



Time-series Prediction

In Energy-harvesting Wireless Sensor Nodes, it is often the case that we would like to know how much energy will be available for harvesting in the future. For example, if we have a solar panel and know a cloud is coming, we might want to conserve energy until the sun appears again. We assume that receiving a weather forecast is not practical: in our setting the node is in a remote area and communication is costly, but often the forecast is not specific enough to be useful at the node's environment.

In this Hackaton Idea, we propose that you help us develop an algorithm for on-node prediction of the environment.

What we have:

- Data on the environment
- Two prediction algorithms: 'dumb' and 'diff'
- A way to compare and evaluate prediction algorithms

What is your task:

- Make your own prediction algorithm - hopefully better than ours

Check out "[ideas/timeseries_prediction.py](#)", and read these notes that will help you start:

- These data are of interest for prediction:
 - Temperatures at all depths, found at "[MHxx_temperature_rock_2017](#)"
 - Net solar radiation, found at "[MH15_radiometer__conv_2017](#)"
 - You can also work with any other data like wind speed average, but this is not of primary importance for energy harvesting.
- The way data is parsed from a file is given in the python file.
- Two simple prediction algorithms are given: *dumb* and *diff*, and you can check them out in the python file. Your job is to come up with new ones, and you can use whatever comes to your mind.
 - Hint about a common mistake: if predicting a value at time t , use only values known and available before time t .
- To evaluate your prediction, use the root-mean-square-error, as shown in the python file. Other metrics exist, but you don't need to worry about them.
- If you use training data to create a model or optimize some parameters, keep in mind the following:
 - Separate training and test data. Only the test data is used to evaluate your prediction scheme.
 - All data is time-stamped, and training data has to be from before the test data.
- Last but not least, there is a handy visualization tool you can use to display your results.

We are eager to see your solutions! If you liked predicting time-series data, consider doing it as a part of your semester or master thesis with us :-)