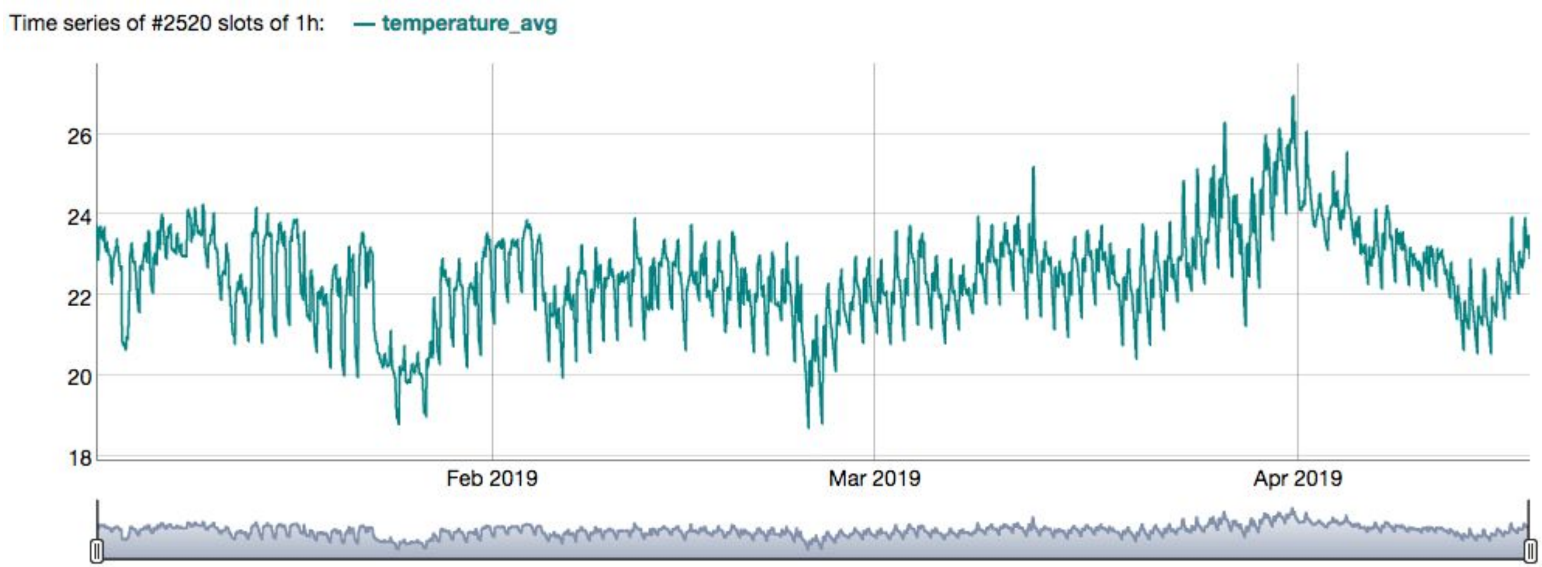
Anomaly detection in time series

Stefano Alberto Russo

Time series data

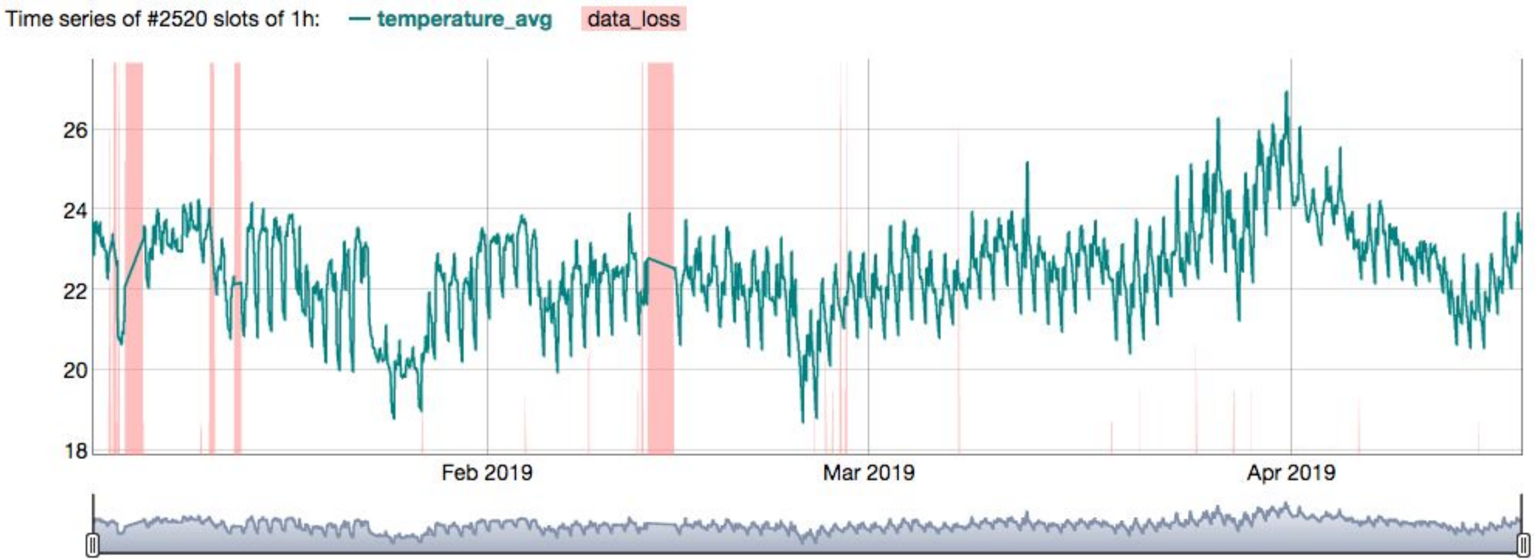
A time series is a series of data points in the time dimension.



Time series data

Time series data points are not necessarily equally spaced, but nearly any data manipulation requires them to be, in order to work in a discrete logic (*t, t+1, t+2, …*)

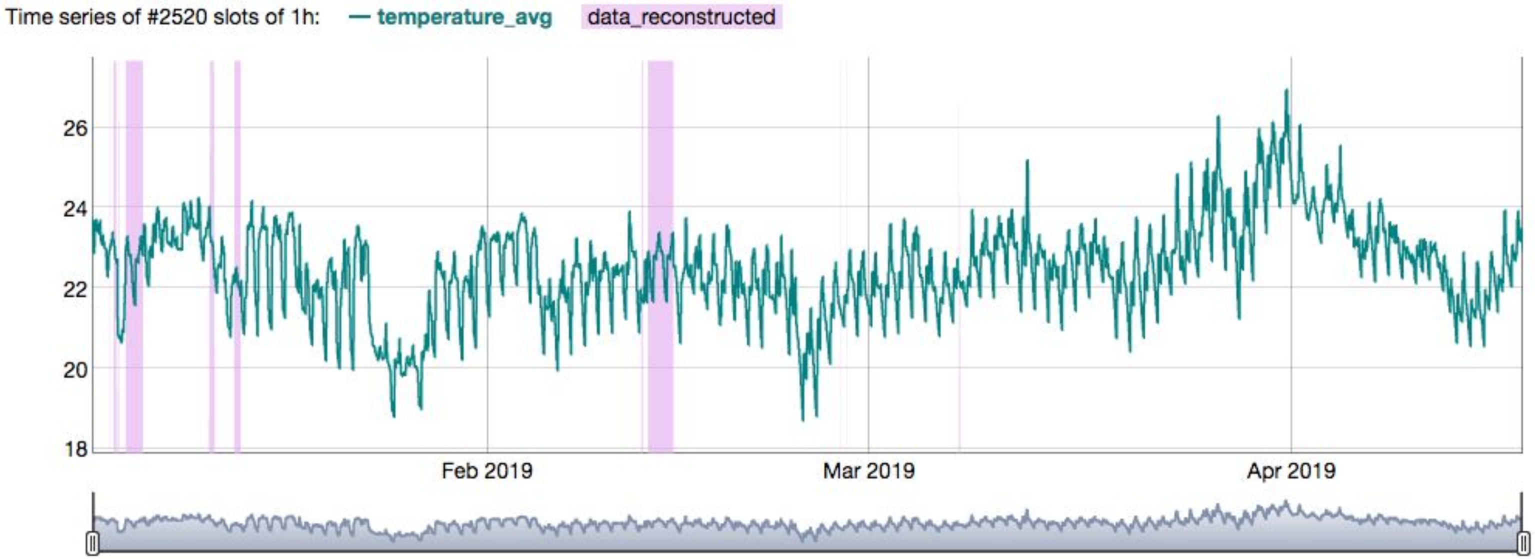
→ *some of the most common tasks are resampling and, interpolation and reconstruction.*



Time series data

Time series data points are not necessarily equally spaced, but nearly any data manipulation requires them to be, in order to work in a discrete logic (*t, t+1, t+2, …*)

→ *some of the most common tasks are resampling and, interpolation and reconstruction.*



What are “anomalies”?

By definition, an ***anomaly*** is something that belongs to the normal behaviour.

What is the **“normal”** behaviour, then?

→ Defined by a human

Thresholds, data tagging, definition of equations…

→ Automatically extrapolated

Based on statistical indicators and probability considerations, usually involving a *normal* distribution.

What are “anomalies”?

By definition, an ***anomaly*** is something that belongs to the normal behaviour.

What is the **“normal”** behaviour, then?

→ Defined by a human

Supervised

Thresholds, data tagging, definition of equations…

→ Automatically extrapolated

Unsupervised

Based on statistical indicators and probability considerations, usually involving a *normal* distribution.

Human-defined anomalies (aka supervised)

Example 1 (threshold-based): if we know that a CPU laptop has to stay below 70°, a temperature of 73° is anomalous

Example 2 (relation-based): if we know that after a rainfall we have an increase of water levels in sewage networks, and in a given pipe the level does not raise then this is an anomaly (i.e. a leak).

Example 3 (learning-based): if we know that some conditions are anomalous but we cannot describe a threshold or a relation, we can tag them as such in our data hoping that a machine learning model will find out the rule for us.

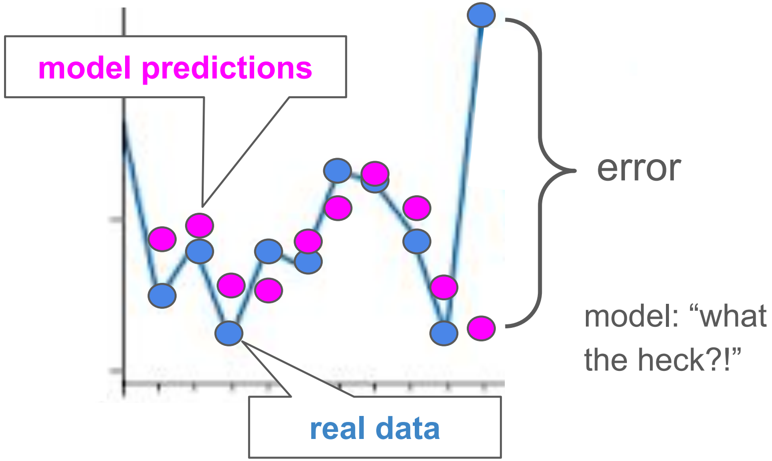
Automatically-defined anomalies (aka unsupervised)

We just capture the pattern of our data in a model.

→ *it does not matter how complex, even an average would do!*

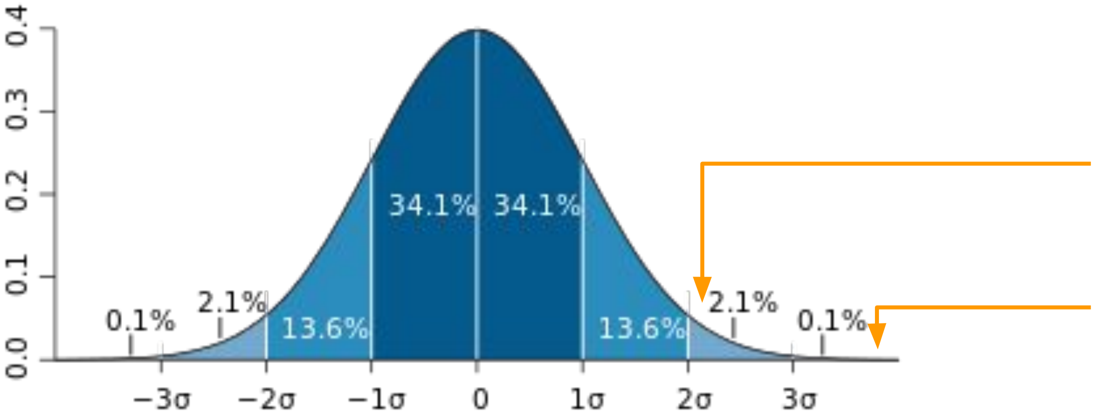
We then compare the real data with the model predictions

Now, if the difference between the prediction and the real data is high, this means that the model failed to capture that behaviour, and thus we have an anomaly.



Automatically-defined anomalies (aka unsupervised)

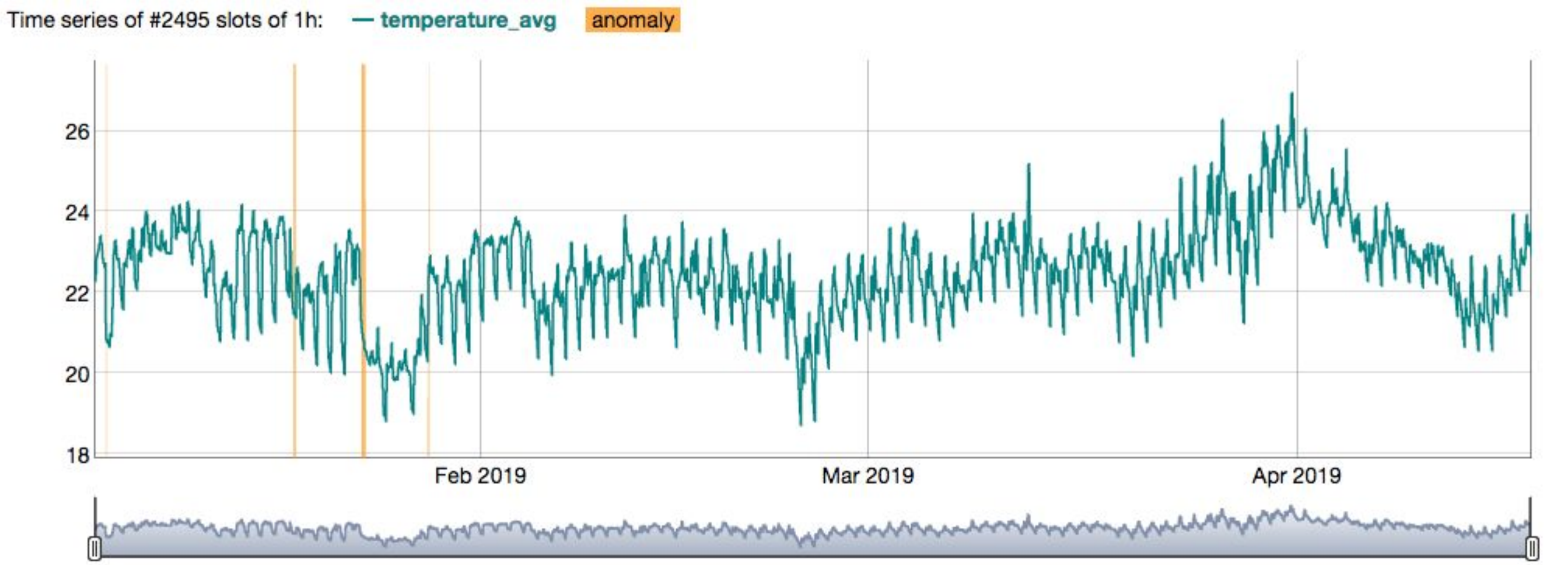
More in detail, we actually first compute the model accuracy, then we set a threshold on the error distribution in terms of standard deviations and only then we can say that above a certain error we have an anomaly



This error *might* be an anomaly

This error is **definitely** an anomaly

Example of fully unsupervised anomaly detection



The model here just computes the periodic averages (i.e. the average temperature value for each hour of the day) and applies an offset to align them with the data.

Open themes

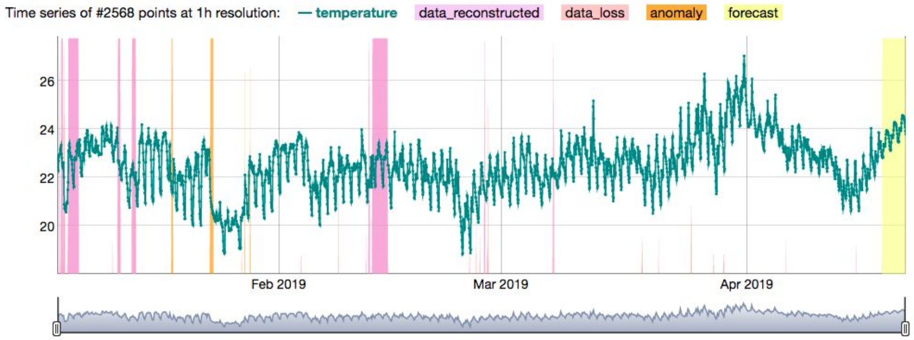
- How do you evaluate different models?
  - How to minimize false positives?
  - How to set anomaly thresholds?
  - Is there something better than brute force, i.e. feature-based model selection?
- Can you take into account domain-specific priors?
  - Bayesian approaches?
  - Physic-based modeling?
- How can you take into account network-effects?
  - Multivariate forecasting models?
  - Which features should be considered?
  - How to handle time-based cross-correlations?

p.s. *Timeseria*: a time series data processing library

Timeseria aims at making it easy to manipulate time series data and to build models on top of it. It comes with a built-in set of common operations (resampling, slotting, differencing etc.) and models (reconstruction, forecasting and anomaly detection).

```
pip install timeseria
```

```
from timeseria.models import AnomalyDetector
anomaly_detector = AnomalyDetector()
anomaly_detector.fit(time_series)
anomaly_detector.apply(time_series)
time_series.plot()
```



I am building it as part of my PhD: https://github.com/sarusso/Timeseria

Thanks!

Stefano Alberto Russo