SOLAR ENERGY TECHNOLOGIES OFFICE
U.S. Department Of Energy

GEDI SI

A Use-case Example: Transients

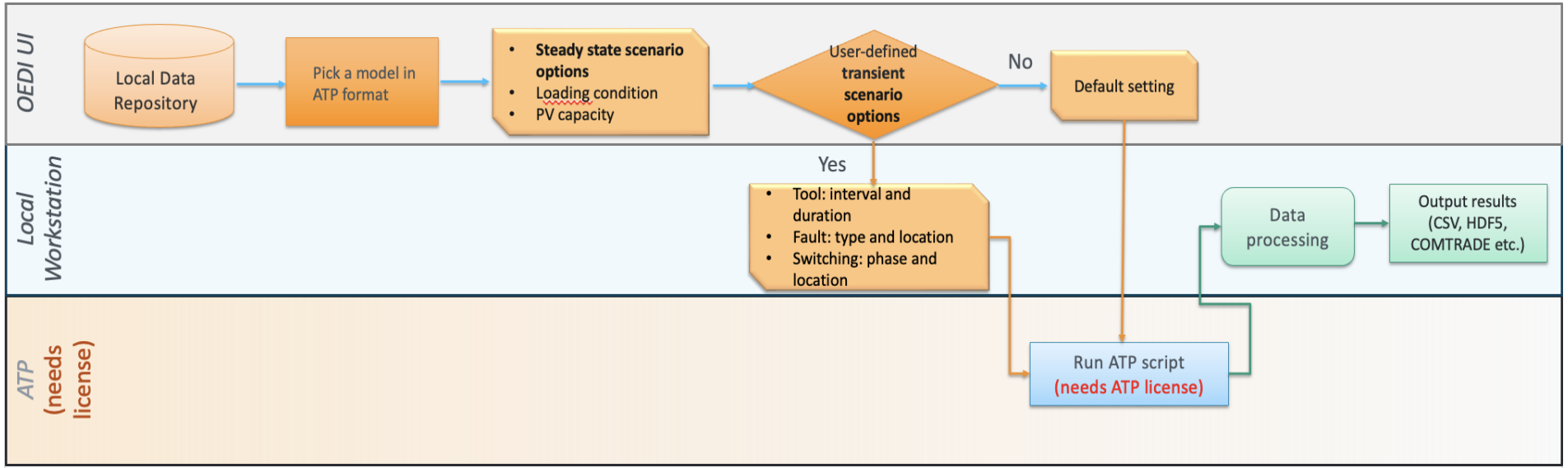
Demonstration on IEEE 123 bus system

8/14/2023

Oak Ridge National Laboratory

Transient Use-case a. Data Generation

- **Objective:** Development of an open-source transient data library
  - Providing POW transient data in distribution models under multiple scenarios
  - Developing algorithms for event detection and classification purpose, based on the datasets in the open-source data library



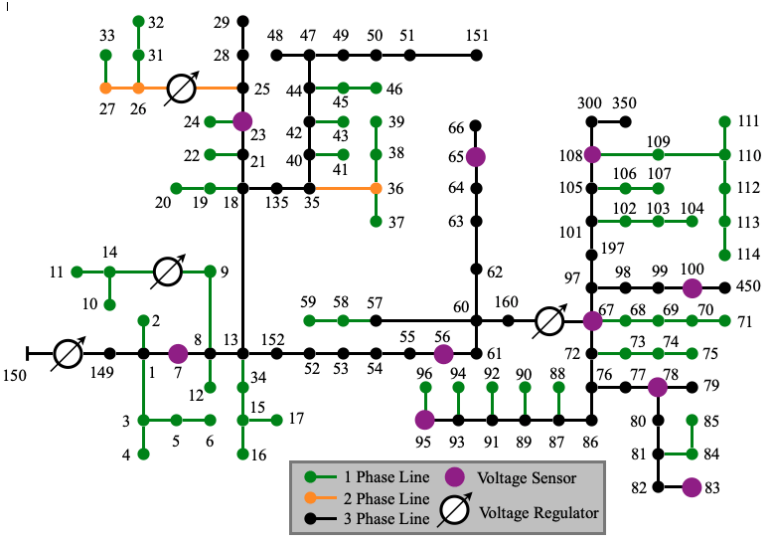
Transient Use-case a. Data Generation

Data generation

- 1584 fault cases in IEEE 123 bus system
- Loading condition: 0.4/1.0
- PV capacity: 0.4/0.6/0.8/1.0
- Fault location: 3 ADJ + 63 three-phase buses
- Fault type: single-phase/line-to-line/three-phase
- Sampling rate: 20 kHz; Time window: [0s, 0.6s]

Data preprocessing

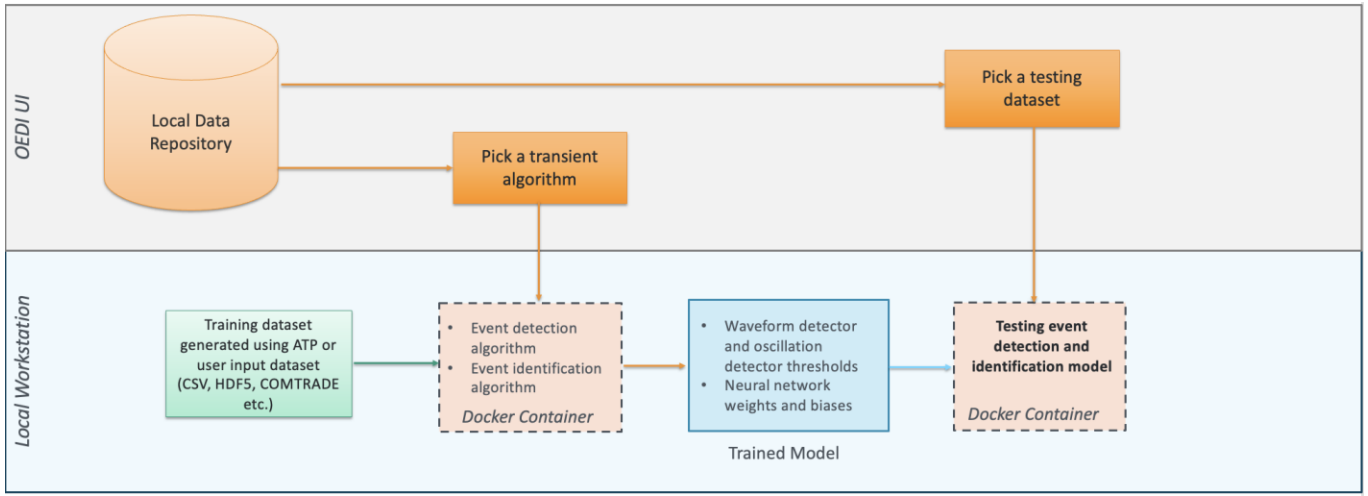
- Avoid the initialization of simulation cases
- Remove [0s, 0.2s] of the data



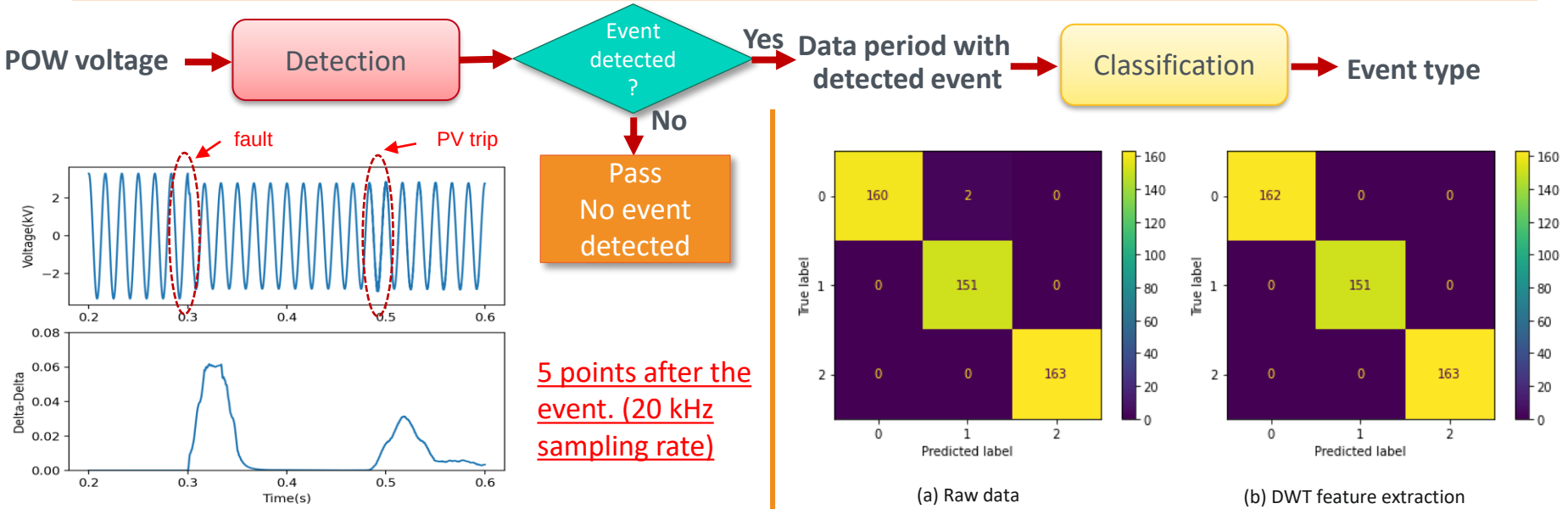
Topology of IEEE 123 bus system

Transient Use-case b. Algorithms

- **Methodology:**
  - Conducting data generation and preprocessing
  - Developing waveform-based event detection algorithms
  - Developing basic machine learning-based event classification algorithms



Transient Use-case b. Algorithms



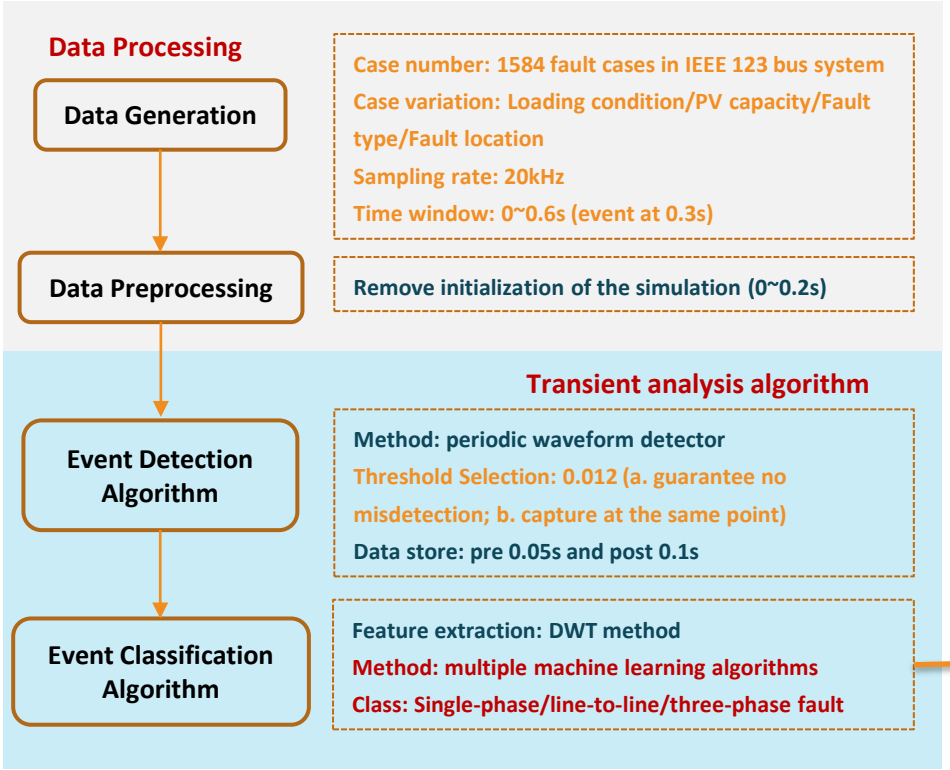
- Input Data: POW *phase A* feeder head voltage of IEEE 123 bus network
  - ✓ Loading condition: 0.4; PV capacity: 0.4
  - ✓ Fault location: ADJ1
- Method: Periodic waveform detector
- Output: 0 (no event) / 1 (pass data during event period)

(a) Raw data (b) DWT feature extraction

Confusion matrix of one ML algorithm (SVM)

- Input Data: Detected event data period
  - ✓ 1584 fault cases (70% for training; 30% for testing)
- Method: Multiple machine learning algorithms
- Output: Identified event type as
  - ❑ 0 (single-phase) / 1 (line-to-line) / 2 (three-phase)

Transient Demo



Naïve Bayes/SVM

- Define model
  - Cross validation
- Show results
  - Error, accuracy, time, confusion matrix

Transient Demo

1. Open a terminal or command prompt, pull the docker image using the following command:

   docker pull liub725/oedi-transient-demo

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 liub725/oedi-transient-demo

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 are saved as csv files and directly loaded in the detection and classification Jupyter notebook examples.