

# Prediction of time series

Practice and Discover

# 1 - Prediction / forecasting

- Many possibilities to format data with `timeseries_dataset_from_array`
  - Univariate or multivariate time series
  - Study the different code blocks in file `lab-Example-TimeseriesGenerator.ipynb` and replace `TimeseriesGenerator` with `times_dataset_from_array` (cf. Keras website)
- Prediction / forecasting of electrical power consumption
  - Try different number of data samples from dataset starting at some day
    - 5000 data samples
    - 10000 data samples
  - Use `household_power_consumption.txt` to be downloaded from filesender ([link](#))
  - Try to complete the file `Empty_Students-Elect-LSTM.ipynb`
  - Write a version that uses `timeseries_dataset_from_array` as

`lab-Elec-timeseries_dataset_from_array-LSTM.ipynb`

## 2 - Prediction / forecasting

- Data provided by [Timeseries-data.csv](#) file
- Use `times_dataset_from_array` (cf. Keras) to generate training or testing data
- You have to design neural networks able to forecast the values of  $T$ 
  - Use data from 01/01/2009 up to 31/07/2009
  - Use 80% as training set and the remaining for testing
  - Plot at the end the data and the predictions for both training and testing sets
- Two prediction models are expected
  - Prediction using a single time series (univariate forecasting)
    - Some past  $T$  values are used to predict the next  $T$  value
  - Prediction based on several time series (multivariate forecasting)
    - Some past  $P$  and  $Rho$  values are used to predict the next  $T$  value
- Important notes:
  - Put the two versions one after the other in your Colab file
  - Do not use too large epoch numbers

### 3 - Prediction / forecasting

- The goal is to reproduce the results obtained in paper  
[Deep Learning and Gradient Boosting for Urban Environmental Noise Monitoring in Smart Cities](#)
- Data
  - Provided by [DataParkmeter.zip](#) file (3 CSV file; one for each Parkmeter of the paper)
  - Preprocess data as explained, use MinMaxScaler from Scikit-learn to normalize
- Use `times_dataset_from_array` to generate training or testing data
- Evaluate the prediction performance of the different deep learning models described in the paper to predict future noise levels (short / long term predict.)
- Once suitable predictions will be obtained try to
  - Use them in order to detect false data injection attacks as done in the paper
- You can also try to deal with pollution data particles...