2024 SOFTWARE MANUAL

Climate Risk Screening Tool (RiSc)

Open-Source Release v0.1

Climate Risk Screening Tool (RiSc)

Open-Source Release

3002028699

Software Manual, March 2024

EPRI Project Manager
E. Lannoye

EPRI
3420 Hillview Avenue, Palo Alto, California 94304-1338 ▪ USA
800.313.3774 ▪ 650.855.2121 ▪ askepri@epri.com ▪ www.epri.com

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

NOTE

For further information about EPRI, call the EPRI Customer Assistance Center at 800.313.3774 or e-mail askepri@epri.com.

Together…Shaping the Future of Energy®

ACKNOWLEDGMENTS

EPRI prepared this report.

Principal Investigators
E. Lannoye
J.C. Martin

This report describes research sponsored by EPRI.

This publication is a corporate document that should be cited in the literature in the following manner: *Climate Risk Screening Tool (RiSc): Product Subtitle goes here.* EPRI, Palo Alto, CA: 2024. 3002028699.

SOFTWARE DESCRIPTION

Climate is increasingly playing an important role in the operations and planning of the power system. Extreme events in the last years are becoming more frequent and less predictable. Accounting for all the climate variability and possible future climate scenarios in power system planning is one of the key challenges to build resilient power systems. As part of the EPRI's Climate Resilience and Adaptation Initiative (Climate READi). researchers have worked together with the industry to define a framework that help planners to build resiliency and adapt to future climate hazards. The Climate RiSc Tool is the result of a combined effort from: EPRI's collaboration on the ISO-NE Operational Impacts of Extreme Weather Events Key Project EPRI Resource Adequacy Initiative, and the already mentioned Climate READi.

Description

To reduce the future uncertainty around climate, planning tools need to consider sufficient weather variability and climate uncertainty to characterize tail. However, this is not always an easy task mainly due to limited computational resources. This highlights the need for an efficient platform that evaluates climate data together with asset vulnerabilities to find periods of high system risk to inform downstream planning areas.

The Climate RiSc Tool is a tool that provides a fast and approximate method for identifying time periods and events of potential high risk of energy shortfalls. It stores and processes large amounts of weather data to identify stressful extreme weather events on the power system. Its outputs can be used to facilitate climate informed capacity expansion processes, the scenario development for production cost models and the simulation of planning and operational power system studies such as adequacy and transmission studies. It also facilitates the understanding of system structural vulnerabilities across different capacity mixes and a potential to assess how climate change will impact the power system under different horizons and scenarios.

Benefits and Value

- Efficient platform to simultaneously evaluate large sets of forward-looking climate data and asset vulnerabilities to find periods of highest system risks

- Identification of extreme events will allow to have climate-informed power system planning to build resiliency and adaptation to future hazards that might pose a significant risk to the security of supply.

- Understanding the nature of system risks (e.g., weather drivers, structural vulnerabilities) is essential to prepare the power system and inform decision-makers to proactively work on mitigation plans or policies that prevent potential risks to power system.

Platform Requirements

The RiSc tool is a server-based application and therefore need to be deployed in a local Linux server with the following pre-requisites.

Software:

- Linux x86_64 (OS):
- SUSE Linux Enterprise Server 15 SP3
- Docker (version 20.10.17-ce) for containerized development and deployment of the application. Docker is a platform for developing, shipping, and running applications and its dependencies in containers. Key details:
    - Python 3.12-slim
    - Docker container “db”: PostgreSQL (14.2) open-source database management system.
    - Docker container “deft”: Risk screening tool API.

Hardware:

- On-premise VM
- 64GB of RAM. Note that RAM requirements becomes crucial to ensure screening across multiple weather years**,** models, scenarios and weather stations.
- 4 to 6 CPU cores.
- Hard disk >1TB
- Remote machine should be accessible through SFTP or “mount” to transfer heavy files (weather and load data) to the application.

Keywords

Extreme weather events, risk screening, resiliency, adaptation, climate scenarios, capacity expansion, generation and transmission adequacy.

CONTENTS

1 INSTALLATION PROCESS ... 1

System Requirements ... 1

Installing the Software ... 1

2 GETTING STARTED ... 4

Introduction ... 4

Input Data ... 4

Climate& Load Data ... 4

Asset Vulnerabilities ... 8

Power System Data ... 9

Tool configuration ... 10

Running a project ... 12

Server execution ... 12

Windows execution ... 12

Output Data ... 13

3 TROUBLESHOOTING ... 14

4 EXAMPLE WALKTHROUGH ... 15

Step 1: Data Collection ... 15

Climate & Load Data ... 15

Asset Vulnerabilities ... 17

Power System Data ... 17

Step 2: Tool configuration ... 18

Settings related to importing data ... 18

Settings related to screening process ... 19

Step 3: Tool execution ... 20

Windows execution ... 20

Server execution ... 22

Step 4: Results ... 24

5 Annex 28
Application diagram 28
Data Base 29

LIST OF FIGURES

Figure 1 : Docker compose file (deft.yml) – Postgres container 2
Figure 2 : RiSc tool initialization. 2
Figure 3 : Stop RiSc tool. 3
Figure 4 : input folder structure. 4
Figure 5 : Historical weather NetCDF format. 5
Figure 6 : Projected weather NetCDF format. 5
Figure 7 : Load file names 7
Figure 8 : Log of failed endpoint with errors 14
Figure 9 : Screenshot of error information 14
Figure 10 : Texas weather station location 16
Figure 11 : Texas load files names for 2021 target year (ERA5 model and scenario) 16
Figure 12 : Texas load profile for weather years 2010 to 2021 16
Figure 13 : Example of Texas asset vulnerabilities (or risk model) 17
Figure 14 : Example of Texas power system data 18
Figure 15 : Extract of input data file and folder names (config.yml) 18
Figure 16 : Extract of input data target location in the RiSc application (sysconfig.yml) 18
Figure 17 : RiSc repository 20
Figure 18 : Configuration file for window execution. 21
Figure 19 : Volumes in backend application. 21
Figure 20 : Windows execution results. 22
Figure 21 : PuTTy connection. 22
Figure 22 : Access the application environment (docker container). 23
Figure 23 : Check input data is being transferred. 23
Figure 24 : Config file for server execution. 23
Figure 25 : RiSc execution log. 24
Figure 26 : Texas summary RiSc results 24
Figure 27 : Texas detailed RiSc results 25
Figure 28 : Extract of aggregated capacity output file for Texas 2021 power system. 25
Figure 29 : Texas generation capacity out during 2021 Uri Storm (RiSc Results) 26
Figure 30 : Texas generation capacity out during 2021 Uri Storm (RiSc Results) 26
Figure 31 : Texas available supply generation and load during 2021 Uri Storm 27
Figure 32 : RiSc schema 28

Figure 33 : RiSc database schema. 29
Figure 34 : Example of database tables input data. 30

LIST OF TABLES

Table 1 : Weather station location format 4
Table 2 : Units in weather data files 6
Table 3: Load format... 7
Table 4 : Asset vulnerabilities format 8
Table 5 : Example of how to include the load in the asset vulnerabilities file 8
Table 6 : Units of variables used in asset vulnerabilities. 8
Table 7 : Power system data format.. 9

MANDATORY SOFTWARE INSTALLATION INFORMATION

Installation of EPRI Software at Client Site

This software uses third party software products, operating systems, and hardware platforms. Over time, security issues may be uncovered in these third-party products. You should review your use of this software with your Information Technology (IT) department to ensure that all recommended security updates and patches are installed to all third-party products when needed.

If you experience difficulties accessing the application

If you experience difficulties accessing the application after standard installation, please consult your IT department personnel to have proper access permissions setup for your use. If the problem cannot be resolved, please call the EPRI Customer Assistance Center (CAC) at 1-800-313-3774 (or email askepri@epri.com).

1 INSTALLATION PROCESS

Installing Climate Risk Screening (RiSc) Tool is a simple process and should normally take less than 15 minutes. These instructions will break this process into a few short steps. The process assumes a moderate amount of Linux knowledge.

System Requirements

The following items are required to use Climate RiSc Tool:

- Windows 10 and on-premise VM with SUSE Linux Enterprise Server 15 SP3 (64-bit).
- 1 TB free hard disk space.
- 64GB (or higher) of RAM. Processing times with many weather years and locations improves with more memory.
- 4 to 6 CPU cores.
- Python 3.10 (client side)
- Docker 20.10.17-ce

To run the application from your local machine, a Windows OS with Python 3.10 and access to remote server (hostname and port) where the tool has been deployed. In addition, a mount is required to transfer (heavy) weather and load input files to the VM..

Installing the Software

Note that you must have administrative privileges (use `sudo su` in bash) to install and uninstall Climate RiSc in your enterprise server.

The software will need to install programs and support files in a designated directory. Follow these instructions to install the software:

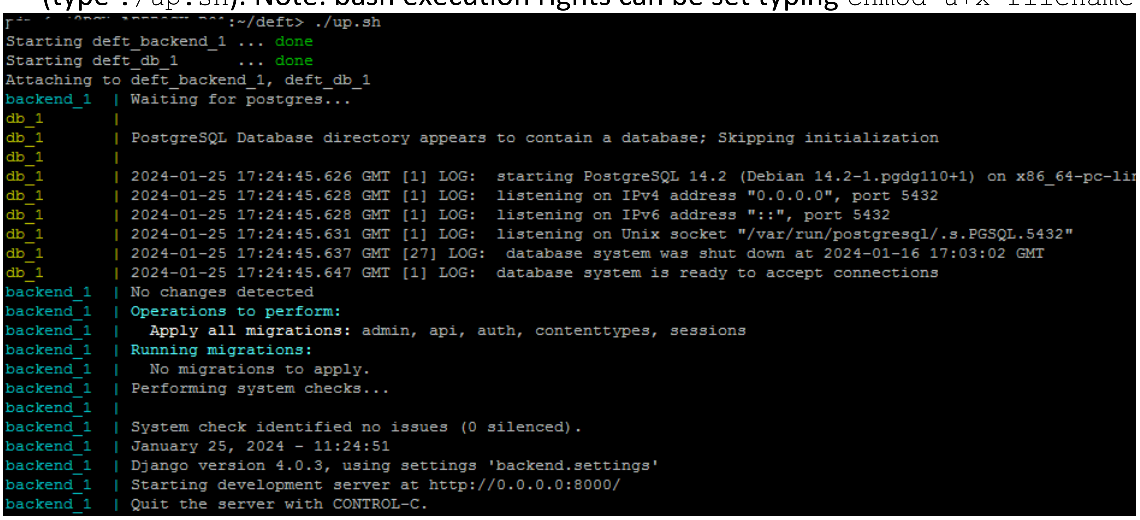
1. Download or clone existing GitHub repository (link to be provided from EPRI when available) to your Windows machine/ enterprise server.
2. Set-up the connection to your remote server (i.e., PuTTY)
3. Set-up the mount or SFTP to transfer files between your windows machine and the remote server.
4. Transfer the unzipped package to the server (Step 3) or pulling from GitHub directly.
5. Use a terminal multiplexer (i.e., tmux) in your server to handle multiple bash windows and open a new session. Type: `tmux attach-session -t 0`
6. In case the docker image is provided as a .tar file, open one tmux bash window and load the docker image in your server `docker load -i my_image.tar.` Once loaded update the deft.yml file image mapping accordingly (See Figure 2). Finally, make sure to change the environmental variables in the .env file before running the tool (Figure below)


```
services:
  db:
    image: postgres:14.11
    tty: true
    container_name: deft_db_1
    volumes:
      - db-data:/var/lib/postgresql/data
      - ./db/dumps:/dumps
    networks:
      - deft-network
    env_file:
      - ./.env
    ports:
      - "54322:5432"
    environment:
      - POSTGRES_DATABASE=deft
      - POSTGRES_USER= deft
      - POSTGRES_PASSWORD=${DB_PASSWORD}
      - TZ=Etc/Greenwich
      - PGTZ=Etc/Greenwich
    healthcheck:
      test: ["CMD", "pg_isready", "-U", "deft", "-d", "deft"]
      interval: 30s
      timeout: 10s
      retries: 3

  backend:
    image: risctool.azurecr.io/risctool:2.0
    tty: true
    networks:
      - deft-network
    env_file:
      - ./.env
```

Figure 1 : Docker compose file (deft.yml) – Postgres container

7. In other tmux bash window, go to the application folder where you have saved the GitHub package (type `cd + your application folder name`) and run the up.sh file (type `./up.sh`). Note: bash execution rights can be set typing `chmod u+x filename`



```
[illegible]:~/deft> ./up.sh
Starting deft_backend_1 ... done
Starting deft_db_1      ... done
Attaching to deft_backend_1, deft_db_1
backend_1  | Waiting for postgres...
db_1       |
db_1       | PostgreSQL Database directory appears to contain a database; Skipping initialization
db_1       |
db_1       | 2024-01-25 17:24:45.626 GMT [1] LOG:  starting PostgreSQL 14.2 (Debian 14.2-1.pgdg110+1) on x86_64-pc-li
db_1       | 2024-01-25 17:24:45.628 GMT [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
db_1       | 2024-01-25 17:24:45.628 GMT [1] LOG:  listening on IPv6 address "::", port 5432
db_1       | 2024-01-25 17:24:45.631 GMT [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
db_1       | 2024-01-25 17:24:45.637 GMT [27] LOG:  database system was shut down at 2024-01-16 17:03:02 GMT
db_1       | 2024-01-25 17:24:45.647 GMT [1] LOG:  database system is ready to accept connections
backend_1  | No changes detected
backend_1  | Operations to perform:
backend_1  |   Apply all migrations: admin, api, auth, contenttypes, sessions
backend_1  | Running migrations:
backend_1  |   No migrations to apply.
backend_1  | Performing system checks...
backend_1  |
backend_1  | System check identified no issues (0 silenced).
backend_1  | January 25, 2024 - 11:24:51
backend_1  | Django version 4.0.3, using settings 'backend.settings'
backend_1  | Starting development server at http://0.0.0.0:8000/
backend_1  | Quit the server with CONTROL-C.
```

Figure 2 : RiSc tool initialization.

8. In a new tmux window ("ctrl+b+c") execute the docker image (type: `docker exec -it deft-backend-1 bash`) and introduce your mount path in deft.yml file. Using `nano` command modify text highlighted in yellow below:

```
backend:
  image: ikimio/deft_backend:1.9
  tty: true
  networks:
    - deft-network
  volumes:
    - ./settings:/code/backend/backend
    - ./logs/backend.log:/code/backend/deft_error.log
    - ./nc_files/:/code/nc_files
    - ./risk_models/:/code/risk_models
    - ./power_systems/:/code/power_systems
    - ./power_system_csv_files/:/code/power_system_csv_files
    - ./load_profiles/:/code/load_profiles
    - ./risk_models_xlsx_files/:/code/risk_models_xlsx_files
    - ./risk_result_reports/:/code/risk_result_reports
    # - /mnt/hgfs/shared/input:/code/share
  ports:
    - "8000:8000"
```

Figure 1 : Docker compose file (deft.yml) – backend container

9. To close the application, go to the application folder execute the down.sh file (type `./down.sh`)

```
:~/deft> ./down.sh
Stopping deft_db_1      ... done
Stopping deft_backend_1 ... done
Removing deft_db_1      ... done
Removing deft_backend_1 ... done
Removing network deft_deft-network
```

Figure 2 : Stop RiSc tool.

2 GETTING STARTED

Introduction

The Climate RiSc Tool is a platform that provide a fast and approximate method for identifying time periods and events of potential high risk of energy shortfalls. It stores and processes big amounts of climate data to identify stressful extreme weather events on the power system. Its outputs can be used to facilitate climate informed capacity expansion processes, the scenario development for production cost models and the simulation of planning and operational power system studies such as generation and transmission adequacy studies. It also facilitates the understanding of system structural vulnerabilities across different capacity mixes and a potential to assess how climate change will impact the power system under different horizons and scenarios.

Input Data

The input data need to use the RiSc tool can be summarize in three blocks:

- Climate & Load Data
- Asset Vulnerabilities
- Power System Data

Data should be stored and transferred respecting the following folder structure:

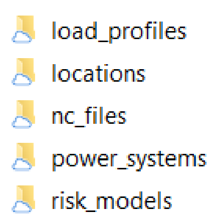


Figure 3 : input folder structure.

Climate& Load Data

Climate data used in RiSc tool must consist of historical and forward-looking climate data This data must be processed in pairs of historical and projected files with the same temporal (typically hourly) and spatial granularities. The spatial resolution for a given region is typically defined in advance based on main load centers or big existing or future wind and solar installations and reported in a csv file to be imported in the tool with the following format:

Table 1 : Weather station location format (Note: Station_0 must be defined)

index	StationName	lat	lon
0	Station_0	34.945	-102.279

1	Station_1	35.255	-101.187

Climate data files must be imported as a pair NetCDF files (historical and projected). Variables not contained in the projected file will be then taken from the historical file. To represent forward-looking climate, the tool uses Global Climate Models (GCMs) and Shared Socio-economic Pathways (SSP) scenarios. The structure and format of each file can be seen below:

```
<xarray.Dataset>
Dimensions:        (time: 105192, station: 26)
Coordinates:
  * time           (time) datetime64[ns] 2010-01-01 ... 2021-12-31T23:00:00
  * station        (station) int32 0 1 2 3 4 5 6 7 ... 18 19 20 21 22 23 24 25
Data variables:
    temperature    (station, time) float32 32.75 35.59 29.43 ... 79.71 79.03
    dewpoint       (station, time) float32 20.63 20.14 20.74 ... 63.95 63.95
    windspeed_10m  (station, time) float32 1.004 2.031 2.419 ... 3.43 2.529
    windspeed_100m (station, time) float32 1.402 2.103 4.735 ... 5.015 5.254
    mslp           (station, time) float32 1.03e+05 1.03e+05 ... 1.003e+05
    precip         (station, time) float32 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0
    snow           (station, time) float32 0.0 0.0 0.0 ... 1.237e-07 1.237e-07
    solar          (station, time) float32 32.33 0.0 0.0 ... 412.9 332.0 175.6
Attributes:
    latitude:   [34.945 35.255 34.05  30.95  32.1   32.36  27.12  27.32  33.7...
    longitude:  [-102.279 -101.187 -101.188 -102.2   -101.45  -100.34   -97.5...
```

Figure 4 : Historical weather NetCDF format.

```
<xarray.Dataset>
Dimensions:       (time: 105192, scenario: 1, station: 26, year: 1, model: 1)
Coordinates:
  * time          (time) datetime64[ns] 2010-01-01 ... 2021-12-31T23:00:00
  * scenario      (scenario) object 'demo'
  * station       (station) int32 0 1 2 3 4 5 6 7 8 ... 18 19 20 21 22 23 24 25
  * year          (year) int32 2021
  * model         (model) object 'ERA5'
Data variables:
    temperature  (scenario, year, station, model, time) float64 32.75 ... 79.03
    dewpoint     (scenario, year, station, model, time) float64 20.63 ... 63.95
Attributes:
    latitude:   [34.945 35.255 34.05  30.95  32.1   32.36  27.12  27.32  33.7...
    longitude:  [-102.279 -101.187 -101.188 -102.2   -101.45  -100.34   -97.5...
```

Figure 5 : Projected weather NetCDF format.

The process to generate the historical file can be summarized in 3 steps:

1. Find historical weather data that is representative of the location of interest and contains all relevant variables (e.g., ERA5, which can be found here: https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=overview)
2. Download data in the available format (ERA5 is available in netcdf format)
3. Format data using the format illustrated in Figure 4. Here the historical weather file is saved as a netcdf with separate Data Arrays for each variable.

The process to generate projected weather data requires more expertise dealing with climate models and data:

1. Use historical weather data that has been found for the historical weather file
2. Find historical and future simulations from selected GCMs and emissions scenarios (e.g., CMIP6 SSPs 1-2.6 and 3-7.0). GCM output is available raw (e.g., https://aims2.llnl.gov/search/cmip6/) or downscaled (e.g., ISI-MIP: https://data.isimip.org/ or NASA NEX-GDDP https://www.nccs.nasa.gov/services/data-collections/land-based-products/nex-gddp). See the CMIP6 data users guidance for more information (https://pcmdi.llnl.gov/CMIP6/Guide/dataUsers.html).
3. Estimate future weather data by shifting historical weather time series according to shifts in variables (e.g., air temperature) projected by GCMs.
4. Export data from the different GCMs and emissions scenarios to the projected weather file. Note: When appending GCMs and SSP scenarios in the projected file, user should avoid naming any of them "historical" as this will create an issue in the tool.

It's important to note that a list of historical weather variables have been pre-selected: temperature, dewpoint, wind speed 10m and 100m height, mean sea level pressure, precipitation, snow, and solar irradiance. However, user can include a different The units should be aligned with the table below:

Table 2 : Units in weather data files

Weather Variable:Units
temperature: F
windspeed_100m: m/s
windspeed_10m: m/s
precip: in
snow: in
mslp (or mean sea level pressure): Pa
solar: w/m^2
dewpoint: F

The values from the historical file will prevail unless there is a projected weather variable. In this case, projected weather values will overwrite historical ones during the execution of the tool.

The tool stores all the data in a PostgreSQL database. The database should store configuration values pre-filling some tables in the climate and load data importing process (e.g., weather variables, target year, models and scenarios). In case the user wants to check or modify these values, please see Annex: DB Configuration.

Aside from the weather variables, the study year and the GCMs and SSPs scenarios are taken from the configuration file (See Configuration). Please make sure that when you define these values, they are aligned with your climate data.

Load data is typically linked to weather conditions. RiSc tool uses system-wide hourly load forecasting data. By default, this data is associated with the zero-index weather station. Climate and load data are both used indistinctively against the screening rules defined in the asset vulnerabilities section.

The tool also incorporates two extra load scenarios to perform sensitivities. Load base plus/minus. These two load scenarios represent a +/-12days shift in the load profile to account for the different workday/weekend conditions. However, in many cases, user can simply set those to 0 or not provide these files. In this case user need to check there is no defined risk screening rule to avoid issues.

The load needs to be transfer to the risk screening tool thought the predefined channel (i.e., mount or SFTP) and keep the following file name convention:

- {*weathermodel*}_{*weatherscenario*}_{*targetyear*}.csv,
- {*weathermodel*}_{*weatherscenario*}_{*targetyear*}_plus.csv (optional file)
- {*weathermodel*}_{*weatherscenario*}_{*targetyear*}_min.csv (optional file)

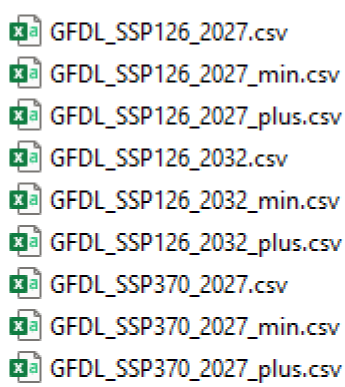


Figure 6 : Load file names

The format within the load files is as follows:

Table 3: Load format

Time	total_load
MONTH/DAY/WEATHER YEAR	Load[MW] – Distributed Generation (i.e, BTM PV)

Note, that there should a coherence between climate and load data time zones. If weather-coherent hourly load profiles are not available and cannot be synthesized , users can run the tool with zero-load hourly profiles as long as the user respect the format and input data requirements. However, results from the RiSc tool should be evaluated with care in this case.

Asset Vulnerabilities

Asset vulnerabilities refer to a set of IF-THEN rules that connect climate data with power system asset-specific derating factors. These bottom-up rules define climate hazards based on different weather conditions. The maximum number of weather variables is set to two (i.e., IF precipitation>0 and temperatures<32F THEN blade icing). These rules are defined in a csv file with the following format:

Table 4 : Asset vulnerabilities format

tech	variable1	operator1	threshold1	variable2	operator2	threshold2	derating	risk
PV	temperature	<=	32	precipitation	>=	0.001	90	Icing
WT_ON	temperature	<	32	precipitation	>=	0.001	20	Blade icing

Note that the weather variable name defined in this table needs to match with the one defined in the climate data files introduced earlier. Similarly, the technology/asset type names need to use the same naming convention as the power system data imported to the tool (See Power System Data).

Load can also be defined (optionally) in the asset vulnerabilities as it is treated as a special case. The load data will be treated similarly to another weather variable. This allows the user to customize the use case as needed (i.e., in systems with a wide gap between summer and winter peak loads, it can be used to offset this gap and have a set of extreme events for both heat/summer and cold/winter conditions)

Table 5 : Example of how to include the load in the asset vulnerabilities file

tech	variable1	operator1	threshold1	variable2	operator2	threshold2	derating	risk
Load	load_base	>=	32000	temperature	>=	65	100	Summer load
Load	load_base	>=	25000	temperature	>=	65	100	Winter load

The units used in this file are aligned with the weather data provided and specified in the table below:

Table 6 : Units of variables used in asset vulnerabilities.

Weather Variable:Units

temperature: F
windspeed_100m: m/s
windspeed_10m: m/s
precip: in
snow: in
mslp (or mean sea level pressure): : Pa
solar: w/m^2
dewpoint: F
Load: MW

Power System Data

The power system data refers to the data describing the generators in the system and contains essential information about each active asset. This information must include at least unique asset id, generator short name, operational date, retirement date (blank if no retirement foreseen), unit type, latitude, longitude, and nameplate capacity in MW. Note that FuelType, Area Code, Zone, and State information is optional. Users can expand the number of columns with additional attributes as they won't impact the execution of the tool. These power system data should be in a csv file with the following format:

Table 7 : Power system data format

Asset ID	Generator Shortname	Operational Date	Retirement Date	Unit Type	FuelType	Latitude	Longitude	Nameplate (MW)
35	35_832_p6251	01/01/1988	01/01/2100	NUC	NUC	28.79	96.04	1280
40	40_836_p10298	01/01/2014	01/01/2100	GT	NG	29.62	95.04	75

Latitude and longitude information from each asset is used to map the asset to the closest weather station. This allows the tool to have a representative proxy of the weather data to be used to evaluate the vulnerabilities on an asset-by-asset basis. In case this information is not available, user can directly use the weather station location of interest for each asset.

Note that this release has not included a sophisticated method to account for transmission risks. However, transmission assets can be also included. The tool treats transmission as a fictitious generator (i.e., if transmission between zone A and zone B is being impacted by a weather hazard the transfer capacity will be used as the nameplate capacity of a fictitious generator Trans_AB). This should allow the user to quantify the transmission and generation risks associated with weather to find events of interest.

Tool configuration

This release does not contain a Graphical User Interphase (GUI). Therefore, a configuration file is provided to the user to define and create projects. Before running the tool, a user needs to modify the configuration file accordingly:

- **Reset all**:
    - "yes": User can define a new project from the start. The database will be emptied, the weather data will be split by model and scenario (computational limits apply and weather data subsets need to be processed sequentially), and finally the weather and load data will be imported to the DB together with the station locations.
    - "no": User wants to reuse already imported weather and load data from a previous project (changes in the power systems and risk models are still allowed)
    - "sys": Advanced users can customize the reset using the sysconfig flags. For instance, user might have already split the weather data and just wants to skip this step from reset all.
- **Delete project_first**: "yes" or "no". Delete project with same name if already exists.
- **Project region**: Study region. It will be reflected in the project name.
- **Scenarios**: List of Shared Socioeconomic Pathways to study. Needs to be consistent with the SSP scenarios defined in the weather data.
- **Models**: List of GCM models to study. Needs to be consistent with the GCM model defined in the weather data.
- **Start year**: Weather year to start screening from included) ,i.e., for *1951* first screening day will be 01/01/1951
- **End year**: Last weather year to screen (excluded), i.e., for *2022* last screening day will be 12/31/2021
- **Weather stations in screening**: Number of weather stations defined in the weather data.
- **Target year**: Study year, i.e., year to which climate data is projected to (e.g., 2050)
- **Power system file**: This file needs to be located in the mount/input/power_system directory and file name has not a fix format (e.g., "input/power_systems/your_power_system_at_study_year.csv")
- **Risk model file**: "input/risk_models/your_asset_vulnerability_matrix.csv"
- **Aggregate capacity file**: "output/aggregated_capacity.csv". This file is generated by the tool. It's based on the mapping of each asset to the closest weather station and provides the aggregated capacity by station.
- **Event day interval**: event interval defined in days for screening (e.g., *21* to analyze 21-day events). Note that for yearly runs you need to set this variable and the event day gap to "yearly". This 'yearly" setting will compare all defined weather years as a whole.
- **Event day gap**: Represent the distance measured in days between events for screening . For instance, event day gap set to 7, it implies a 21-day- events shifted by one week -or two

weeks overlap- from one another. Note that for yearly runs you need to set this variable and the event day interval to "yearly".

- **Top events percentage**: Retain the worst top events as a percentage of the whole number of events evaluated (i.e., if top 10% in screening performed with 21-day events, 1 week shift, and weather years from 2020 to 2021, excluded, this will lead to 5 events retained)
- **Threshold MW**: Retain events where the average risk calculated per event is above this number (i.e., if set 50GW retained events should have an average risk above this number). Note that this can be used in combination with top events percentage. In case users don't want to use it, set to 0.
- **Sync files enable**: "yes" or "no". Typically used for windows execution where no interaction with the remote machine (and docker container) is needed. It is used to sync input data located in the mount with remote server (docker container). If set to "no", parameters below are disregarded.
- **Sync type**: "local" or "remote_share". "local" is used when executing on the server directly. "remote_share" is used when syncing from windows machine. Note that input data need to be in the dedicated mount target folders.
- **Src_predictive**: projected weather NetCDF file (all scenarios and models). It must be located in the mount within folder nc_files.
- **Src_historical**: historical weather NetCDF file (all scenarios and models). It must be located in the mount within folder nc_files.
- **Weather station file**: weather station location file (csv). It must be located in the mount within the folder locations.
- **Load folder name**: load folder containing the load files. It must be located within the folder load_profiles.

System settings are placed in a different configuration file called sysconfig.yml:

- **Clean database**: "yes" or "no". It deletes the entire DB Note: If set to "yes", this will require confirmation from user during execution.
- **Split weather files enable**: "yes" or "no". It splits the projected weather file by scenario and model to facilitate processing.
- **Import weather files enable**: "yes" or "no". It imports splitted weather files into the DB.
- **Import station locations**: "yes" or "no". It imports station locations.
- **Src predictive target**: Target paths to dedicated location in the application ../../nc_files/FILE_NAME
- **Src historical target**: Target paths to dedicated location in the application ../../nc_files/FILE_NAME
- **Weather station file target**: Target paths to dedicated location in the application ../../power_system_csv_files/FILE_NAME

- **Load folder name target**: Target paths to dedicated location in the application ../../load_profiles/FILE_NAME
- **Host name:** User should provide here the connection details of the server if running in remote mode or use "localhost" when running from the server.
- **port:** This parameter is set to 8000. In case, advanced user wants to change it, make user docker-compose file deft.yml is also changed.

Running a project

After the tool is set-up correctly the user can run the tool directly from the server (within the docker environment) or remotely using your windows machine and the python package downloaded from GitHub executing the demo.py file (LINK TO BE PROVIDED BY EPRI).

Server execution

Users that prefer to run the tool though a client (i.e., PuTTY) will need to follow the process described below:

1. Make sure the input data you need has been transferred to the container. If you have set-up a mount you can transfer the data from the mount typing the following line: `docker cp /mnt/your_import_folder/input/ deft-backend-1:/code/backend/demo/`
   This will replace the input folder located where the code lies with the data you have specified. Make sure to preserve the input folder structure (See Input Data).
2. Set-up your project execution parameters in the config.yml. You can transfer and replace this file from your mount or type nano config.yml to modify the parameters accordingly (See Section: Configuration)
3. From step 6 in the installation process, open a new tmux window (same tmux session as the one in installation) typing: "Ctrl+b+c". This window will be used to launch the docker container environment where the RiSc tool will be executed. Type : `docker exec -it deft-backend-1 bash` to launch the docker environment.
4. Run the tool typing: `python demo.py`. This will start the execution process that you can follow on the console.
5. Check solution files in the output folder of the demo directory within the container. If you want to transfer these files to your windows machine type:
   `docker cp deft-backend-1:/code/backend/demo/output /mnt/deft_imports/`

Windows execution

Users that prefer to run the tool in their windows machine will need to follow the process described below:

1. Go to RiSc repository in your windows machine and access the demo folder:

2. Set-up your project execution parameters in the config.yml and set the parameter sync_type to "remote_share", hostname to your server (where application is running) hostname and your mount path connecting server and your machine for data transfers (See Configuration)
3. Make sure the input data you need has been transferred to the mount and you keep the input folder structure. (See Input Data).
4. Execute `python demo.py` from the demo directory folder
5. Check results in the output folder within the working directory “demo”

Output Data

The output of the risk screening tool is generated by project (project name generated automatically by the tool based on specified user configuration), each project being a combination of scenarios and models. The generated files are 2 per combination (i.e., for 2 SSP scenarios and 2 GCMs there will be 4 GCM/SPP combined projects each with 2 files):

- **Summary results (csv file):** this file provides essential information about the X% events that have been screened (start and end date, scenario, model, and average capacity at risk in MW)
- **Detailed results (xlsx file):** this file provides detailed information about the weather and the active risks or vulnerabilities for each event reported in different tabs (screened events). The detailed data include hourly weather and risk information linked to each resource type and weather station. The risks or derating capacity information is reported in MW and the weather variables units are defined in the table below:

Table 8 : Units reported in the detailed output file

Weather Variable:Units
temperature: K
windspeed_100m: m/s
windspeed_10m: m/s
precip: in
snow: in
mslp: Pa
solar: w/m^2
dewpoint: K

An additional file (“aggregated_capacity.csv”) is generated to report the aggregated capacity used by the tool per weather station.

This release does only provide the raw output data coming from the RiSc tool. EPRI post-processing and clustering step to facilitate the data handling and analysis of results for potential case studies is not provided as part of this open-source release. Please reach out to the EPRI team for more information.

3 TROUBLESHOOTING

Debugging RiSc can be challenging. For that reason, there are two log files that users need to be familiar with:

- log_timestamp.log: This file, located in the application demo folder, contains the log of RiSc application where most errors are reported as server error. However, this can make it challenging to detect api endpoint error. For detailed information about api errors use deft_error.log.

```
Endpoint export_average_risk_result_split failed with error
<!doctype html>
<html lang="en">
<head>
  <title>Server Error (500)</title>
</head>
<body>
  <h1>Server Error (500)</h1><p></p>
</body>
</html>
```

Figure 7 : Log of failed endpoint with errors

- deft_error.log: This file, located in the application backend folder, provides detailed information about API endpoint errors (server error - 500). You can type nano deft_error.log to see more information about the error (see screenshot).

```
Internal Server Error: /api/import_station_locations/
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/site-packages/django/db/backends/utils.py", line 89, in _execute
    return self.cursor.execute(sql, params)
psycopg2.errors.UniqueViolation: duplicate key value violates unique constraint "stations_name_71acd296_uniq"
DETAIL:  Key (name)=(Station_0) already exists.
```

Figure 8 : Screenshot of error information

4 EXAMPLE WALKTHROUGH

In this section, a step-by-step example is provided to help users get acquainted with the basic features of RiSc. Steps followed in this example are representative of a normal execution of the tool. We invite the user to reach out to EPRI for specific support or customized implementation of the tool including event selection and visualization, scenario building (wind, solar& load profiles, maintenance, weather dependent outages) for operational simulations tools, or other customized implementation coming from the outputs of RiSc. Data provided as part of the software package does not intent to represent actual power system data and should be used a placeholder data to be replaced by users as described in following sections. Below users can see an application of the RiSc tool to the Texas Power System for the study year 2021:

Step 1: Data Collection

Climate & Load Data

Climate data for the region of Texas has been obtained from public reanalysis data: ERA5. As mentioned in Section 2 – Climate Data, RiSc tool need a pair of NetCDF files: historical and projected data files. In this example, both historical and projected contain the same ERA5 data to illustrate the process. However, user should also complement this with forward-looking climate projections (combination of GCMs and SSPs) in the projected file.

There are 12 weather years represented in this example (from 2010 to 2021). Obviously, the more weather years you consider in your runs the more processing time you will require to screen across all of them. This is also true for the level of geospatial resolution your weather data contains. In this case, 26 weather stations are considered across the Texas region. The choice should be a trade-off between accuracy and processing time. In this case, the selection of the stations has been driven by the 10 most relevant existing and future wind installations, 12 solar installations and 4 main load centers (Austin, Houston, Dallas and San Antonio)

	A	B	C	D
1		StationName	lat	lon
2	0	Station_0	34.945	-102.279
3	1	Station_1	35.255	-101.187
4	2	Station_2	34.05	-101.188
5	3	Station_3	30.95	-102.2
6	4	Station_4	32.1	-101.45
7	5	Station_5	32.36	-100.34
8	6	Station_6	27.12	-97.54
9	7	Station_7	27.32	-99
10	8	Station_8	33.7	-99.5
11	9	Station_9	33.05	-98.4
12	10	Station_10	33.45	-95.9
13	11	Station_11	32.7	-102
14	12	Station_12	31	-98
15	13	Station_13	30.8	-103.4
16	14	Station_14	29.5	-96
17	15	Station_15	29.3	-100
18	16	Station_16	27.4	-98.4
19	17	Station_17	34.5	-101.8
20	18	Station_18	29.3	-98.4
21	19	Station_19	32.1	-95.25
22	20	Station_20	31.2	-102.2
23	21	Station_21	33	-99.5
24	22	Station_22	30.267	-97.733
25	23	Station_23	29.75	-95.36
26	24	Station_24	32.779	-96.809
27	25	Station_25	29.424	-98.491

Figure 9 : Texas weather station location

The system-wide load is the result of applying the weather data mentioned above to the EPRI load forecasting models. We encourage users to use weather-coherent load profiles. If this is not possible, we suggest users to contact EPRI for support. Load files need to have the convention name described in Section 2.

- ERA5_historical_2021
- ERA5_historical_2021_min
- ERA5_historical_2021_plus

Figure 10 : Texas load files names for 2021 target year (ERA5 model and scenario)

	A	B	C
1	Time	total_load	
2	01/01/2010 00:00	34466.96	
3	01/01/2010 01:00	35312.42	
4	01/01/2010 02:00	35672.66	
5	01/01/2010 03:00	37512.05	
6	01/01/2010 04:00	37666.79	
7	01/01/2010 05:00	37494.45	
8	01/01/2010 06:00	38445.62	
9	01/01/2010 07:00	39078.27	

Figure 11 : Texas load profile for weather years 2010 to 2021

Asset Vulnerabilities

In the context of EPRI Climate READi, this table have been generated thanks to the input of subject matter experts, manufacturers, historical events, and data. Asset vulnerabilities in this example are specific to the Texas region. We recommend users to update this table based on expert knowledge on the study region.

	A	B	C	D	E	F	G	H	I
1	tech	variable1	operator1	threshold1	variable2	operator2	threshold2	derating	risk
2	PV	temperature	<=	32	precipitation	>=	0.001	90	Icing
3	PV	temperature	<=	32	snow height	>=	1	100	Snow
4	PV	temperature	<=	-29				20	Extreme cold equipment damage
5	PV	wind speed at 10m	>=	60				40	Hurricane/tornado force wind damage
6	PV	temperature	>	14	temperature	<=	41	4	Minimal temperature derating
7	PV	temperature	>	41	temperature	<=	68	10	Mid-low temperature derating
8	PV	temperature	>	68	temperature	<=	95	16	Mid temperature derating
9	PV	temperature	>	95	temperature	<=	122	22	Mid-high temperature derating
10	PV	temperature	>	122	temperature	<=	149	28	High temperature derating
11	PV	temperature	>	149	temperature	<=	165	34	Extreme high temperature derating
12	PV	temperature	>	165				100	High Temperature Rating Limit
13	PV	solar irradiance	<	800	solar irradiance	>=	600	20	Sub-max sun
14	PV	solar irradiance	<	600	solar irradiance	>=	400	40	Mid-high sun
15	PV	solar irradiance	<	400	solar irradiance	>=	200	60	Mid-low sun
16	PV	solar irradiance	<	200	solar irradiance	>=	50	80	Low sun
17	PV	solar irradiance	<	50				100	No sun
18	PV	precipitation	>=	0.25	precipitation	<	0.7	5	Possible Flood Risk
19	PV	precipitation	>=	0.7				10	Major Flood Risk
20	WT_ON	wind speed at 100m	<	4				100	Low wind output
21	WT_ON	wind speed at 100m	>=	4	wind speed at 100m	<	6	88	Low windspeed
22	WT_ON	wind speed at 100m	>=	6	wind speed at 100m	<	8	68	Mid-low windspeed
23	WT_ON	wind speed at 100m	>=	8	wind speed at 100m	<	10	32	Mid-high windspeed
24	WT_ON	wind speed at 100m	>=	10	wind speed at 100m	<	12	12	Suboptimal windspeed
25	WT_ON	wind speed at 100m	>=	18	wind speed at 100m	<	23	12	High windspeed furl
26	WT_ON	wind speed at 100m	>	23				100	High speed cutout
27	WT_ON	temperature	<	32	precipitation	>=	0.001	20	Blade icing
28	WT_ON	temperature	<	-12				100	Low Temperature Rating Limit (1)
29	WT_ON	temperature	<	6	temperature	>=	-12	20	Low Temperature Rating Limit (2)
30	WT_ON	temperature	>	96	temperature	<=	123	20	High Temperature Derating
31	WT_ON	temperature	>	123				100	High Temperature Rating Limit

Figure 12 : Example of Texas asset vulnerabilities (or risk model)

When considering adaptations, we suggest the user to create new unit types such as (CC_Winterize, GT_InletChiller,…) adapting the corresponding thresholds and IF-THEN rules as needed based on the hardening options you might be able to include in your study. In this example, adaptation and hardening options have not been considered.

Power System Data

Synthetic power system data for Texas has been used converted to RiSc format as shown in Figure 13:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Asset ID	Generator Shortname	Operational Date	Retirement Date	Unit Type	FuelType	AreaCode	Zone	State	Latitude	Longitude	Nameplate (MW)	
2	1	1_828_p3460	01/01/1970	01/01/2025	ST	NG	7	Coast	TX	29.7501	-94.9256	746	
3	2	2_828_p3460	01/01/1972	01/01/2025	ST	NG	7	Coast	TX	29.7501	-94.9256	749	
4	3	3_829_p3464	01/01/1976	01/01/2024	GT	NG	7	Coast	TX	29.8222	-95.2194	54	
5	4	4_829_p3464	01/01/1976	01/01/2024	GT	NG	7	Coast	TX	29.8222	-95.2194	54	
6	5	5_829_p3464	01/01/1976	01/01/2024	GT	NG	7	Coast	TX	29.8222	-95.2194	54	
7	6	6_829_p3464	01/01/1976	01/01/2024	GT	NG	7	Coast	TX	29.8222	-95.2194	50	
8	7	7_829_p3464	01/01/1976	01/01/2024	GT	NG	7	Coast	TX	29.8222	-95.2194	64	
9	8	8_829_p3464	01/01/1976	01/01/2024	GT	NG	7	Coast	TX	29.8222	-95.2194	54	
10	9	9_830_p3469	01/01/1974	01/01/2030	CC	NG	7	Coast	TX	29.9418	-95.5306	104	
11	10	10_830_p3469	01/01/1974	01/01/2030	CC	NG	7	Coast	TX	29.9418	-95.5306	104	
12	11	11_830_p3469	01/01/1972	01/01/2030	CC	NG	7	Coast	TX	29.9418	-95.5306	57	
13	12	12_830_p3469	01/01/1972	01/01/2030	CC	NG	7	Coast	TX	29.9418	-95.5306	57	
14	13	13_830_p3469	01/01/1972	01/01/2030	CC	NG	7	Coast	TX	29.9418	-95.5306	57	
15	14	14_830_p3469	01/01/1972	01/01/2030	CC	NG	7	Coast	TX	29.9418	-95.5306	57	

Figure 13 : Example of Texas power system data

Note that Unit Types can be different as long as they are aligned with the asset vulnerabilities file. Also, remember that FuelType, AreaCode, Zone or State are optional columns for RiSc tool.

Step 2: Tool configuration

Settings related to importing data

After collecting the data that is needed to run the tool, it is important to remember to set-up a mount or SFTP with the remote sever where the tool is hosted. This mount will make heavy data files (climate & load data) available to the tool. The user should always keep the input data folder structure as follows:

```
# Settings related to importing weather data
sync_files_enabled: "no"
sync_type: "local" # local, remote_share
src_predictive: "Texas_Projected_Data_2021.nc" # must be in folder nc_files/
src_historical: "Texas_Historical_Data.nc" # must be in folder nc_files/
weather_station_file: "station_location_texas.csv" # must be in folder locations/
load_folder_name: "2021_TEXAS" # must be in folder load_profiles/
```

Figure 14 : Extract of input data file and folder names (config.yml)

```
src_predictive_target: "../../nc_files/Texas_Projected_Data_2021.nc"
src_historical_target: "../../nc_files/Texas_Historical_Data.nc"
weather_station_file_target: "../../power_system_csv_files/station_location_texas.csv"
load_folder_name_target: "../../load_profiles/2021_TEXAS"
```

Figure 15 : Extract of input data target location in the RiSc application (sysconfig.yml)

Make sure that when setting up the tool, you set the sync_files_enable to “yes” for both the server and windows execution.

- Server execution: Set and sync_type to “local” and transfer the data from your windows machine to the application (docker container). Type:
  `docker cp /mnt/your_import_folder/input/ deft-backend-1:/code/backend/demo/`

- Windows execution: Set sync_type to “remote_share” and the hostname and mount point in the sysconfig.yml

When setting reset_all to “no” you are planning to customize the weather data importing process. The customized parameters are defined in the sysconfig file: Clean_database, Split_weather_data_enable, import_weather_files_enable, and import_station_locations. By default, these are set to “yes”, unless user has already imported the weather and load data and wants to re-run or run different sensitivities using different power systems or risk models (all parameters set to “no”).

As opposed to climate and load data, power system and asset vulnerabilities (or risk models) paths are defined in the config.yml file because this allows more flexibility to the user who can decide to change and run sensitivities using different capacity mixes and asset vulnerabilities preserving the already imported climate and load data (reset_all set to “no”). In this example the name of the files needs to coincide with the ones stored in the mount and transferred to the server.

- power_system_file
- risk_model_file

A dedicated file is generated depending on the power system used in each project, where the capacity is aggregated by weather station depending on the distance of the asset to give the user a clear view of how the tool treat risks within a given region.

- Aggregated capacity_file

When data is imported make sure that the tool settings are aligned with the data.

- **target year** is set to the year you are studying (i.e, weather data projected to that year, representative load and power system of the study year). In this example: 2021.
- **Weather stations in screening** need to coincide with the weather data. Therefore, for this example it will be 26.
- **Start and end weather years** need to be in the weather data yearly range. In this case, weather data contains data from 2010 to 2021 (included) but the screening range has been set to 2 years: 2020 to 2021 just to reduce computational resources in this example.
- **Scenarios**: SSP scenarios need to be aligned with the weather data (projected file). Typically, SSP scenarios are defined here: SSP126, SPP370. In this example to make the data publicly available, only ERA5 reanalysis data is used.
- **Models**: Global Climate Models need to be defined in the weather data (projected file). Typically, there are a set of models to screen across (e.g., UKESM, GFDL. MRI, MPI or IPSL) but in this example, as scenarios, only ERA5 reanalysis data has been used.

Settings related to screening process

Screening parameters need to be defined depending on the type of study you want to run:

- **Event_day_interval**: Define the length of the events. In this example we have used a 21-day period to identify extreme events based on average risk across the 3-week interval.
- **Event_day_gap**: Define the gap or distance between events. In this example, the tool will evaluate all 3-week events shifted 1 week from one another across 2020 to 2021 weather years.
- **Top_events_percentage**: Defines the number of events to be retained based on the average risk calculation.

In this example, this all results in 2years * 52weeks/year *5% ~ 6 events to be retained.

Step 3: Tool execution

Before executing the tool, make sure that the application is up and running (See Installation).

Windows execution

1. Go to RiSc repository in your windows machine and access the demo folder:

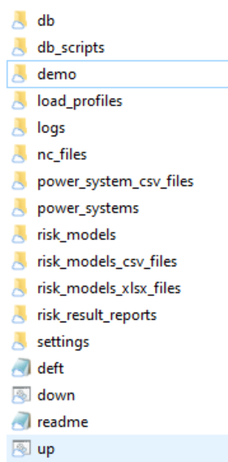


Figure 16 : RiSc repository

2. Set-up your project execution parameters in the config.yml and set the parameter sync_type to "remote_share" , hostname to your server hostname and your mount path(See Configuration)

```
reset_all: "yes"
delete_project_first: "yes" # "yes" or "no", delete project before creating it
project_region: "TEXAS"
scenarios:
  - "demo"
models:
  - "ERA5"
start_year: 2020
end_year: 2022
weather_stations_in_screening: 26
target_year: 2021
power_systems_file: "input/power_systems/generatorData_ACEP_REF_2021.csv"
risk_model_file: "input/risk_models/risk_models_texas.csv"
aggregate_capacity_file: "output/aggregated_capacity_2021.csv"
event_day_interval: 21
event_day_overlap: 7 # rename to event_day_gap
top_events_percentage: 5
threshold_MW: 0

# Settings related to importing weather data
sync_files_enabled: "yes"
sync_type: "remote_share" # local, remote_share
src_predictive: "Texas_Projected_Data_2021.nc" # must be in folder nc_files/
src_historical: "Texas_Historical_Data.nc" # must be in folder nc_files/
weather_station_file: "station_location_texas.csv" # must be in folder locations/
load_folder_name: "2021_TEXAS" # must be in folder load_profiles/
```

Figure 17 : Configuration file for window execution.

3. Make sure the input data you need has been transferred to the mount and you keep the input folder structure. (See Input Data). Check the deft.yml file to see the volume has been mapped correctly:

```
backend:
  image: ikimio/deft_backend:1.9
  tty: true
  networks:
    - deft-network
  volumes:
    - ./settings:/code/backend/backend
    - ./logs/backend.log:/code/backend/deft_error.log
    - ./nc_files/:/code/nc_files
    - ./risk_models/:/code/risk_models
    - ./power_systems/:/code/power_systems
    - ./power_system_csv_files/:/code/power_system_csv_files
    - ./load_profiles/:/code/load_profiles
    - ./risk_models_xlsx_files/:/code/risk_models_xlsx_files
    - ./risk_result_reports/:/code/risk_result_reports
    # - /mnt/hgfs/shared/input:/code/share
  ports:
    - "8000:8000"
```

Figure 18 : Volumes in backend application.

4. Execute `python demo.py` from the demo directory folder.
5. Check results in the output folder within the working directory “demo”

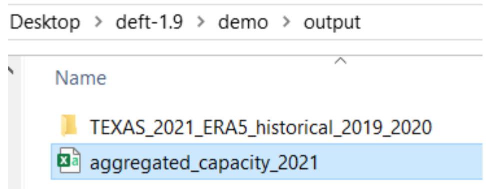


Figure 19 : Windows execution results.

Server execution

1. Open your PuTTY connection though Citrix Workspace. This will open a command prompt, then open a new tmux session: `tmux attach-session -t 0`.

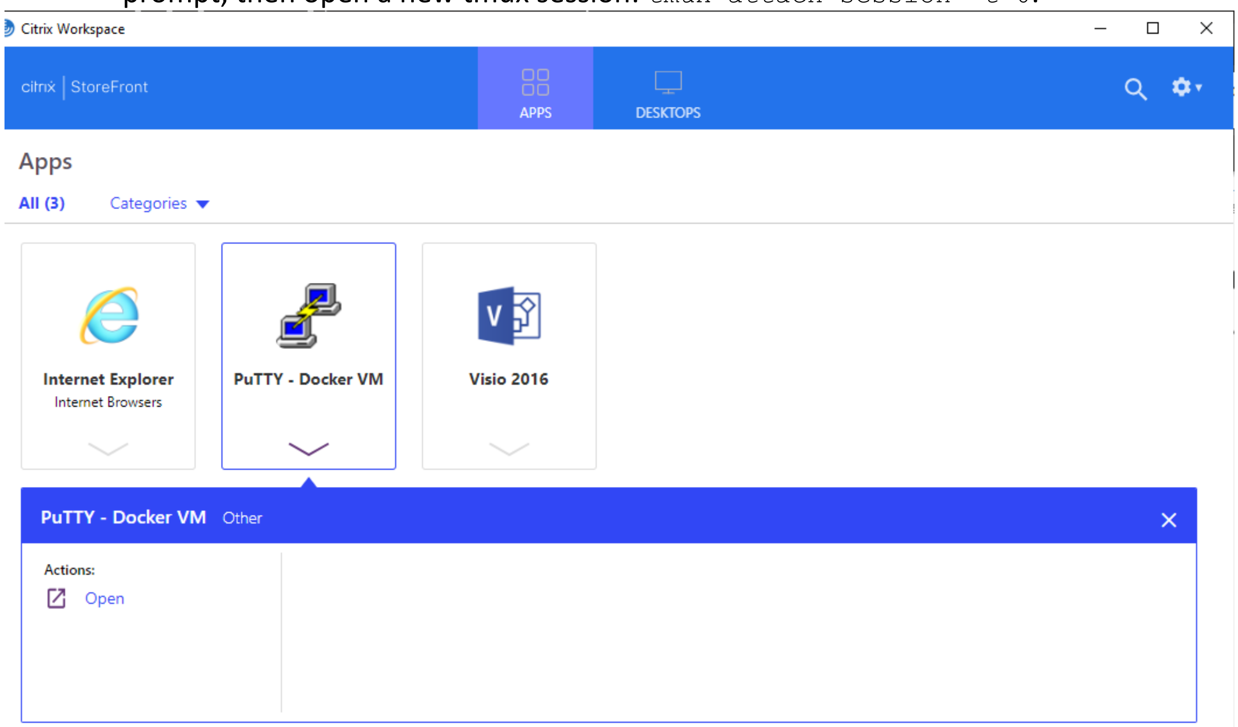


Figure 20 : PuTTy connection.

2. Run the application if not already up: ./up.sh (See Installation)
3. Make sure the input data you need has been transferred to the application. If you have set-up the mount, you can transfer the data from the mount typing the following line in bash:
   `docker cp /mnt/your_import_folder/input/ deft-backend-1:/code/backend/demo/`
   This will replace the input folder with the data you have specified. Make sure to preserve the input folder structure and set-up correctly the target paths (sysconfig.yml)

4. In another tmux window ("ctrl+b+c") open the application environment

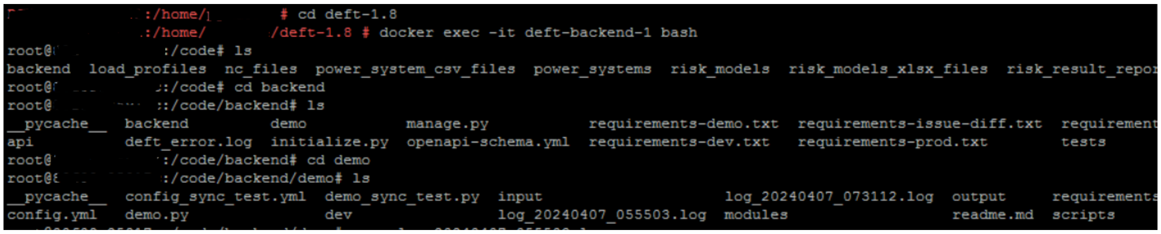


Figure 21 : Access the application environment (docker container)

5. In the application, you can verify that the data has been correctly transferred: Go to backend/demo/input and check the data

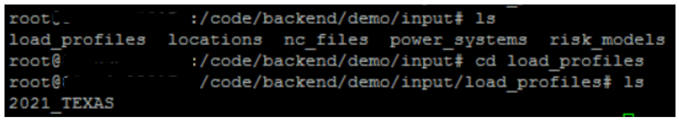


Figure 22 : Check input data is being transferred.

6. Once the data is transferred to the application, set-up your project execution parameters in the config.yml. You can transfer and replace this file from your mount or type nano config.yml to modify the parameters accordingly (See Configuration)

```
reset_all: "yes"
delete_project_first: "yes"
project_region: "TEXAS"
scenarios:
  - "demo"
models:
  - "ERA5"
start_year: 2020
end_year: 2022
weather_stations_in_screening: 26
target_year: 2021
power_systems_file: "input/power_systems/generatorData_ACEP_REF_2021.csv"
risk_model_file: "input/risk_models/risk_models_texas.csv"
aggregate_capacity_file: "output/aggregated_capacity_2021.csv"
event_day_interval: 21
event_day_overlap: 7 # rename to event_day_gap
top_events_percentage: 5
threshold_MW: 0

# Settings related to importing weather data
sync_files_enabled: "yes"
sync_type: "local" # local, remote_share
src_predictive: "Texas_Projected_Data_2021.nc" # must be in folder nc_files/
src_historical: "Texas_Historical_Data.nc" # must be in folder nc_files/
weather_station_file: "station_location_texas.csv" # must be in folder locations/
load_folder_name: "2021_TEXAS" # must be in folder load_profiles/
```

Figure 23 : Config file for server execution.

7. Run the tool typing "python demo.py". This will start the execution process that you can follow on the console.

```
2024-06-03 11:40:57,046 - INFO - Getting access token
2024-06-03 11:40:57,265 - INFO - Access token:

Are you sure you want to delete the data? (Y/N): Y
2024-06-03 11:41:17,738 - INFO - Deleting all data, please wait...
2024-06-03 13:30:07,540 - INFO - Database cleaned.
2024-06-03 13:30:07,542 - INFO - {"success":true,"data":{"message":"Database cleaned successfuly.","data":null}}
2024-06-03 13:30:07,542 - INFO - Creating aggregate capacity file.
2024-06-03 13:30:08,842 - INFO - Aggregate capacity file saved to: output/aggregated_capacity_2021.csv
2024-06-03 13:30:08,842 - INFO - Converting power systems to json
2024-06-03 13:30:08,859 - INFO - Power systems json file saved to: input/power_systems/ps-demo-input.json
2024-06-03 13:30:08,859 - INFO - Converting risk models to json
2024-06-03 13:30:08,868 - INFO - Risk model json file saved to: input/risk_models/rm-demo-input.json
2024-06-03 13:30:08,868 - INFO - Loading power systems and risk models
2024-06-03 13:30:08,869 - INFO - Power systems and risk models loaded.
2024-06-03 13:30:08,869 - INFO - Syncing files enabled, type set to local, starting sync...
2024-06-03 13:30:08,869 - INFO - Copying historical nc file input/nc_files/Texas_Historical_Data.nc to backend target: ../../nc_
2024-06-03 13:30:08,933 - INFO - Historical file transferred.
2024-06-03 13:30:08,933 - INFO - Copying predictive nc file input/nc_files/Texas_Projected_Data_2021.nc to backend target: ../..
2024-06-03 13:30:08,963 - INFO - Predictive file transferred.
2024-06-03 13:30:08,963 - INFO - Copying stations and locations file input/locations/station_location_texas.csv to backend target
2024-06-03 13:30:08,964 - INFO - Stations and locations file transferred.
2024-06-03 13:30:08,965 - INFO - Copying load profiles from input/load_profiles/2021_TEXAS to backend target: ../../load_profile
2024-06-03 13:30:09,009 - INFO - Load profiles transferred.
2024-06-03 13:30:09,009 - INFO - Splitting NC files.
2024-06-03 13:30:09,009 - INFO - Creating split nc files: TEXAS_2020_2022_historical.nc, TEXAS_ERA5_historical_2020_2022_project
```

Figure 24 : RiSc execution log.

8. Check solution files in the output folder of the demo directory within the container. If you want to transfer these files to your windows machine, follow the similar importing instructions as in step 1 typing:

```
docker cp  deft-backend-1:/code/backend/demo/output /mnt/your_import_folder/
```

Step 4: Results

The output of the risk screening tool is generated by project (project name generated automatically by the tool based on specified user configuration), each project being a combination of scenarios and models. In this example, only one combination is possible: ERA5 and historical leading to only one project folder.

TEXAS_2021_ERA5_historical_2020_2022_2024-06-04
aggregated_capacity_2021

Results for each project contain 2 files:

- **Summary results (csv file):** this file provides essential information about 5% top worst events that have been screened (start and end date, scenario, model and average capacity at risk in MW)

	A	B	C	D	E	F	G	H
1	start	end	average	model	scenario	year	report	
2	2021-02-03T00:00:00	2021-02-24T00:00:00	70507.9857	ERA5	historical	2021	ERA5_historical_1	
3	2021-02-10T00:00:00	2021-03-03T00:00:00	69591.1258	ERA5	historical	2021	ERA5_historical_2	
4	2020-08-12T00:00:00	2020-09-02T00:00:00	65421.6186	ERA5	historical	2021	ERA5_historical_3	
5	2021-01-27T00:00:00	2021-02-17T00:00:00	65094.7645	ERA5	historical	2021	ERA5_historical_4	
6	2020-08-05T00:00:00	2020-08-26T00:00:00	63435.411	ERA5	historical	2021	ERA5_historical_5	
7	2021-07-14T00:00:00	2021-08-04T00:00:00	63173.8601	ERA5	historical	2021	ERA5_historical_6	
8								

Figure 25 : Texas summary RiSc results

- **Detailed results (xlsx file):** this file provides detailed information about the weather and the active risks or vulnerabilities for the 5% top worst events, where each event is reported in a different tab . The detailed data include hourly weather (column C) and risk (column D) information in JSON format linked to each resource type and weather station. The risks or derating capacity information is reported in MW and the weather variables units are defined in the table below:

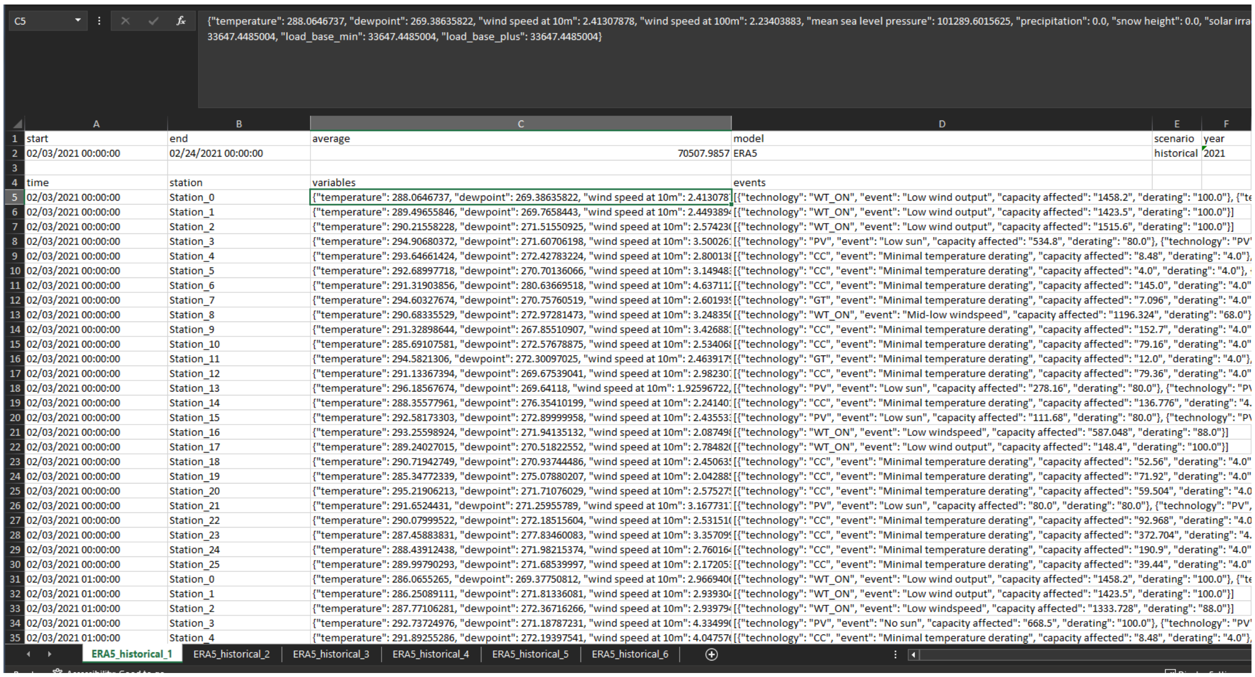


Figure 26 : Texas detailed RiSc results

An additional file is generated to report the aggregated capacity used by the tool per weather station. "aggregated_capacity_2021.csv"

	A	B	C
1	Unit Type	station	Nameplate
2	CC	Station10	1979
3	CC	Station12	1984
4	CC	Station14	3419.4
5	CC	Station18	1314
6	CC	Station19	1798
7	CC	Station20	1487.6
8	CC	Station22	2324.2
9	CC	Station23	9317.6
10	CC	Station24	4772.5
11	CC	Station25	986
12	CC	Station4	212
13	CC	Station5	100
14	CC	Station6	3625
15	CC	Station9	3817.5

Figure 27 : Extract of aggregated capacity output file for Texas 2021 power system.

This release does only provide the raw output data coming from the RiSc tool. EPRI post-processing and clustering step to facilitate the data handling and analysis of results for potential case studies is not provided as part of this open-source release. You can see here some visualization that have been generated to better interpret and understand the results. Please reach out to the EPRI team for more information.

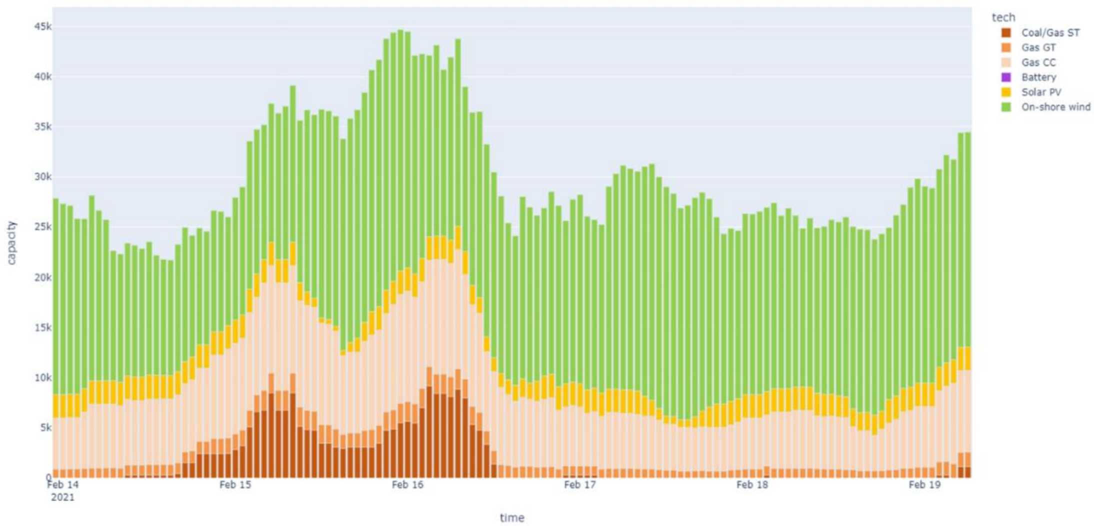


Figure 28 : Texas generation capacity out during 2021 Uri Storm (RiSc Results)

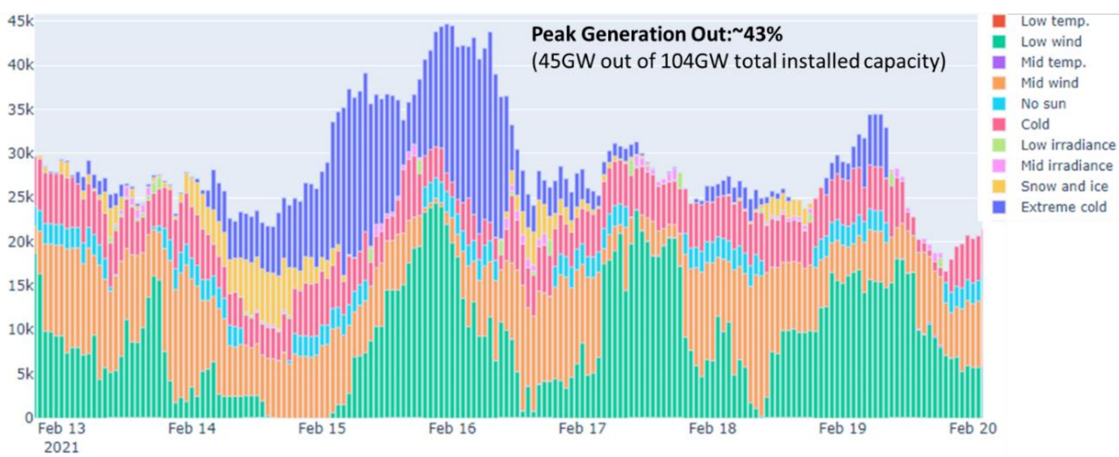


Figure 29 : Texas generation capacity out during 2021 Uri Storm (RiSc Results)

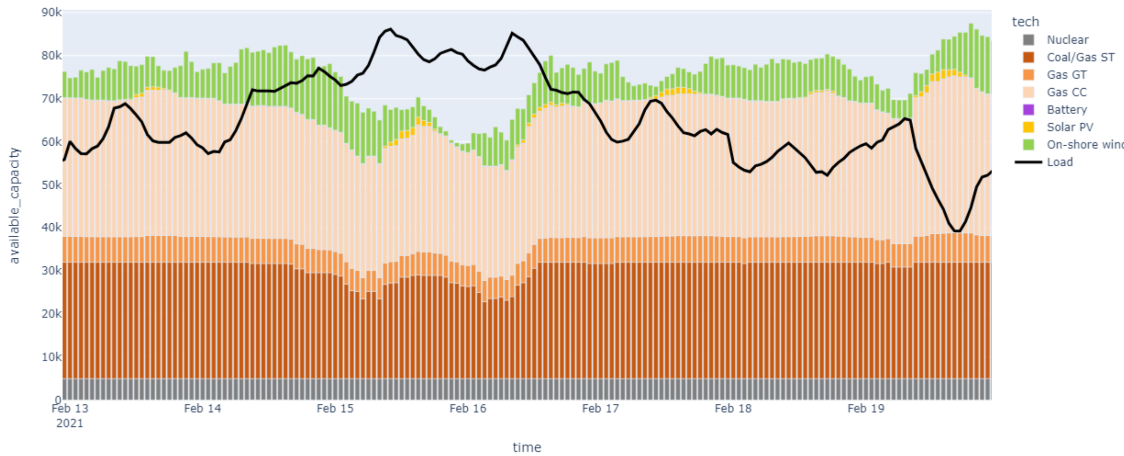


Figure 30 : Texas available supply generation and load during 2021 Uri Storm

5 ANNEX

Application diagram

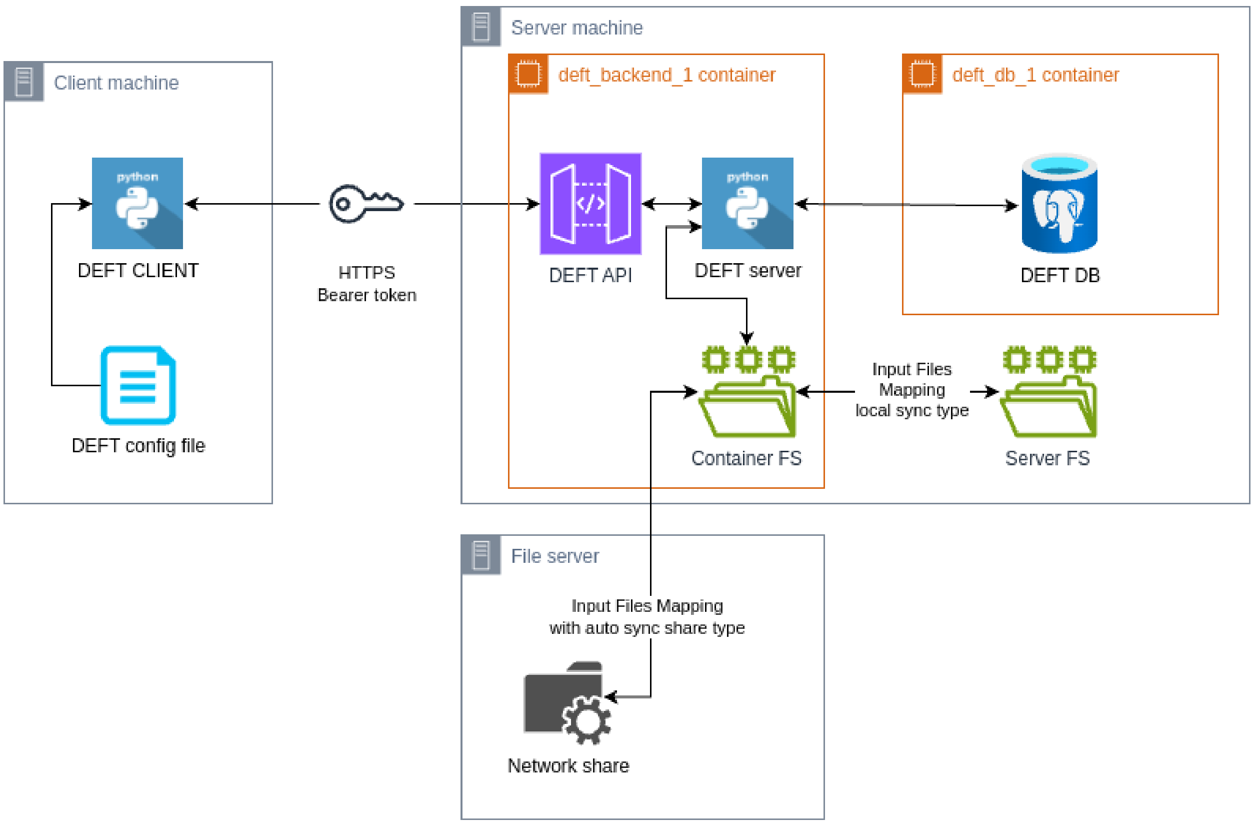


Figure 31 : RiSc schema.

Data Base

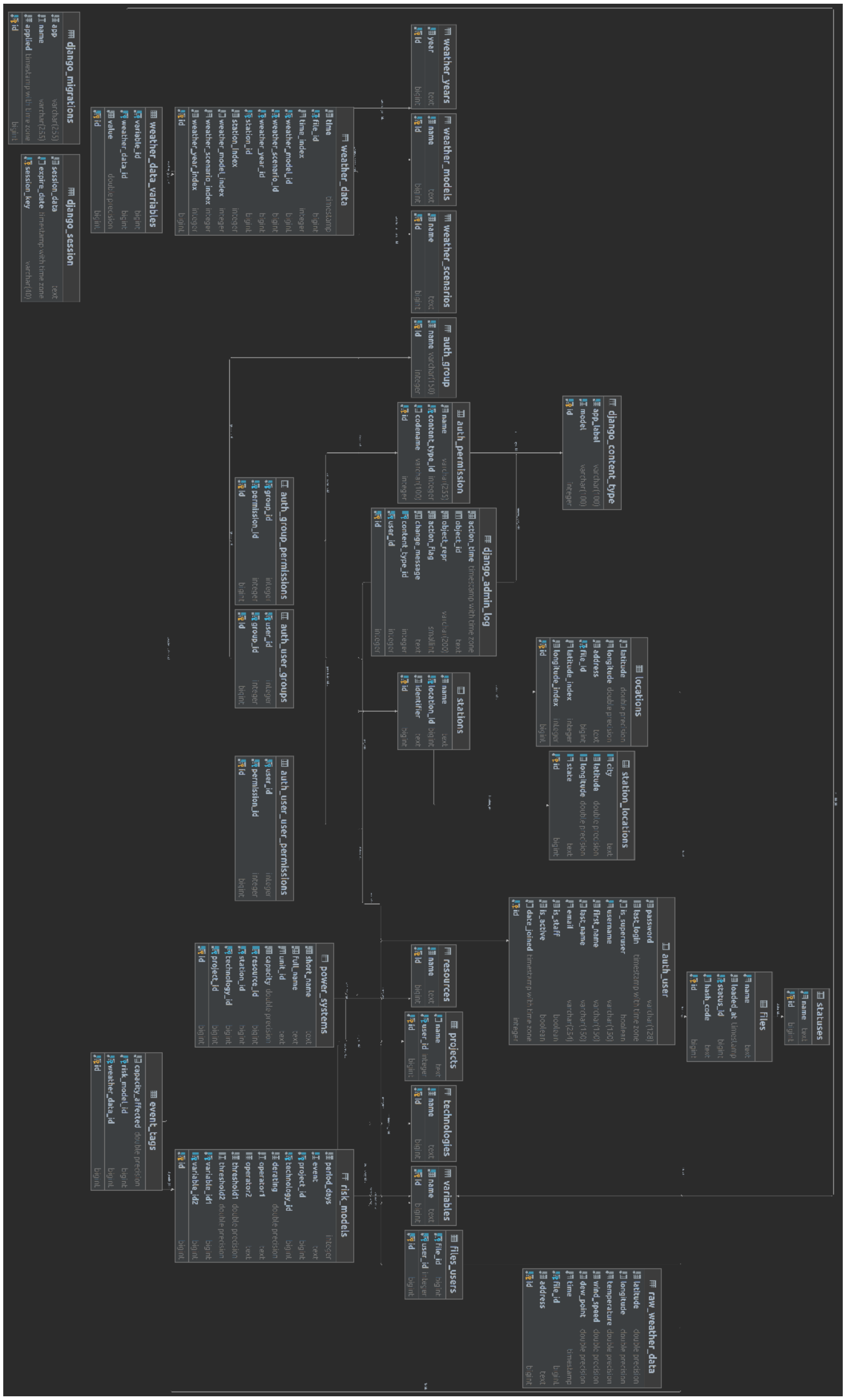


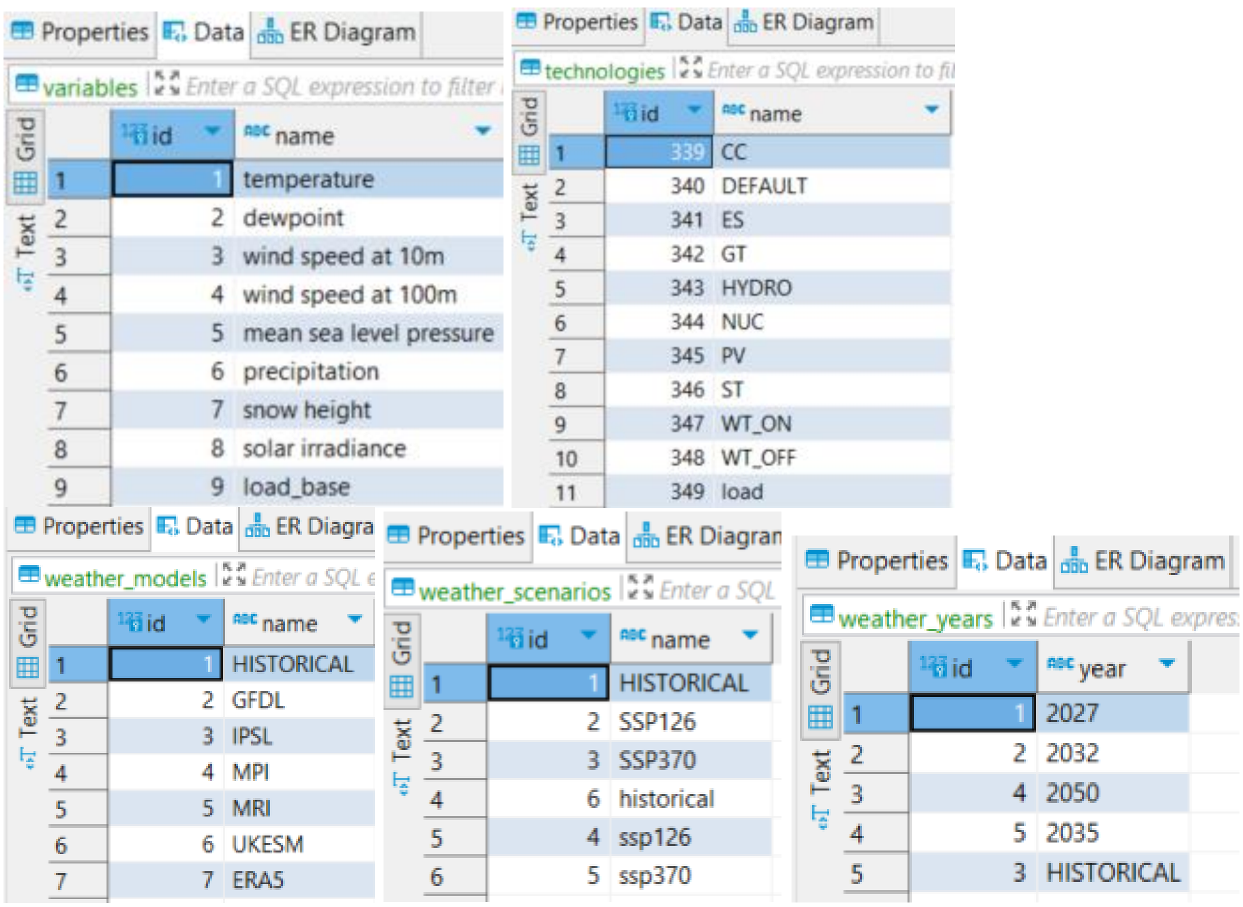
Figure 32 : RiSc database schema.

Connect DB from Windows

Use a client application (e.g., DBeaver) to interact from your laptop with the DB. To connect download Postgres drivers and define:

- Database name: deft
- Port: 54322
- Host name: XXXXX.your_organization.com
- Password located in db/db-password.txt
- Before doing any operation it is recommended to do a backup of the current DB. Execute deft>db_scripts>export_db.sh (type: ./export_db.sh). This will dump db into a deft>db>dumps folder.

For configuration purposes, it's always a good practice check that the data has been imported correctly into the DB. Below you can find some examples where tables are directly linked to the imported weather, risk models, and power system data.



variables

	id	name
1	1	temperature
2	2	dewpoint
3	3	wind speed at 10m
4	4	wind speed at 100m
5	5	mean sea level pressure
6	6	precipitation
7	7	snow height
8	8	solar irradiance
9	9	load_base

technologies

	id	name
1	339	CC
2	340	DEFAULT
3	341	ES
4	342	GT
5	343	HYDRO
6	344	NUC
7	345	PV
8	346	ST
9	347	WT_ON
10	348	WT_OFF
11	349	load

weather_models

	id	name
1	1	HISTORICAL
2	2	GFDL
3	3	IPSL
4	4	MPI
5	5	MRI
6	6	UKESM
7	7	ERA5

weather_scenarios

	id	name
1	1	HISTORICAL
2	2	SSP126
3	3	SSP370
4	6	historical
5	4	ssp126
6	5	ssp370

weather_years

	id	year
1	1	2027
2	2	2032
3	4	2050
4	5	2035
5	3	HISTORICAL

Figure 33 : Example of database tables input data.

Export Control Restrictions

Access to and use of this EPRI product is granted with the specific understanding and requirement that responsibility for ensuring full compliance with all applicable U.S. and foreign export laws and regulations is being undertaken by you and your company. This includes an obligation to ensure that any individual receiving access hereunder who is not a U.S. citizen or U.S. permanent resident is permitted access under applicable U.S. and foreign export laws and regulations.

In the event you are uncertain whether you or your company may lawfully obtain access to this EPRI product, you acknowledge that it is your obligation to consult with your company's legal counsel to determine whether this access is lawful. Although EPRI may make available on a case-by-case basis an informal assessment of the applicable U.S. export classification for specific EPRI products, you and your company acknowledge that this assessment is solely for informational purposes and not for reliance purposes.

Your obligations regarding U.S. export control requirements apply during and after you and your company's engagement with EPRI. To be clear, the obligations continue after your retirement or other departure from your company, and include any knowledge retained after gaining access to EPRI products.

You and your company understand and acknowledge your obligations to make a prompt report to EPRI and the appropriate authorities regarding any access to or use of this EPRI product hereunder that may be in violation of applicable U.S. or foreign export laws or regulations.

About EPRI

Founded in 1972, EPRI is the world's preeminent independent, non-profit energy research and development organization, with offices around the world. EPRI's trusted experts collaborate with more than 450 companies in 45 countries, driving innovation to ensure the public has clean, safe, reliable, affordable, and equitable access to electricity across the globe. Together, we are shaping the future of energy.

Program(s):

Climate READi

Resource Adequacy Initiative

P173C

3002028699

EPRI
3420 Hillview Avenue, Palo Alto, California 94304-1338 ▪ USA
800.313.3774 ▪ 650.855.2121 ▪ askepri@epri.com ▪ www.epri.com