NoICE - Relation between icing conditions and train delays

May 14, 2019

Petri Välisuo and Thileepan Paulraj

Vaasan yliopisto
UNIVERSITY OF VAASA

region västerbotten

Regional Council of Ostrobothnia

EUROPEISKA UNIONEN

Interreg
Botnia-Atlantica
Europeiska regionala utvecklingsfonden

1 The data and the libraries

The analaysis in this document is made using Python which many different libraries for data analysis (Pandas, NumPy, SciPy) and for visualization (Matplotlib, Seaborn) and scientific data storage (H5Py). All the source code used for the analysis is shown.

First we need to read needed libraries and data:

```
In [16]: import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         from matplotlib.pyplot import subplot, grid, xlabel, ylabel, figure, plot, title
         from numpy import linspace, hstack, arange
         from datetime import datetime
         from scipy.interpolate import interp1d
         from scipy.signal import savgol_filter

         # Time resolution for the statistics
         TRES='3H'

         # Time period to be covered
         t='2017-02'

         TT=pd.read_hdf('alldata.h5')
         TTall=pd.read_hdf('TTdata.h5')
```

2 Power loss of a wind turbine

One of potentially interesting weather parameter is the power loss of the wind turbine, since it may be caused by icing, which can be also one of the resons for train delays and other traffic problems.

To define the power loss of a turbine, the typical power curve is first estimated using statistics. Here it is done by selecting a lot of wind turbine data, smoothing it with Savitzky-Golay filter and defining an interpolator for smoothed data.

```
In [3]: # Calculate the power curve, f
        Ts=TTall.dropna().sort_values(by='U1')
        U=hstack((0, Ts.U1.values, 20))
        P=hstack((-70, Ts.P1.values, 3302))
        u=linspace(0,18);
        f=interp1d(U, savgol_filter(P, 191, 1))
```

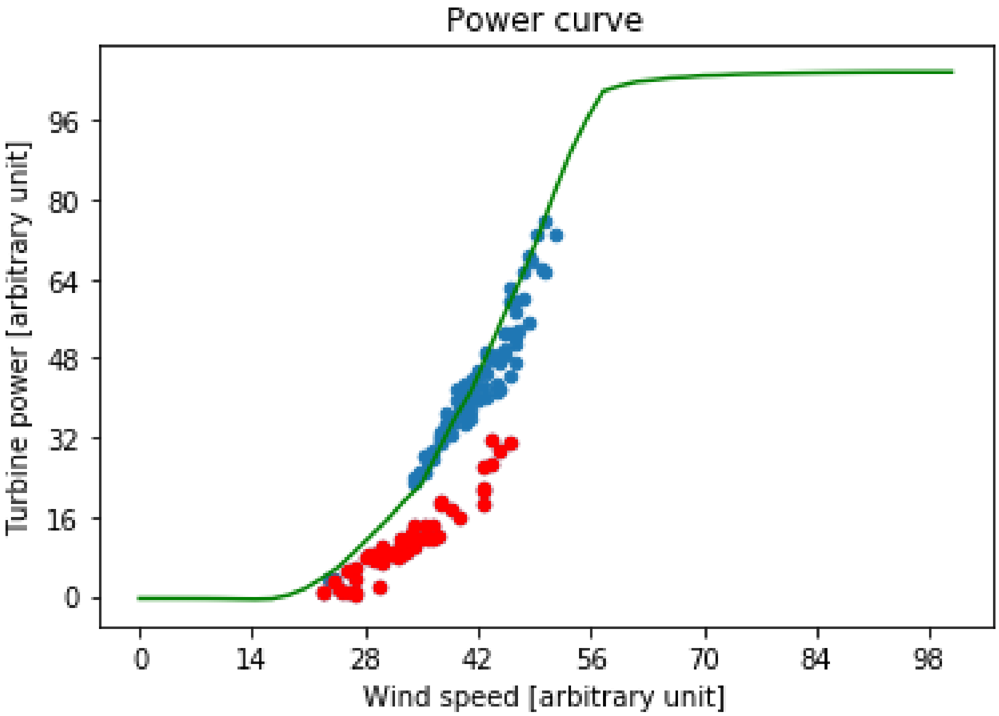
The power lost is now the observed wind turbine power minus the expected due to power curve. The following Figure shows the time incidents when power is lost (red dots), possibly due to icing.

```
In [44]: ax=subplot(1,1,1)
         TTall['2017-02-11'].plot.scatter('U1', 'P1', ax=ax)
         i=(TTall.P1/f(TTall.U1))<0.75
```

```
print len(i)
TTall[i]['2017-02-11'].plot.scatter('U1', 'P1', ax=ax, c='red')
ax.plot(u, f(u), 'green')
ax.set_yticklabels(arange(-16,100,100/6));
ax.set_xticklabels(arange(-14,100,100/7));
title('Power curve')
xlabel('Wind speed [arbitrary unit]')
ylabel('Turbine power [arbitrary unit]')
```

```
8496
```

```
Out[44]: Text(0,0.5,u'Turbine power [arbitrary unit]')
```



2.1 Correlation of variables

In addition to the power loss in wind turbine, we have plenty of weather parameters measured by the weather station of the Finnish Meteorological Institute (FMI) or by the wind turbines.

1. DL_IC Delay of intercity trains
2. DL_S Delay of Pendolino trains
3. Df Wind direction, measured by FMI weather station in Seinäjoki
4. Dt Wind direction in 140 m height in Torkkola, measured in Wind turbine nacelles
5. Pl Loss of production power, when compared with the standard power curve (kW)

6. Rh Relative humidity measured by FMI weather station in Seinäjoki
7. Tf Temperature measured by FMI weather station in Seinäjoki
8. Tt Temperature in 140 m height in Torkkola, measured in Wind turbine nacelles
9. Uf Wind speed measured by FMI weather station in Seinäjoki
10. Ut Wind speed in 140 m height in Torkkola, measured in Wind turbine nacelles

3 Correlation

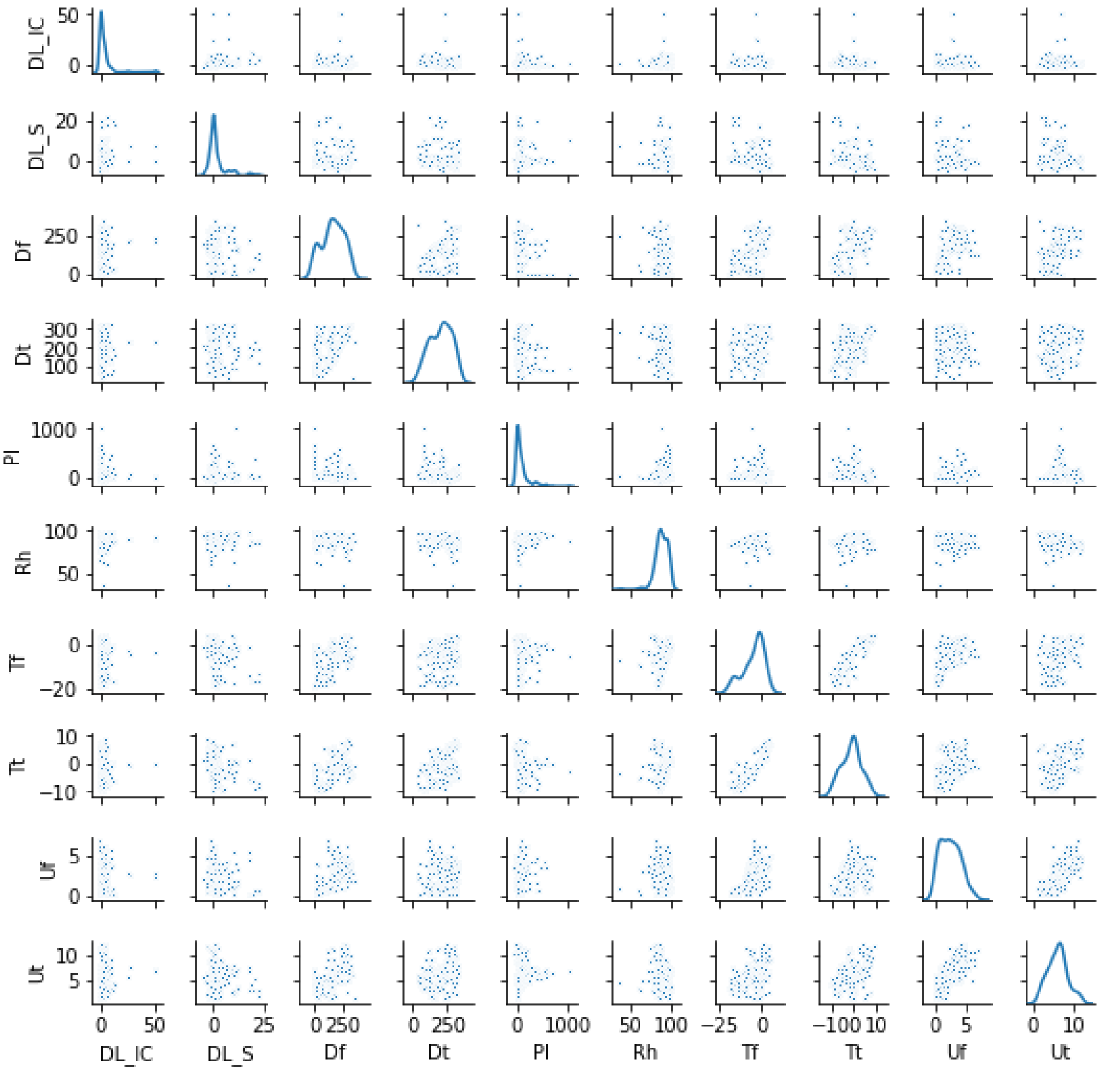
Correlation and dependency can be studied visually using so called pairplot. The diagola of the plot represent the kernel density esimate of the distribution of variables. The distributions are hardly gaussian. Visible dependencies are not very strong, except in obvious cases, between the weather variables masured in different locations.

For example temperature measure by FMI (Tf) is strongly correlated with the temperature measured in a wind park (Tt), and so are the wind speeds Uf and Ut.

It is not easy to say, based on visual examination that train delays would be dependent on any measured weather parameters.

```
In [10]: sns.pairplot(TT, diag_kind='kde', plot_kws={"s": 5}, size=0.8)
```

```
Out[10]: <seaborn.axisgrid.PairGrid at 0x7fc93509c890>
```



3.1 Numerical correlation

Numerical analysis reveal quite the same which was already observed in visual analaysis. There are correlations between the weather parameters measured in different locations but no strong correlation between the train delays and individual weather parameters.

```
In [6]: TT.corr()
```

```
Out[6]:            DL_IC      DL_S        Df        Dt        Pl        Rh        Tf  \
        DL_IC   1.000000  0.186331  0.095084  0.024291 -0.104932  0.004202 -0.019514
        DL_S    0.186331  1.000000 -0.113332 -0.082395  0.044704 -0.038699 -0.249587
        Df      0.095084 -0.113332  1.000000  0.641652 -0.189949 -0.042115  0.447378
        Dt      0.024291 -0.082395  0.641652  1.000000 -0.263441 -0.180018  0.109137
        Pl     -0.104932  0.044704 -0.189949 -0.263441  1.000000  0.186885  0.062083
```

```
Rh      0.004202 -0.038699 -0.042115 -0.180018  0.186885  1.000000  0.265532
Tf     -0.019514 -0.249587  0.447378  0.109137  0.062083  0.265532  1.000000
Tt     -0.037353 -0.246235  0.449079  0.295061 -0.069601  0.217375  0.861544
Uf      0.036686 -0.184741  0.518120  0.049715  0.259860  0.040879  0.554113
Ut      0.025821 -0.182203  0.482556  0.217543  0.167502 -0.088296  0.342262

             Tt        Uf        Ut
DL_IC -0.037353  0.036686  0.025821
DL_S  -0.246235 -0.184741 -0.182203
Df     0.449079  0.518120  0.482556
Dt     0.295061  0.049715  0.217543
Pl    -0.069601  0.259860  0.167502
Rh     0.217375  0.040879 -0.088296
Tf     0.861544  0.554113  0.342262
Tt     1.000000  0.399859  0.373163
Uf     0.399859  1.000000  0.696053
Ut     0.373163  0.696053  1.000000
```

4 Time domain analysis

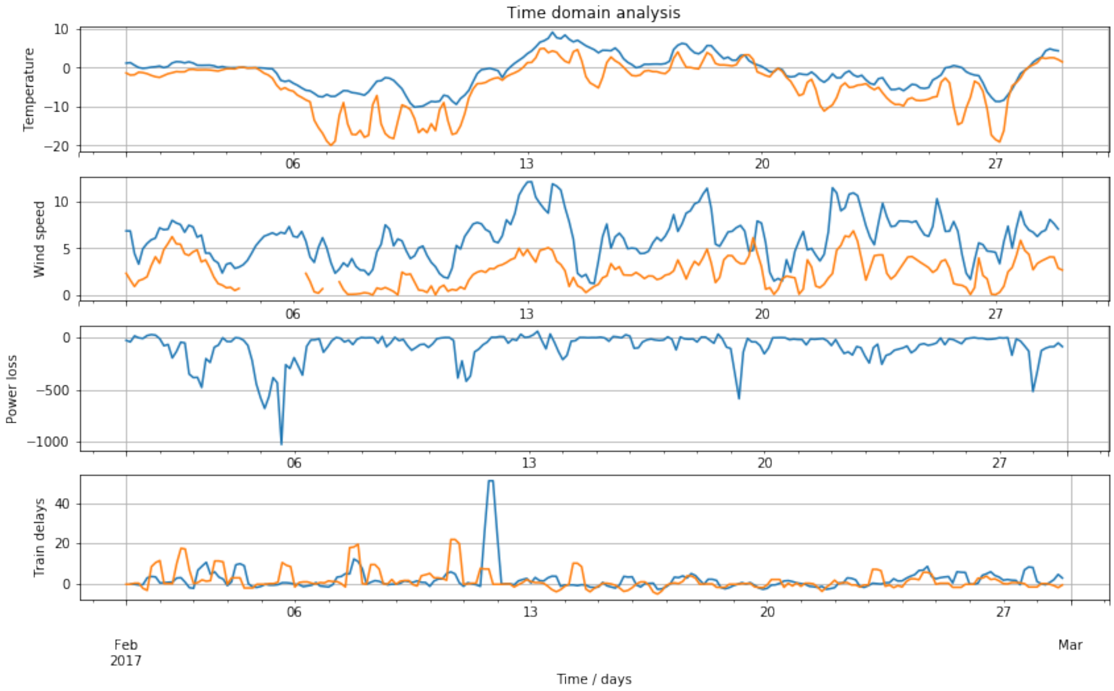
The weather parameters, wind turbine data and train delays can also be analyzed in time domain, to see if they repeat some similar patterns.

Time domain analysis already reveals that wind speeds and temperatures are sometimes quite different in ground level and wind turbine nacelle level, and sometimes they are close to each other. This may provide deeper insight into the current weather conditions, and may be relevant to train delays too.

```
In [48]: figure(figsize=(14,8))

         ax=subplot(4,1,1)
         TT.Tt.plot(ax=ax)
         TT.Tf.plot(ax=ax)
         grid()
         ylabel('Temperature')
         title('Time domain analysis')

         ax=subplot(4,1,2)
         TT.Ut.plot(ax=ax) # Blue
         TT.Uf.plot(ax=ax) # Orange
         grid()
         ylabel('Wind speed')

         ax=subplot(4,1,3)
         (TT.Pl*-1).plot(ax=ax)
         grid()
         ylabel('Power loss')
```

```
ax=subplot(4,1,4)
TT.DL_IC.plot(ax=ax)   # Blue
TT.DL_S.plot(ax=ax)    # Orange
grid()
ylabel('Train delays')
xlabel('Time / days')
```

```
Out[48]: Text(0.5,0,u'Time / days')
```



5 Conclusion

This far, we have collected data over one month from different sources and combined it together to visualize it and to make simple analysis of the dependences. This far, the analysis didn't reveal clear dependencis.

The next step is to include data over wider time range and to build a model for predicting the delays. The model makes it also possible to analyze the importance of different input variables for the prediction.