T4C: A Framework For Time-Series Clustering-As-A-Service

CPSWS 2022

Alessandro Falcetta
Manuel Roveri

Politecnico di Milano

September 2022

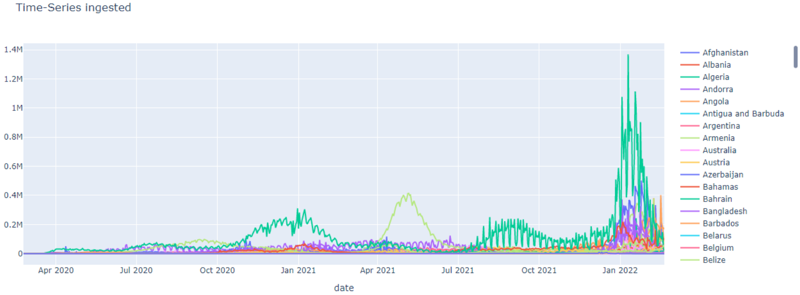
Cloud and Services

- Recent years showed more and more integration between Machine (ML) and Deep Learning (DL) and Cloud computing;
- This led to the *as-a-service* paradigm, where services based on ML and DL are offered directly to end-users;
- Notable examples include image recognition[1], text-to-speech, speech-to-text[2].

[1] Myeongsuk Pak and Sanghoon Kim. "A review of deep learning in image recognition". In: *2017 4th international conference on computer applications and information processing technology (CAIPT)*. IEEE. 2017, pp. 1–3.

[2] M Shamim Hossain and Ghulam Muhammad. "Cloud-assisted speech and face recognition framework for health monitoring". In: *Mobile Networks and Applications* 20.3 (2015), pp. 391–399.

Time-series clustering



How can we offer time-series clustering models, following the as-a-service paradigm, directly to end users?

Related literature

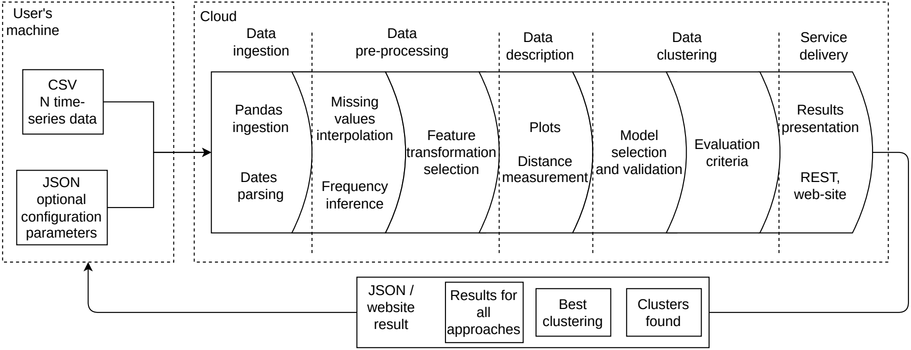
- There are various software libraries which offer helper functions to use time-series clustering models in a simple way, like *tslearn*[3];
- Nonetheless, they are not meant to be used in a as-a-service manner;
- AWS Forecast[4] and TIMEX[5] (on which this study is based) are two solutions for time-series forecasting which can be used in a as-a-service manner.

[3] Romain Tavenard et al. "Tslearn, A Machine Learning Toolkit for Time Series Data". In: *Journal of Machine Learning Research* 21.118 (2020), pp. 1–6. URL: `http://jmlr.org/papers/v21/20-091.html`.

[4] Amazon Web Services Inc. *Amazon Forecast*. 2021. URL: `https://aws.amazon.com/forecast/`.

[5] Alessandro Falcetta and Manuel Roveri. "TIMEX: an Automatic Framework for Time-Series Forecasting-as-a-Service". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. The Sixth International Workshop on Automation in Machine Learning.* 2022.

T4C

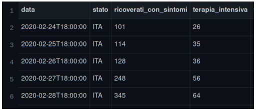


Software implementation released as an open-source repository[6] and on PyPI.

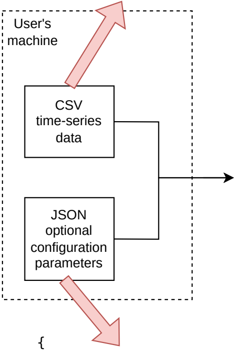
[6] https://github.com/uGR17/T4C_CLUSTERING

T4C

User's machine



data	stato	ricoverati_con_sintomi	terapia_intensiva
2020-02-24T18:00:00	ITA	101	26
2020-02-25T18:00:00	ITA	114	35
2020-02-26T18:00:00	ITA	128	36
2020-02-27T18:00:00	ITA	248	56
2020-02-28T18:00:00	ITA	345	64

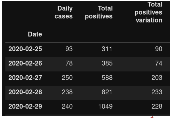


```
{
  url: "https://..../file.csv",
  distance: "dtw",
  ...
}
```

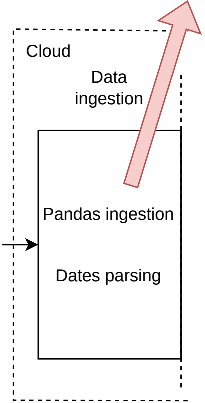
- The time-series clustering pipeline starts on the user's machine;
- It is assumed that the user has a dataset of time-series to cluster, and an optional JSON file containing configuration parameters to tune T4C, if desired.

T4C

Cloud - Data ingestion



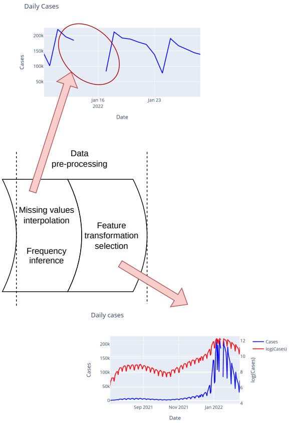
Date	Daily cases	Total positives	Total positives variation
2020-02-25	93	311	90
2020-02-26	78	385	74
2020-02-27	250	588	203
2020-02-28	238	821	233
2020-02-29	240	1049	228



- The CSV is loaded and transformed in a `Pandas DataFrame`;
- The first column is used as time-index, with its values automatically parsed in `datetime` objects.

T4C

Cloud - Data pre-processing



Data pre-processing

Missing values interpolation

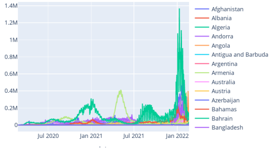
Frequency inference

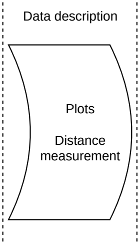
Feature transformation selection

- The frequency of the time-series is estimated, if not specified by the user;
- Any missing value is obtained with interpolation.
- To ease the *clustering* step, various feature transformations are applied (e.g., logarithmic one).

T4C

Cloud - Data description





Data description

Plots

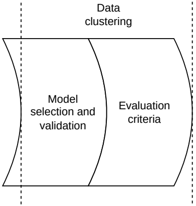
Distance measurement

`[ euclidean, dtw... ]`

- In this step various plots are automatically generated with the `Plotly` library;
- Distance metrics are computed for the time-series in the datasets.

T4C

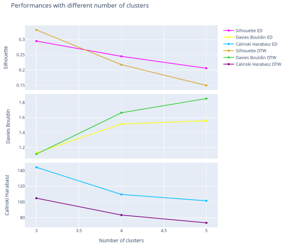
Cloud - Data prediction



Data clustering

Model selection and validation

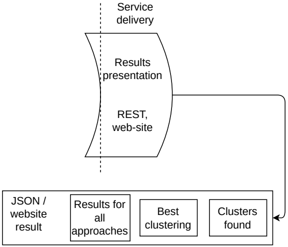
Evaluation criteria



- The available models are all tested on the dataset, such as K-Means and Gaussian Mixture;
- The models are tested with different feature transformations and with different amounts of clusters.

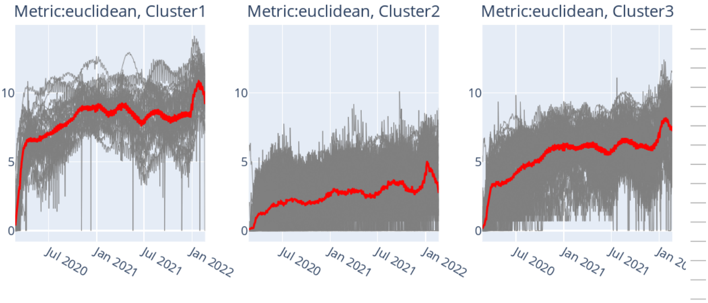
T4C

Cloud - Service delivery

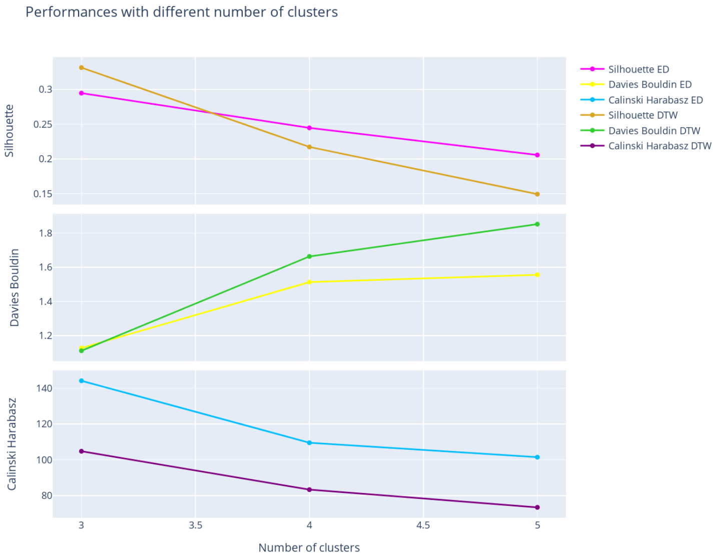


- The final result, which includes the plots and the clustering, can be returned to the user through a REST endpoint (as JSON/ZIP format), or through a website.
- The latter solution is implemented using the `Dash` library.

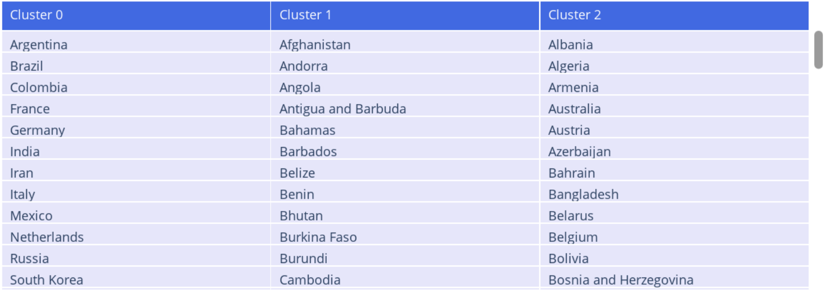
Use case: `https://clustering.covid-timex.it`



Use case: `https://clustering.covid-timex.it`



Use case: `https://clustering.covid-timex.it`



Cluster 0	Cluster 1	Cluster 2
Argentina	Afghanistan	Albania
Brazil	Andorra	Algeria
Colombia	Angola	Armenia
France	Antigua and Barbuda	Australia
Germany	Bahamas	Austria
India	Barbados	Azerbaijan
Iran	Belize	Bahrain
Italy	Benin	Bangladesh
Mexico	Bhutan	Belarus
Netherlands	Burkina Faso	Belgium
Russia	Burundi	Bolivia
South Korea	Cambodia	Bosnia and Herzegovina