Load curve model – Hands on

# Installation

* Install Anaconda you can find here : <https://www.anaconda.com/products/individual> or in the *Software center* of the IEA.
* You need to install the following packages to run the model.
* Numpy
* Matplotlib
* Pandas
* Pytorch lighting
* Pytorch ( see here what version to install – depends on the computer <https://pytorch.org/get-started/locally/> )
* Omegaconf
* Holidays
* Yaml
* Scipy
* Seaborn
* Sklearn
* Cvxpy ( always struggling with this one, if it is the case don’t install – as it is only use in “res\_non\_res\_separation.py” which is a side function of the prediction not useful)
* **The local package which contains the source code** (run this command : pip install –e .)

In order to install these packages open *spyder* or *jupyterlab* in *anaconda* and run this command : pip install *package\_name*

* Open *JupyterLab* in *anaconda.* It is the interface used to run the load curve model without getting into the source code. Now the useful files to open with *JupyterLab* are : Model manager.pynb & models/Results analysis.ipynb
* Model manager.ipynb is the interface to make dataset ready for the neural network, run the algorithm on the *train set* and make predictions on the *test set*. You can define the algorithm parameters from this notebook and you have a quick explanation on how the model works actually
* Results analysis is the notebook which contains the analysis of the experiment you want. When running this notebook, all graphs and metrics are saved in folders specified.
* **Everything is explained in model manager.ipynb and results analysis.ipynb**

# Organisation of the folder *Load\_curve\_modeling*

* models : Where you can find every experiments you have performed here models/logs & results analysis.ipynb
* data : All data files and sources
* docs : all documents and presentations done
* picture : useful pictures
* BoxWEO2021 : All files to make the graphs for the load curve WEO 2021 box
* src : Where you can find the source code of the model.
* data : all codes to get download and process data.
* features : codes to build features useful to train the model. Useful to build new ones.
* models : the model itself (the trainer, the predicter and the architecture of the neural network)
* visualisation : not use for now.

# Data folder organisation.

**External** : “weo\_results”contains hourly load data by subsector from WEO model runs. It can be used to make comparisons, validity checks, etc.

**Mapping :**

**country\_mapping.csv** contains all the data regarding countries to process, and what data to use. Rows that are not complete (e.g. Iceland, as we are missing hourly weather data) will be ignored

- “country” is how the country will be referred. It is also the country identifier used to load data from the etp\_inputs file.For non-total loads (e.g. enedis sector loads in France), a row can be added with a different “country” name, but adjustments to the load will have to be made during data processing

- “ISO\_2\_digits” used to load the country weather data

- “edc\_country\_name” the country's identifier in EDC data

- “timezone” can be found here https://en.wikipedia.org/wiki/List\_of\_tz\_database\_time\_zones

- “load\_data\_name” name of the load data column in the processed hourly load dataset.

- “pop\_percentage” is the population percentage of the country into the continent (used for WEM load curves disagregation)

- economy : label of the economic state can be either advanced\_economies or developing economies

**sector\_mapping.csv** maps EDC subsectors (e.g Iron and steel) to our subsectors names (e.g IND\_IS). Only subsectors listed in this file are used (which is why there is no RES/SER data, as we are using ETP data for that).

**annual\_demand :** annual demand data for historical and future years. For historical years, currently using EDC data for industry/transport (to have a country-level spatial resolution), and ETP data for buildings (to have more details in the subsectors). ETP scenario data is required for future years for all sectors, so some spatial disaggregation may be necessary in Excel here.

**explanatory\_variables :** hourly inputs to the model (temperature, solar irradiance, etc). Weather files for every country can be downloaded at [https://www.renewables.ninja/country\_downloads/<CC>/ninja\_weather\_country\_<CC>\_merra-2\_population\_weighted.csv](https://www.renewables.ninja/country_downloads/%3cCC%3e/ninja_weather_country_%3cCC%3e_merra-2_population_weighted.csv) where `<CC>` is the 2 digits country code, e.g. `FR` (or the region code e.g. `US.CA` for California). The `src/data/download\_data.py` script automates this. Spatial aggregation is population weighted, and underlying data comes from the MERRA-2 reanalysis dataset

**hourly\_demand :**

- “Singapore” : Excel files with hourly demand for a whole week can be found at “https://www.ema.gov.sg/statistic.aspx?sta\_sid=20140826Y84sgBebjwKV”.This dataset <https://data.gov.sg/dataset/half-hourly-system-demand> has already done the aggregation (using the same data source) from 2012 up to June 2016. For more recent data, we aggregate the excel files with week-long hourly data in `process\_hourly\_load.py`

- “time\_series\_60min\_singleindex.csv” : hourly demand data (along with other variables that we don't use) for some European countries, based on ENTSO-E data. https://data.open-power-system-data.org/time\_series/

**scenario\_anomalies :**  csv files from Chiara, with monthly temperature anomalies for countries/ETP regions, for a given future year and RCP scenario