CLOUD-BASED WEB SERVICE FOR OUTLIER ANALYSIS IN ENVIRONMENTAL TIME SERIES SERVED VIA SOS

Doron Goldfarb, Johannes Kobler, Johannes Peterseil

This work was done in the context of the EOSC-HUB (grant agreement no 777536) project which has received funding from the European Union's Horizon2020 research and innovation programme.
BACKGROUND

- Research Infrastructures (RI) seek to provide standardised facilities, resources, processes and services for specific research domains, such as Long Term Ecological Research (LTER)

- Ongoing digitalisation puts focus on data aspects across the whole data life cycle

- RI increasingly use cloud data repositories for sharing and publishing data
  SOS is a natural choice for provision of geo-referenced time-series

- Data quality assessment is an important aspect of high relevance across and beyond RI
ELTER RI

- European Long-Term Ecosystem, critical zone and socio-ecological systems ESFRI-RI
- Mission: Research into ecosystem structures and functions
  site-based, multi-scale and cross-disciplinary
- Systematic coverage of major European terrestrial, freshwater and transitional water environments
  ~250 selected research sites
- eELTER Service Portal – access to data and sites

  DEIMS-SDR (Site and dataset registry)
  DIP (Data Integration Portal)
  CDN (Central Data Node – SOS based)

  Current development: Virtual Data Labs
OUTLIER DETECTION IN TIME SERIES

● Data quality assessment is prerequisite for aggregation/re-use of data from different sources
  Outlier detection is one important aspect

● Separate technical reasons (Resulting in missing or incorrect measurement data)
  ● Covered sensor, Overflowing container, Bad calibration, Flawed sensor, Data corruption during transmission
    from „unusual“ observations (Correct measurements with values outside the norm)
  ● Wind speed during storm, High temperature during heatwave, Water level during heavy rain

● Many different methods to detect outliers, desire to standardise approaches within the same RI
OUTLIER DETECTION SERVICE FOR ELTER RI

● Support data aggregation for standardised eELTER RI Data Products
  ● Offer general quality assessment service for data providers

● Standardised „non blackbox“ array of outlier detection methods
  ● Centralised maintenance and lower usage threshold

● Offer R-Script based workflow encapsulated behind REST-API
  ● Operates on data offered via SOS or file based cloud repositories
  ● R-script environment can also be provided for local or offline use

● Support data provenance, traceability and trust for eELTER RI workflows (and those of others)
REST-API BASED ON OPENAPI SPECIFICATION

- Standardised description of REST API functions and parameters
- Automatic generation of server stubs from specification
- Connexion framework for Python/Flask based Web Servers
SYNCHRONOUS (BLOCKING) REQUESTS

- Processing of time-series with many values can take a long time
ASYNCHRONOUS (NONBLOCKING) REQUESTS

- Initiate processing, do not wait but occasionally poll status and pick up results once available
OUTLIER DETECTION

● Selection of methods made available via existing R-packages

● Detection usually based on moving window with caller-specified width and interval

● Currently 9 different methods
  ● Outlier identification by classifying the forward and backward absolute change
  ● Rosner’s Test for Outliers (R-package::function = EnvStats::rosnerTest)
  ● Univariate outlier detection with bounds based on robust location and scale estimates (R-package::function = univOutl::LocScaleB)
  ● An implementation of the LOF algorithm (R-Package::function = DMwR::lofactor)
  ● Outlier detection using Robust Kernel-based Outlier Factor (RKOF) algorithm (R-Package::function = OutlierDetection::dens)
  ● Outlier detection using Mahalanobis Distance (R-Package::function = OutlierDetection::maha)
  ● Outlier detection using k Nearest Neighbours Distance method (R-Package::function = OutlierDetection::nn)
  ● Outlier detection using kth Nearest Neighbour Distance method (R-Package::function = OutlierDetection::nnk)
OUTLIER DETECTION

- Results made available in different files, packaged in a ZIP file
  - Annotated original time series (Each value has flags set if determined as outlier by a specific detection method)
  - More detailed analysis of outliers wrt moving window etc in separate files
LESSONS LEARNED SO FAR

- Identify individual time series in SOS?
  Parameter + FoI + Begin + End?
  Parameter + Procedure + FoI + Begin + End?
  Offering + Begin + End?
- Configuration dependent, no universally valid approach ➔ Potentially problematic for generic application
- Consider specification or reuse of dedicated profile ➔ within RI, or even beyond

- Performance
  - Linear increase in retrieval/processing time
  - Outlier analysis consumes most of the time
  - Improve code efficiency, parallel execution
  - Consider caching of results

---

Cloud-based web service for outlier analysis in environmental time series served via SOS

![Graph of number of observations (x) vs. execution time in seconds (y)](chart)

- SOS retrieval
- Outlier analysis
OUTLOOK

- Service will play important role in eLTERRI
- Implement provenance trace
- Evaluate additional, especially multivariate, outlier detection methods
- Data caching strategies, Performance improvements
THANK YOU

- Contact: doron.goldfarb@umweltbundesamt.at
- Source Code: https://github.com/d0rg0ld/OutlierDetection4EOSC