DATA MINING METHODS FOR FAILURE CLASSIFICATION ON PV-MODULES UNDER FIELD-CONDITIONS

G. Behrens, A. Dercho, H. Quakernack, T. Wächter, S. Hempelmann, I. Kruse

Munich, 23th of June 2016
• **Data Mining** ist the computational process of discovering patterns in large data sets

• **Big data** sets can come from
  • monitoring PV-plants - best way module based data, ✔
  • picturing and video streaming in PV-fields – IR, EL,
  • environment data – climate, irradiation, … ✔
  • Module specification data – data sheet

• **Goals by data Mining in PV-fields:**
  • Failure classification ✔
  • Failure prediction
  • Predictions for optimal component selection
4 Steps of Data Mining-Process

1. Data Collection
   - Module based SunSniffer®-Data from 3 PV-Fields
   - Weather API from Internet

2. Preprocessing
   - Data-Filtering
     - Cleaning
     - Interpolation
   - Normalization

3. Analysis
   - Correlation
   - Visualisation
   - Interview with Experts

4. Classification
   - Neural Network Learning
   - Experiments
   - Interpretation


Oral Presentation 5DO_12_4 – PVSEC 2016, Munich 23th of June 2016  Prof. Dr.-Ing. Grit Behrens
1. Step: Data Collection with SunSniffer®

- Sunsniffer®-Technology
  - Modulbased measuring of temperature and voltage
  - Powerline- data communicaction
  - Integrated in junction box

1. Step: Data Collection by Weather-API

- **weather data for example:**
  - CloudeCover [0,1]
  - Temperature [°F]

- **Worldwide locations**

- **Request-Types:**
  - Current conditions
  - Forecasts
  - historical data

- **1000 requests a day for free**
  >1000 requests: 0.0001$ each

**HTTP-Request:**
https://api.forecast.io/forecast/APIKEY/LATITUDE,LONGITUDE,TIME

**HTTP-Response (JSON):**
All weather data for given day, hour-by-hour resolution

Http://www.forecast.io, MUNICH 23.06.2016
2. Step: Data PreProcessing

Filtering

Example module temperature:
- **outliers** correction
- **Interpolation** outliers are missing
  - by next neighbours
  - time equidistant values

\[ x_f^{(i)} = \frac{x_f^{(i-1)} + x_f^{(i+1)}}{2} \]

Modul temperature values before (left) and after interpolation operations (right)

Oral Presentation 5DO_12_4 – PVSEC 2016, Munich 23th of June 2016
Prof. Dr.- Ing. Grit Behrens
2. Step: Data PreProcessing
Normalisation I/II

• Individual features have very often very different ranges of values.
• Numerical classification methods would ignore features with less value ranges

Feature vector in project:

<table>
<thead>
<tr>
<th>feature</th>
<th>unit</th>
<th>Value range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Temperatur</td>
<td>[ºC]</td>
<td>20 - 60</td>
</tr>
<tr>
<td>Module Voltage</td>
<td>[V]</td>
<td>0 - 100</td>
</tr>
<tr>
<td>String Current</td>
<td>[A]</td>
<td>0 - 20</td>
</tr>
<tr>
<td>Weather Temperatur</td>
<td>[ºC]</td>
<td>-20 - 50</td>
</tr>
<tr>
<td>Solar efficiency *</td>
<td></td>
<td>~(0 – 10)</td>
</tr>
<tr>
<td>Cloud Covering</td>
<td></td>
<td>0 - 1</td>
</tr>
</tbody>
</table>

* Solar efficiency = (sunset – sunrise)* (1- cloud covering)
2. Step: Data PreProcessing
Normalisation II/II

For example:
- **reziprocal transformation** of values of module temperatures
- Gives values between [-1,1]

\[
f : R \setminus \{0\} \rightarrow R \setminus \{0\} \\
f(x) = f^{-1}(x) = 1 / x
\]

- Modul temperature values after filtering [20, 60]
- After reziprocal calculations [-0.1, 0.04]
3. Step : Data Analysis

**Feature vector optimisation**

- Find out *dependencies* between components and *most significant* features
- Vizualisation
- Expert interivies

**Correlation matrix indices for relations between feature i and j :**

\[
c_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} (x_k^{(i)} - \bar{x}^{(i)})(x_k^{(j)} - \bar{x}^{(j)}), \quad \text{mit} \quad i, j = 1, \ldots, p
\]

* n – number of features vectors, p – number of components of feature vector

<table>
<thead>
<tr>
<th>Feature 1</th>
<th>Feature 2</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modul Temp.</td>
<td>Temperatur</td>
<td>82,02</td>
</tr>
<tr>
<td>Modul Temp.</td>
<td>String Current</td>
<td>43,98</td>
</tr>
<tr>
<td>Modul Temp.</td>
<td>Module Vol.</td>
<td>13,42</td>
</tr>
<tr>
<td>String Current</td>
<td>Temperatur</td>
<td>7,26</td>
</tr>
<tr>
<td>Module Voltage</td>
<td>Temperatur</td>
<td>2,34</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- yellow – Module Temperature,
- orange – String Currency
4. Step: Classification by Neural Network Classifier I/II

- **WEKA** – *Waikato Environment for Knowledge Analysis* [3]
  - Open Source Library for Java
  - Preprocessing, normalisation, classification, visualisation,

Results for training on 3 classes, all data:
- Defect diode
- No failure
- Mechanically destroyed

**Orange**: right classified 81,57%
**Yellow**: bad classified 18,42%

- false positives:
- some modules with failure classified in the „no failure“ result class

4. Step: Classification by Neural Network Classifier II/II

Results for training on 4 classes, 3 PV-Fields monitored by Sunsniffer

Only sunny days:

- Defect bypass diode
- No failure
- Mechanically destroyed
- Hailstorme damage

- Shadowing can recognized with 99% accuracy (SVM better than NN) [4]

<table>
<thead>
<tr>
<th>Right classified</th>
<th>Epochs</th>
<th>Momentum</th>
<th>Learning Rate</th>
<th>Training Data Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>584</td>
<td>0,1</td>
<td>0,1</td>
<td>98%</td>
</tr>
<tr>
<td>96%</td>
<td>69</td>
<td>0,2</td>
<td>0,5</td>
<td>66%</td>
</tr>
</tbody>
</table>

Results

- **Data Mining Algorithms are useful** on large data sets from *module based* monitoring data taken with *low-cost SunSniffer®-Sensor*.
- Weather data *taken from low cost - Internet API* are useful.
- Our Software application is *easy to adopt to any monitoring system with possibility of failure labeling*.
- *High recognition rates* give up to 96% accuracy in failure detection.
- Recognition of *different failure types is possible partly based based on additional information from weather sensors*.
- *Different module types* in the 3 PV-plants were not a problem.
- *The recognition rate results are better than better are the training data sets*:
  - Quantity: than more labeled failure types exist (up to 100%)  
  - Quality: than more optimized is the training data set (excellent conditions, significant features)
Thank You for Your Attention!