Pacific Northwest
NATIONAL LABORATORY

PNNL-35875

Survey of Use Cases and Scenarios on the Open Energy Data Initiative Solar Systems Integration (OEDI SI) Platform

April 2024

Bhaskar Mitra
Meghana Ramesh
Rounak Meyur
Rabayet Sadnan
Tylor Slay
Andrew Reiman
Jim Follum

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-35875

Survey of Use Cases and Scenarios on the Open Energy Data Initiative Solar Systems Integration (OEDI SI) Platform

April 2024

Bhaskar Mitra
Meghana Ramesh
Rounak Meyur
Rabayet Sadnan
Tylor Slay
Andrew Reiman
Jim Follum

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Summary

The Open Energy Data Initiative Solar Systems Integration (OEDI SI) Data and Modeling Platform offers a comprehensive set of use cases tailored for power systems analysis. Each use case is centered around a specific power system analysis problem, supported by composite input data and reference algorithms. These composite input datasets are meticulously assembled using OEDI SI's data preprocessing tools, which integrate raw data from various sources. The primary objectives of the OEDI SI Platform include facilitating access to composite input data through widely accepted input/output formats and verified results. This accessibility enables power system network researchers and developers to validate their algorithms and showcase their applications' capabilities to the broader community. Moreover, the platform strives to promote reproducible, robust, replicable, and generalizable solar systems integration research.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy under the Solar Energy Technologies Office (SETO) Award Number 38411.

We extend our deepest gratitude to the invaluable contributions made by Rui Yang (NREL), Karthikeyan Balasubramaniam (ANL), Jin Dong (ORNL), and their respective team members to this project. Their dedication and expertise have significantly enriched our endeavors.

Special thanks to Dr. M. Kemal Çelik, the SETO/SI Technology Advisor, for the unwavering support and generous sponsorship that made this project possible. Dr. Çelik's guidance and vision have been instrumental in shaping the trajectory of our work.

Acronyms and Abbreviations

AI	artificial intelligence
ANL	Argonne National Laboratory
ATP	alternate transient program
CNN	convolutional neural network
DER	distributed energy resources
DOPF	distribution optimal power flow
DSSE	Distribution State Estimator
EKF	Extended Kalman Filter
ML	machine learning
NREL	National Renewable Energy Laboratory
OEDI SI	Open Energy Data Initiative System Integration
ORNL	Oak Ridge National Laboratory
PNNL	Pacific Northwest National Laboratory
POW	point-on-wave
PV	photovoltaic
SETO	Solar Energy Technology Office

Contents

Figures

Tables

1.0 Introduction

The Open Energy Data Initiative Solar Systems Integration (OEDI SI) Data and Modeling Platform is a cutting-edge solution designed to tackle the complex challenges of electric power systems analysis. This platform was developed under the National Lab Core Capabilities for FY 22-24 from the Solar Energy Technologies Office (SETO) and is accessible through a website.[1] The OEDI SI platform is a collaborative effort among a team that includes the National Renewable Energy Laboratory (NREL), Pacific Northwest National Laboratory (PNNL), Oak Ridge National Laboratory (ORNL), and Argonne National Laboratory (ANL). The integration of renewable energy and local inverter-based controls in the modern distribution system power grid has posed new control challenges to the system operator. This necessitates advanced algorithms that can perform complex tasks such as state estimation, optimal power flow, and event detection for the large distribution feeders in a computationally efficient manner. A common challenge in this validation task is the lack of a framework that automates this comparison process. To perform a detailed validation of the proposed algorithm, a researcher must gather test feeders of different sizes, identify multiple state-of-the-art algorithms, and generate a work pipeline to perform a comparative study of these methodologies. OEDI SI offers a comprehensive collection of use cases, each crafted to address specific issues within the realm of solar systems integration. These use cases are constructed around distinct general power system analysis problems, leveraging a composite input dataset and a reference algorithm to provide robust solutions.

Central to the effectiveness of the OEDI SI platform is its ability to seamlessly integrate raw input data from different sources. Through advanced data preprocessing tools, OEDI SI harmonizes and consolidates diverse datasets, ensuring a cohesive and comprehensive foundation for analysis. This integration process is crucial, as it enables the platform to draw insights from a wide array of data streams, ranging from solar irradiance measurements to grid infrastructure specifications.

The composite input data, synthesized by OEDI SI's sophisticated data preprocessing tools, serves as the bedrock for addressing various challenges encountered in electric power systems analysis. The use cases developed in the OEDI SI platform have been tailor-made to provide insights into optimizing grid performance, assessing renewable energy integration, or mitigating grid instabilities.

Several algorithms have been developed and validated to ensure accuracy and reliability in analysis outcomes through several peer-reviewed publications. The workflow facilitates users in seamlessly integrating an algorithm model into the co-simulation framework, followed by thorough testing and validation using curated power grid datasets. The workflows demonstrate the interaction between the different federates in the OEDI SI platform. In this way, the platform provides stakeholders across the energy sector with the essential tools and resources required to navigate the intricate challenges posed by modern power systems, ensuring precision and effectiveness in their operations and decision-making processes.

In this report we describe the use cases developed by different participating labs. Each of the use cases addresses a specific power system analysis problem and contains scenarios that demonstrate how the problem can be addressed with a particular algorithm, network model, and set of input conditions. This report provides details for the use cases and scenarios developed

[1] https://openei.org/wiki/OEDI-SI/Overview

by PNNL, along with brief summaries of scenarios developed by other lab partners to reflect the full scope of the OEDI SI platform.

The report is structured in the following manner. Section 2.0 discusses the overall OEDI SI platform. Section 3.0 discusses the distribution system state estimator use case as developed by PNNL and NREL. Section 4.0 discusses the distribution optimal power flow use case as developed by PNNL and NREL. Section 5.0 discusses the transient data use cases developed by PNNL and ORNL. Section 6.0 concludes the report with deliverables planned for the remainder of the year and planned developments in the upcoming project cycle.

2.0 OEDI SI Platform

OEDI SI is a co-simulation framework that interfaces Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS) [1] components for dynamic testing of algorithms and data models. OEDI SI provides common Pydantic power system models to ensure interoperability between individual components. These components are known as federates and we will use these terms interchangeably. An example consisting of two federates in a co-simulation setup is shown in Figure 1. The platform offers a repository for algorithms capable of executing complex tasks such as distribution state estimation, distribution optimal power flow, and event detection efficiently. The OEDI SI platform provides a pivotal advancement in the field. Well curated and diverse power grid datasets are of utmost importance to test or validate different control algorithms before their actual implementation and deployment in the power grid. The OEDI SI platform helps to create a data lake with access to open-source power grid data and power system analysis algorithms.

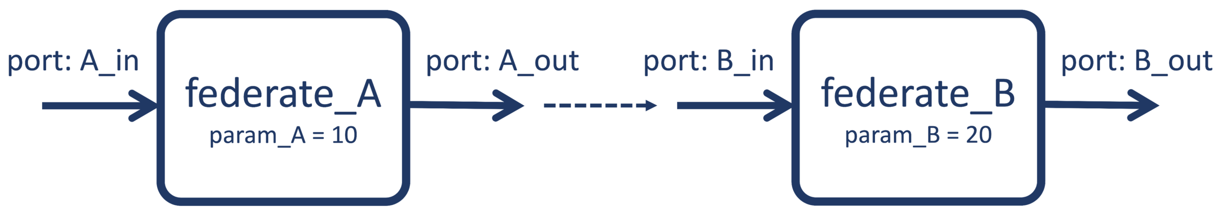


Figure 1. Wiring diagram of an example system composed of two federates. The output ‘A out’ from ‘federate A’ is used as input ‘B in’ for ‘federate B’.

The use of docker containers allows for the encapsulation of dependencies of the OEDI SI federates to ensure interoperability between operating systems. They help to promote scalability, ease of deployment and portability. Single container implementation runs all federates defined by a given scenario within one docker container. Figure 2 shows and example dockerfile that contains all the dependencies of different federates.

```
FROM python:3.10.6-slim-bullseye
ARG SCENARIO
RUN apt-get update && apt-get install -y git ssh
RUN mkdir -p /root/.ssh

# Libraries specifically required for Mac machines
RUN apt update && apt install -y \
  libboost-dev \
  libboost-filesystem-dev \
  libboost-program-options-dev \
  libboost-test-dev \
  libzmq5-dev python3-dev \
  libopenblas-dev \
  build-essential swig cmake git

WORKDIR /simulation
COPY . .
RUN pip install -r requirements.txt
```

Figure 2. Example dockerfile showing the dependencies.

Once a container has been built and published to docker hub the user can select their desired scenario from OEDI and pull the corresponding container. Figure 3 highlights the process of downloading a container, running it on your local machine, and reviewing the results using a Jupyter notebook.

Repository overview

PNNL Linear Distribution Optimal Power Flow (LinDistFlow)

Load container

```
docker pull openenergydatainitiative/pnnl-dopf-lindistflow:0.0.0
```

Running container

```
docker run -it -p 8888:8888 openenergydatainitiative/pnnl-dopf-lindistflow:0.0.0
```

Analysis

Open the Jupyter notebook *workflow.ipynb* at 127.0.0.1:8888 and follow the scenario selection steps.

Figure 3. Screenshot for a single container workflow showing PNNL's LinDistFlow on OEDI SI website.

3.0 Distribution System State Estimator Use Case

Figure 4 presents the workflow for the distribution system state estimation (DSSE) use case. The use case provides containerized DSSE co-simulation scenarios that ingest a test feeder dataset and a choice of DSSE algorithm on the workstation. The DSSE algorithm is executed inside the docker container to estimate voltage magnitudes and angles at different buses. Upon completion of the co-simulation, the recorded results are copied from the docker container to the local workstation for post-processing and plotting operations.

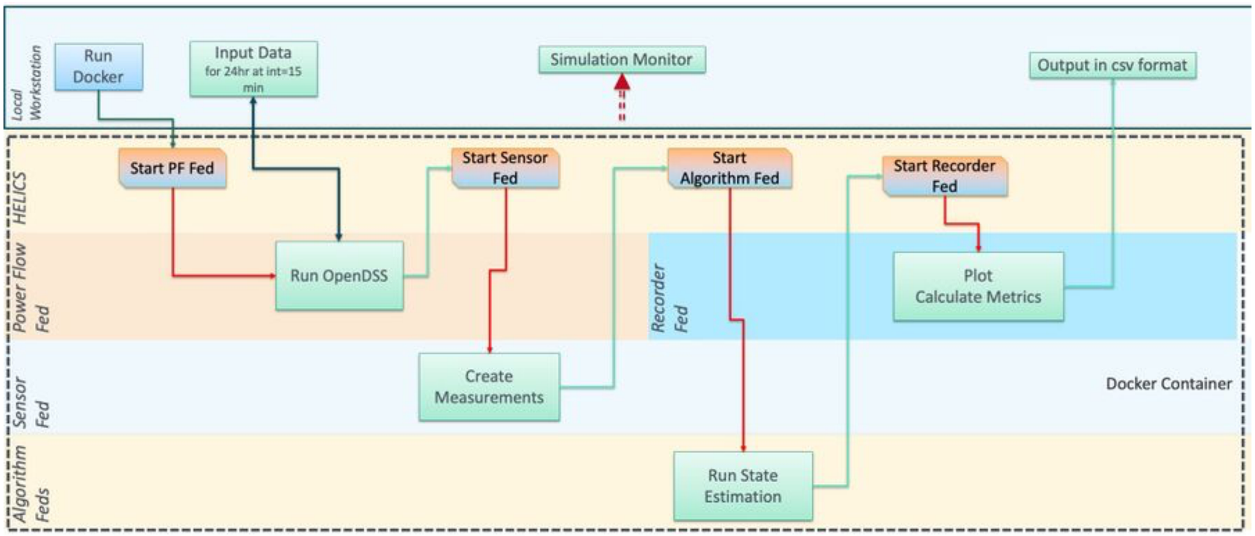
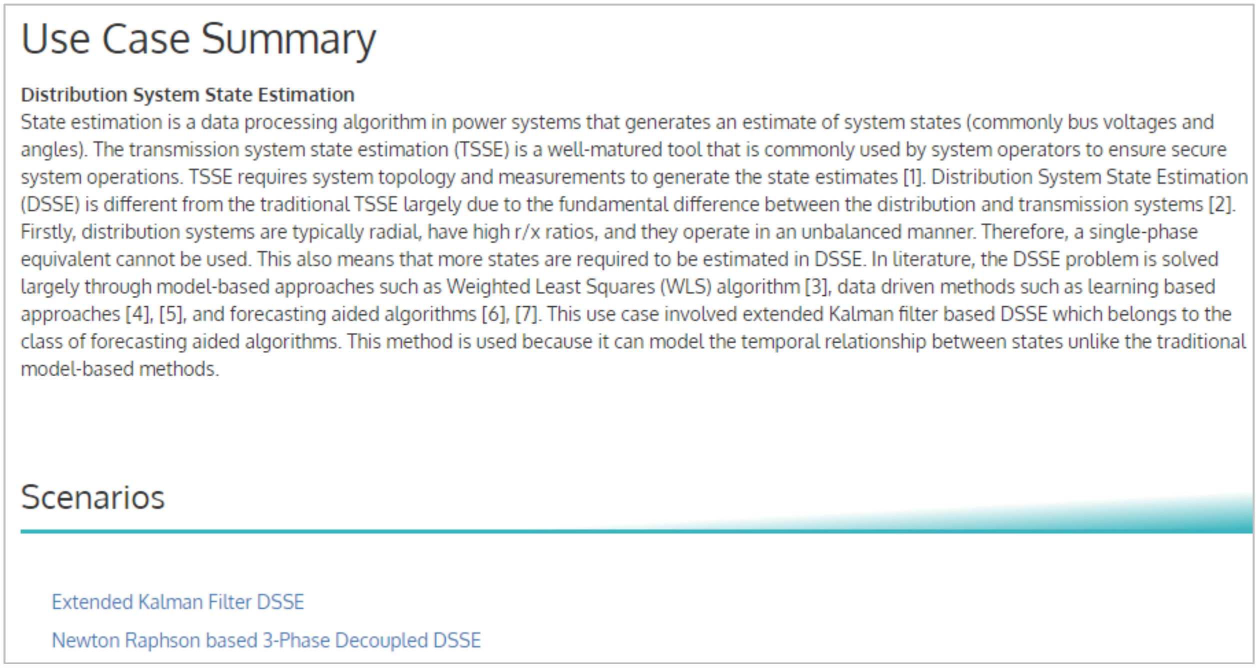


Figure 4. Overview of the DSSE use case workflow.

This use case provides relevant scripts and information to perform DSSE algorithms that estimate voltage magnitudes and angles at different nodes in the distribution network. Figure 5 provides the screenshot of the use case developed by PNNL on the OEDI SI website.


Use Case Summary

Distribution System State Estimation

State estimation is a data processing algorithm in power systems that generates an estimate of system states (commonly bus voltages and angles). The transmission system state estimation (TSSE) is a well-matured tool that is commonly used by system operators to ensure secure system operations. TSSE requires system topology and measurements to generate the state estimates [1]. Distribution System State Estimation (DSSE) is different from the traditional TSSE largely due to the fundamental difference between the distribution and transmission systems [2]. Firstly, distribution systems are typically radial, have high r/x ratios, and they operate in an unbalanced manner. Therefore, a single-phase equivalent cannot be used. This also means that more states are required to be estimated in DSSE. In literature, the DSSE problem is solved largely through model-based approaches such as Weighted Least Squares (WLS) algorithm [3], data driven methods such as learning based approaches [4], [5], and forecasting aided algorithms [6], [7]. This use case involved extended Kalman filter based DSSE which belongs to the class of forecasting aided algorithms. This method is used because it can model the temporal relationship between states unlike the traditional model-based methods.

Scenarios

Extended Kalman Filter DSSE

Newton Raphson based 3-Phase Decoupled DSSE

Figure 5. Screenshot of DSSE use case summary from the OEDI SI website.

3.1 Extended Kalman Filter DSSE Scenario

This scenario helps to demonstrate the OEDI SI use case for DSSE and was developed by PNNL. The goal is to estimate the bus voltage magnitudes and angles given power and voltage measurements at different buses. This scenario employs an extended Kalman filter (EKF)-based method for DSSE [2]. Figure 6 provides the screenshot of this scenario from the OEDI SI website.

Scenario Summary

Extended Kalman Filter DSSE

- Objective: An Extended Kalman Filter (EKF)-based State Estimation algorithm
- Use Case: Distribution System State Estimation
- Methodology
  - This scenario employs an extended Kalman filter (EKF) based method for DSSE. The EKF method has two-steps i.e., prediction step and an update step. Note that the voltage magnitudes and voltage angles are the states to be estimated. Therefore, the number of states is twice as many compared to the total number of nodes in the system.
  - The DSSE algorithm is integrated in OEDI-SI as an independent federate, where HELICS maintains a message queue. This allows each federate to move at their own pace. The use case represents a time series analysis at 15-minute intervals for 24 hours. The sensor federate generates the measurements using the power flow results generated by the power flow federate. At each time step, the EKF algorithm (DSSE) federate generates the voltage magnitude and angle estimates by using the measurement set and topology data. Results are logged by the recorder federate as well as written in the .csv files. They are also plotted using the post-processing scripts.
- Inputs
  - Node names, nominal node voltages, and angles
  - System Y-bus matrix
  - Location of source bus
  - Nominal active and reactive power loads at all nodes (used for pseudo-measurements)
  - Measurements of voltage magnitudes
  - Measurements of real and reactive powers
  - Location of all measurements. Note that the measurements are randomly generated at 20 % of the total nodes (as configured in the sensor federate). For the rest of the nodes, DSSE federate generates the pseudo-measurements.
- Outputs
  - Estimated voltage and the estimated angle at all nodes
- Configuration
  - The user is not required to manipulate the internal contents of this image. To run the image, the user needs to follow the instructions in the readme file.
- Webinars

  Webinar_Demo_Presentation_PNNL_DSSE.pdf

Figure 6. Screenshot of PNNL's DSSE scenario summary from the OEDI SI website.

The EKF method has two steps: prediction and update. The co-simulation runs at 15-minute intervals for 24 hours. The sensor federate generates the measurements using the power flow results generated by the feeder federate. At each time step, the EKF algorithm (DSSE) federate generates the voltage magnitude and angle estimates by using the measurement set and topology data. Results are logged by the recorder federate as well as written in the csv files. They are also plotted using the post-processing scripts.

The PNNL-developed, EKF-based DSSE scenario has been containerized in a Docker container. It consists of relevant python scripts and C++ executable to implement the EKF-based DSSE algorithm for the IEEE 123 bus feeder. Results of EKF based DSSE implementation are shown in Figure 7. The left plot shows the estimated voltage profile (red line plots) for five nodes in the network. The estimated voltages are compared with the true voltages (blue line plots). The right plot shows the mean absolute error of voltage estimation at different nodes in the network for a single time instant.

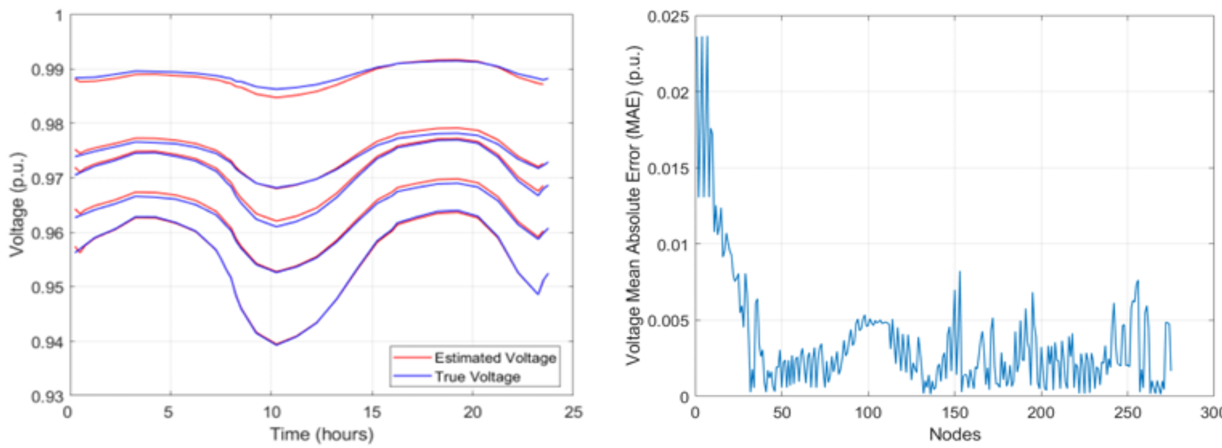


Figure 7. Results of EKF-based DSSE implementation.

3.2 Newton-Raphson-Based Three-Phase Decoupled DSSE Scenario

The Newton-Raphson based three-phase decoupled DSSE developed by NREL [3] is a nonlinear weighted least square problem where nonlinear algebraic equations are used to solve the problem by the Newton-Raphson method. The three-phase measurements are concatenated into a single variable and conventional transmission system's single-phase state estimation algorithm is leveraged to solve the problem. Figure 8 shows the screenshot of this scenario from the OEDI SI website.

Scenario Summary

Newton Raphson based 3-Phase Decoupled DSSE

- Objective: Newton Raphson based 3-Phase Decoupled DSSE
- Use Case: Distribution System State Estimation
- Methodology
  - This scenario models DSSE as a nonlinear weighted least square (WLS) problem. The formulated nonlinear WLS problem will be converted to a nonlinear algebraic equation using the first-order optimality condition, and then solved by the Newton-Raphson method. In the 3-phase decoupled strategy, it involves concatenating the three-phase measurements into a single variable and directly utilizing the conventional transmission system single-phase state estimation algorithm to solve it.
- Inputs
  - Node names, nominal node voltages
  - System Y-bus matrix
  - Location of source bus
  - Nominal active and reactive power loads at all nodes (used for pseudo-measurements)
  - Measurements of voltage magnitudes
  - Measurements of real and reactive powers
  - Location of all measurements
- Outputs
  - Estimated voltage and angle at all nodes
- Configuration
  - The user is not required to manipulate the internal contents of this image. To run the image, the user needs to follow the instructions in the readme file.

Figure 8. Screenshot of Newton-Rapson-based decoupled DSSE scenario summary from the OEDI SI website.

4.0 Distribution System Optimal Power Flow Use Case

Figure 9 presents the workflow for the distribution optimal power flow (DOPF) use case. This use case provides containerized DOPF co-simulation scenarios that ingests a test feeder dataset and a choice of DOPF algorithm on the workstation. The DOPF algorithm is executed inside the docker container to evaluate optimal photovoltaic (PV) inverter setpoints for the test feeder. Upon completion of the co-simulation, recorded results are copied from the docker container to the local workstation for post-processing and plotting operations.

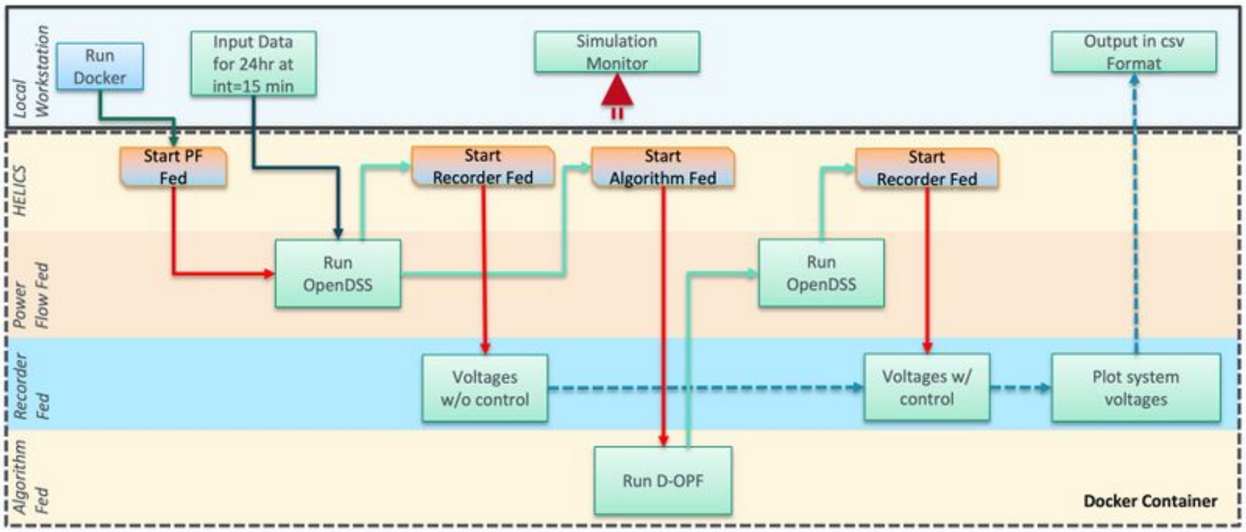


Figure 9. Overview of the DOPF use case workflow.

The optimal power flow for the three-phase power distribution systems attains optimal control sequence/solutions of the distributed energy resources (DER) inverters to maintain operational feasibility of the system while minimizing some cost function for the system.

This use case provides relevant scripts and information to perform DOPF algorithms to evaluate optimal DER inverter set points. Figure 10 provides the screenshot of the use case on the OEDI SI website. Under this use case there are scenarios created by PNNL and NREL.

Use Case Summary

Distribution Optimal Power Flow

The optimal power flow problem (OPF) is an optimization problem that determines the optimal setpoints of controllable devices to achieve certain control objectives subject to operational constraints in the power system. Mainly due to the nonlinearity of the power flow equations, the OPF problem is nonconvex and NP-hard in general [1]. In transmission systems, OPF methods are typically based on DC power-flow models and on the assumption that the system is balanced. In distribution systems, however, these two simplifying assumptions are no longer valid, since distribution systems generally have high R/X ratio and are unbalanced with a variety of different types of connections. Therefore, distribution optimal power flow (D-OPF) algorithms have been developed to address the AC OPF problem for multiphase distribution networks [2].

Scenarios

Feedback-based D-OPF

Linear Distribution Power Flow

Figure 10. Screenshot of DOPF use case summary from the OEDI SI website.

4.1 Linear Distribution Power Flow Scenario

This DOPF scenario was developed by PNNL [4]. The goal is to formulate the optimal power flow problem in the distribution system for active and reactive power setpoints of PV systems using topology information and voltage measurements. The co-simulation runs every 15 minutes for the given feeder configuration. Figure 11 provides the screenshot of this scenario from the OEDI SI website.

Scenario Summary

Linear Distribution Power Flow

- Objective: This research is to meant to demonstrate the OEDI SI use case for distributed optimal power flow (DOPF). The goal was to formulate the optimal power flow problem in the distribution system for active and reactive power setpoints of PV systems using topology information and voltage measurements. The co-simulation runs every 15 minutes as outlined within the scenario file for the given feeder configuration.
- Use Case: Distribution Optimal Power Flow
- Methodology
  - Check the validity of the data for possible feasible solutions
  - Formulate an optimal power flow problem for the 3-phase unbalanced distribution system
  - Control variables: active and reactive power setpoints of distributed energy resources (DERs)
  - Multi-objective: Optimize the voltage profile and attain network level objectives (Loss Minimization, DER curtailment minimization, CVR, etc.) •Constraints: Linear-approximated three-phase power flow equations, operational boundaries
- Inputs
  - branch real/reactive power flows
  - bus voltages
  - system topology
- Outputs
  - real/reactive power setpoints for controllable DERs
- Configuration
  - pf_flag: allow voltage violations
  - control_type: Watt, VAR, Watt-VAR

Figure 11. Screenshot of PNNL's DOPF scenario summary from the OEDI SI website.

The scenario example posted on the OEDI SI website by PNNL contains the LinDistFlow-based DOPF and the NREL-developed feedback based Online Multi-objective Optimization (OMOO) DOPF. The scenarios have been containerized in separate docker containers and they consist of relevant python scripts to implement the algorithm. Additionally, they consist of test scenarios such as IEEE 123 bus feeder, SMART-DS test feeders. The relevant scripts to execute the DOPF scenarios are available at https://github.com/pnnl/oedisi_dopf.

Each scenario has a "*system.json*" file, that depicts the wiring diagram for the OEDI SI platform or the manner the DOPF federate is connected to the other federates in the co-simulation framework. Users have the option to modify this file to add new federates or remove unwanted federates. We use feeder federate to obtain the network topology and power injection information at each time step. Thereafter, we run the DOPF algorithm to evaluate the optimal real and reactive power set points for the PV inverters and send these control commands to the feeder federate to be implemented in the succeeding time step. The recorder federate is used to record the bus voltages in the network after implementing the optimal control set points at each time step.

Users have the option to test the DOPF on multiple test feeders by modifying contents of the "*system.json*" file. This can be executed by altering the name of the feeder file in the feeder federate. Additionally, the LinDistFlow algorithm can be swapped with some other algorithm such as the feedback based OMOO power flow algorithm developed by NREL.

Furthermore, post-analysis python scripts have been developed by PNNL to construct voltage heatmaps of the distribution network at a particular time instant as well as voltage trees for the different phases in the network. These scripts can be used by the user to generate plots for comparing different DOPF algorithms and between different test cases. Examples of such plots are presented in Figure 12 and Figure 13.

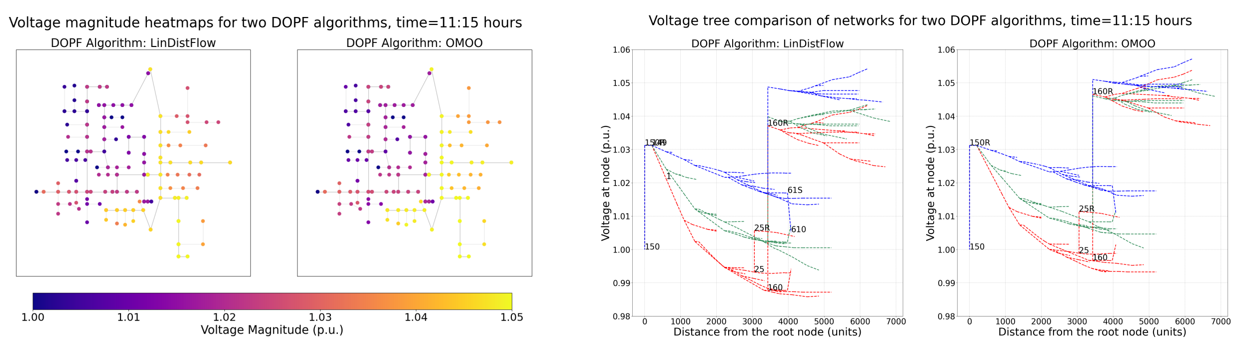


Figure 12. Voltage heat map and voltage tree comparison for two DOPF algorithms implemented on the IEEE 123 bus test feeder at time step with peak solar irradiance.

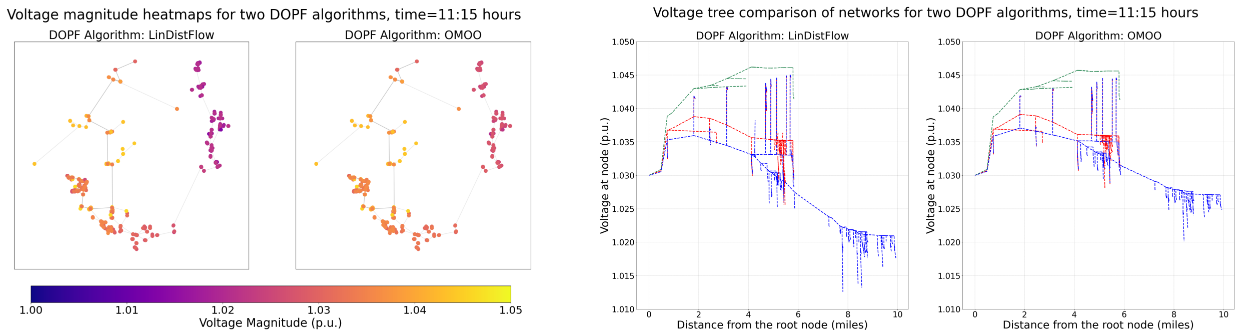
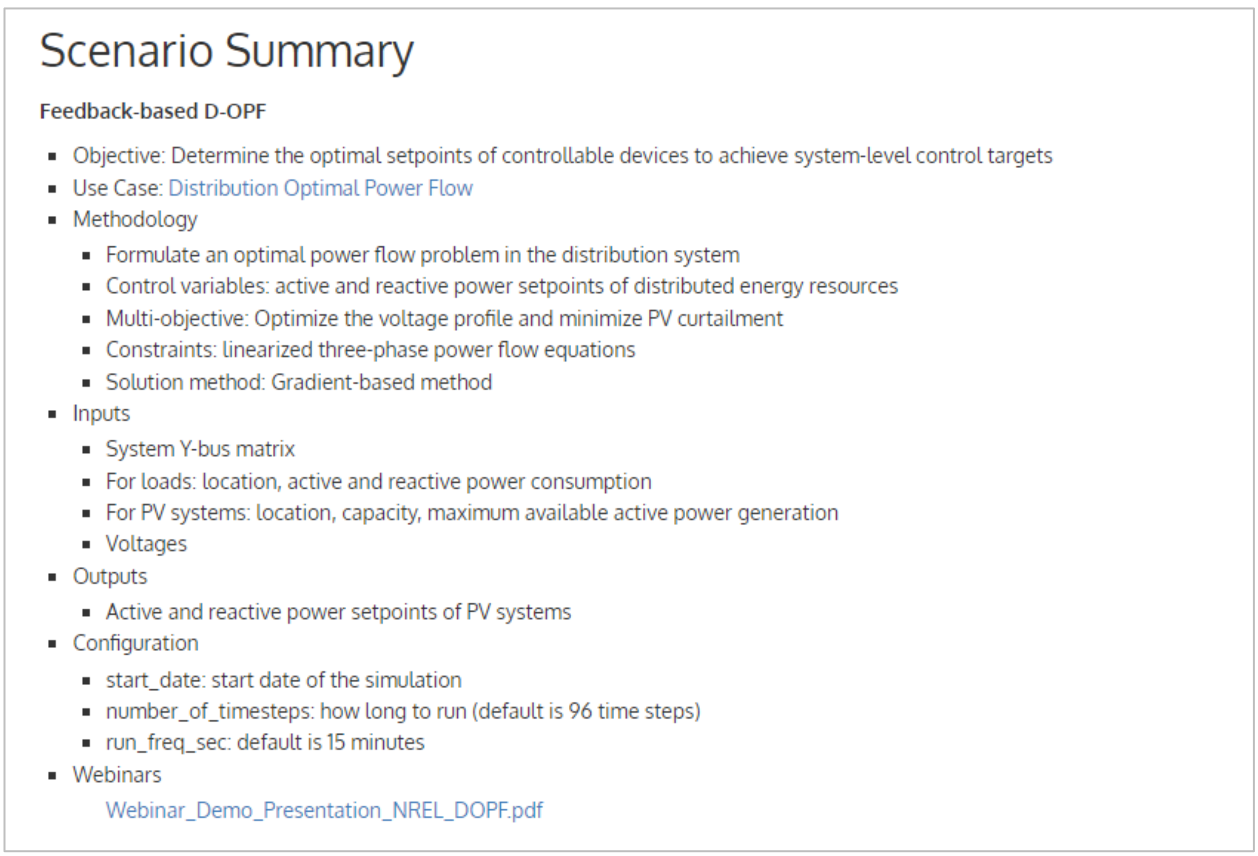


Figure 13. Voltage heat map and voltage tree comparison for two DOPF algorithms implemented on an example SMART DS test feeder at time step with peak solar irradiance.

4.2 Feedback-based Distributed Optimal Power Flow Scenario

The DOPF algorithm developed by NREL is based on a projected-gradient method, suitably modified to accommodate appropriate measurements [5]. The dynamic optimization method essentially leverages feedback from the system using the developed bilevel control paradigm. Figure 14 provides the screenshot of this scenario from the OEDI SI website.



Scenario Summary

Feedback-based D-OPF

- Objective: Determine the optimal setpoints of controllable devices to achieve system-level control targets
- Use Case: Distribution Optimal Power Flow
- Methodology
  - Formulate an optimal power flow problem in the distribution system
  - Control variables: active and reactive power setpoints of distributed energy resources
  - Multi-objective: Optimize the voltage profile and minimize PV curtailment
  - Constraints: linearized three-phase power flow equations
  - Solution method: Gradient-based method
- Inputs
  - System Y-bus matrix
  - For loads: location, active and reactive power consumption
  - For PV systems: location, capacity, maximum available active power generation
  - Voltages
- Outputs
  - Active and reactive power setpoints of PV systems
- Configuration
  - start_date: start date of the simulation
  - number_of_timesteps: how long to run (default is 96 time steps)
  - run_freq_sec: default is 15 minutes
- Webinars

  Webinar_Demo_Presentation_NREL_DOPF.pdf

Figure 14. Screenshot of NREL's DOPF scenario summary from the OEDI SI website.

5.0 Transient Data and Algorithms Use Cases

DERs have significant effects on distribution networks, both in terms of their opportunities and challenges. While DERs bring about numerous benefits, their integration requires careful planning, technological advancements, and policy frameworks to ensure a reliable, resilient, and efficient distribution network that can effectively harness the potential of DERs. In 2018, IEEE Standard 1547 [6] was revised to, among other things, allow DER to ride through voltage and frequency disturbances. This amendment [7] can have significant effects on transient studies that are conducted for distribution network. It impacts system stability study, grid recovery and resynchronization, fault ride-through analysis, protection and co-ordination study, advanced control strategy, inclusion of DERs in transient models of distribution feeders, microgrid and islanding issues, and many more.

The objective of transient-based use cases is to provide point-on-wave (POW) transient data and algorithms to perform various transient based studies to understand the effects of DERs. Under this task, we have developed two use cases: data generation and event detection and identification algorithm. The data generation use case can be standalone to generate transient POW data for any transient application. It can also be used as a prerequisite for the event detection and identification use case. Scripts are provided to generate additional data (labels for the dataset) and to post-process the data into various formats required by the algorithm. The event detection and identification use case provides two ML-based scenarios for different applications.

5.1 Data Generation Use Case

Data generation is the foundation of the complete transient use case and provides the details about data generation for transient algorithms using ATP as shown in **Error! Reference source not found.**.

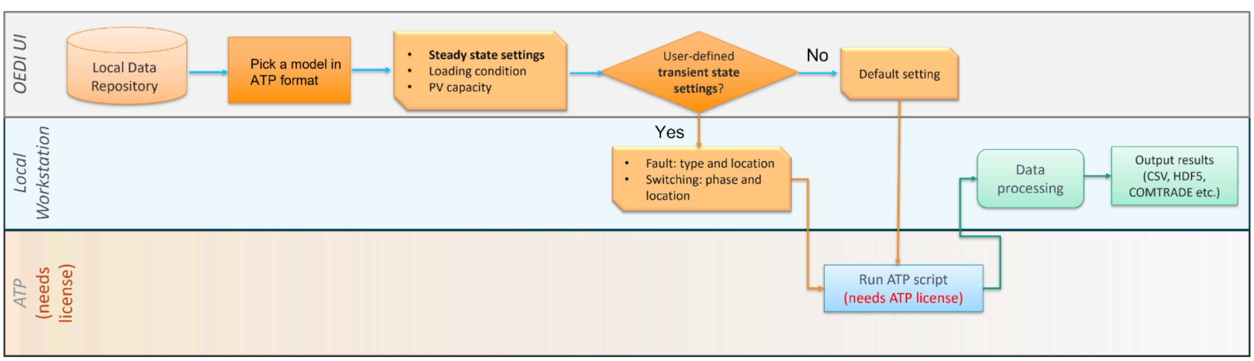


Figure 15. Overview of transient dataset generation use case.

This use case provides relevant scripts and information to generate transient data to perform various transient-based studies to understand the effects of DERs. Figure 16 provides the screenshot of the use case on the OEDI SI website. Under this use case there is a scenario created by PNNL.

Use Case Summary

Transient Data Generation

The increasing integration of Distributed Energy Resources (DERs) offers numerous benefits, including emissions reductions, grid resilience, and enhanced energy independence, it also presents challenges related to grid stability, operational complexity, and regulatory adaptation that must be addressed to realize the full potential of DERs in distribution networks. In 2018, IEEE Standard 1547 [1] underwent revisions, including provisions permitting DERs to ridethrough voltage and frequency fluctuations [2]. This amendment has significant implications on transient studies conducted for distribution networks. It impacts system stability study, grid recovery and resynchronization, fault ride through analysis, protection and co-ordination study, advanced control strategy, inclusion of DERs in transient models of distribution feeders, microgrid and islanding issues and many more [3, 4]. Hence, there is a need for transient data to perform various transient based studies to understand the effects of DERs. Transient data generation use-case is the foundation for the scenarios in event detection and identification use case. It provides the details about POW transient data generation for distribution feeder models. This use case provides sufficient training/validation dataset for the follow-up transient analysis algorithms. To generate the dataset for a specific distribution model with certain level of PV penetration, the user needs to pick a model in ATP format from the OEDI SI repository. The test model originates with different steady state settings, including the loading condition and PV capacity, which forms multiple ATP net files. Then the user has the option to modify the transient state settings or not. Documentations are provided for the user to make edits to transient state settings. After all the cases are designed and prepared, the ATP simulation will generate the datasets in the corresponding ATP format. Lastly, the user has the option to implement data processing step and then output the dataset in required data formats.

Scenarios

Data Generation for CNN Based Protection Zone Identification

Figure 16. Screenshot of transient data generation use cases from the OEDI SI website.

5.1.1 Data Generation for CNN-Based Protection Zone Identification Scenario

This scenario helps to generate POW transient data for any distribution feeder model available in the ATP format. This scenario is intended to generate datasets (including the labels) that can be directly used for CNN-based protection zone identification scenario under the event detection and identification use case. Figure 17 provides the screenshot of this scenario from the OEDI SI website.

Scenario Summary

Data Generation for CNN Based Protection Zone Identification

- Objective: Provide scripts and documentation to generate transient POW data using Alternate Transients Program (ATP) software.
- Use Case: Transient Data Generation
- Methodology
  - This scenario provides Python scripts to generate transient POW data using Alternate Transients Program (ATP) software. Distribution feeder models in ATP format is used as the input. Documentations are provided to help users to edit steady state and transient settings of the feeder model. Scripts to post-process the data into CSV, HDF5 and COMTRADE are available. The dataset generated using this scenario can be used to train and validate ML model presented in CNN Based Protection Zone Identification scenario.
- Inputs
  - Distribution feeder model in ATP format. IEEE 123 bus model is available here https://github.com/pnnl/oedisi_transients
- Outputs
  - ATP generates the data in its native format .pl4. Python scripts are available to post-process the data into CSV, HDF5 and COMTRADE formats.
- Configuration
  - The scripts are generalized to be used for any distribution feeder model in ATP format.

Figure 17. Screenshot of data generation for CNN-based protection zone identification scenario from the OEDI SI website.

PNNL has demonstrated this scenario for IEEE 123 bus model. The relevant python scripts and ATP network files to generate the data are available in the repository https://github.com/pnnl/oedisi_transients. The scripts can be easily edited to generate the data for any other feeder model. Users have the option to select different steady-state settings for different loading conditions and solar PV capacity. The base case (default) ATP file uses 90% loading, and 100% PV capacity is labeled IEEE123_PV_net.atp. Table 1 provides details of the various ATP steady-state cases available to the user for the IEEE123 bus network.

Table 1. IEEE123 bus circuit steady-state configurations.

ATP File	Total Load Multiplier	Total Solar PV Capacity
IEEE123_PV_net.atp	90%	100%
IEEE123_PV_L1C1_net.atp	36%	20%
IEEE123_PV_L1C2_net.atp	36%	100%
IEEE123_PV_L2C1_net.atp	63%	20%
IEEE123_PV_L2C2_net.atp	63%	100%
IEEE123_PV_L3C1_net.atp	90%	20%
IEEE123_PV_L3C2_net.atp	90%	100%

In this scenario, we have provided an option for the user to select preferred simulation transient settings, i.e., fault angles, types and locations or select the default settings. These parameters may be reconfigured via manipulation of the python script *ATPLoopFaults_all_IEEE123_PV.py* or left as-is for a default case. This python script is designed to call ATP and runs all fault types (L-G, LL-G, LLL-G) in all zones of the network. The user must set directory paths to the ATP files using `atp_path` for the successful execution of the script. The output files will be in .pl4 format.

The python script checks the phase count of the bus that is selected to simulate different fault cases and runs a maximum of seven fault cases (if its three phase – LLL-G, LL-G, and L-G for each phase combination), three cases if the bus selected contains only two phases, or a minimum of one fault case if the bus selected is single phase only. An example of the ATP transient waveform plots for a test case run on bus 51 (three-phase node) using the script ATPLoopFaults_all_IEEE123_PV.py are shown in Figure 18, Figure 19, Figure 20. As noted above, a three-phase fault is conducted initially, followed by three double line-to-ground faults for each phase combination (only BC shown) and three single line-to-ground faults for each phase (only A shown). Output files are plotted below using PlotXY.

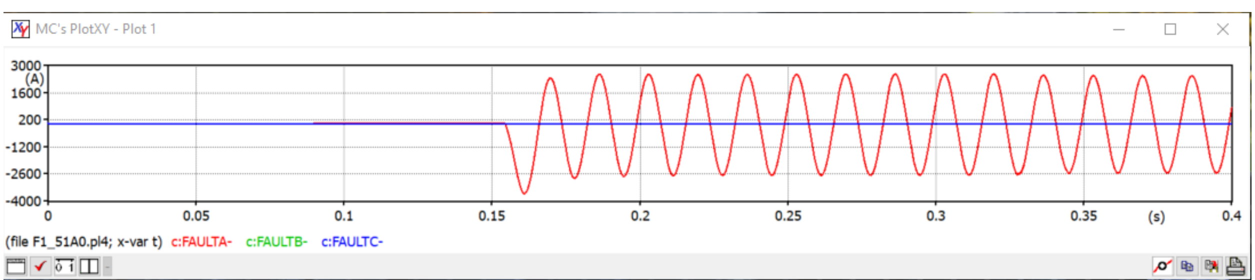


Figure 18. Single line-to-ground fault phase A.

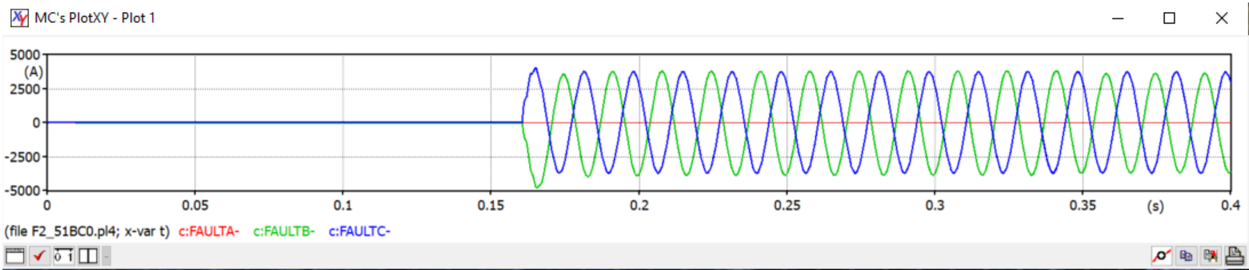


Figure 19. Double line-to-ground fault phases B, C.

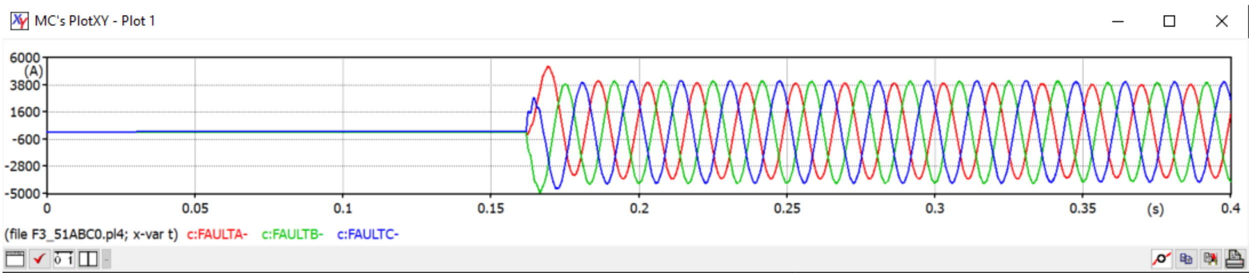


Figure 20. Three phase line-to-ground fault.

After the transient simulations have been completed, the user has the option of converting the ATP output files in *.pl4* format to point-on-wave COMTRADE files or compressed CSV via .mat file conversion. Python scripts to achieve the user-preferred data format are provided in the scenario. After the transient data generation and conversion, the POW data can be used for any application. If the user wants to use the data for fault identification and protection zone classification algorithm, then it is essential to rearrange the data in a particular format along with the labels for supervised training.

5.2 Event Detection and Identification Use Case

Event detection and identification, shown in Figure 21, develops transient algorithms using machine learning (ML) techniques. This use case provides a containerized data-driven algorithm that takes the dataset on workstation and trains the transient algorithm inside the docker container. Upon the completion of training and testing, trained model, training and testing results, and plots will be copied from the docker container to the local station. Two data-driven algorithms are available for this use case:

- Convolutional neural network (CNN)-based protection zone identification algorithm was developed by PNNL. It uses a CNN to classify fault locations by protective zone in distribution systems with 100% penetration of PV.
- Artificial intelligence (AI)-based event identification algorithm was developed by ORNL. This algorithm extracts the features from an input dataset using discrete wavelet transform. Then multiple ML algorithms are used to classify the events. These algorithms are available on OEDI SI website.

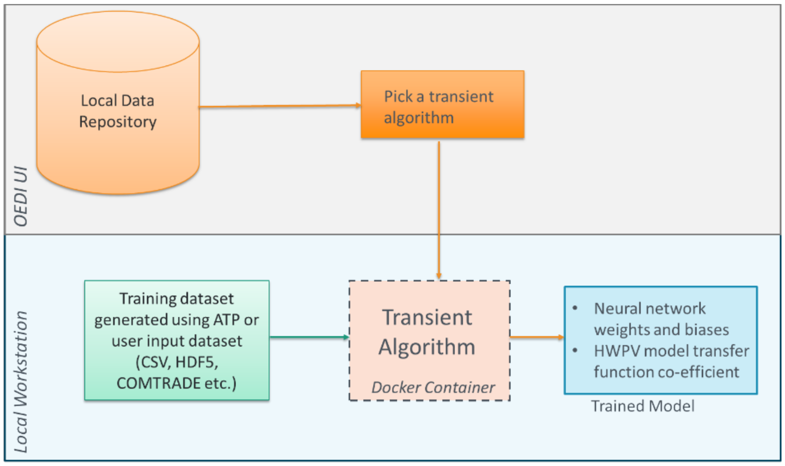


Figure 21. Overview of event detection and identification use case.

This use case provides two learning-based event detection and identification scenarios prepared by PNNL and ORNL. Figure 22 provides the screenshot of the use case on the OEDI SI website.

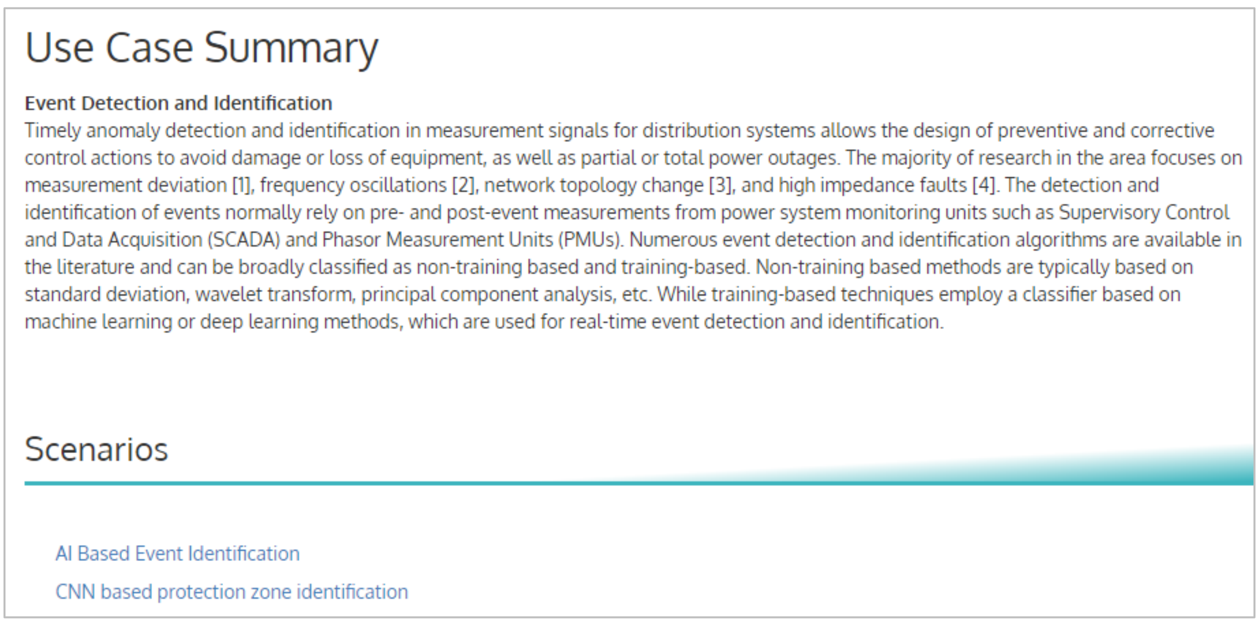
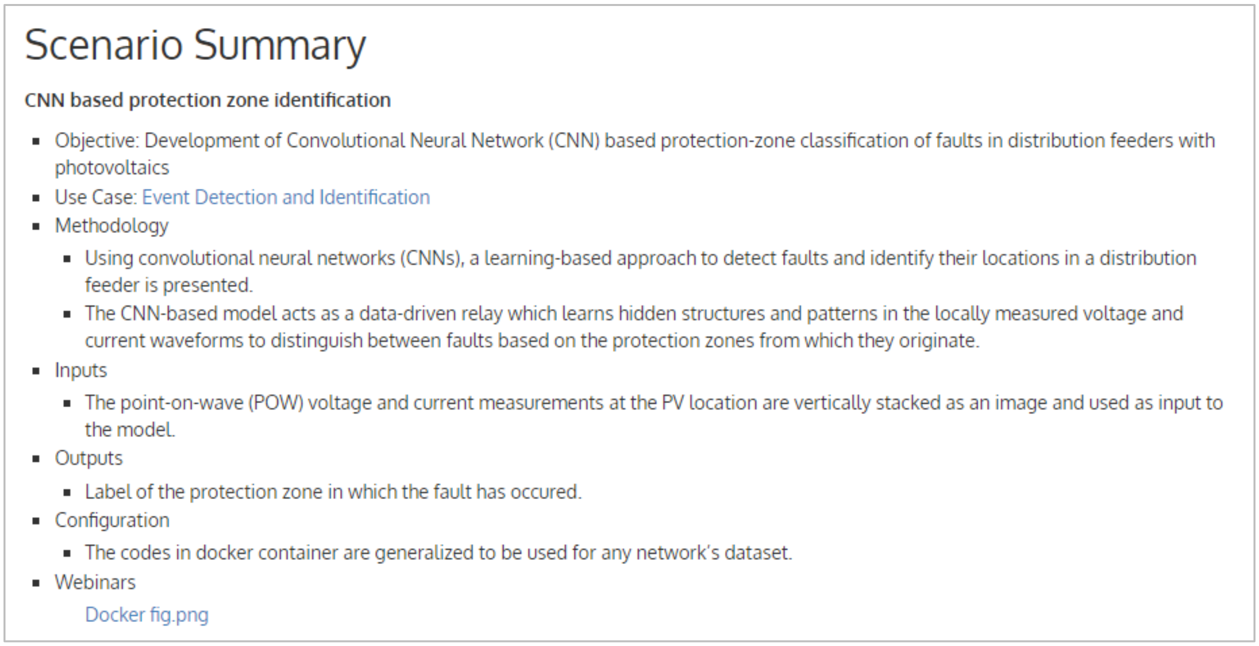


Figure 22. Screenshot of event detection and identification use cases from the OEDI SI website.

5.2.1 CNN-Based Protection Zone Identification Scenario

This scenario presents a data-driven protection algorithm. It uses CNN to classify fault locations by protective zone in distribution systems with 100% penetration of PV [8, 9]. Figure 23 provides the screenshot of this scenario from the OEDI SI website. Python scripts, documentation, and ATP network files are available in the repository https://github.com/pnnl/oedisi_transients.



Scenario Summary

CNN based protection zone identification

- Objective: Development of Convolutional Neural Network (CNN) based protection-zone classification of faults in distribution feeders with photovoltaics
- Use Case: Event Detection and Identification
- Methodology
  - Using convolutional neural networks (CNNs), a learning-based approach to detect faults and identify their locations in a distribution feeder is presented.
  - The CNN-based model acts as a data-driven relay which learns hidden structures and patterns in the locally measured voltage and current waveforms to distinguish between faults based on the protection zones from which they originate.
- Inputs
  - The point-on-wave (POW) voltage and current measurements at the PV location are vertically stacked as an image and used as input to the model.
- Outputs
  - Label of the protection zone in which the fault has occured.
- Configuration
  - The codes in docker container are generalized to be used for any network's dataset.
- Webinars

  Docker fig.png

Figure 23. Screenshot of CNN-based protection zone identification scenario from the OEDI SI website.

The algorithm is demonstrated on single-phase PVs in IEEE 123 bus model and three-phase PV in EPRI J1 model and the scripts are containerized. This work is available in the proceedings of IEEE GreenTech Conference 2024 [10]. This use case provides docker containers for single-phase and three-phase PVs. The container (Figure 24) takes a dataset on workstation and trains the CNN algorithm inside the docker container. Upon completion of training and testing of CNN, the trained model, plots and training, and testing results will be copied from docker container to local station.

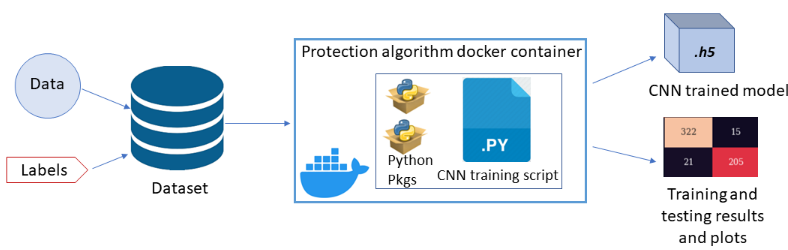


Figure 24. Docker container for event detection use case.

5.2.2 AI-Based Event Identification Scenario

This algorithm, developed by ORNL, is a data-driven event identification algorithm based on the periodic waveform detector [11]. The algorithm extracts the features from an input dataset using discrete wavelet transform, then multiple ML algorithms are used to classify the events. Figure 25 provides the screenshot of this scenario from the OEDI SI website. All the scripts, documentation, and data for this scenario are available at https://github.com/openEDI/oedisi-transient .

Scenario Summary

AI Based Event Identification

- 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
- Use Case: Event Detection and Identification
- Methodology
  - Event detection algorithm is based on the periodic waveform detector. Its basic idea is to calculate the voltage magnitude difference between two adjacent cycles twice to remove the off-nominal frequency and periodic characteristics. A threshold is designed to detect the fault based on the distribution of the detector.
  - Event identification algorithm first utilizes discrete wavelet transform to extract the features, and then use multiple machine learning algorithms to classify the events.
- Inputs
  - Detection input: Measurements of feeder head voltage waveforms
  - Detection input: Measurements of feeder head current waveforms
  - Detection input: Pre-selected detection threshold
  - Identification input: Detected event data period
- Outputs
  - Detection output: 0 (No event) / 1 (Pass data during event period)
  - Identification output: Identified event type as 0 (single-phase) / 1 (three-phase) / 2 (phase-to-phase)
- Configuration
  - The user is not required to manipulate the internal contents of this image.
  - To run the image, the user needs to follow the instructions.
  - The user can test multiple identification algorithms by running different blocks of code in the Jupyter Notebook.
- Webinars

  Webinar_Demo_Presentation_ORNL_Transient.pdf

Figure 25. Screenshot of AI-based event identification scenario from the OEDI SI website.

6.0 Conclusion and Future Work

The report highlights the development of the OEDI SI platform to tackle the challenges encountered in electric power systems analysis, particularly in integrating renewable energy and local inverter-based controls within modern distribution systems. It offers a repository for algorithms capable of executing complex tasks such as distribution state estimation, distribution optimal power flow, and event detection efficiently. The platform allows for easy integration with a data lake consisting of several well curated and diverse test feeders. This report highlights the different use cases that were curated around the scenarios described in Section 3.0, 4.0 and 5.0.

The platform is continuously going through development and improvement to make a broader set of solar systems integration research more reproducible, robust and replicable. As distribution power systems evolve, unreported behind-the-meter PV panels can challenge the operation of the grid. A state estimation –based PV detection algorithm is being developed that can identify the behind the meter PVs with some certainty. With the capability to solve larger and complex electric power grid it is essential for optimal power flows to use distributed computing features to be integrated in its development. A distributed method to solve DOPF is currently in the works to accomplish such tasks. And finally, a learning-based algorithm for transients is being integrated for increased penetration of DERs using transient POW data. Currently the use of the OEDI SI framework requires a deep knowledge of how the federates need to be developed and connected through component definitions files and system files. A graphical user interface (GUI) is under development to expedite the generation of scenarios using available algorithms and models found withing the OEDI SI database. The GUI will also allow users to run already configured scenarios and view their results without any knowledge of the underlying OEDI SI framework.

References

[1] B. Palmintier, D. Krishnamurthy, P. Top, S. Smith, J. Daily, and J. Fuller, “Design of the HELICS high-performance transmission-distribution-communication-market co-simulation framework,” in 2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), pp. 1–6, 2017

[2] F. B. dos Reis, A. P. Reiman, and G. D. Black, “Distributed multi-area state estimation for distribution systems,” in 2022 IEEE Power Energy Society General Meeting (PESGM), 2022, pp. 1–5.

[3] Newton Raphson based 3-Phase Decoupled DSSE, Available at: https://openei.org/wiki/OEDI-SI/Scenarios/Newton_Raphson_based_3-Phase_Decoupled_DSSE

[4] Sadnan, Rabayet, Poudel, Shiva, Mckinsey, Joseph, and Slay, Tylor. 2023. "Linearized Distribution Optimal Power Flow for OEDI SI". United States. https://dx.doi.org/10.25984/2202772. https://data.openei.org/submissions/5972.

[5] A. Bernstein and E. Dall’Anese, “Bi-Level Dynamic Optimization with Feedback,” the 5th IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, Quebec, Canada, Nov. 2017. https://ieeexplore.ieee.org/document/8308704

[6] “IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems,” IEEE Std 1547-2003, pp. 1–28, 2003.

[7] “IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces,” IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003), pp. 1–138, 2018.

[8] T. McDermott, M. Ramesh, N. Shepard et al., “Protection of distribution circuits with high penetration of solar PV,” Pacific Northwest National Laboratory, Tech. Rep. PNNL-32230, 2021.

[9] T. McDermott, S. Meliopoulos, M. Ramesh, G. Cokkinedes, J. Doty, J. Kolln, N. Shepard et al., “Protection of Radial Circuits with High Penetration Distributed Photovoltaics,” Proceedings of the 7th Annual Georgia Tech Protective Relaying Conference, April 28-April 30, 2021, Virtual., pp. 1–5, 2021.

[10] M. Ramesh, K. Chatterjee, D. Glover, J. Follum, T. McDermott, A. Reiman,"Convolutional Neural Network-Based Protection-Zone Classification of Faults in Distribution Feeders with Photovoltaics" accepted for IEEE GreenTech 2024.

[11] AI Based Event Identification, Available at: https://openei.org/wiki/OEDI-SI/Scenarios/AI_Based_Event_Identification

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov