Transient Data Library and Use Case for Analysis Algorithms

Yuqing Dong
ydong22@utk.edu
Background

Open Energy Data Initiative for Solar Data and Analytics

(OEDI Systems Integration)
Access to Composite Data Sets for Solar Integration Analysis

- Distributed solar generation is increasing exponentially
- There are several potentials to be harnessed and challenges to be addressed
- Visibility of DERs is limited
- Different technologies house different types of distribution edge information (Multiple data sets from multiple silos)
  - Traditional SCADA data
  - Smart Meter data
  - Smart inverter data
  - PMU/micro-PMU data
- Access, confidentiality, synchronization, imputation, mapping into composite data sets for analysis are time-consuming, tedious tasks
OEDI-SI Program Goals

• Make generic test data for solar generation integration to power system analysis publicly available for testing and verification of new analytics
• Provide data anonymization, imputation, synchronization, and bad data pre-processing to consistently map and integrate the individual data coming from different systems
• Provide open-source algorithms to enable using the integrated data for different power system algorithms
  • Physics-/network- & data-/ML-based solar analytics
  • Make their verified results and metrics available to researchers for comparing their algorithms
Provide Easy Access to Data and Algorithms for Solar Integration Simulations

- Adaptation of power systems analytics for distribution networks with high distributed solar generation participation
- Robust physics/network model-based algorithms
- New machine learning algorithms based on large data sets
- Steady-state and transients’ analysis
- Data interfaces (CIM, OpenDSS, Gridlab-D)
Main Functions: 1 of 4

- **Use OEDI SI** to reproduce simulation results
  - Go to a use case of interest
  - Pick a certain scenario
  - Download the **public composite input** data
  - You do **not** need to run data pre-processing. The composite (integrated & cured) data is ready to be used
  - Download the **public algorithms container(s)**
  - Run the executable(s). You should be able to **reproduce** the results (also posted on **OEDI SI**)

- **Main goals of this function**
  - Compare **different scenarios (algorithms)** for a specific use case (distribution state estimation, volt/VAR, etc.)
  - Build confidence that a particular approach works
  - **Contribute** to a **public data & algorithms repository** for future R&D work
Main Functions: 2 of 4

• Use OEDI SI to replicate simulation results
  • Go to a use case of interest
  • Pick a certain scenario
  • Download the public algorithms container(s)
  • Go to the raw data page for that scenario
  • Download the public raw input data
  • Swap (some of) the public raw data with your private data
  • You will have to run data pre-processing. You will get a notification when the composite (integrated & cured) data is ready
  • Run the executable
  • Use the metrics posted on OEDI SI to check how a particular approach performs with your own data

• Main goals of this function
  • Compare different scenarios (algorithms) for your data
  • Pick a particular approach works best with your data
Main Functions: 3 of 4

• Use OEDI SI to ensure your algorithm/approach is robust
  • Go to a use case of interest
  • Pick a certain scenario
  • Download the public algorithms container(s)
  • Swap (some of) the public algorithm(s) with your own algorithm/approach
  • Download the public composite input data
  • You will not need to run data pre-processing
  • Run the new executable/container
  • Use the metrics posted on OEDI SI to check/verify your approach works

• Main goals of this function
  • Ensure your algorithm works under different scenarios for a specific use case
  • Share your algorithm with other researchers on OEDI SI
Main Functions: 4 of 4

- Use OEDI SI to ensure your algorithm and data is generalizable/scalable
  - Go to a use case of interest
  - Pick a certain scenario
  - Download the public algorithms container(s) & swap some with your own
  - Go to the raw data page for that scenario
  - Download the public raw input data
  - Swap (some of) the public raw data with your private data
  - You will have to run data pre-processing.
  - Run the new executable/container
  - Use the metrics posted on OEDI SI to check/verify your results

- Main goals of this function
  - Ensure your algorithm works under different scenarios with your own data
  - Share your algorithm and data with other researchers on OEDI SI
# OEDI SI – OEDI FY22 Lab Call, Core Topic

<table>
<thead>
<tr>
<th>Public Data</th>
<th>Private Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Network models</strong></td>
<td><strong>Network models</strong></td>
</tr>
<tr>
<td>123 IEEE network</td>
<td>Confidential network data</td>
</tr>
<tr>
<td>SmartDS networks</td>
<td><strong>Confidential complementary data</strong></td>
</tr>
<tr>
<td><strong>Complementary data</strong></td>
<td>Confidential load/solar PV profiles</td>
</tr>
<tr>
<td>Load/solar PV profiles</td>
<td>Confidential smart meter data, etc.</td>
</tr>
<tr>
<td>AMI/Smart meters</td>
<td><strong>Steady-state &amp; Transients</strong></td>
</tr>
<tr>
<td>Smart inverters</td>
<td>Distribution State Estimation</td>
</tr>
<tr>
<td>PMUs</td>
<td>Volt/Var optimization</td>
</tr>
<tr>
<td>Smart sensors</td>
<td>Transient analysis, etc.</td>
</tr>
<tr>
<td><strong>Public Algorithms</strong></td>
<td><strong>Private Algorithms</strong></td>
</tr>
<tr>
<td>Verified algorithms using</td>
<td>To test proprietary algorithms locally</td>
</tr>
<tr>
<td>123 IEEE network</td>
<td>Using OEDI SI data pre-processing</td>
</tr>
<tr>
<td>SmartDS networks</td>
<td>Using OEDI SI public data</td>
</tr>
<tr>
<td><strong>Steady-state &amp; Transients</strong></td>
<td><strong>Private</strong></td>
</tr>
<tr>
<td>Distribution State Estimation</td>
<td><strong>User interface is rudimentary</strong></td>
</tr>
<tr>
<td>Volt/Var optimization</td>
<td><strong>Actively promoting to public</strong></td>
</tr>
<tr>
<td>Transient analysis, etc.</td>
<td><strong>In BP3, more data &amp; UI &amp; algorithms/data from other SETO/SI programs/projects</strong></td>
</tr>
</tbody>
</table>

- **SETO Core Lab Call Program**
  - National Labs collaboration
  - ANL, NREL, ORNL, PNNL
  - Approaching end of 2\textsuperscript{nd} Budget P
  - User interface is rudimentary
  - But private data/private algorithm proof-of-concept implemented & tested
  - Actively promoting to public
  - Users group kick-off
  - In BP3, more data & UI & algorithms/data from other SETO/SI programs/projects
OEDI SI – Web Portal

https://openei.org/wiki/OEDI-SI/Overview
A Use-case Example: Transient Data Generation and Event Identification

Demonstration on IEEE 123 bus system
Use-case Overview

• Objective: Develop algorithms for event detection and identification based on the generated datasets in the open-source data library

• Methodology
  • Data generation: multi-scenario POW data in ATP-EMTP (free license)
  • Data preprocessing: remove the initialization of EMT simulation
  • Event detection: periodic waveform detector
  • Event identification: (1) extract feature based on discrete wavelet transform (2) classify the events by multiple machine learning algorithms

• Advantage
  • Speed and accuracy guaranteed
Data Generation Flow Chart

Steady state setting
- Loading condition
- PV penetration

Contingency setting
- **Fault type:** Single-phase to ground (0)/line-to-line (1)/three-phase to ground fault (2)
- **Fault location:** Adjacent nodes + all three-phase buses

Output: Voltage/current waveforms in IEEE 123 bus test system
Sampling rate: 20kHz; Time duration: 0~0.6s (event at 0.3s)

Model conversion

Contingency setting and dynamic simulation

Open-source data

https://github.com/openEDI/oedisi-transient/tree/main/input
Transient Algorithms Flow Chart

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>Preprocessed feeder head voltage/current waveforms</td>
<td>Detected event data period</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Pre-selected detection threshold</td>
<td></td>
</tr>
<tr>
<td>parameter</td>
<td>Detected event data period</td>
<td>Event type (0/1/2)</td>
</tr>
</tbody>
</table>

Local Data Repository

Training dataset generated using ATP or user input dataset (CSV, HDF5, COMTRADE etc.)

OEDI UI

Pick a transient algorithm

Local Workstation

Training dataset

• Event detection algorithm
• Event identification algorithm

Docker Container

Pick a testing dataset

Trained Model

• Waveform detector and oscillation detector thresholds
• Neural network weights and biases

Testing event detection and identification model

Docker Container
Demo Instructions

1. Open a terminal or command prompt, pull the docker image using the following command:
   
   ```
   docker pull openenergydatainitiative/oedisi-transient-demo:0.0.1
   ```

2. Run the Docker image using the following command which maps port 8888 of the container to the host's port 8888:
   
   ```
   docker run -p 8888:8888 openenergydatainitiative/oedisi-transient-demo:0.0.1
   ```

3. You should see the Jupyter Notebook server starting up. Look for a URL with a token in the terminal output, similar to:
   
   ```
   http://127.0.0.1:8888/?token=<TOKEN>
   ```

4. Copy the URL and paste it into a web browser. This will open the Jupyter Notebook interface in your browser, where you can access and work with the notebook.

For demonstration, the preprocessed data have been saved as csv files and directly loaded in the detection and identification Jupyter notebook examples.
### Use-case Output

#### Confusion matrix of SVM algorithm

(a) Raw data

(b) After feature extraction

#### Confusion matrix of Naive Bayes algorithm

(a) Raw data

(b) After feature extraction

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Index</th>
<th>Naive Bayes</th>
<th>Decision Tree</th>
<th>SVM</th>
<th>KNN</th>
<th>Random Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Raw data</strong></td>
<td>Accuracy (%)</td>
<td>98.74</td>
<td>100</td>
<td>99.58</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Exec. Time (s)</td>
<td>2.60</td>
<td>9.15</td>
<td>26.11</td>
<td>2.02</td>
<td>7.07</td>
</tr>
<tr>
<td><strong>After feature extraction</strong></td>
<td>Accuracy (%)</td>
<td>99.58</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Exec. Time (s)</td>
<td>0.72</td>
<td>2.24</td>
<td>4.37</td>
<td>0.51</td>
<td>4.96</td>
</tr>
</tbody>
</table>
Conclusion

• Easy access to data integration for power system analysis with verified benchmark algorithms
  • Publicly available datasets to test and compare algorithms to improve their accuracy and scalability
  • Small IEEE network models with complementary data
  • Large synthetic network models with results
  • Algorithms only for data preprocessing (for private data/algorithms), DSSE, DOPF and some transient analysis
• Demos can be found on OEDI-SI web portal

https://openei.org/wiki/OEDI-SI
Thank you!