

EUDP IEA Task 41

DELIVERABLE 2.2 – PYTHON LIBRARY DOCUMENTATION FOR
OPEN SOURCE DISTRIBUTION NETWORK DATA WITH LARGE
SHARE OF WIND

AEISHWARYA BAVISKAR, ANCA D. HANSEN

IEA Task 41: Denmark D2.2 for EUDP

Python Library Documentation

Report 2025

By

Aeishwarya Baviskar, Anca D. Hansen

Copyright:	Reproduction of this publication in whole or in part must include the customary bibliography citation, including author attribution, report title, etc.
Cover Photo:	None
Published by:	DTU, Department of Wind and Energy Systems, Fredriksborgvej 399, Building 118, 4000 Roskilde, Denmark www.windenergy.dtu.dk
ISSN:	[0000-0000] (electronic version)
ISBN:	[000-00-0000-000-0] (electronic version)

CONTENT

Summary

1. Introduction

1.1. EUDP IEA Task 41: Distributed Wind

2. Description of the Python Library

2.1. Brief overview of the dataset

2.2. Key features of the library

2.3. Library Functions

2.4. Installing the Python Library

2.5. Using the Python library

SUMMARY

This report constitutes part of the work package deliverable D2.2 of the EUDP IEA Task 41 project funded by Energy Technology Development and Demonstration Program (EUDP) of the Danish Energy Agency. The report presents a brief snapshot of the python library published to extract and modify the open source distribution network with large share of wind.

1. INTRODUCTION

This report has been produced with the intention of giving an overview of the python library published to extract and modify the open-source distribution network dataset with a large share of distributed wind and solar power plants. The report is providing first an overview of the salient features of the dataset, and functions in the library to download and use the dataset. The distribution network model is a 7000 bus network spanning across three voltage levels 60-10-0.4 kV. The distribution network model embodied with a large share of RES, with generation time-series simulated from meteorological data is an open-source network model. It is entitled as the DTU 7k Bus Active Distribution Network (DTU-ADN) and is accompanied by load time-series aggregated at 60 kV, 10 kV and 0.4 kV. This active distribution network model, flexible to furthermore incorporate other assets such as electric vehicle charging stations, storage, can be used to study, optimize, and control the effects of weather-dependent generation and other network assets in future distribution networks. This report presents an overview of the dataset and the accompanying python library to access the dataset.

1.1. EUDP IEA Task 41: Distributed Wind

The overall objective of this project is to identify and explore studies of Danish interest of Distributed Wind (DW) for cost effective technology development and integration into a continuously evolving energy system. This is done by collaborating and contributing to the IEA Wind TPC Task 41 international activities on DW turbine technology development and assessment in a series of dedicated work packages (WPs). IEA Wind TPC Task 41 is an international network centered on international collaboration and coordination in the field of DW. The purpose is to accelerate the development and deployment of DW technology as one of the leading generation sources in global renewable markets, the facilitation of easier and faster DW integration into electrical grids, increasing thus the competitiveness of wind and accelerating the replacement of fossils fuels. The IEA collaboration is enforced partly by exchange of information, sharing of results, and conducting analyses and explorative studies in the form of reports and publications and partly by implementing a strong cross IEA Wind TPC Tasks collaboration effort.

2. OVERVIEW OF THE DATASET

The DTU 7k Bus Active Distribution Network (*DTU-ADN*) is a completely open-source multi-voltage network model. It is accompanied by load time-series aggregated at 60 kV, 10 kV, and 0.4 kV. The network topology for the DTU-ADN are generated from geographical data and represent real distribution networks. The objective behind the development of the DTU-ADN is developing control and optimization algorithms for active management of future distribution network with a large share of renewable energy resources (*RES*). The following table gives key features for the network model

The *DTU-ADN* spans across 7000 buses. However, the network dataset is modular and can be used to create multiple network combinations with varying number of buses. The dataset is designed in independent blocks of sub-networks which can be connected to the main 60 kV network to create a larger 2 voltage level network, i.e. 60 kV-10 kV network, or a three voltage level network, i.e. 60 kV-10 kV -0.4 kV. It is also possible to use the 10 kV network as the base network and connect it to one or multiple 0.4 kV networks. However, each modular network dataset block has its own unique place to be connected in the larger network. This is because the loading conditions and the time-series for each nodes of the network should be respected.

2.1. Contents of the dataset

The dataset collection contains three type of datasets that together describe the network and the load profiles. The three types of datasets are,

1. Grid Topology: Describes the physical network connections and locations for the loads, transformers and lines.
2. Load and Generation Time-Series Data: Time-series data with 1 hour resolution detailing the load and generation profiles at particular buses. The load and generator time-series are in per unit.
3. Load and Generation Type Data: These tables list the type of load connected at each bus nodes and the generator types at each generator node. The type of bus loads can be household, industrial, agricultural. The different type of generator nodes are wind and solar photovoltaic.

The distribution network spans across three voltage levels. The 60 kV network is the medium voltage network that is a fundamental building block for the network. Figure 1. shows the 60 kV network topology. The 10kV-400V medium voltage networks may be

connected to the 60kV medium voltage network to build a large network depending on the case-study.

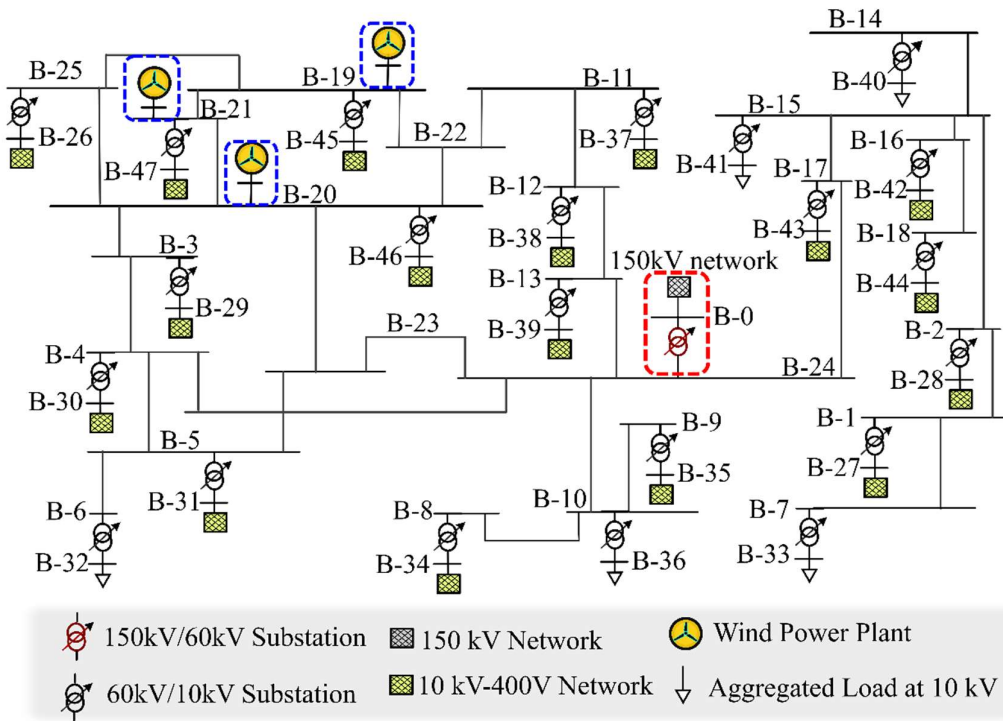


Figure 1. 60 kV medium voltage network topology

In total 17 of the 10kV-400V medium voltage networks can be connected to the main 60 kV network.

2.2. Key Features of the library

The python library is designed to access, extract, and modify the 7000 bus active distribution network. Some of the key features of the library are as follows,

- Extracts relevant network topology data
- Builds the network data in the required format
- Functions to connect one or more 10kV-400V networks to the main 60 kV network
- Function to update the network with load and generation information at a given time-stamp
- Runs load flow on the entire 60kV network

2.3. Installing the Python Library

How to install the python library:

Run the following command in the command prompt or Anaconda prompt:

```
pip install dtu-adn
```

The above command will install the complete dataset and the functions. The dataset can directly be accessed from the library functions.

2.4. Using the library – Example

This section provides two examples to use the library showcasing the main functions. Two examples are explained which use the most important functions from the library. Example 1 demonstrates the procedure to load the main building block of the network dataset, i.e. the 60 kV network, and Example 2 demonstrates how to expand the network with 10kV-400V networks.

2.4.1 Example 1: Powerflow on 60 kV network

The python library is imported with the command `'import dtu-adn'`. The subroutine `'DTU_ADN'` contains the required functions to load the network topology data. Next, the function `'DTU_ADN'` imported as `'network_class'` is used to load the 60 kV network topology in the variable `'net_dict'`. The network dictionary consists of all the necessary datasets to define the 60 kV network. Furthermore, `net_dict` is an instance of the the class `network_class`, hence all the functions defined for `network_class` are inherited by the class instance. The function `'network_class.gen_and_denmand_net_60(t)'` loads the generation and demand data in the `bus` and `gen` datasets for the 60 kV network. The function takes the time variable `t` as an argument. To run the power flow, `network_class` has an inbuilt `runpf` function which outputs the result of the power flow.

Example 1

```
#*****
# File: PowerFlow60kV.py
#.....
# Description:
#
# Step 1: Load the DTU_ADN 60 kV network.
# Step 2: Update the load and generation values for the
# 60 kV network corresponding to the timestamp
# Step 3: Run a power flow for the 60 kV network
#
# ---end of file
#.....
#
# Author: Aeishwarya Baviskar
# email: aeish@dtu.dk
#-----

# User Defined functions

import dtu_adn.DTU_ADN as network_class

net_dict = network_class() # One instance of the network class as a dictionary

# select a time-stamp for power flow
t0 = 1500

# load the generation and demand values at that time stamp
net_dict.gen_and_demand_net_60(t0)

# run power flow and save results in pf_result network dataset
pf_result = network_class.runpf(net_dict.net)

## *** end of file *** ##
```

2.4.2 Example 2: Expanding the 60kV network with 10kV-400V networks

The main focus for example 2 is connecting 10kV-400V network to the main 60kV network. An instance of the *network_class* class is invoked as *net_main*. The array of network nodes provides all the nodes where the 10kV-400V networks can be connected. The method for *network_class*, '*connect_10kV400V_network*' is used to connect 10kV-400 V network to the main 60kV network. The node at which the network should be connected is passed as an argument to the method. In example 2, two nodes are selected for expansion, namely node 37, and node 46. To load the load and demand generation data for the multi-voltage level network, the method '*gen_and_demand_net*' is used with the timestamp as an argument to the method. Same class method '*runpf*' is used to run power flow in the multi-voltage network.

Example 2

```
#####
# File: PowerFlowMultiV.py
#.....
#
# Description:
# Step 1: Load the DTU_ADN 60 kV network.
# Step 2: Update the load and generation values for the 60 kV network corresponding to the timestamp
# Step 3: Run a power flow for the 60 kV network
#
# ---end of file
#.....
# Author: Aeishwarya Baviskar
# email: aeish@dtu.dk
#-----

# User Defined functions

import dtu_adn.DTU_ADN as network_class
import numpy as np
from pypower.api import poption

net_main = network_class() # One instance of the network class as a dictionary

## 10-0.4 kV network dataset
network_nodes = np.array([43, 39, 30, 38, 46,
                          26, 47, 45, 29, 37,
                          28, 31, 34, 35, 27,
                          42, 44])

print(f"Number of buses in the 60kV network: {len(net_main.net['bus'])}")
```

```

## Connect the 10-0.4kV network at node 37
net_main.connect_10kV400V_network(37)

print(f"Number of buses in the 60-10-0.4kV network with 10-0.4kV network at bus 37: {len(net_main.net['bus'])}")

## Connect another 10-0.4kV network at node
net_main.connect_10kV400V_network(46)

print(f"Number of buses in the 60-10-0.4kV network with 10-0.4kV network at bus 37 and 28:
{len(net_main.net['bus'])}")

# select a time-stamp for power flow
t0 = 1500

# load the generation and demand values at that time stamp
net_main.gen_and_demand_net(t0)

opt = ppoption(ENFORCE_Q_LIMS=1, VERBOSE=1, OUT_ALL=1)

# run power flow and save results in pf_result network dataset

pf_result = network_class.runpf(net_main.net, opt)[0]

## *** end of file *** ##

```

2.5. Resources

- Link to the Python Library: <https://pypi.org/project/dtu-adn/>
- Link to download the dataset: https://data.dtu.dk/collections/DTU_7k-Bus_Active_Distribution_Network/5389910/1

2.6. How to cite the dataset?

Baviskar, Aishwarya; Hansen, Anca Daniela; Das, Kaushik; Koivisto, Matti Juhani (2021). DTU 7k-Bus Active Distribution Network. Technical University of Denmark. Collection. <https://doi.org/10.11583/DTU.c.5389910.v1>.