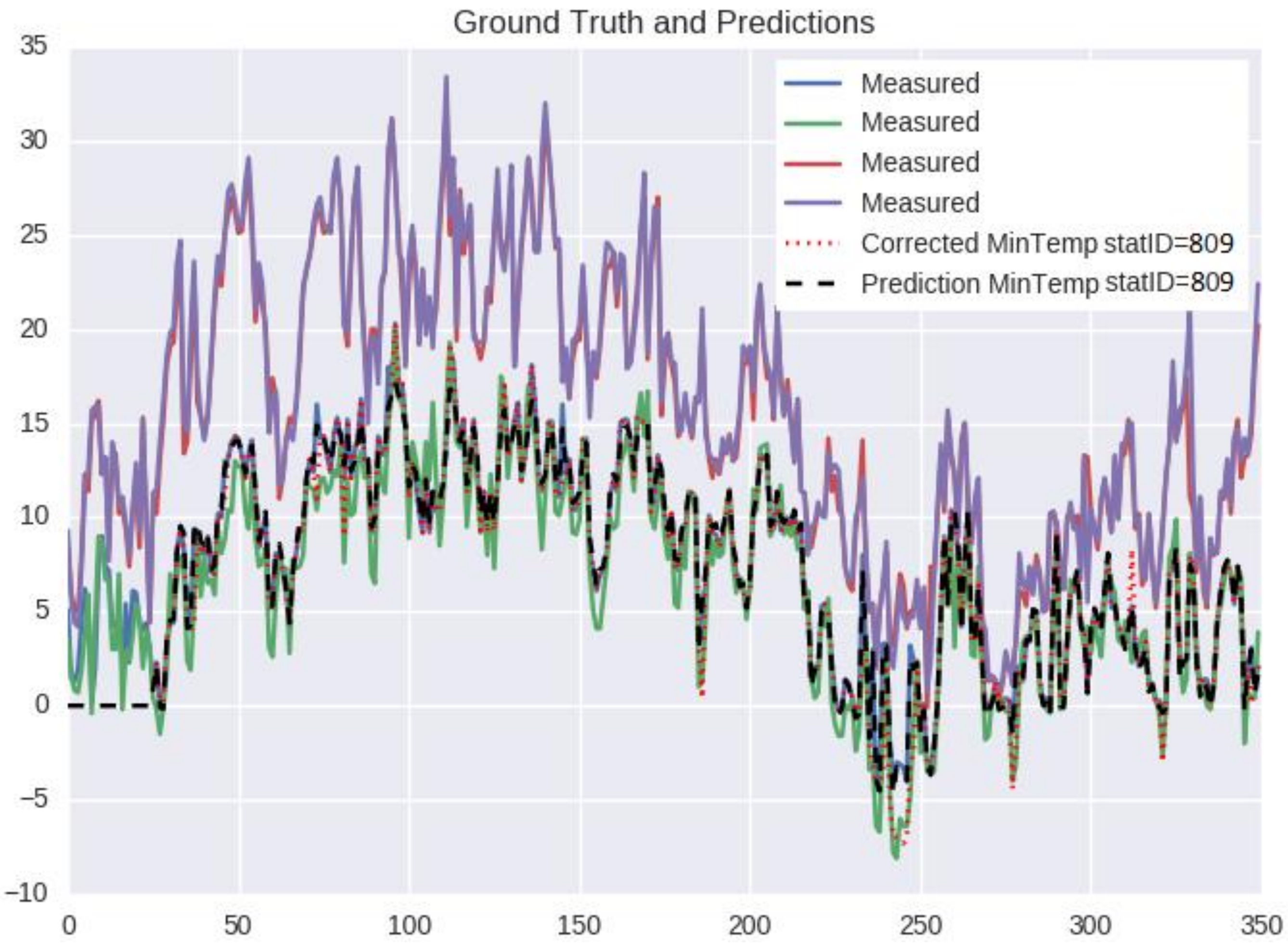


For most of the signal (Apr 1989-Apr-1990) both predictions and corrections are the same

There are several discrepancies that range from +/- ~5 degrees

Experiments & Discussions

Preliminary results in station 809: predicting correct MIN temp values from raw MIN/MAX values and 1 reference



For most of the signal (Apr 1989-Apr-1990) both predictions and corrections are the same

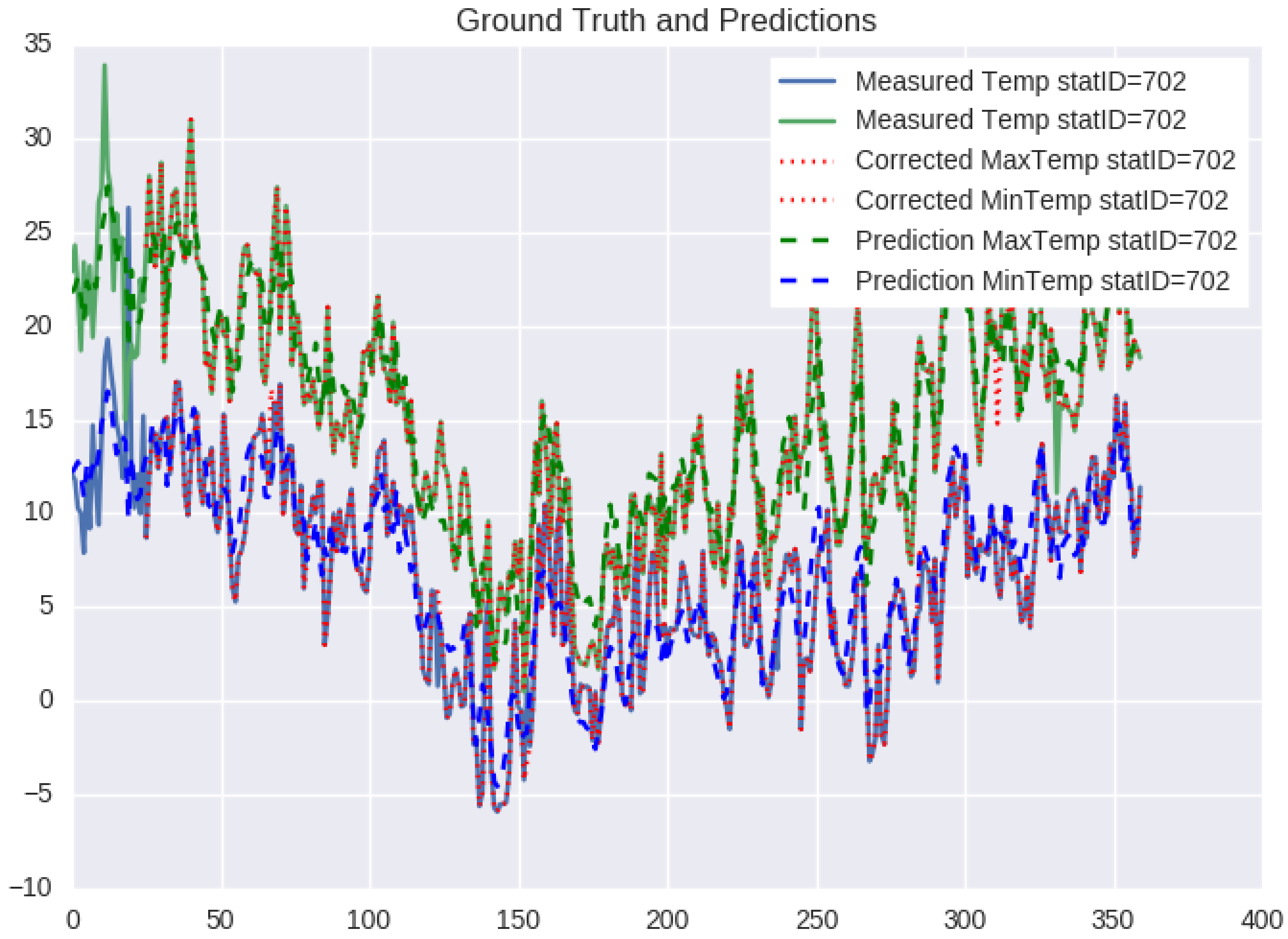
There are several discrepancies that range from +/- ~5 degrees

Experiments & Discussions

Preliminary results in station 702: predicting correct MIN/MAX temp values from raw MIN/MAX values and no reference

Not clear if its possible to exploit co-relation between Max/Min values by predicting both at once

Periodicity and variations on Tmax and Tmin values seems to be different



Outline

Introduction

An overall idea of our QC approach

Selecting the right Neural Network model for daily temperatures QC

Preliminary results in Belgian climatological stations

Final remarks and practical aspects

Conclusions and Next steps

Final remarks & practical aspects

With regards to the predictions/corrections

For the moment it seems challenging to try to go beyond < 5 degrees in terms of precision of the corrections

Developers perspective

Our implementation makes use of a deep learning backend named Tensorflow, accessed through Python.

Tensorflow was released by Google 2 years ago, is opensource, have a growing community of users and is still supported by Google.

It was implemented in c++ and is highly portable. Google provides an interface of it to Python, though interfaces from others languages are possible; for instance to c++ or Java.

Once decided the basic things such as the network architecture, loss function to use, optimization strategy, then the Tensorflow backend does most of the work for you.

As the processing progress (in terms of iterations), some checkpoints are saved. The last checkpoint is the only thing you need to keep. It is basically the “model F” that we referred at the beginning.

Outline

Introduction

An overall idea of our QC approach

Selecting the right Neural Network model for daily temperatures QC

Preliminary results in Belgian climatological stations

Final remarks and practical aspects

Conclusions and Next steps

Conclusions

As summary

- Our approach is not based on fixed-rules and basically learns them from the training data
- Provides both the outliers and their possible corrections and hence is a step closer to full automatization
- Can work with individual (i.e. isolated/separated) stations without any reference when necessary

Back to the question: Is it possible to do something similar in weather timeseries ?

The experiments suggest it is certainly possible to learn by showing the raw signals and its corresponding correction.

Is it comparable to human performance ? Proper assessment and benchmarking is needed ….

What’s next

To do

- Augment data by adding more outliers in the timeseries
- Consider different selection criterion for the reference station(s)
- A carefully designed benchmarking, possibly relying in data gathered with automated stations

Other types of RNN models

- Maybe deeper architectures allow to better capture relationships between the stations and the corrections
- Specific/custom components in the RNN to model dependencies between the networks

* Many possibilities in terms of modeling

Thanks

Köszönöm!