Predicting and reporting temperature changes measured by an on-skin monitoring device

UNIVERSITY of STIRLING

Peter Henriksen

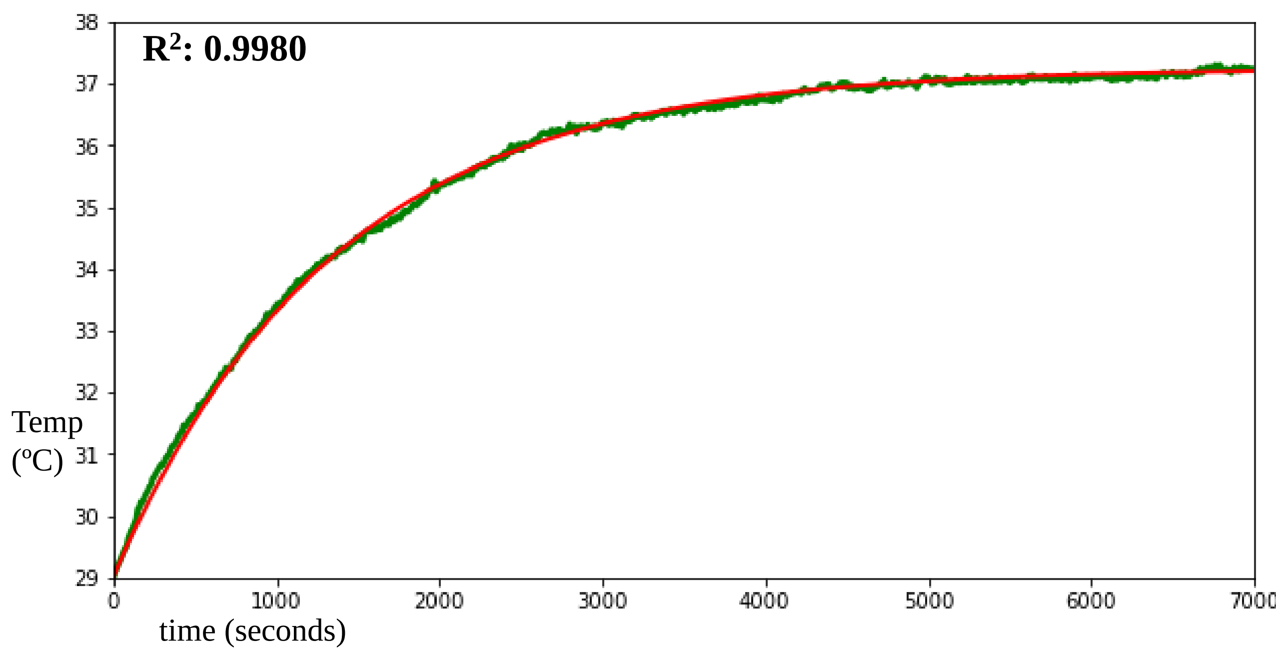
MSc Big Data

Project aims and challenges:

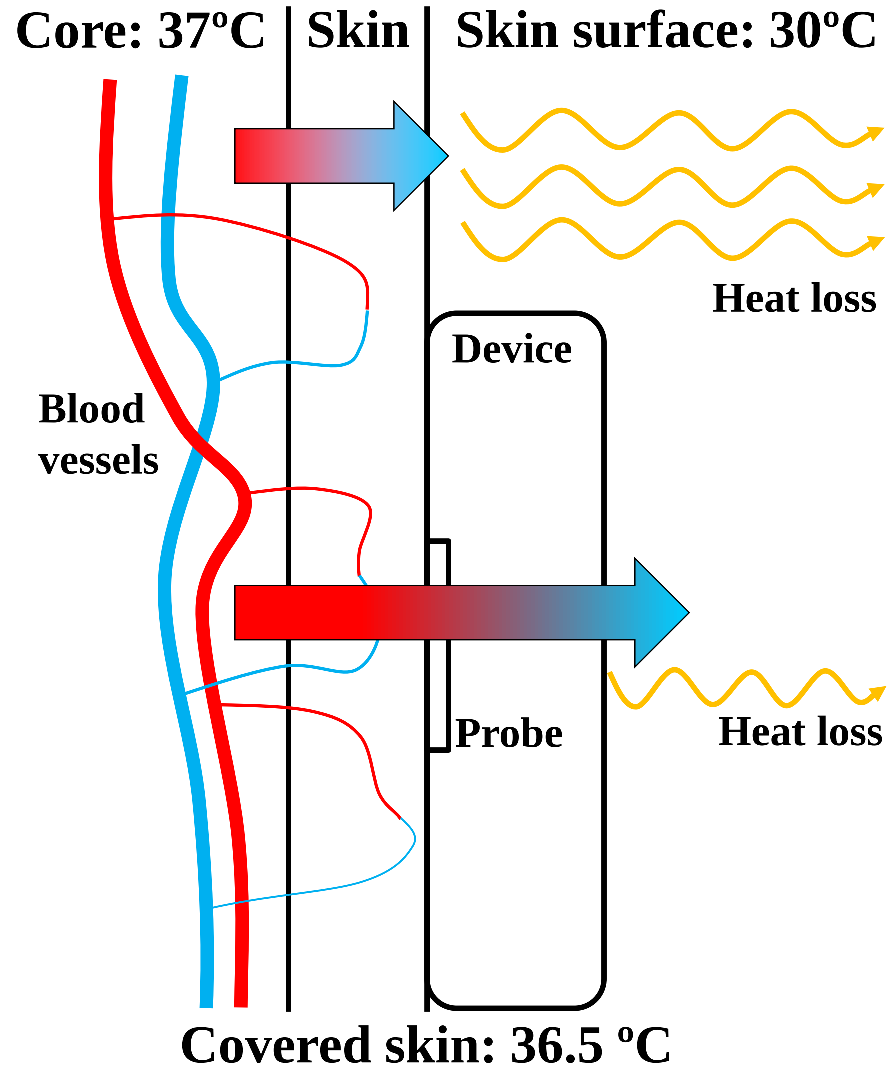
- **Predicting core temperature as early as possible**
- **Detect deviations in temperature trajectories**
- **Coping with deviations from main skin-heating trajectory**

snap40 monitoring device The device is secured to the upper arm with an elastic arm band; allowing continous monitoring of a patient.

```python
def newtons_cooling(t, temp_core, temp_0,k):
    return (temp_core - (temp_core-temp_0)/(np.exp(k*t)))
```



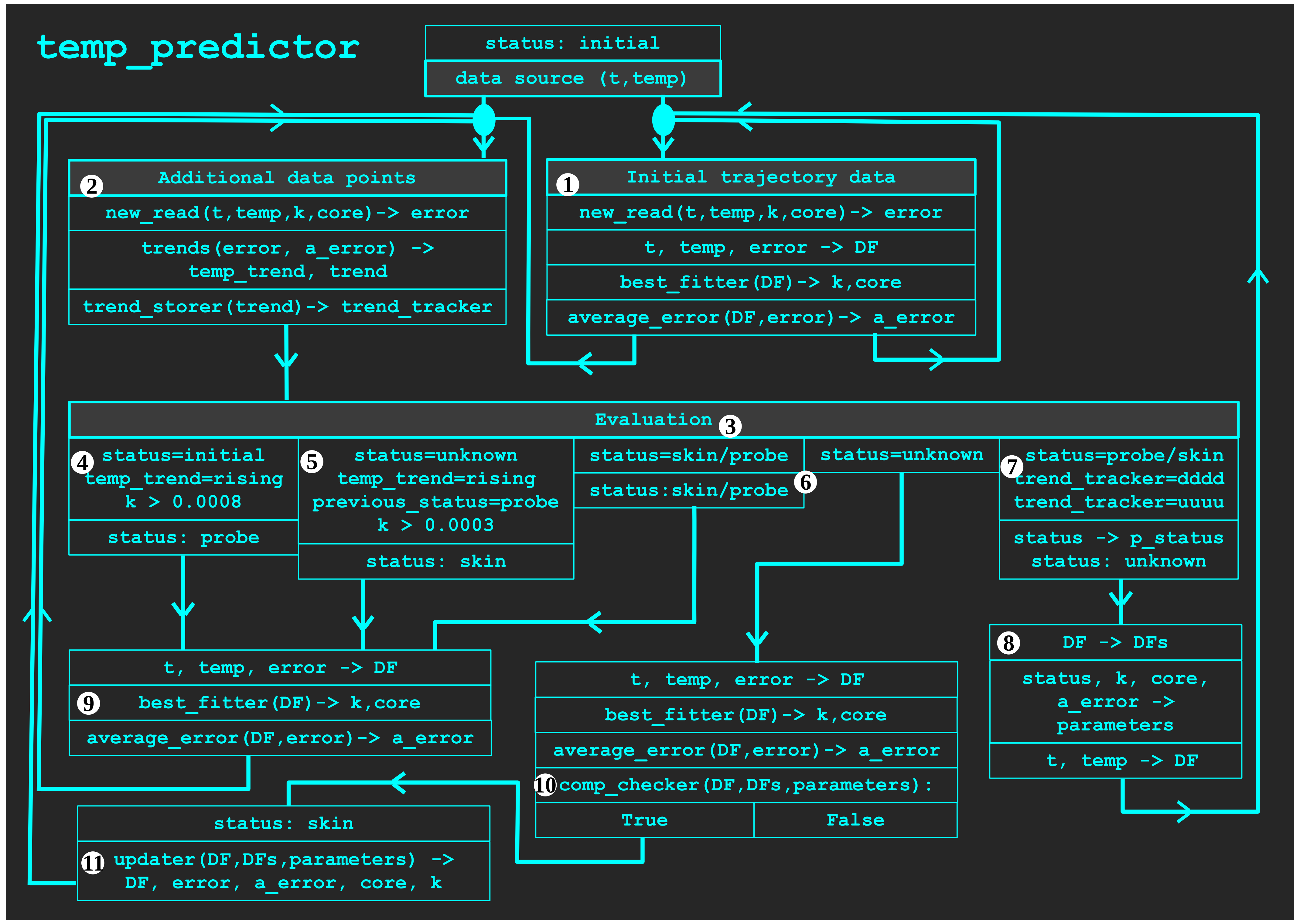
Temperature trajectory with Newtons Law of Cooling fitted. Newtons Law of Cooling (NLC) written as python code (upper). Device temperature reads plotted against time (green) together with a curve (red) derived from NLC fitted to the data (lower). Temperature increases as the device is deployed onto the patients arm until an equilibrium is reached, close to patient core temperature. R squared (R^2) for the fit is shown at the top.



An example model of the main factors influencing covered-skin temperature. The snap40 device insulates the skin and prevents heat loss. This shifts the temperature gradient towards the surface of the device, resulting in a covered skin temperature close to the core temperature. Another important factor influencing the covered skin temperature is the heat convection generated by blood flowing to the skin. In both cases variations may occur independently of the core temperature. For instance, clothing or room temperature can affect the heat loss while physical workout or fevers can affect heat convection. When a device is applied to a patient, the gold-plated device probe quickly heats to the temperature of the skin and then distributes the heat to probe-proximal device parts. These events cause rapid increasingly temperature trajectories which must be dealt with by temp_predictor

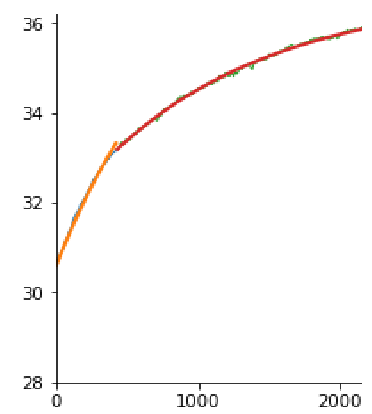
temp_predictor: overview

Simplified diagram outlining the major functional parts of the temp_predictor algorithm. temp_predictor compares each incoming data-point to previous data, using NLC to fit data. The status variable define the current type of trajectory; allowing only skin-heating (status: skin) trajectories to report core temperature. ①Temperature (temp) and time (t) data is stored in a data frame (DF) and the core temperature (core) and k value of NLC is estimated by the best_fitter function. Error is the difference between temp and model temp, while a_error is the average error for the current trajectory. ②Once DF contains 6 data-points, deviations from the model-trajectory are being calculated by the trends function while the temp_tracker variable keeps track of recent devitions. Calculated parameters are evaluated③, leading to changes in status ④⑤, a continuation of the current trajectory ⑥ or the beginning of a new unknown trajectory ⑦. The new unknown trajectory is started when trend_tracker indicates 4 downwards (dddd) or 4 upwards deviations. This prompts storage of DF and relevant parameters⑧ while starting a new DF which continues at ① .

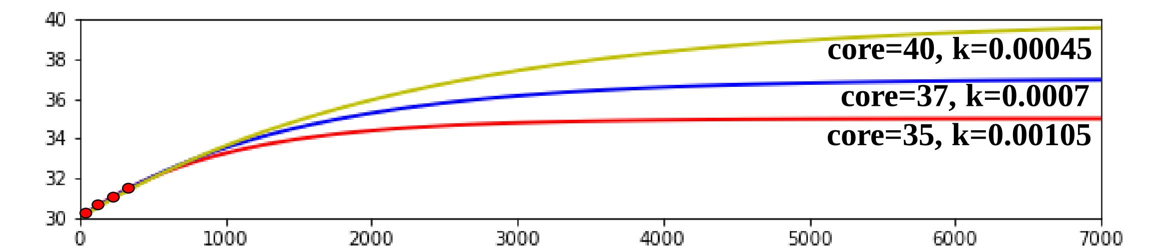


Probe-heating related trajectories

A probe-heating trajectory (yellow) followed by a skin-heating trajectory (red). The probe status is given to trajectories which have the status: initial, a rising temperature and a k value greater then 0.0008 (④). Data is then added to DF while k and core are updated.

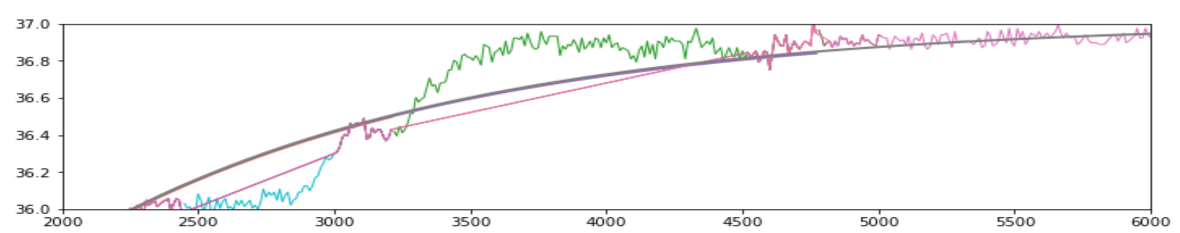


Estimation and reporting of core temperature



Example trajectories with different core and k values.⑨The three curves all seem appropriate for the initial data points (red dots), but their parameters values differ. This shows the limits of early core temperature estimation. temp_predictor only starts estimating core for skin_heating when core=37 and estimation of k stops to be efficient for trajectory modelling. The switching point is determined by a function (not shown) which sums error for the most recent 15 points and compares it to the average error.

Coping with trajectory deviations



Plots of temp vs t for input data and model trajectories. The pink line represents data estimated to be on the main skin-heating trajectory by temp_predictor while remaining data (cyan, green) are classified as unknown deviations. Model curves for the skin-heating trajectories (purple, grey) are also shown. The comp_checker function checks compatibility between unknown trajectory parts and previous skin-heating trajectories (⑩) while updater merges DF for the unknown and skin-heatingtrajectory⑪.