OPENTUNITY power flow developments (v1)

5 November 2024

opentunity

OPENTUNITYproject.eu

Deliverable details

Title	WP	Version
OPENTUNITY power flow developments (v1)	5	1.0

Contractual delivery date	Actual delivery date	Delivery type*	Dissemination**
M22 (October 2024)	M22 (October 2024)	R	PU

*Delivery type: R: Document, report; DEM: Demonstrator, pilot, prototype; DEC: Websites, patent fillings, videos, etc; OTHER; ETHICS: Ethics requirement; ORDP: Open Research Data Pilot.

Dissemination Level: **PU: Public; **CO**: Confidential, only for members of the consortium (including the Commission Services)

Author(s)	Organization
Lucas Pons	ETRA I+D

Version	Date	Person	Action	Status***
0.1	18.07.2024	Álvaro Nofuentes	Table of Content	Draft
0.2	25.10.2024	Lucas Pons	Draft version ready to peer review	Draft
0.3	29.10.2024	Sara Vieira, Federico Giani, Janez Gregor Golja	Peer review	Draft
1.0	31/10/2024	Lucas Pons	Final version	Final

***Status: Draft, Final, Approved, Submitted (to European Commission).

Authors (organization)
Lucas Pons (ETRA I+D); Álvaro Nofuentes (ETRA I+D); Janez Gregor Golja (UL), Dimitris Lagos (ICCS), Themistoklis Xygkis (ICCS); Kyriakos Andresakis (ICCS); Sara Vieira (ANELL), Federico Giani (AEM).

Keywords

DSO, Topology identification, Topology detection, Low-cost Real-Time thermal rating, Dynamic Line Rating, State estimation, Fuse burn detection, Critical Point detection, DER impact

Executive Summary

This document presents the design and first version of the OPENTUNITY modules providing services and technologies for grid operators.

The summary and purpose of each of the modules is presented below:

Topology identification tool:

Aims to determine both the connections and line impedances, without knowledge of line infrastructure.

Topology detection tool:

The system operator knows the line infrastructure and their impedances and needs to determine the ones that are currently energized.

Fuse burn detection tool for early outage and islanding recovery:

In certain cases, the normal operation of a triphasic line gets disrupted when a fuse from one phase blows. The aim of the task is to detect the blown fuse through the monitoring of the voltage at end user level and the appropriate calculations.

Enhanced state estimation tool:

Aims to retrieve the unknown system state, that is, the complex voltages at all buses and connection points.

Critical point detection tool:

The aim of this tool is to detect critical points in a branch given the capacity limits of the different sections of cable in the same line.

Short term analysis of the impact of DER in the Distribution grid:

The aim of this tool is to study the voltage variations due to DER fluctuations that might impact the stability in the transient state.

Real-Time Thermal Rating (RTTR) tool:

The aim of this tool is to develop a Dynamic Line Rating algorithm based on weather condition estimation.

The modules presented will be integrated to the Advanced Distribution Management System (ADMS) developed by ETRA called **ÉTER.**

Each module is described in detail, focusing on design, implementation, and preliminary mock-ups.

Finally, the conclusion of the development performed and the next steps are explained in the "Conclusions" section.

Copyright statement

1 INDEX

List of Figures

List of tables

2 INTRODUCTION

2.1 Purpose of the document

The purpose of this deliverable is to provide a clear explanation about the developments of Task 5.2 "Upgrading topology identification and state estimation" and Task 5.3 "Low-cost Real-Time thermal rating". Firstly, an overview of how all the different modules and developments interact is provided and then, the detailed individual explanation is shown.

However, the information included in this deliverable may evolve and will be enhanced later during the refinement phase and those improvements will be reported in the deliverable D5.2 "OPENTUNITY power flow developments (v2)".

2.2 Scope of the document

The scope of this deliverable will be circumscribed to the technical description of the different modules developed under Task 5.2 "Upgrading topology identification and state estimation" and Task 5.3 "Low-cost Real-Time thermal rating". In this deliverable, the description of the design and implementation of the modules is provided, together preliminary mock-ups of the User Interface. D5.2 "OPENTUNITY power flow developments (v2)" will focus on showing the functionalities of the final version of the modules and their User Interface, without the need of explaining again the design and implementation aspects.

2.3 Structure of the document

Apart from this introductory section, the current document is structured as follows:

The document initiates its content with an overview of all the developed technologies and how they interact in order to provide valuable functionalities to the Distribution System Operator.

Then, each of the modules are described focusing on: a) Design, b) Implementation and c) Preliminary mock-ups.

Finally, the conclusion of the development performed, and the next steps are explained in the "Conclusions" section.

3 OVERVIEW OF THE DEVELOPED TECHNOLOGIES

This document presents the design and first version of the OPENTUNITY modules providing services and technologies for grid operators. The Figure 1 OPENTUNITY grid operator services architecture illustrates how these modules are related among each other and with the main distribution system components and applications.

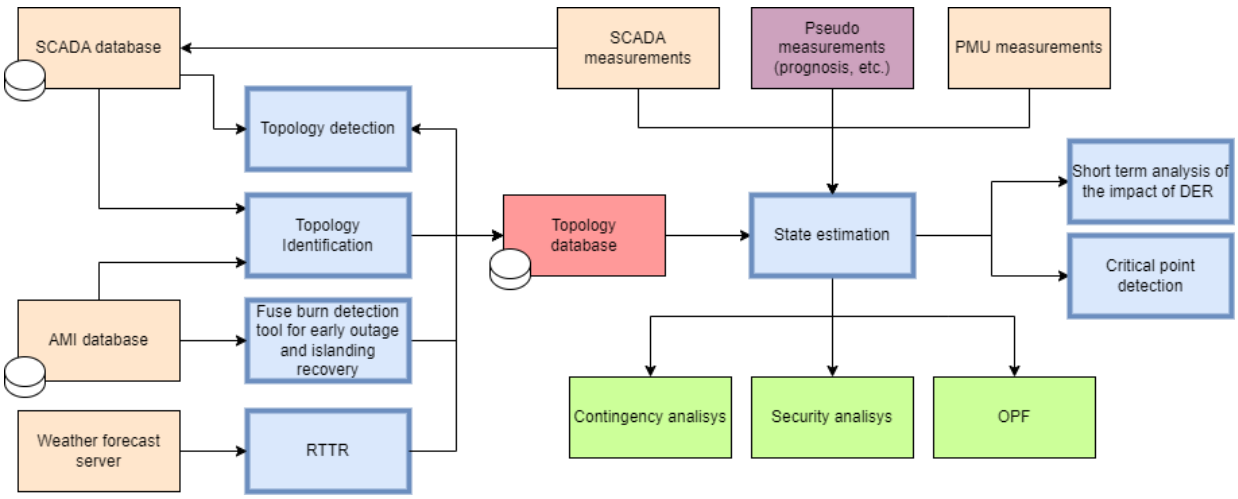


Figure 1 OPENTUNITY grid operator services architecture

The OPENTUNITY tools are marked in blue. Orange boxes represent data sources, like the SCADA or AMI databases. Green colour is for the existing services of the DSO not directly covered by OPENTUNITY tools but that can benefit from its results.

At the center of the grid management is the topology database. This database contains the structure of the grid and the characteristics of its assets.

There are four modules in OPENTUNITY that modify the structure of this topology. Two of them, the **Topology detection tool**, and the **Fuse burn detection tool for early outage and islanding recovery**, aim at identifying errors or inaccuracies on the existing topology based on the analysis of the data received from the SCADA and AMI systems.

Conversely, the **Topology identification tool** aims to determine the structure and connections of the low voltage network without any previous knowledge, just by analysing the smart meters' data received by the AMI system. This could be used to define from scratch the topology during the initial deployment and configuration of a low voltage network.

The last module affecting the topology structure is the **Real-Time Thermal Rating (RTTR) tool**. The aim of this tool is to develop a Dynamic Line Rating algorithm based on weather condition estimation.

By making use of the topology and the measurements from SCADA and PMUs, two versions of **Enhanced state estimation tools** will be developed. One that will focus on predicting the state of the unknown pseudo-measurements by making use of the historical smart meter and SCADA measurements, and the other more focused on the analysis of PMU data and the study of the most beneficial deployment option of these assets for observability purpose.

The obtained state estimation results could be used for the typical management DSO task like security analysis of the current and forecasted situation, optimal power flow calculation or contingency analysis. Two tools will be added to this set of functionalities. The **Critical point detection tool** aims at detecting critical points in a branch given the capacity limits of the different sections of cable in the same line and could be used to simulate extreme situations and assess the correct behaviour of the grid. The **Short-term analysis of the impact of DER in the Distribution grid** aims at studying the voltage variations due to DER fluctuations that might impact the stability of the grid by causing congestions or affecting the protection scheme.

The modules presented will be integrated to the Advanced Distribution Management System (ADMS) developed by ETRA called **ÉTER.**

4 MODULES' DEVELOPMENT

4.1 Topology identification tool

One problem that often affects the low networks (LV) is the incorrect topological information. Such networks are far more complex than the MV and HV networks, because they are composed of a myriad of small cable segments that are deployed following the structures of cities and towns towards the different connection points. This structure must be reflected in the topology database of the system operator, but there are situations where this topology is not accurate or does not even exist and should be defined, for distinct reasons:

- Some LV networks were deployed many years ago, and the digitalization of these networks rely on schematics and diagrams that might be inaccurate.
- The topology may have been defined by the DSO with the support of a SCADA provider selected for the monitoring of the grid, but the related SCADA product might have been discontinued or outdated, so the topology database modification or the export to a new SCADA system might be problematic.
- LV network topology change frequently due to normal power engineering activities aiming at reducing line losses, managing outages or accepting more intermittent distribution generators, but these operations might not be properly logged-in the topology database. This might end up with a topological structure completely different from the one stored by the DSO.

The **topology identification tool** will try to identify the LV network topology from scratch, just making use of the available LV data, mainly the smart metering data.

One of the duties assigned to the DSO is the collection of the power curves from the end users. This was historically done using electrical meters, but nowadays Smart Meters are almost ubiquitous, and this task is automatically done using advanced metering infrastructure (AMI) systems. Smart meters in the AMI system periodically collect data from energy users and send these data to the utility, including real/reactive power, voltage magnitude and energy consumption. Also, there are smart meters installed at the second side of each distribution transformer to measure the total power consumption of all users energized by this transformer, with the purpose of managing line losses. The **topology identification tool** (also known in literature as topological learning) will make use of the residential data at feeder buses received in the AMI to try to build the network topology from scratch.

There are different approaches in literature that investigate on the topic of topological learning. Most of them make use of the Voltage Correlation (VC). VC uses one of the most basic principles of electrical engineering. When a current flows through an impedance, a voltage drop occurs. Hence, in a power system, a similar voltage profile over time suggests that the two metering sites are closely connected electrically.

4.1.1 Design

The topology identification tool is a process that is meant to be running off-line. It does not require up-to date measurements, but just accurate and time-synchronised historical data.

Within OPENTUNITY, the tool is included as a feature of the broader scope 'OPENTUNITY topology converter' tool. This tool aims at converting among different known topological definition formats. A new feature will be added to the tool for identifying the topology making use of the historical meter data. The next picture illustrates how this process is included in the topology converter tool (that will be reported in Deliverable 5.5):

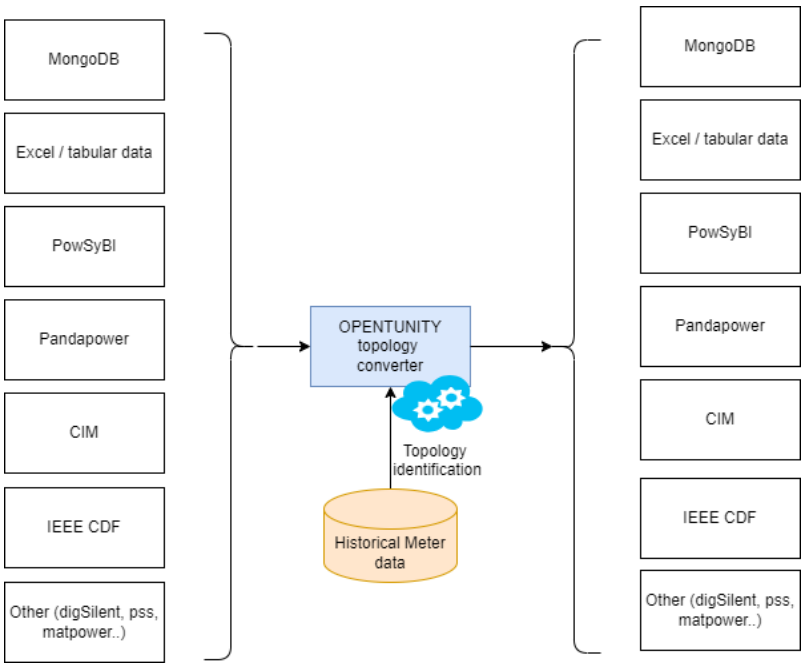


Figure 2 Topology identification within topology converter tool

The identified topology could then be used by the DSO operator for the operation of the grid. The Figure 4 Topology identification related communication mechanismspresents the entire process from the smart meters power curve capturing to the DSO management.

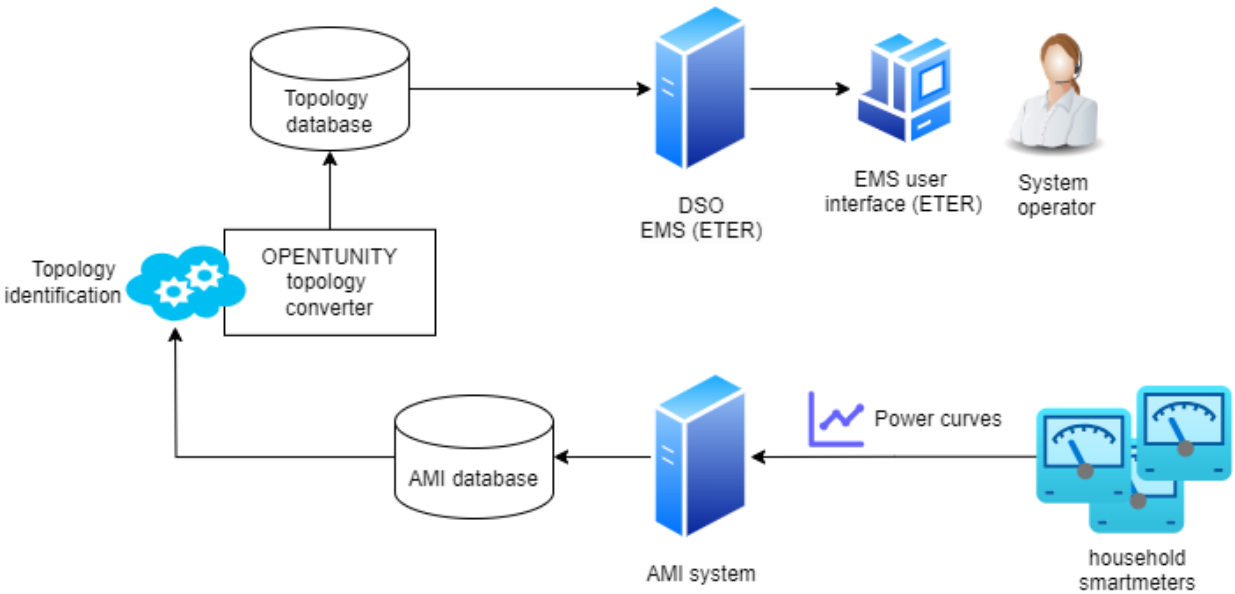


Figure 3 Topology identification algorithm in DSO environment

From the point of view of the deployment and communication, the historical problem of lack of appropriate communication channels is solved by making use of the AMI power line communication (PLC) infrastructure. This technology utilizes existing electrical power lines to transmit data and communication signals between the smart meters and the control centre (AMI system). More in detail. each smart meter features a PLC modem that modulate and demodulate data signals onto and off the power lines. This data is collected by different data concentrators, often located at substations or other central points in the network, which forwards this data to the control center.

The process of AMI network creation is tightly coupled to the smart meter rollup and configuration done by the workforce operators of the DSO. When connected to a LV line and configured, the smart meters start measuring electricity consumption and properties of the connection point (normally a household) and sending this data over the PLC network. Normally this data is used for billing purposes, but in the case of OPENTUNITY, it will be used to feed the topology identification tool and build the topology.

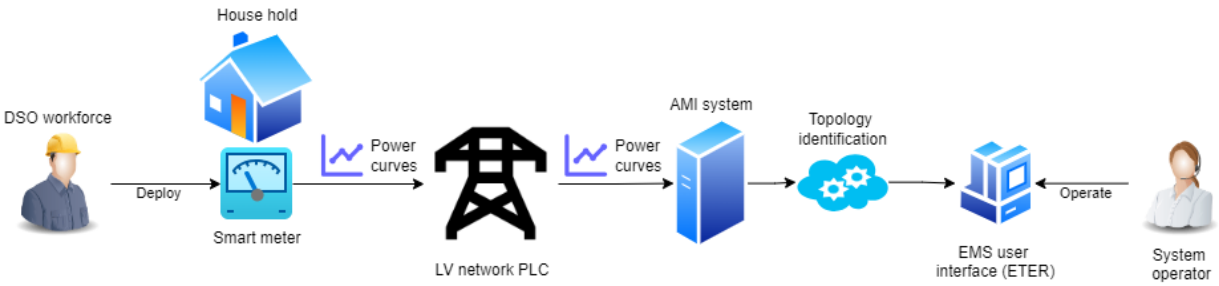


Figure 4 Topology identification related communication mechanisms

4.1.2 Implementation

The tool has been developed as a functionality embedded in the topology converter tool. This tool has the form of a Python program that can be invoked through the command line using the appropriate parameters.

Depending on these parameters, the appropriate topological conversion is executed, from one format to another. In the case of this tool, the input parameters will be:

- The connections details to the smart meter database
- The identifiers of the smart meters in the network to identify
- The name of the network to create
- The format and location of the resulting topology

After analysing all smart meter data and applying the aforementioned techniques, the resulting topology will be returned in the format specified in the execution parameters.

4.1.3 Mock-Ups

The tool will not have a user interface, as the output of the tool is the topology of the LV network. The representation of this topology in the ETER tool is presented in the section 4.2 'Enhanced state estimation tool'

4.2 Enhanced state estimation tool

Power system state estimation (PSSE) is a crucial function within the energy management system (EMS), offering operators reliable snapshots of the operating system conditions based on real-time field measurements [1], [2]. In general, a state estimation (SE) module is capable of converting unrefined measurement data into structured information about the system state. For a power grid, knowing its real-time state requires estimating two key components

a) Its operating topology, i.e., the configuration of buses (nodes) and their interconnected lines (branches), and
b) Its state vector, which typically comprises either the positive sequence complex bus voltages or branch currents [3]. To discriminate the individual problems, the former one is referred to as topology identification (TI) and described in section 4.1 - Topology identification tool. Using the estimated topology and state vector, all the other grid variables (mainly referring to active/reactive power flows and injections) can then be evaluated, thus, providing the operators with vital insights into system behaviour. Concisely, the SE output can be exploited for real-time grid operation and control, e.g., volt-var optimization, fault detection, transmission-distribution interface coordination, cybersecurity etc., and planning tasks, such as proactive decision-making, contingency analysis, support of energy markets, forecasting, etc [4].

Since its introduction at late 1960s by Schweppe, PSSE has been formulated and solved as a model-based optimization problem [5]. Specifically, a measurement model based on Kirchhoff's circuit laws has been employed to represent the measured electrical quantities as functions of the state vector, while also incorporating the associated measurement errors. By applying maximum likelihood estimation (MLE) and assuming these errors follow a normal (Gaussian) distribution, the SE problem amounts to a weighted least-squares (WLS) task [2], [5]. Given that the measurement model is generally nonlinear, the WLS task can be iteratively solved using the Gauss-Newton algorithm.

The WLS method has long been the most widely employed approach for PSSE in energy control centers (ECC) at power transmission level [2], [3]. Moreover, it remained a focal point of related academic research until early 2010s. Research efforts primarily aimed to optimize PSSE performance to meet the rigorous standards required for real-time transmission system monitoring, while also developing viable distribution system state estimation (DSSE) algorithms, since the medium and low voltage (MV, LV) parts of the power grids have practically been unmonitored due to their passive behaviour [6]. This scarcity of measurements problem for state estimation calculation can be addressed using various techniques. This section presents one such technique: **ML-assisted estimation of statuses from unmonitored injections**. The next section describes an alternative technique: **Enhancement in the DSSE through the installation of PMUs and micro-PMUs in selected network locations**.

Given a limited set of power measurements acquired by supervisory control and data acquisition (SCADA) and distribution automation systems, state estimation aims to recover the unknown system state, that is, the complex voltages across the network. To enhance observability DSSE has to rely on the so-called **pseudo measurements** [7], that can be generated via load, generation, and voltage forecasting tools. In this OPENTUNITY tool, the pseudo measurements generation technique for magnitudes such as the forecasted loads and voltages is postulated via deep recurrent neural networks (RNNs), that are capable of capturing complex nonlinear dependencies present in time

series data. Subsequently, the DSSE will be fed with this pseudo measurements and the accuracy of the results will be assessed with the observed measurements.

4.2.1 Design

Consider an unbalanced distribution network comprising $N+1$ buses indexed by $n \in N := \{0,1,\dots,N\}$, and assuming the distribution grid is functionally radial with the substation bus numbered by $n = 0$.

To perform DSSE, we measure M system variables collected in the vector $z := [z_1,\dots,z_M]^T$, being $M < N$. DSSE aims to retrieve the state vector $v := \left[v_1^r, v_1^i, \dots, v_N^r, v_N^i\right]^T \in R^{2N}$ from the generally noisy and uncomplete vector z. v^r and v^i stands for the real and imaginary parts of the complex voltages.

Learning-based DSSE seeks the function mapping from the measurement vector z to v based on historical/simulated data. As the relationship between z and v is complex, the function mapping from z to v is nonlinear.

For this enhanced state estimation tool described, we will make use of a particular deep neural network technology, the **Physics-Aware Proximity Linear Neural Network (Prox-Linear Net)**. In the realm of power system state estimation (PSSE), this technique has been demonstrated empirically to be successful in estimating the states of transmission networks [8].

The Prox-Linear Net integrates the flexibility of neural networks with the robustness of proximal gradient methods, creating a powerful tool for tackling the complexities of PSSE. Traditional neural networks excel in modelling non-linear relationships, but they can struggle with constraints and non-smooth optimization problems. This is where the proximal gradient methods come into play. These methods are optimization techniques that extend the capabilities of gradient descent to handle functions that are not smooth or have specific constraints. In our particular case, the specific constraints are dictated by the physical laws, ensuring that the outputs are physically plausible. Examples of physical laws constraining the results could include Kirchhoff's laws in electrical circuits, conservation of mass and energy in fluid dynamics, etc.

The Prox-Linear Net architecture is as follows: The input layer takes in the measurements from the power system, such as power flows, power injections, and voltage magnitudes. These measurements are then processed through multiple hidden layers. Each hidden layer performs a linear transformation followed by a non-linear activation function, typically a ReLU (Rectified Linear Unit). This setup allows the network to capture and model the complex, non-linear relationships between the measurements and the state variables.

Mathematically, the forward pass through a hidden layer can be described by the equation:

$$\boldsymbol{a}^{(l)} = \sigma\left(\boldsymbol{W}^{(l)}\boldsymbol{a}^{(l-1)} + \boldsymbol{b}^{(l)}\right)$$

Where $\boldsymbol{a}^{(l)}$ is the activation of the l-th layer, $\boldsymbol{W}^{(l)}$ is the weight matrix, $\boldsymbol{b}^{(l)}$ is the bias vector, and σ is the activation function.

What sets the Prox-Linear Net apart are its proximal layers. These layers incorporate proximal operators—mathematical functions designed to enforce constraints and handle non-smooth optimization problems. The proximal operator for a function f is defined as:

$$prox_f(\boldsymbol{v}) = \arg min_u \left(f(u) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{v}\|_2^2\right)$$

By integrating these operators, the Prox-Linear Net can effectively manage the constraints inherent in power systems, such as physical limits on state variables, and improve the stability and speed of convergence during training.

The final output layer of the network maps the processed information to the estimated state variables, providing the crucial data needed for real-time monitoring and control of the power system.

The advantages of this approach are manifold. By combining the strengths of neural networks and proximal gradient methods, the Prox-Linear Net achieves higher accuracy in state estimation. The proximal operations not only enforce constraints but also enhance the network's robustness and convergence speed, making the training process more efficient. Moreover, the architecture is scalable, capable of handling large power systems with numerous measurements and state variables without sacrificing performance.

To illustrate, consider a power system with three measurements and two state variables. Using historical data, the Prox-Linear Net can be trained to learn the mapping from these measurements to the state variables. The state estimation problem can be formulated as finding the state vector $\boldsymbol{x}$ that minimizes the difference between the measured values $\boldsymbol{z}$ and the estimated values:

$$\boldsymbol{h}(\boldsymbol{x}): min_x \|\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})\|_2^2$$

Once trained, the network can quickly and accurately estimate the state from new measurements, providing real-time insights that are essential for effective power system management.

In practice, implementing a Prox-Linear Net involves several steps. First, historical measurements from the power system are collected and pre-processed to ensure they are suitable for training. The network is then trained on this data, learning the complex relationships and constraints. The loss function used during training can be expressed as:

$$\boldsymbol{\mathcal{L}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{i+1}^{N} \|\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i\|_2^2$$

Where $\boldsymbol{y_i}$ is the true state vector, $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ is the estimated state vector from the network, and N is the number of training samples.

This general loss function can be enhanced by including terms that represent physical constraints so that the loss function will penalize violations of power flow equations. In our case, we will embed Kirchhoff's Laws:

- **Kirchhoff's Current Law (KCL)**: Ensures that the sum of currents entering a node equals the sum of currents leaving the node.
- **Kirchhoff's Voltage Law (KVL)**: Ensures that the sum of voltage drops around any closed loop in a network equals zero.

The new loss function can then be modified in such a way:

$$\boldsymbol{\mathcal{L}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{i+1}^{N} \|\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i\|_2^2 + \lambda \sum_{j=1}^{M} \left\|KCL_j(x)\right\|_2^2 + \mu \sum_{k=1}^{L} \|KVL_k(x)\|_2^2$$

Where λ and μ are regularization parameters, KCL_j and KVL_k are the residuals of Kirchhoff's laws for nodes and loops, respectively.

Once trained, the Prox-Linear Net can be used to estimate state variables from new measurements, aiding in the real-time monitoring and control of the power system. The gradient descent update rule with a proximal operator can be written as:

$$\boldsymbol{x}^{k+1} = prox_{\lambda f}\left(\boldsymbol{x}^{k} - \alpha \nabla g(\boldsymbol{x}^{k})\right)$$

Where $\boldsymbol{x}^{k}$ is the state vector at iteration k, α is the learning rate, $\nabla g(\boldsymbol{x}^{k})$ is the gradient of the smooth part of the objective function, and λ is a regularization parameter.

Upon learning the weight coefficients during the off-line training stage, the Prox-Linear Net can be employed for inferring the distribution system states in real time. This mechanism has proved to enhance the convergence speed and stability and also ensure the network's predictions are not only accurate but also physically plausible.

Unlike transmission networks where metering devices are adequate, distribution grids suffer from partial observability due to limited instrumentation. To enhance observability, data vector z has to be augmented with **pseudo measurements**. In this tool we advocate for a predicting pseudo measurements technique based on **deep recurrent neural networks (RNN)**.

RNNs are a class of neural networks designed to handle sequential data, making them well-suited for time-series predictions inherent in power systems. They can capture temporal dependencies and patterns in historical data to forecast future states. An RNN processes input sequences by maintaining a hidden state vector that evolves over time:

$$\boldsymbol{h}_t = f\ (\boldsymbol{x}_t, \boldsymbol{e}_t, \boldsymbol{h}_{t-1}, \boldsymbol{W}, \boldsymbol{b})$$

Where:

- $\boldsymbol{h}_t$ is the hidden state at time t,
- $\boldsymbol{e}_t$ are the values of the exogenous variables at time t,
- $\boldsymbol{x}_t$ is the input at time t,
- $\boldsymbol{w}$ and $\boldsymbol{b}$ are the weight matrices and bias vectors,
- f is the activation function,

The exogenous variables, or external parameters, can significantly enhance the performance of RNNs in state predictions. These variables provide additional context that helps the RNN capture more complex relationships and dependencies in the data. The selection of the most appropriate and important variables to include depend on the specific case, but some of the typically relevant exogenous variables are:

- Time of Day: Variations in load and generation patterns.
- Seasonality: Seasonal changes affecting power consumption and generation.
- Weather Conditions: Temperature, humidity, wind speed, and solar irradiance.
- Electricity Prices: Market-driven price variations influencing demand and generation.
- Economic Activity: Industrial and commercial activity levels.

The RNN can predict future states $\hat{\boldsymbol{x}}_{t+1}$based on past states:

$$\hat{\boldsymbol{x}}_{t+1} = g(\boldsymbol{h}_t, \boldsymbol{W}_o, \boldsymbol{b}_o)$$

Where g is the output function, and $\boldsymbol{W_o}, \boldsymbol{b_o}$ are the output weights and biases.

The predicted states $\hat{\boldsymbol{x}}_{t+1}$ from the RNN serve as pseudo-measurements, augmenting the traditional measurement vector $\boldsymbol{z}$. This augmented measurement vector $\boldsymbol{z_{aug}}$ is given by:

$$\boldsymbol{z}_{aug} = \begin{bmatrix} \boldsymbol{z} \\ \hat{\boldsymbol{x}}_{t+1} \end{bmatrix}$$

This technique could be applied to different non-metered elements in the grid, so that we could predict the statuses at different points and levels. The scenarios considered in the tool are:

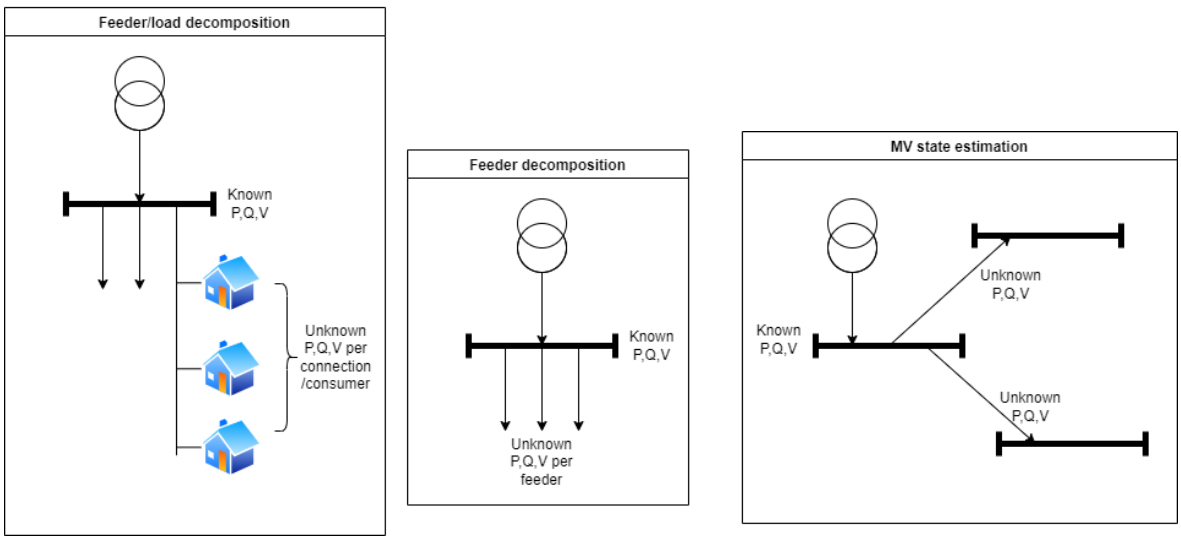


Figure 5 Pseudo-measurement calculation scenarios

Considering these scenarios, AI models for generating pseudo-measurements for the following elements will be built:

- For each connected **load or generator**, one for modelling the active power.
- For each connected **load or generator**, one for modelling the reactive power.
- For each **feeder**, one for modelling the whole feeder active power.
- For each **feeder**, one for modelling the whole feeder reactive power.
- For each **MV line**, one for modelling the whole MV line active power.
- For each **MV line**, one for modelling the whole MV line reactive power.

4.2.2 Implementation

For the **DSSE calculation**, the Prox-Linear Net models will be built based on TensorFlow/Keras and using Python programming language. The process will take as inputs the instantaneous known electrical measurements of the grid. These can be:

- The current values (the most updated values read).
- The values of a certain past period (stored in database).
- The values forecasted for future period.

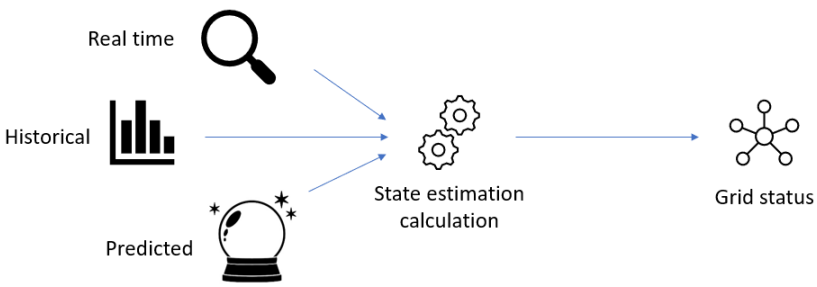


Figure 6 DSSE calculation variants.

This data will be obtained from the following systems:

- The **distribution SCADA** system will provide RT status of main MV assets. In the case of Spanish pilot, this will be received as MQTT messages.
- **Usage points/loads** consumption profiles data will be received from AMI system. This will include data from households obtained by smart meters.
- The instantaneous electrical quality measurements (P, Q, V, I) are obtained from **LV network** by configuring test cycles in the PLC concentrator.

Other data required for the state estimation will be gathered from different sources:

- **Current Weather** will be retrieved from Open Weather [9]
- Calendar data manually will be introduced by operators

The process will have the form of a microservice that will calculate the state estimation from the input parameters. These parameters will be the topology of the network, the known network elements real time measurements and some network elements known measurements from past periods.

The microservice could be invoked manually from the DSO EMS tool (ETER) for testing specific scenarios or network reconfigurations, but it will be invoked automatically by two different processes:

1. The real time processor will gather current measurements from network elements and will request the calculation of the state estimation every few seconds.
2. The short-term forecast will predict the evolution of the network elements measurements for the next hours and will request the calculation of state estimation based on this forecast every few minutes. This will be used to warn operator upon the prediction of a security problem for the near future.

The output will be the set of measurements (some of them estimated) for the all the elements in the network. This result can feed several DSO power flow-related tools:

- Optimal power flow
- Security and contingency analysis
- Topology detection
- Critical point detection
- Short term analysis of the impact of DER in the distribution grid

For the calculation of the **pseudo measurements** associated to network elements, several RNN models will be built on TensorFlow/Keras using Python programming language: One for each load/generator, one for each feeder and one for each MV line. This will generate a huge number of individual models, and that the reason for using MLFlow [10] as a repository of models.

MLflow is an open-source platform designed to manage the complete machine learning lifecycle. It can be integrated with various machine learning libraries and tools and provides a central repository for models that can be used in the training and exploitation of the AI models. This is an example of some OPENTUNITY models deployed in the platform:

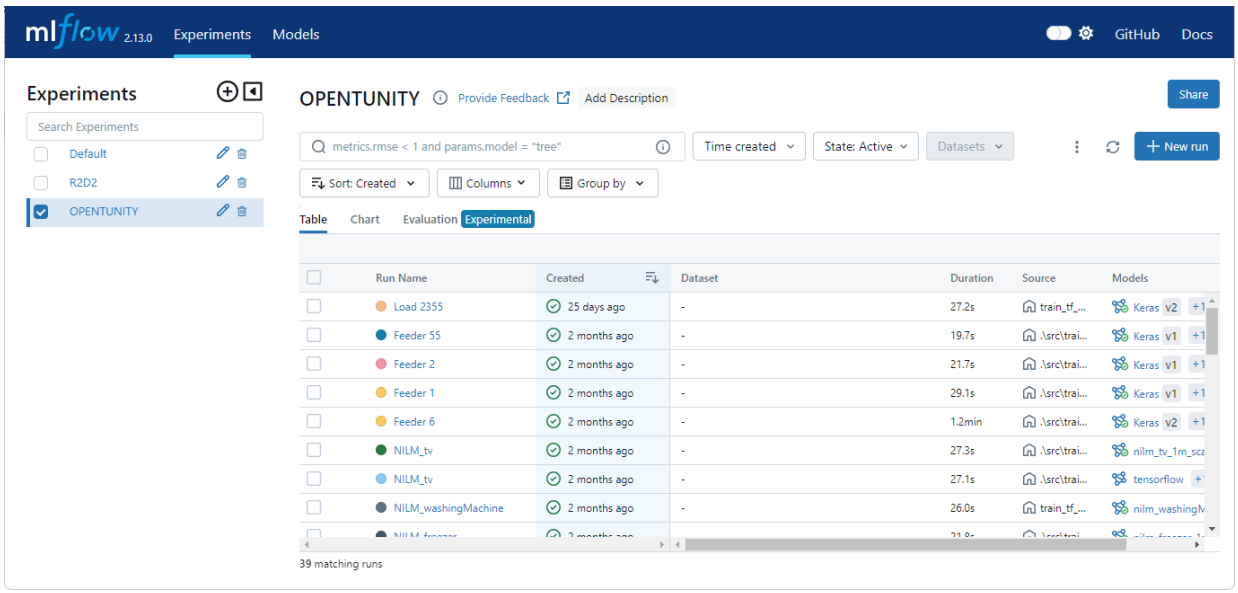


Figure 7 MLFlow user interface

The process for creating the machine learning models is composed of the following phases:

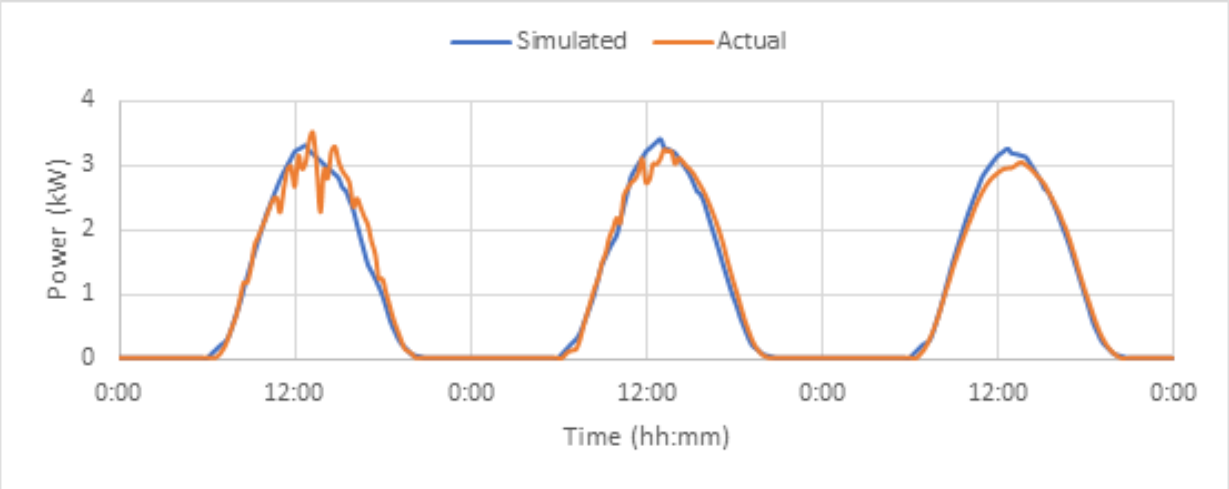
1. **Data Cleaning**: In this phase, the raw data collected for the project is carefully processed to eliminate errors, inconsistencies, and missing values. This involves tasks such as imputing missing data, handling outliers, and rectifying any inaccuracies that could adversely affect model performance. Data cleaning ensures that the subsequent analysis is based on accurate and reliable data.

2. **Feature Extraction**: Feature extraction involves selecting and transforming the relevant attributes or features from the cleaned dataset that will serve as inputs for the machine learning models. During this phase, domain knowledge is crucial for identifying the most informative features, which can sometimes mean either reducing the data's complexity or introducing new variables to account for previously overlooked factors. Extracted features should capture meaningful patterns and relationships within the data, improving the model's ability to generalize from the training data to new, unseen data.

3. **Model Creation**: During the model creation phase, we choose and apply machine learning algorithms to the prepared dataset, which includes the features used for training and the target variable—the active and reactive power. Our objective here is real-time anomaly detection by predicting these measurements. Various algorithms, such as regression, decision trees, neural networks, or ensemble methods, are explored to identify the best fit for the problem. The dataset is divided into training and validation sets for model training and performance evaluation.

4. **Model Deployment**: Once a satisfactory model is created and evaluated, the next step is to deploy it into a production MLflow environment. This phase involves integrating the trained model into the system where it will be used to make predictions on new data. Model deployment requires careful consideration of factors like scalability, real-time performance, and robustness. Monitoring the model's performance in production and implementing mechanisms to retrain or update the model as new data becomes available is also part of this phase.

The accuracy of the models strongly depends on the quality and availability of the input data, these are some examples of the model results for the pseudo measurement calculation compared to the real data:



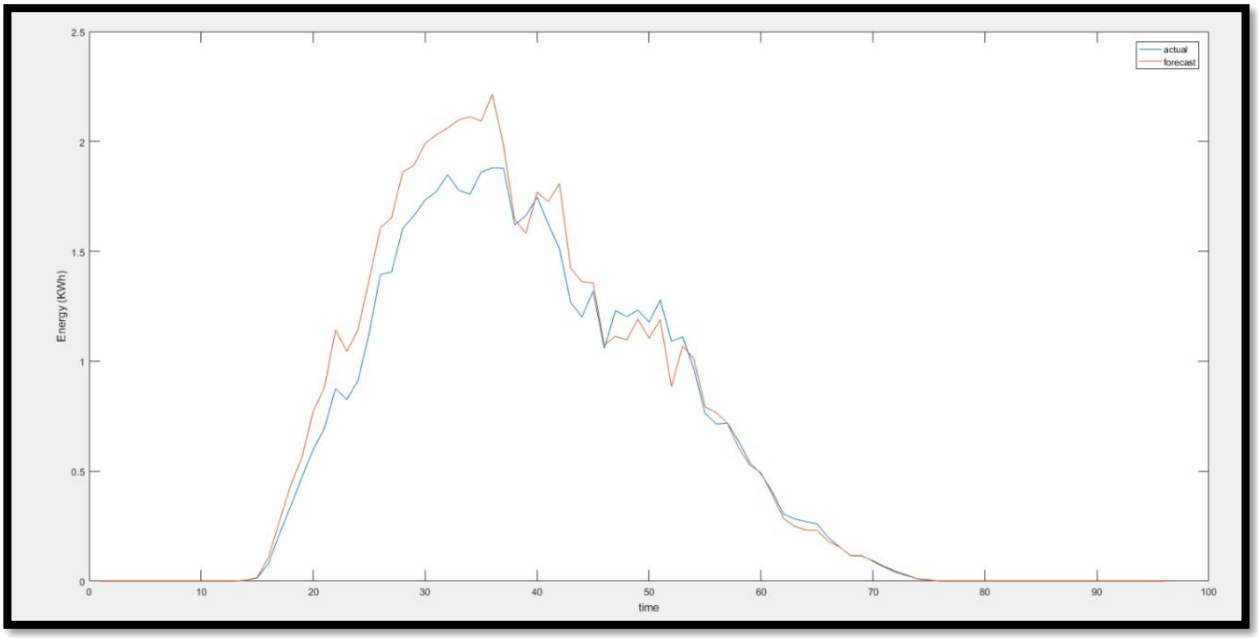


Figure 8 Examples of forecasted load consumption.

4.2.3 Mock-Ups

The DSSE itself does not have a graphical output as it just estimates a number of measurements. Nevertheless, its results are presented in other DSO tools that will indirectly reflect the output of the tool. In the case of this tool, the ETER ADMS software for DSOs (developed by ETRA) will be used. This software integrates the DSSE results and presents them in a graphical way. These is a screenshot of the tool:

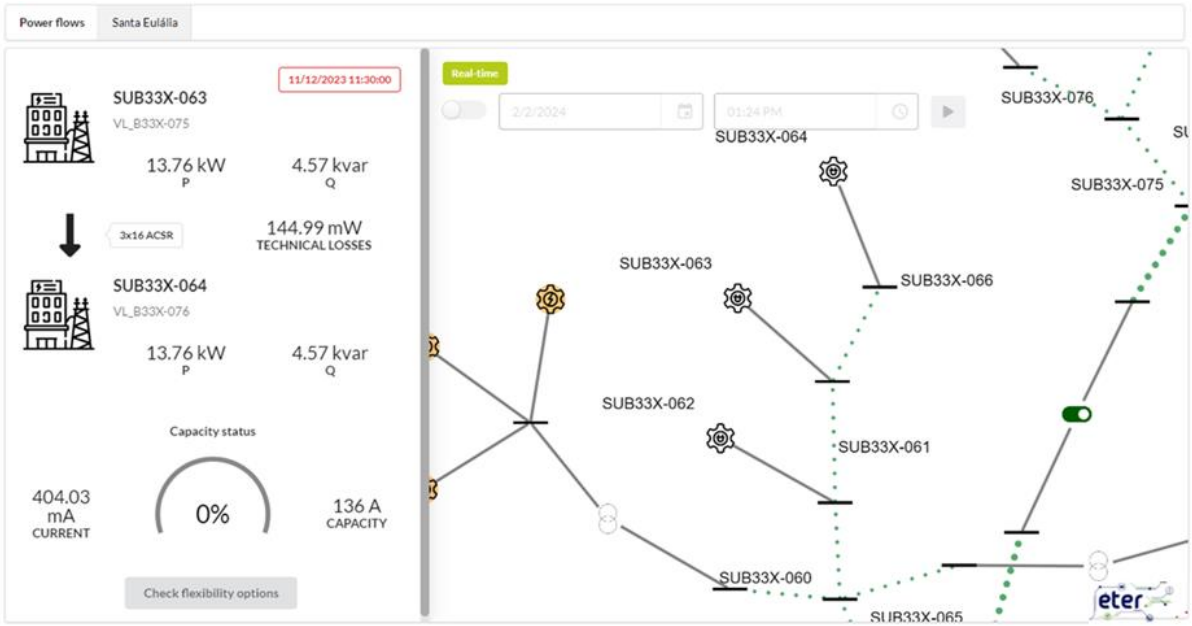


Figure 9 ETER GUI for monitoring electrical networks.

The topology is presented in the ETER tool, the measurements of the network elements are obtained from the DSSE and presented appropriately, like the line losses.

4.3 Enhanced state estimation tool (Greek demo version)

As described in previous section DSSE calculation is hindered by the lack of sensors in MV and specially in LV networks. Over the last decade, the attention of utilities and operators has turned to the real-world application of synchrophasor technology, which has marked a significant milestone in power grid measurement systems [11]. Currently, phasor measurement unit (PMU) infrastructures are prevalently deployed at transmission systems. These advances create a new operation framework for PSSE worldwide. Ultra-fast input streams of highly accurate, synchronized voltage and current phasor measurements can enhance PSSE performance, enabling an unprecedented level of real-time power grid monitoring. Yet, at the same time, several challenges mainly stemming from the emerging multi-rate data environment, including measurements from SCADA systems, PMUs, intelligent electronic devices (IEDs) etc., and the massive volume of PMU information raising big data issues, need to be addressed to build reliable PSSE functions [12] [13].

In parallel, low-cost PMUs are gradually developed to enable the introduction of the technology into distribution networks [14]. Given that the feeders originating from primary substations have typically lacked sufficient instrumentation, the installation of PMUs at strategic MV nodes will significantly boost the availability of real-time measurements for DSSE [15]. This advancement is timely as the role of DSSE in contemporary, active distribution networks (ADNs) with high penetration of distributed energy resources (DER) and customer engagement, is crucial for efficient real-time grid operation [16].

However, the issue of low observability that has traditionally plagued distribution networks for the past decades, is still present [17]. On the one hand, the number of MV nodes is so huge that it is neither technically feasible nor economically justifiable to install metering units at all of them. On the other hand, the measurement heterogeneity is even more intense, as most of the available data downstream primary substations are derived from smart meters (SMs) at customer premises with

their update rates and accuracy being drastically lower than that of PMU and SCADA recordings. Hence, unless pseudo-measurements. i.e., artificially generated data, are utilized for unmeasured nodes (to the detriment of DSSE accuracy) and sophisticated techniques are exploited to reconciliate multi-rate and unsynchronized actual measurements, the dataset is incomplete, thus, rendering the SE problem unsolvable using the WLS method (or other model-based approaches) [18] [19].

In this framework, the employment of machine learning (ML) for enhancing existing WLS-based DSSE algorithms or constructing from scratch new ones, has been regarded a well-suited solution [20] [21] [22]. In particular, utilizing ML methodologies to develop DSSE procedures offers the following merits:

- the observability conditions imposed in the context of conventional approaches do not apply, thus, the SE problem can be solved without any need for pseudo-measurements
- since ML schemes are essentially training-based models relying on data-driven processes, they can exploit the abundant raw data and information obtained from the various, operating measurements systems in an efficient and robust manner
- provided that a limited set of selected measurements is sufficient to implement ML based DSSE, synchronized data from strategically placed PMUs can exclusively be utilized, thus, bypassing the problem of lack of synchronization.

The majority of the proposed ML methodologies for DSSE leverage deep learning (DL) to train and implement deep neural networks (DNNs), a subclass of artificial NNs [23]. NNs are capable of handling the nonlinearities and intricate relationships among power system operating parameters, providing high levels of accuracy and computational efficiency. Recently, graph neural networks (GNNs) have also gained significant attention in this research area due to their ability to process data structured as graphs, which aligns well with the topological representation of power systems [24].Both DNNs and GNNs are promising approaches for estimating the operating topology and state vector of ADNs.

4.3.1 Design

Recognizing the substantial benefits of applying of ML to the challenging task of DSSE, the proposed approach principally aims to utilize DNNs for the development of an enhanced SE tool that is designed specifically for ADNs. This tool integrates functionalities for both TI and SE, constituting a comprehensive solution for quality DSSE. The basic concepts of the approach can be found in [25] . Moreover, synchronized data from selected network locations comprise the primary input of the tool. Sophisticated ML techniques, namely, principal component analysis and random forest, are deployed for the selection of the locations for PMU installation, targeting an optimal trade-off between the number of the installed units and the desired levels of TI and SE accuracy [26]. To effectively train the ML models used within the proposed framework, the dataset is generated by means of a thorough procedure based on extensive Monte Carlo trials.

4.3.1.1 Utilized ML methods for DSSE

4.3.1.1.1. Deep neural networks

A standard ANN consists of multiple layers, each with its own parameters, which are updated during training to minimize the error between the target and predicted outputs. It typically includes an input layer that receives the features (data), followed by one or more hidden layers, where computations allow the network to identify patterns and relationships among the features. The output layer

provides the final result, based on the specific problem. Adjacent layers are connected through weights, and in a simple ANN, each neuron in one layer is linked to every neuron in the next.

Neurons, the building blocks of each layer, process information by receiving a weighted sum of inputs from the previous layer, adding a bias, and passing the result through an activation function. The output is then sent to the next layer. During training, the network learns to predict the output by passing input data forward and calculating the error between the predicted and target values. The weights and biases are updated through backpropagation, which uses the chain rule of calculus to adjust each parameter in reverse order.

The difference between a simple ANN and a DNN is the depth of the architecture. For DNNs, the number of hidden layers is multiple (typically 1 or more). This allows these structures to identify more complex patterns hidden within the input data. In practical terms, DNNs often have several hidden layers, sometimes ranging from 5 to hundreds or even thousands of layers, depending on the specific architecture and task. These deep architectures enable the network to learn complex and hierarchical representations of data, capturing intricate patterns and relationships.

The applicability of ANNs to a wide variety of problems is founded upon the universal approximation theorem, which essentially suggests that a feedforward ANN (where information moves in one direction—from the input to the output layer) with at least one hidden layer, using a non-linear activation function, can approximate any continuous function to a desired degree of accuracy, given sufficient neurons in the hidden layer.

In particular, DNNs are routinely used to approximate minimum mean squared error (MMSE) estimators in the context of Bayesian inference, which amounts to minimizing the error between the predicted (or estimated) and target output quantified by the following metric [27] [28]:

$$\min_{\hat{x}} \mathbb{E}(\|x - \hat{x}(z)\|^2) \tag{1}$$

where x and $\hat{x}$ are the target and predicted output, respectively, z denotes the input data and $\mathbb{E}(\cdot)$ is the mean value operator.

In this framework, DL can comprise an effective approach to model and solve TI and SE tasks [29] [30]. On one hand, the TI task can be conceptualized as a classification problem where each topology is treated as a different class. A DNN is then employed to determine the operating configuration of an ADN. The primary input of the DNN is a subset of the line current flows, where a nonzero value indicates that the corresponding branch is active. The output represents the statuses of the grid switching devices, with a value of one indicating a closed switch (branch in operation) and zero indicating an open switch (branch out of operation).

On the other hand, the SE task can be modelled as a regression problem. A separate DNN is utilized to estimate the grid state vector. The input of this DNN is the estimated topology by the DNN based TI and all the available measurements, such as voltage magnitudes/ angles, line current flows/ injections etc. The output is the set of nodal complex voltages or, alternatively, of branch complex currents.

Both DNNs need to be trained using a dataset that captures a multitude of the grid operating conditions considering different topologies, loading, DER generation etc. Apparently, this dataset should be aligned with the availability of real-world measurements during the actual operation of the grid. The performance of the DNNs is likely to degrade in the presence of poorly synchronized or low accuracy data. As previously stated, the deployment of synchrophasor technology can contribute to

high performing DNN based DSSE, even with a relatively small number of measurements compared to the size of the grid. Optimally placed PMUs designed for ADNs, such as microPMUs and D-PMUs, can provide enough data to support the training and operation of the DNNs.

4.3.1.1.2. Principal component analysis (PCA)

PCA is a dimensionality reduction technique that transforms high-dimensional data into a smaller set of uncorrelated variables called principal components. It captures the most significant variance in the data by projecting it onto new axes, with the first few components retaining the majority of the original data variability. In this way, the initial features of the studied problem are transformed to new fewer ones, retaining a percentage information of the former.

The capabilities of PCA can be leveraged for feature selection in DNNs, which is the process of identifying and selecting the most relevant input features from a dataset to improve the performance, efficiency, and generalization of the model. Specifically, given that the input of the DNN consists of an initial set of *N* features, PCA transforms these into *N* new, independent features. The contribution of each original feature to each new, transformed feature can then be calculated. By aggregating (summing) these contributions across all transformed features, a contribution score for each original feature is obtained. Based on this list, features with the highest scores can then be selected as the most relevant input for the DNN.

In view of the DNN based DSSE, the problem of optimally allocating PMUs over grid nodes is modelled as a feature selection task. In most cases, an installed microPMU or D-PMU (for example, at an MV/LV transformer) measures the nodal voltage phasor (angle and magnitude) and one current phasor through an incident branch. Hence, each PMU is assumed to deliver four measurements. By treating each phasor as a different input feature for the DNNs, PCA can be used to assign a score to each PMU measurement. This ranking essentially helps determine the optimal PMU locations. The measurements with the highest scores are more beneficial to DNN performance and, thus, should be selected. In case a measured angle is selected, the associated magnitude is chosen as well and vice versa. For the TI task, only current measurements are considered as candidates for selection, whereas both voltage and current phasors are examined for the SE task.

4.3.1.1.3. Random forest (RF)

RF is a powerful ensemble learning method used for classification and regression tasks. It operates by constructing a multitude of decision trees during training and outputs the mode (classification) or mean prediction (regression) of the individual trees. Each tree in the forest is built using a random subset of the training data and a random subset of features. It is often used as a standalone model for classification and regression tasks due to its robustness and ability to handle large datasets with high dimensionality.

Thanks to its ability to evaluate feature importance, RF is also used as a data preprocessing technique. In particular, RF can assess the importance of each feature in the dataset. Hence, features with higher importance scores are considered more informative and may be prioritized for further analysis, while less important ones can be discarded as part of feature selection. Also, by ranking features based on their importance scores, RF can be used to reduce feature space by selecting only the most informative features for subsequent modelling. In this way, the computational complexity of the associated task is reduced and the performance of ML algorithms in the next steps can be improved.

RF is also utilized to perform feature selection for the proposed DNNs. It generates a list of ascending importance metrics of all input features, that is, PMU measurements, thus, allowing their optimal selection. Typically, the input features are chosen independently from each other. However, in our case, if a phasor magnitude is selected, the corresponding angle must also be included. In order to decide on the pair of measurements instead of single features, the importance metrics of the phasor angle and magnitude are aggregated to obtain a combined importance score. This procedure provides the importance ranking of PMU measurements. Similar to PCA, current phasor measurements are exclusively studied for feature selection in the TI task, while both voltage and current phasors are assessed in case of the SE task.

4.3.2 Implementation

The enhanced SE tool is designed as follows:

- Scheduling stage (offline learning)
    1. Gathering of information about the ADN: topology, switching devices, line capacities, load profiles, historical measurement data from SCADA, SMs, advanced metering equipment etc.
    2. Dataset generation for the training-based ML models
    3. Optimal selection of PMU measurements via RF- or PCA-based feature selection
    4. Training and evaluation of the DNNs for TI and for SE in a sequential manner.
- Operation stage (real-world application)
    1. Installation of actual PMU devices at the grid locations dictated by RF or PCA
    2. Execution of TI and SE using the corresponding, trained DNNs in a sequential manner.

A detailed implementation scheme has been devised for the scheduling stage of the proposed SE tool, encompassing the extraction of the dataset, the feature selection procedure and the training of the DNNs which carry out the TI and SE tasks. The scheme is described in the sequel.

4.3.2.1 Dataset generation

To generate a comprehensive dataset for feature selection and model training, an extensive Monte Carlo trial scheme is used. The scheme can account for various operating scenarios for the ADN under study, incorporating multiple topologies, loading conditions and DER generation. The conducted trials are based on actual field measurement data, such as SCADA, SMs etc., to ensure that the dataset reflects realistic conditions.

To consider different load scenarios, the following algorithm is devised:

1. The active power loads of the grid obtained from the DSO database (nominal transformer capacity or average values from load profile) are set as the base values for calculations.
2. Every bus is assigned a random number between 0 and 1 (using a uniform distribution).
3. In case the number assigned to the bus is greater than 0.5, its active power load is obtained by assuming a uniform distribution with mean value equal to the corresponding base value (step 1) and standard deviation of 30% of the mean value.

4. Provided that all loads operate at a fixed power factor, the corresponding reactive power are directly computed from the active power value using this power factor.

The above steps are repeated 1000 times for each assumed topology. Each time, after the random updates of the loading conditions are performed, power flow analysis is carried out to acquire the nodal voltage and branch current phasors. Random noise following normal distribution with standard deviation of 0.1% for both magnitude and phase angle, is subsequently added to the above phasor data, in order to simulate the actual measurement data provided by PMUs.

Overall, the procedure described guarantees the generation of a large enough dataset including a multitude of loading scenarios. A key feature is the inclusion of step 2, which contributes to extra variability. Adding errors to model actual measurements provides the ML algorithms with training data that capture real-world operating conditions more reliably.

4.3.2.2 Feature selection

The feature selection procedure is based on two separate, iterative ML algorithms for the TI and SE tasks, respectively. The block diagrams of the algorithms are illustrated in Figure 10 and Figure 11. The ML algorithms can be based on either PCA or RF. As showed, their implementation relies on training and evaluation of multiple DNNs. The procedure is initiated by considering all available PMU measurements from the grid, as candidate features. A single feature, i.e., PMU measurement, is chosen from the candidate set per iteration; the selected feature is the maximiser of the computed score depending on the utilized ML technique (PCA or RF). The algorithm is terminated when the already selected PMU measurements ensure that the desired level of DNN accuracy (for TI or SE) is fulfilled. The performance indicator TI_{acc} used to assess the accuracy of the associated DNN, is computed below:

$$TI_{acc} = \frac{\textbf{number of} \text{ correctly estimated topologies}}{\textbf{total number of} \text{ topologies}} 100\% \quad (2)$$

Furthermore, the performance indicators considered to evaluate the accuracy of the DNN used for SE, are the mean absolute percentage error ($MAPE_V$) for the nodal voltage magnitudes:

$$MAPE_v = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{V_{m,i}-\hat{V}_{m,i}}{V_{m,i}}\right| 100\% \quad (3)$$

and the mean absolute error (MAE_a) for the corresponding phase angles:

$$MAE_a = \frac{1}{n}\sum_{i=1}^{n}\left|V_{a,i} - \hat{V}_{a,i}\right| \quad (4)$$

where $V_{m,i}$ ($V_{a,i}$) and $\hat{V}_{m,i}$ ($\hat{V}_{a,i}$) are the actual and estimated values of the estimated voltage magnitudes (phase angles), and n is the number of states variables for the whole dataset.

It is noted that no budget limit is considered for the PMU selection. Additionally, in case a grid node cannot host a PMU, e.g. due to technical difficulties or high associated cost, the corresponding measurements can be excluded from the candidate set of features.

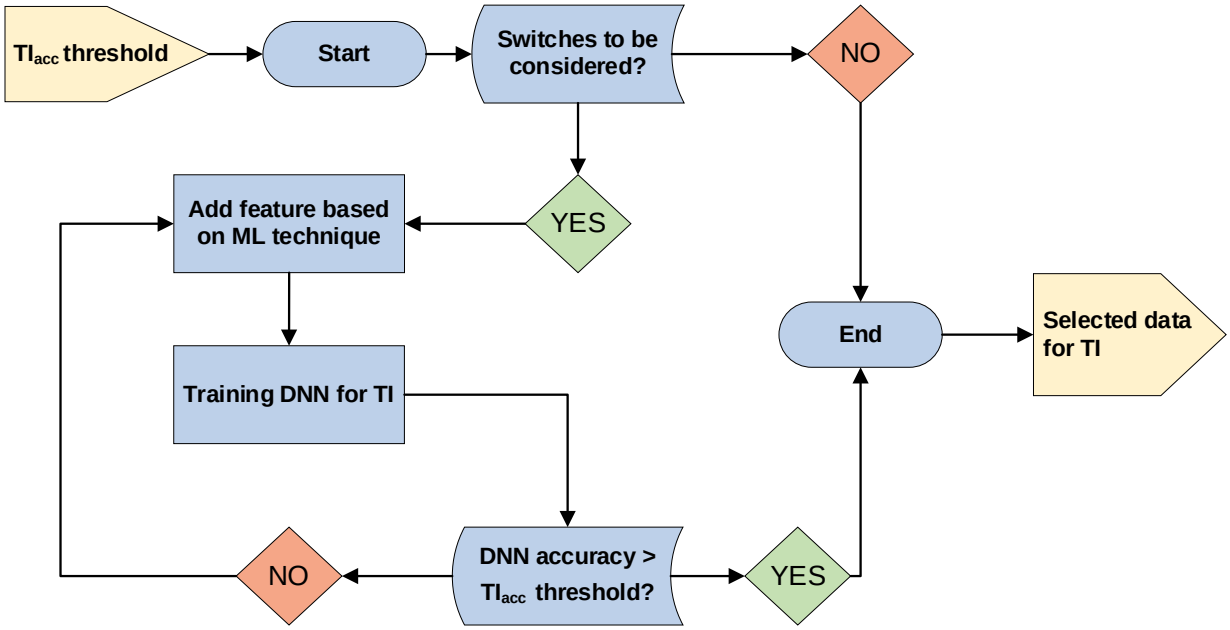


Figure 10: Algorithm of feature selection for the TI task.

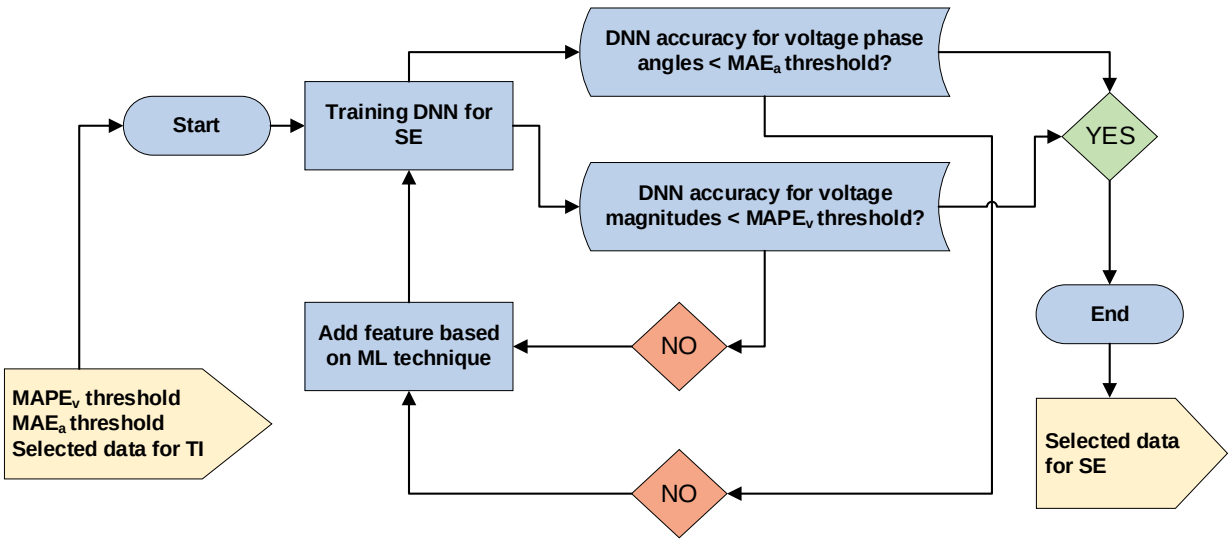


Figure 11: Algorithm of feature selection for the SE task.

4.3.2.3 Training procedure

In order to train the DNNs for the TI and SE tasks, the development of multiple ML models is required. Besides, hyperparameter optimization is performed for both tasks. For all training cases, the train-validation-test ratio is 80-10-10.

To solve the TI task, a classification DNN is built, trained and evaluated. The measurements from the optimally allocated PMUs based on the feature selection process (Figure 10), are utilized as input to track the switch statuses of the grid. As far as its architecture is concerned, the number of neurons in the output layer is equal to the number of feasible topologies of the ADN. The number of hidden layers and neurons per layer are hyperparameters that require tuning. The ReLU activation function is used in the input and hidden layers, and the softmax activation function is employed for the output

layer. While training the model, the categorical crossentropy is set as the loss function and the optimizer utilized is adam. Its learning rate is experimented upon. Lastly, dropout layers are also introduced between the hidden layers to avoid overfitting. The DNN is evaluated based on its estimations on the test set, using the indicator TI_{acc}.

For hyperparameter optimization, the following parameters are investigated: the number of hidden layers, the number of neurons per hidden layer, the number of epochs, batch size, and learning rate. Additionally, the dropout rates in dropout layers are experimented with, and adjustments to the number of neurons per hidden layer are also examined.

With a view to tuning the hyperparameters, a combination of random and grid searches is implemented. First random combinations are examined and, subsequently, a number of grid searches are executed. The latter guarantees optimal results within the specified search grid, since it systematically explores all possible combinations.

For the purpose of SE, the developed DNN is trained using the selected PMU measurements for the TI task along with the ones obtained from the feature selection for the SE task (Figure 11). The above regression model could be swapped with 2 separate DNNs, with the former one estimating the nodal voltage phase angles and the latter one the voltage magnitudes. This approach, albeit more effective in terms of accuracy, causes more computational burden, as it requires 2 DNNs to be implemented and then, appropriately tuned. Hence, a compromise solution is to develop a model which splits into 2 symmetrical branches. In this framework, the number of neurons in the output layer of each branch is equal to the number of the grid nodes. The linear function is chosen as the activation function per output branch, while the activation functions for the hidden layers are the same as the ones used in the case of TI. The mean squared error, given in (1), is set as the loss function and adam remains the optimizer employed. Finally, the indicators $MAPE_v$ and MAE_a are leveraged to evaluate the performance of the trained DNN for SE.

Concerning hyperparameter optimization, the settings of the following hyperparameters are examined: the number of neurons of each hidden layer, the number of the hidden shared layers and the number of the branch layers. The number of epochs, the batch size for the training and the learning rate of the adam optimizer are also investigated. Lastly, similar to the DNN for TI, properly setting the number of neurons is successively considered.

4.3.2.4 Software and coding languages employed

The proposed DSSE scheme was implemented using:

- the MATLAB software (release version 2024a) along with the open-source toolbox for electric power system simulation and optimization MATPOWER (release version 7.1) [31], to conduct power flow studies and generate the dataset
- the Python programming language (release version 3.10) along with the libraries Keras, TensorFlow and scikit-learn, to develop, train and evaluate the ML models.

4.3.3 Mock-Ups

Overall, a UI will be utilized for two purposes. The first purpose is for system operator to upload the system topology and historical measurements. This, will be similar to the interface presented in section 4.8.3 where an appropriate format of input file will be uploaded, giving the feeder topology

and the location and type of available measurements. In the UI the results of PMU placement algorithm will be also presented, similar to the results presented in the following subsection. The other functionality of the UI is to define the communication protocol settings (MQTT) with the existing measurements and the communication settings with the PMUs, as well as the communication interface for sending the results, via MQTT back to the system operator. This will be detailed described in v2 of the deliverable.

In the remaining of this section, some numerical results on an IEEE benchmark network will be presented, which will also serve as mock-ups on results that will be presented in the UI to the user for the optimal location of PMUs and the performance of the enhance state estimation algorithm.

4.3.3.1Numerical studies

To demonstrate the use and investigate the effectiveness of the proposed methodology, preliminary numerical studies are performed using the IEEE 33-bus benchmark test system [32], which is depicted in next Figure. As observed, the test system can be reconfigured using the tie-line branches that are marked with blue, dashed lines. A set of 15 radial and meshed topologies are taken into account for the simulations. The radial topologies result from a combined operation of tie-line (switched in) and fixed (switched out) branches as follows: 25–29 with 28–29, 21–8 with 7–8, and 12–22 with 11–12.

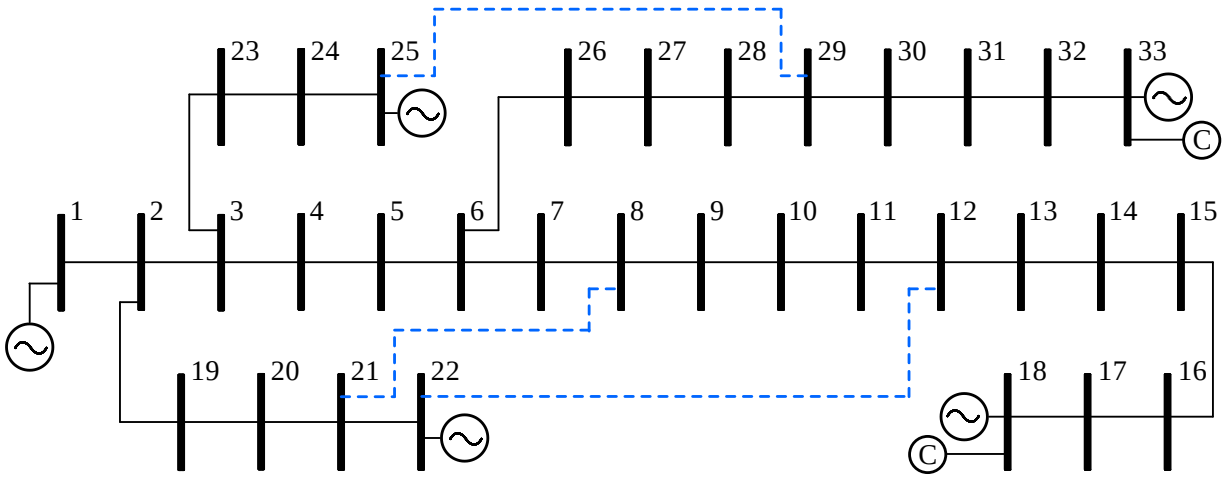


Figure 12: The IEEE 33-bus benchmark test system.

All nodes are considered as candidate locations for PMU installation. Each unit is assumed to measure one nodal voltage and one branch current phasor. The list of assumed measurement data per PMU is provided in Table 1. As seen, the PMUs 33, 34 and 35 pertain to the tie-line branches.

As far as the training dataset is concerned, the MATPOWER toolbox is utilized to apply the Monte Carlo trials by conducting the necessary power flow calculations [33]. To account for the measurement errors, a total vector error (TVE) of PMU data below 1% is taken into consideration, as reported in the IEEE standard [34]. As commonly postulated, the synchrophasor data errors follow a Gaussian distribution. A maximum error of ±0.1% is assumed for voltage and current magnitude measurements, while a maximum error of 0.018° is considered for the voltage and current phase angles. Finally, the threshold values for the accuracy indices used for the feature selection process, are TI_{acc}=95%, $MAPE_v$=0.35% and MAE_a=0.05°.

Table 1 Candidate PMU measurements

PMU #	Nodal voltage, branch current	PMU #	Nodal voltage, branch current	PMU #	Nodal voltage, branch current	PMU #	Nodal voltage, branch current
1	V_1, I_{1-2}	**10**	V_{10}, I_{10-11}	**19**	V_{19}, I_{19-20}	**28**	V_{28}, I_{28-29}
2	V_2, I_{2-3}	**11**	V_{11}, I_{11-12}	**20**	V_{20}, I_{20-21}	**29**	V_{29}, I_{29-30}
3	V_3, I_{3-4}	**12**	V_{12}, I_{12-13}	**21**	V_{21}, I_{21-22}	**30**	V_{30}, I_{30-31}
4	V_4, I_{4-5}	**13**	V_{13}, I_{13-14}	**22**	V_3, I_{3-23}	**31**	V_{31}, I_{31-32}
5	V_5, I_{5-6}	**14**	V_{14}, I_{14-15}	**23**	V_{23}, I_{23-24}	**32**	V_{32}, I_{32-33}
6	V_6, I_{6-7}	**15**	V_{15}, I_{15-16}	**24**	V_{24}, I_{24-25}	**33**	V_{21}, I_{21-8}
7	V_7, I_{7-8}	**16**	V_{16}, I_{16-17}	**25**	V_6, I_{6-26}	**34**	V_{12}, I_{12-22}
8	V_8, I_{8-9}	**17**	V_{17}, I_{17-18}	**26**	V_{26}, I_{26-27}	**35**	V_{25}, I_{25-29}
9	V_9, I_{9-10}	**18**	V_2, I_{2-19}	**27**	V_{27}, I_{27-28}		

The optimal PMU locations obtained from the proposed RF and PCA for both the TI and SE tasks are illustrated in the next Figures. For reasons of convenience, the arrows indicate the branch current flows measured by the selected units.

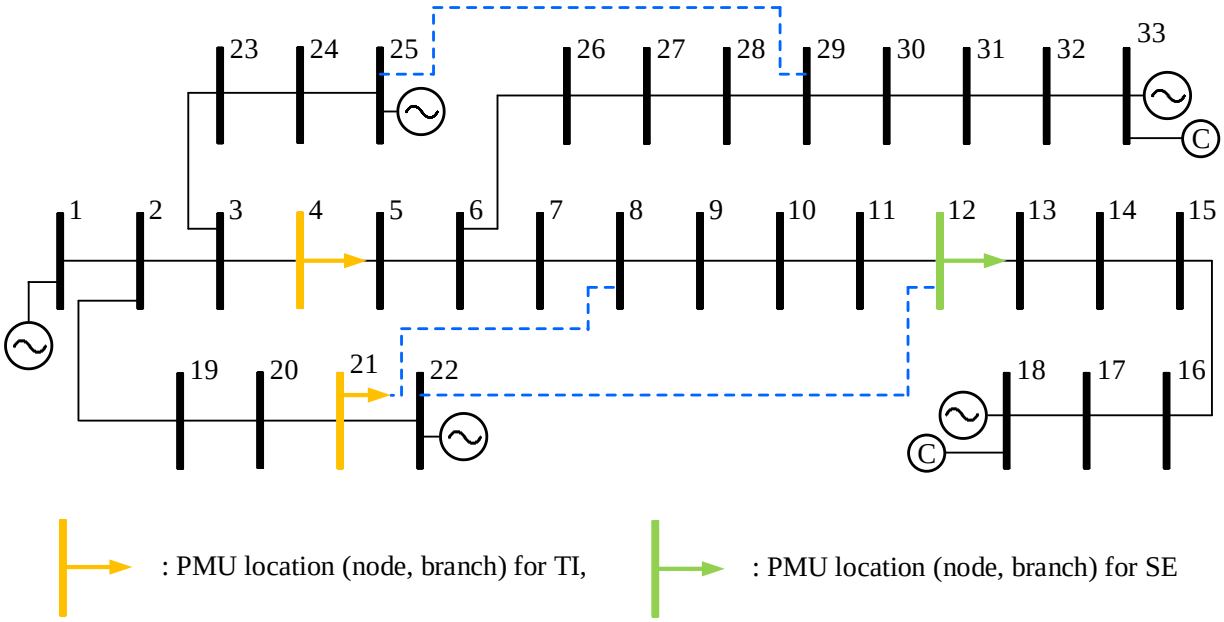


Figure 13: Selected PMU locations using the RF technique.

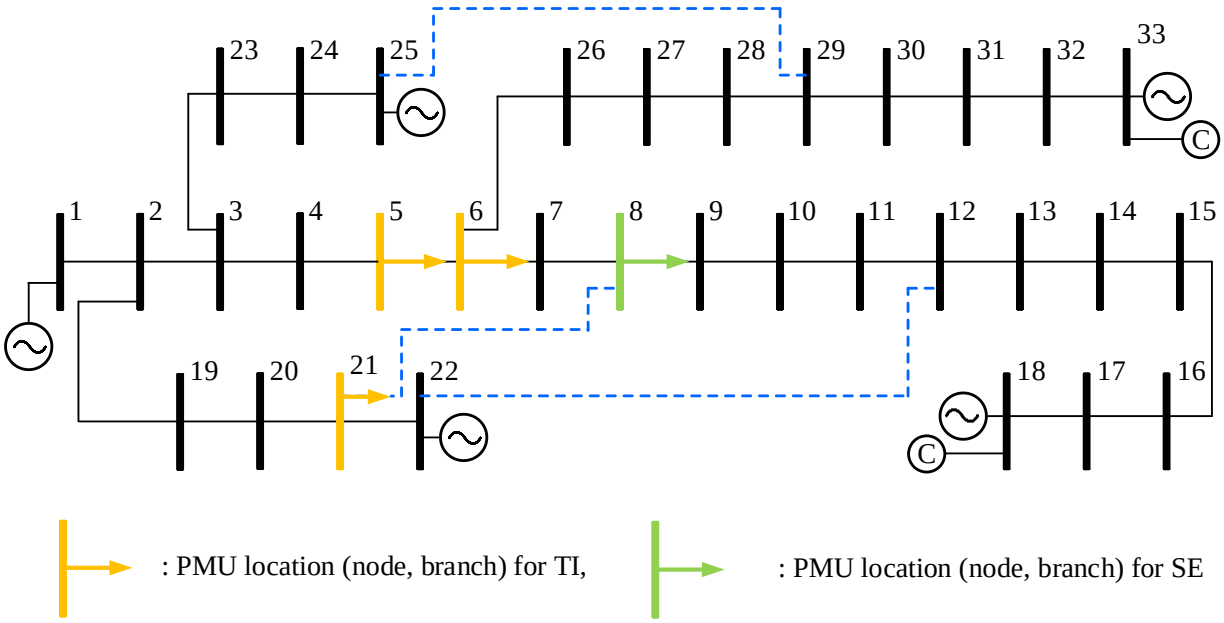


Figure 14: Selected PMU locations using the PCA technique.

As displayed in Figure 13, using the RF technique ensures that the desired TI and SE accuracy standards are met by means of 3 PMUs allocated at nodes 4, 12 and 21. On the other hand, as showed in Figure 14 to fulfill the same accuracy standards using the PCA technique, the installation of 4 PMUs at nodes 5, 6, 8 and 21 is required. The selection of node 21 for placing a PMU by both ML techniques highlights its strategic position for quality DSSE. Additionally, both ML techniques select features, i.e., PMU measurements, from the segment between nodes 2 and 18, a finding which also stresses the importance of these locations for DSSE performance.

The computation time of the ML techniques is also a parameter of increased interest. The corresponding results are provided in Table 2. All computation times have low and comparable values. Therefore, the proposed feature selection techniques operate seamlessly on the IEEE 33-bus test system and can be extended to larger ADNs for testing.

Table 2 Computation time per PMU placement technique for the TI and SE tasks

ML technique	Computation time (s)	
	TI task	SE task
RF	23.5	88.2
PCA	**23.9**	**53.1**

Following their training using the placed PMUs, the performance of the DNNs for the TI and SE tasks is evaluated on the test set (10% of the dataset). As regards the TI task, the accuracy of the DNN using the PMU selections by the RF, is 98.6%, while the corresponding value in case the PMUs selected by the PCA are used, is 99.8%. Hence, the performance of the DNN in both cases, is excellent.

Concerning the SE task, the bar diagrams of the attained $MAPE_v$ and MAE_a values per node, are illustrated in Figure 15 and Figure 16.

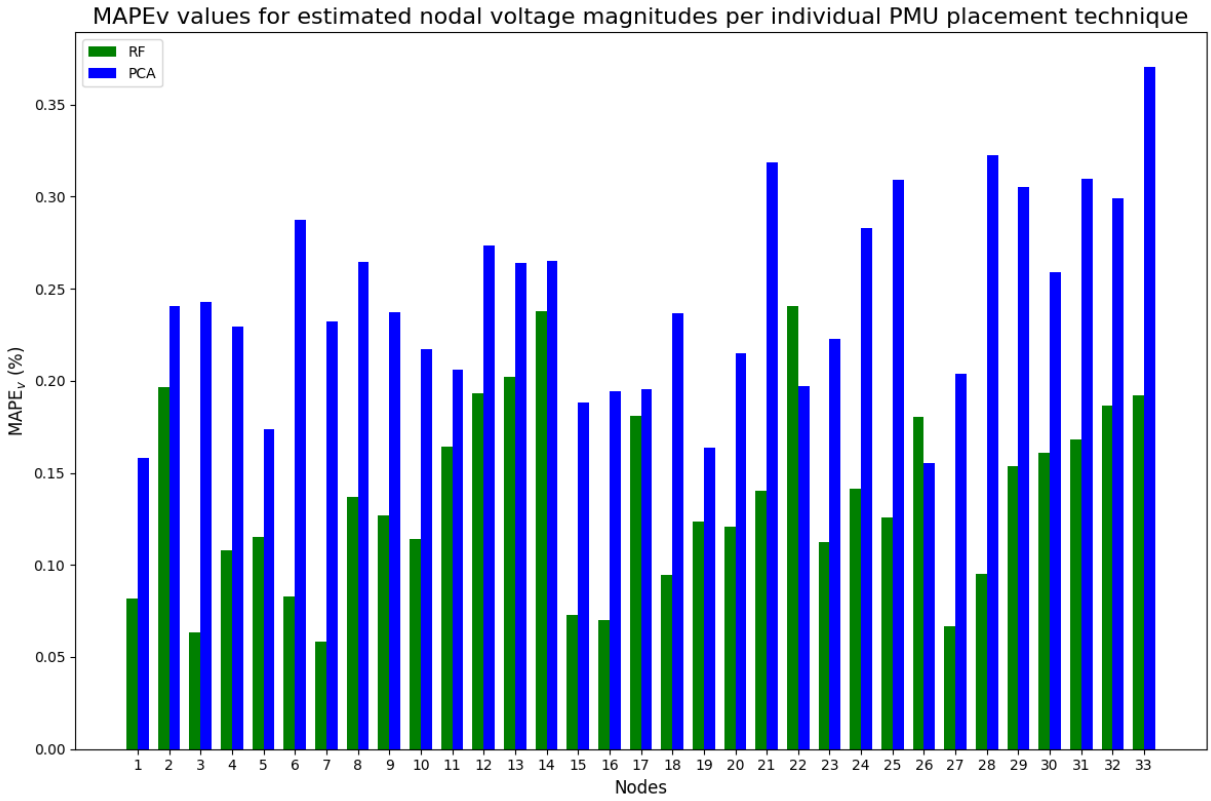


Figure 15: Attained SE accuracy in terms of nodal voltage magnitudes per PMU placement technique.

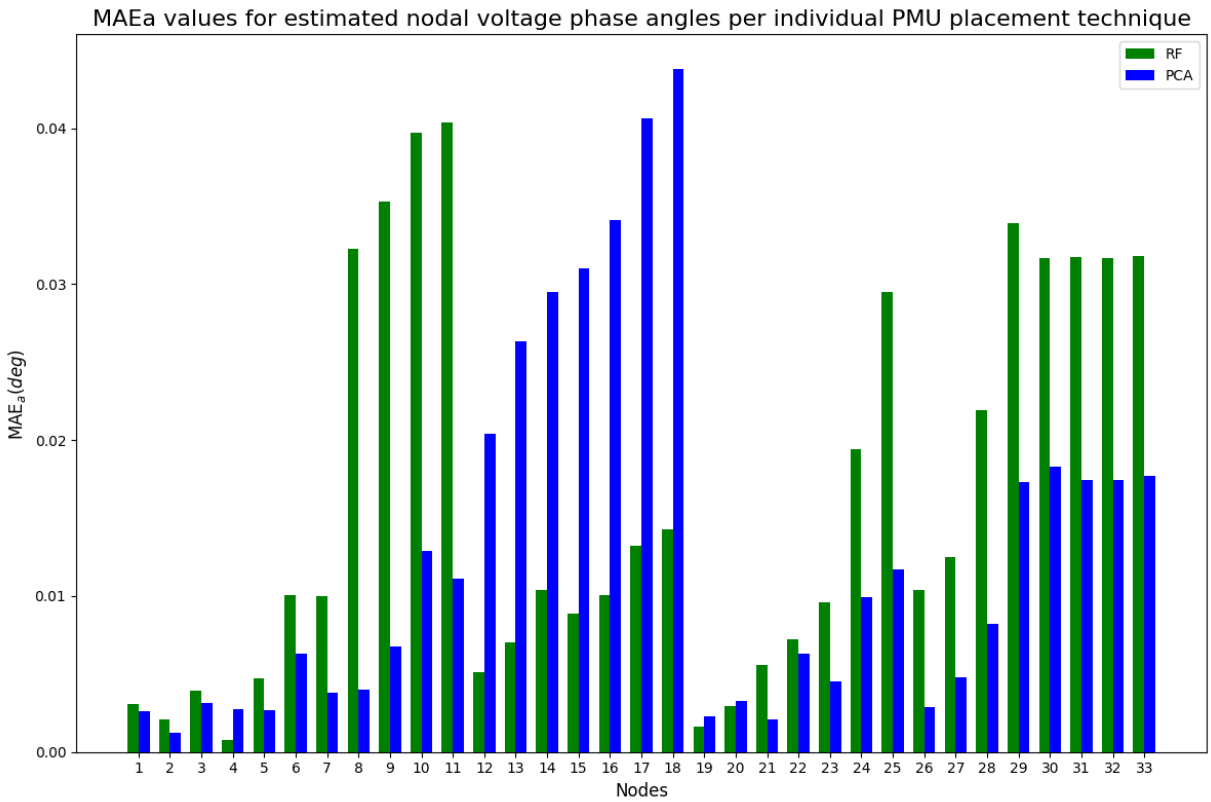


Figure 16: Attained SE accuracy in terms of nodal voltage phase angles per PMU placement technique.

As perceived, the DNN for the SE task delivers more precise estimates of voltage magnitudes using the PMUs selected by the RF, since the corresponding $MAPE_v$ values are lower compared to the ones obtained based on the PCA. Conversely, the results for phase angles are less straightforward, as there is no clear trend in the MAE_a values that consistently favors either RF or PCA.

Overall, the developed DNN models provide quality DSSE for the reconfigurable IEEE 33 test system only utilizing a limited number of PMUs. This is an advantageous outcome for the applicability of the proposed ML based framework to the enhanced SE tool. Moreover, performing feature selection using the RF technique is preferable since high TI and SE accuracy is attained with fewer PMUs compared to the PCA.

4.4 Topology detection tool

Distribution system operators (DSO) operate and control the medium and low voltage networks. For them, understanding the topological structure of a power grid and laws of changes within it in a timely manner based on measurements is of paramount importance: it is required to have reliable information about the network topology and up to date measurements of electrical parameters. This situational awareness is normally more developed in the medium voltage (MV) side than in the low voltage (LV) side, but in both cases, it is far from being perfect, due to the limited placement of real-time metering devices. This is in stark contrast to high voltage (HV) transmission systems that usually enjoy full observability and breaker/switcher statuses on lines are reported in real-time or identified jointly while estimating system states via the generalized power system state estimators.

The biggest problem are the LV lines (also known as feeders). Here the lack of measurements, due to the lack of sensors or an appropriate communication channel makes almost impossible to have a reliable network monitoring system: There may be thousands of buses in a distribution system where only a hundred of them are monitored with real-time measurements. This scarcity of sensors is linked to the fact that LV networks are huge and contain several connections, that are (economically) impossible to monitor in real time. This lack of observability could lead to inaccuracies in the topological information for different reasons:

- The LV network may be affected by public works and incidents that might end up with not logged-in changes in the topology, made on a rush to restore service for customers.
- LV network topology change frequently due to normal power engineering activities aiming at reducing line losses, handling outages or accepting more intermittent distribution generators, but these operations might not be properly logged-in the topology database.
- LV Breakers and protections do not normally report status in real time, so the topology might have automatically changed as a result of a critical event without the DSO noticing.

In OPENTUNITY, there will be tree different tools oriented to solve or alleviate for such problems and improve the situational awareness:

- The **enhanced state estimation tool** presented in the previous sections aims to retrieve the unknown system state, that is, the complex voltages at all buses and connection points. Two different approaches were presented for enhancing the results of state-of-the art, one developed for the Slovenian and Spanish pilot, and other developed for the Greek pilot based on micro-PMUs.

- The **topology detection tool** will try to identify the inaccuracies on the topological models of the utility by analysing the data received from the available network sensors and checking its compatibility with the stored topology.
- The **fuse burn detection tool for early outage and islanding recovery tool** will focus on identifying burn fuses on specific phases that might impact the quality of supply and even end up in islanding problems in specific areas.

4.4.1 Design

The connectivity of nodes in the topology is usually represented by an **adjacency matrix**. This is a square matrix used to represent a graph, where the elements of the matrix indicate whether pairs of vertices (nodes) are adjacent or not in the graph. In the context of power systems, the nodes represent buses (or substations), and the edges represent transmission lines or circuit breakers connecting these buses. This is an example of these:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

This is an example of a 4 buses system with and the following connections:
- Bus 1 is connected to Bus 2.
- Bus 2 is connected to Bus 3 and Bus 4.
- Bus 3 is connected to Bus 4.

This matrix represents the physical connections, but there are switching and breaking elements that may change the connectivity of the buses. There has to be noted that, compared to the topology identification problem, where the full adjacency matrix must be discovered, here the connections are known, and the challenge is to determine which of these connections are not actually energized (probably due to the actuation of a switch or protection), so the number of actual connections in the adjacency matrix could be reduced, but not increased.

There has to be noted that the concept of bus in LV is different from the one in MV. LV topology is defined so that every cable section connects two different buses. In many circumstances these are not Buses in the physical sense of pieces of equipment for connecting cables and transformers used in substations, but can be fictitious buses defined for modelling different LV network situations, such as:

- Junctions of different cable types
- Conversion from aerial to subterranean
- Connection points for more than two cable, like branching
- Connection to loads.

4.4.2 Implementation

The topology detection tool is tightly integrated with the state estimation tool. The state estimation will produce a set of measurements (v, p, q, I, etc.) in all network elements that are coherent with the known measurements. Anomalous measurements, non-convergent power flows, huge power losses or unrealistic voltages are examples of situations identified by the state estimators that might be indicative of topological inaccuracies. There will be some thresholds linked to these DSSE results that will trigger the topology detection functionality when something strange is detected.

Upon the detection of potential problems, the DSSE results gathered, and n-1 power flow analysis is performed on them. This analysis simulates the effects of changing the topology by removing each

and every connection in the adjacency matrix. In case any of these changes produce a result better than the original DSSE result, the corresponding line or breaker status is marked as incorrect.

4.4.3 Mock-Ups

The topology detection is a tool that does not feature a graphical user interface. Nevertheless, the plan is to present the detected problems in the DSO EMS software ETER, for the DSO operator to notice and eventually accept or not the proposed topological change.

This is an example of how this could be presented in the ETER tool:

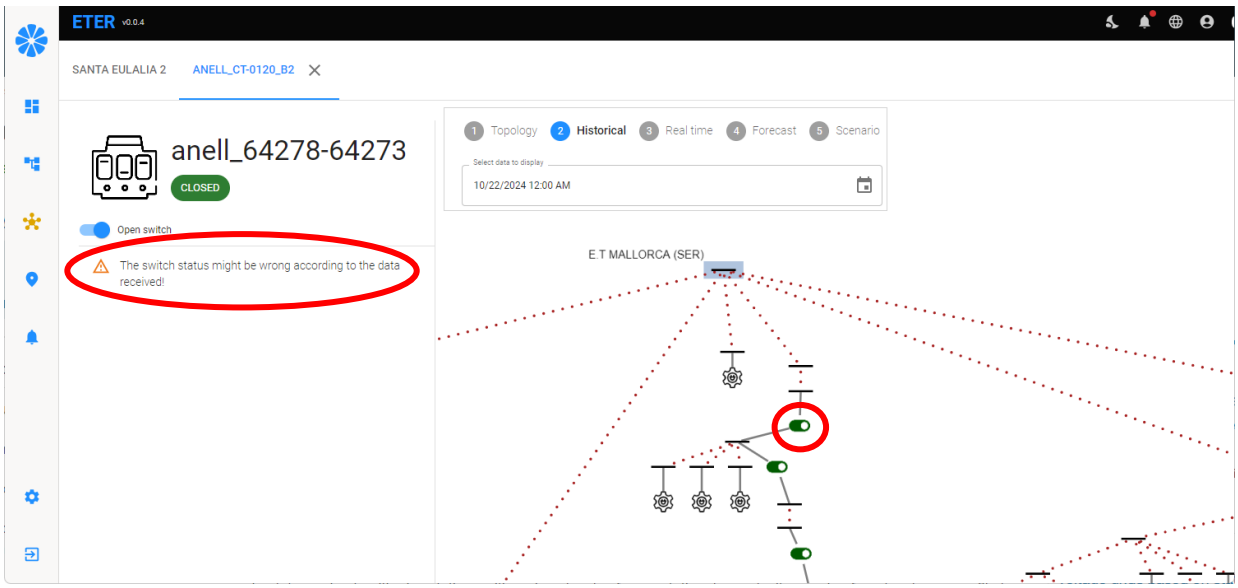


Figure 17 Topological error detected user interface.

Upon the selection of the switch or line, the tool could display a message with the possible incorrect status. If necessary, the switch or line status could be changed to reflect the real topology.

4.5 Fuse burn detection tool for early outage and islanding recovery.

A fuse in low-voltage (LV) networks is a protective device designed to safeguard electrical circuits from overcurrent conditions, such as short circuits or overloads. It consists of a metal wire or strip that melts when the current flowing through it exceeds a predetermined level, thereby interrupting the circuit and preventing damage to the electrical system or connected equipment. Fuses are commonly used in residential, commercial, and industrial settings to protect wiring, appliances, and other electrical components from potential hazards caused by excessive current.

Detecting a blown fuse in a low-voltage (LV) three phase power grid, particularly in a complex environment like a multi-apartment building, presents a multifaceted challenge. When a fuse blows, it disrupts the electrical continuity in one of the phases, potentially causing an imbalance in the system. This can lead to partial power outages, malfunctioning equipment, and even safety hazards.

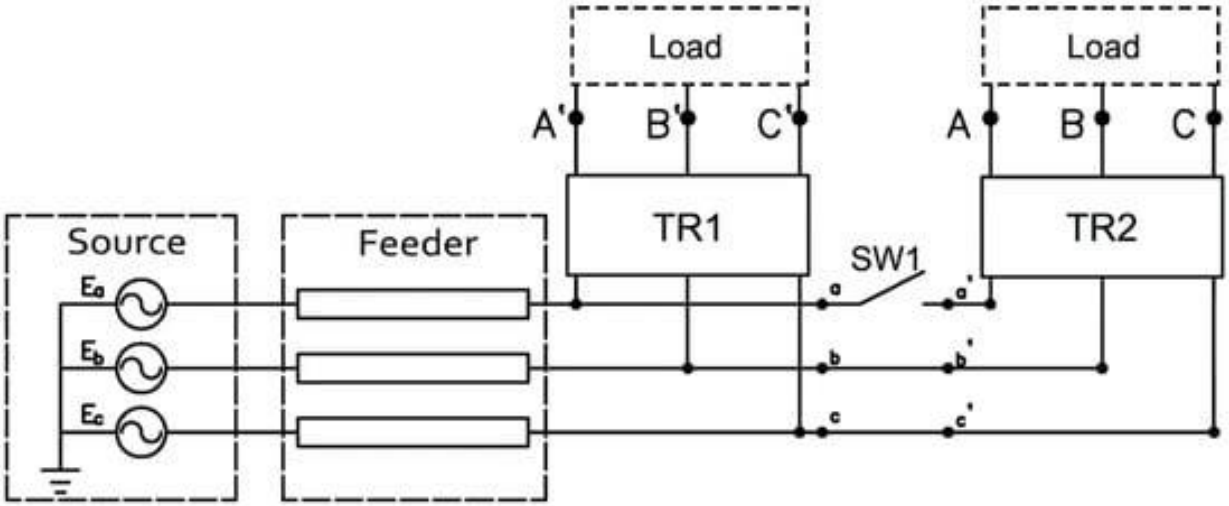


Figure 18 Connection of loads to individual phases in LV

Islanding occurs when a portion of the electrical grid, typically containing distributed energy resources (DER) like solar panels, continues to be energized and operate independently despite being disconnected from the main utility grid. This situation can pose safety hazards to utility workers, cause damage to equipment, and disrupt the proper functioning of the grid. Anti-islanding protection mechanisms are essential to detect and prevent islanding, ensuring that DERs shut down or disconnect when the main grid power is lost. This is the schema of the connection of a simple domestic PV system:

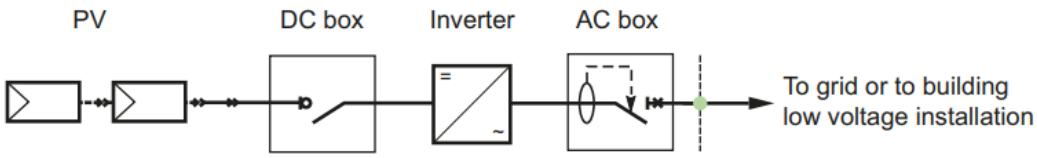


Figure 19 PV connection scheme in domestic environments

PV are arranged in arrays and producing DC current. This goes through an inverter that transforms the electricity to AC and connects to the LV grid. The PV systems always feature protection mechanisms that prevents islanding; however, these mechanisms are primarily designed to detect a complete loss of grid power. When only one phase is curtailed, detection can be more complex, and it could happen that the PV energy dangerously feeds a client connected to the curtailed phase:

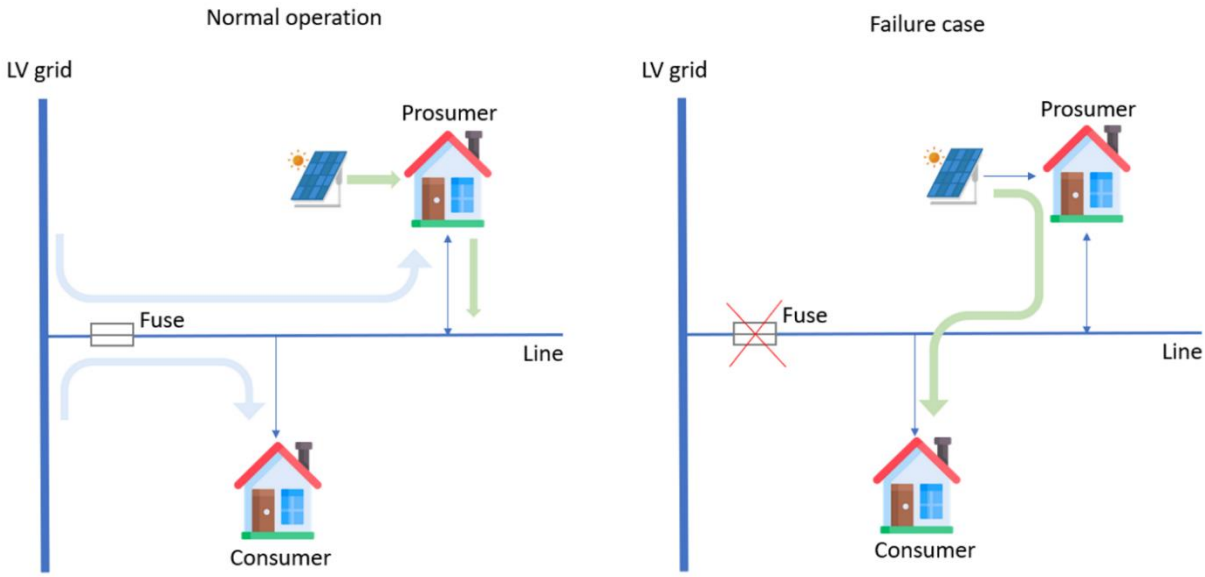


Figure 20 Example of islanding caused by blown fuse

The aim of the tool described here is the **detection of blown fuses and pinpoint its exact location and phase through the monitoring of the voltage at end user level**. This especially important when it occurs in remote or less accessible parts of the LV network. Traditional methods of detection, such as manual inspection, are time-consuming and impractical for large-scale or critical applications.

4.5.1 Design

The blown of a fuse because of a network error can happen io any part of the LV network.

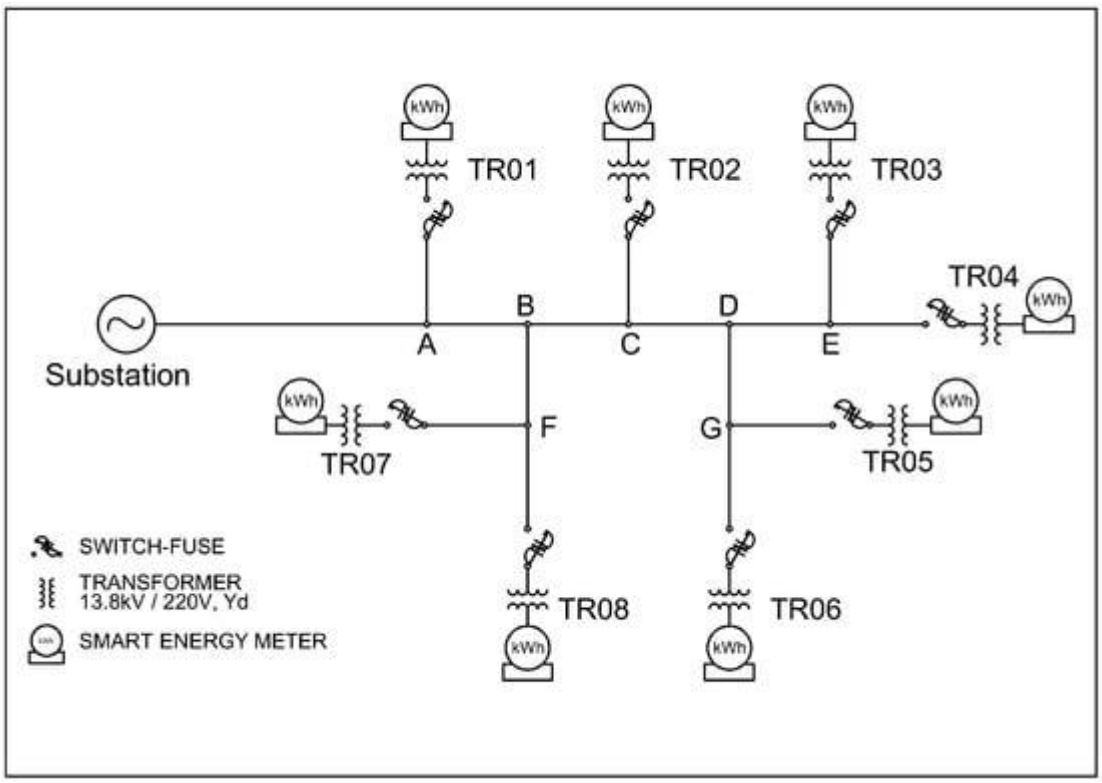


Figure 21 LV radial network with fuse location

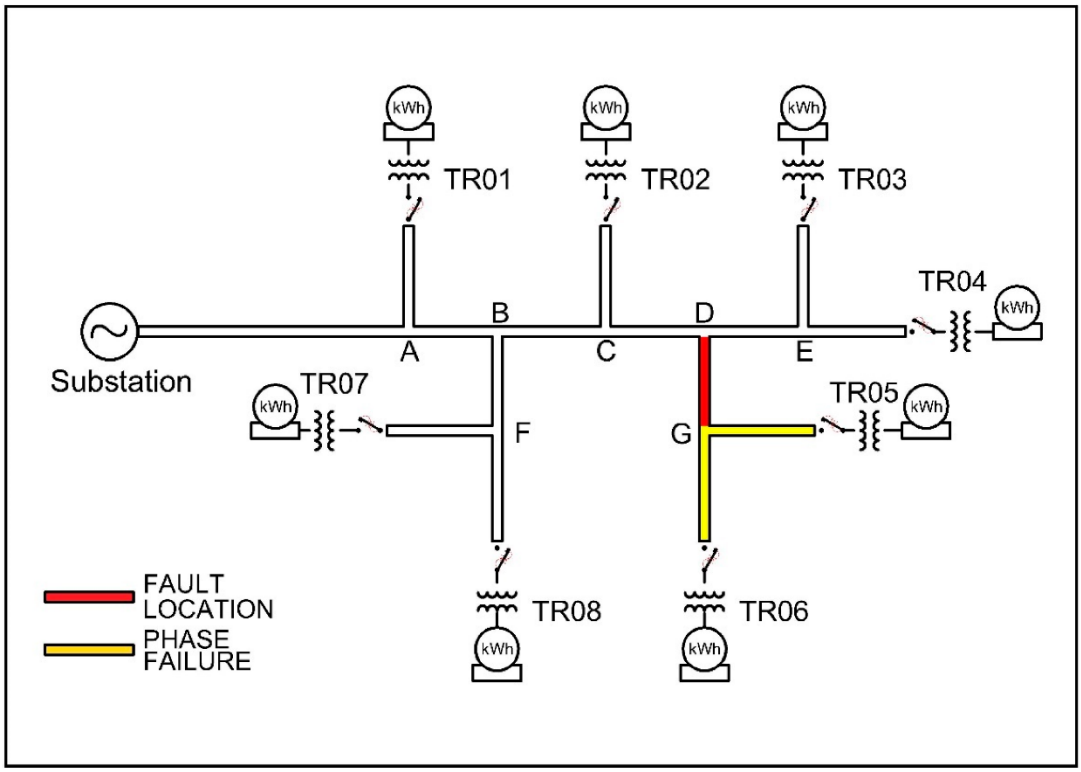


Figure 22 Effect of fuse closing in LV network

The location of blown fuses will be done by interrogating the smart metering devices across the LV feeder. The analysis of the data for such meter will be used to identify for the faulty phases. The two main mechanisms for this are:

- **Voltage Imbalance Detection**: Look for significant voltage drops or imbalances between phases. A blown fuse in one phase will result in a lower or zero voltage reading for that phase at the affected smart meters.
- **Current Discrepancies**: Detect sudden drops to zero current in one phase while other phases continue to carry current, indicating a possible fuse blow.

The interrogation of the smart meters using PLC channel is a process that takes quite a lot of time. OPENTUNITY will investigate the most optimal PLC interrogation strategy to locate the failure.

4.5.2 Implementation.

To detect blown fuses, the LV network's PLC concentrators will be configured to interrogate a set of smart meters for current electrical measurements, storing this data in a database. The detection process will be done by a microservice written in Python. This microservice will activate every few seconds to analyse the stored measurements.

If it identifies indicators of a blown fuse, such as voltage imbalance or current discrepancy in any of the smart meters, two actions can occur:

1. **Sufficient Data for Fault Location**: If enough measurements are available to pinpoint the fault, the potential location of the blown fuse is stored in the database, and the DSO operator is notified through the ETER tool.

2. **Insufficient Data**: If more data is needed, the microservice will request additional smart meters to be included in the interrogation cycle. This process will repeat iteratively until the fault location can be unambiguously detected.

By leveraging this automated and iterative approach, the system ensures efficient and accurate detection of blown fuses, enhancing network reliability and operational efficiency.

4.5.3 Mock-Ups

The result of the blown fuse will be presented in the ETER tool.

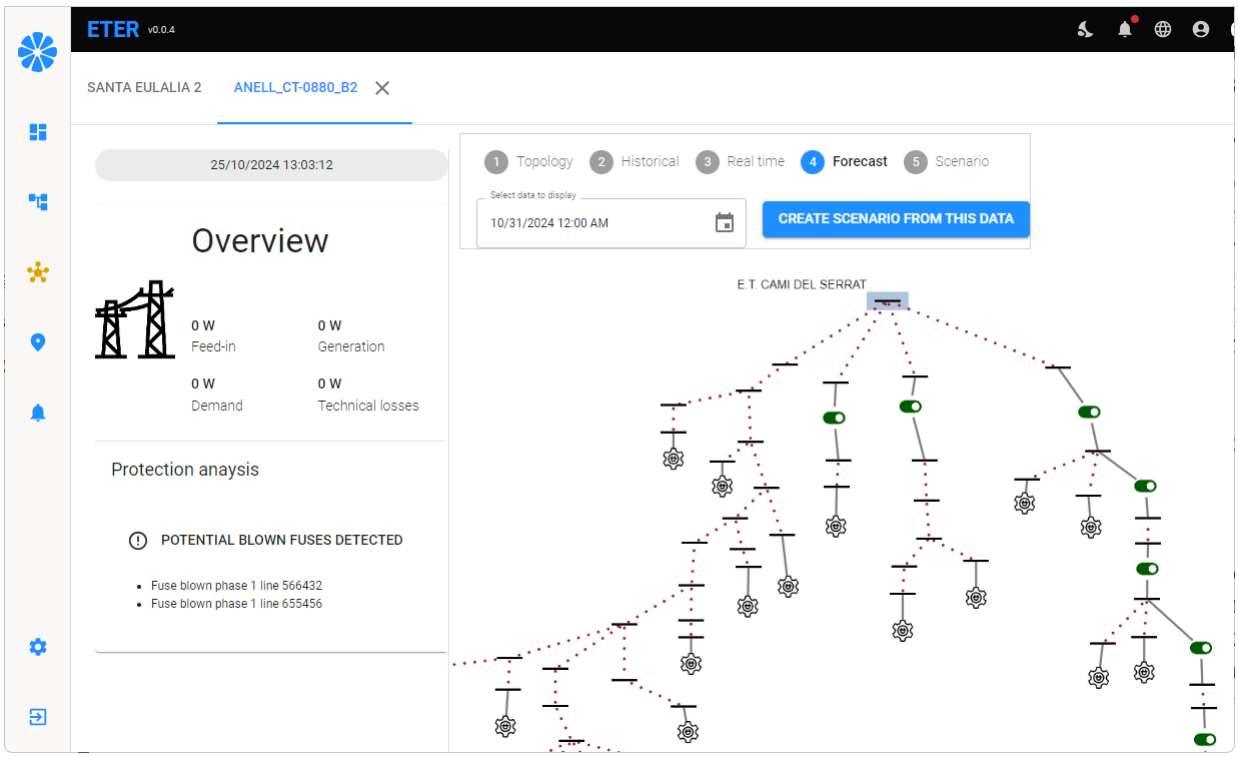


Figure 23 result of blown fuse detection in ETER

By clicking on the overall network view, the result of the analysis of protections will be presented. In the example, the analysis results in the identification of two potential situations of blown fuses in two different lines. The final version of the GUI will also highlight the affected lines in the SLD.

4.6 Critical point detection tool

DSO's LV networks have usually a radial structure from the secondary bus of the electrical transformer. The infrastructure used for supplying electricity to all clients is composed by several small cable segments that are deployed following the structures of cities and towns towards the different connection points. This is an example of how the feeder span from the MV/LV substation to reach all clients in dense populations.

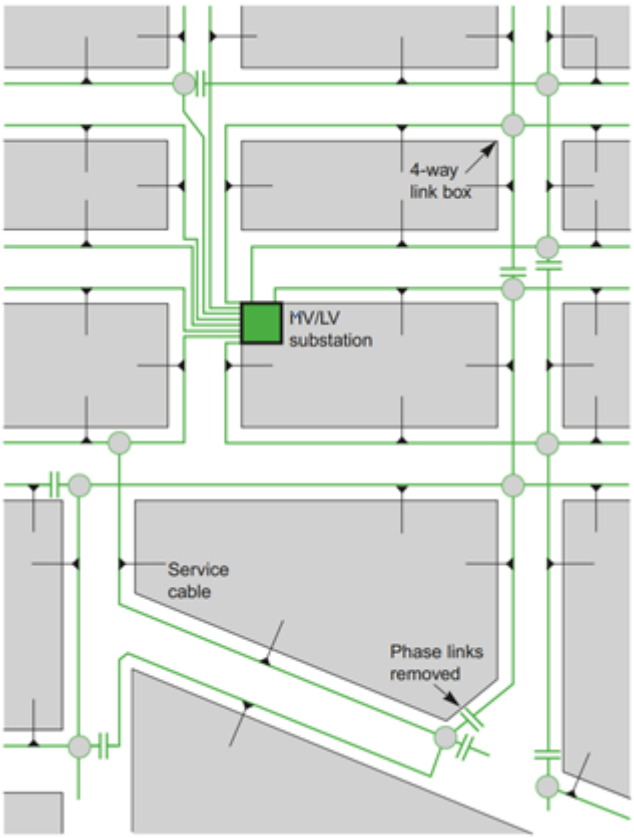


Figure 24 one of several ways in which a LV distribution network may be arranged

These cables could have different characteristics: section size, composition, insulation, aerial or subterranean, etc. These characteristics determine the maximum capacity of the network and must be carefully considered to allow for new electrical connections.

One problem linked to this is the fact that since the cable section gets smaller as it goes further away from the transformer and the penetration of PV and EV is normally higher, there's a potential risk of **congestion** in parts of the line with lower sections of cable. Also, the **voltage drop** could be a problem in the sections of the network far away from the primary bus.

The LV cable infrastructure also determines the protection devices required. These devices are essential for ensuring the safety, reliability, and efficiency of electrical systems. These devices, including circuit breakers, fuses, Residual Current Devices (RCDs), and Surge Protection Devices (SPDs), are strategically installed to safeguard against various electrical faults such as overcurrent, short circuits, earth faults, and voltage surges. Circuit breakers and fuses protect wiring and equipment from damage caused by excessive current, preventing potential fire hazards and equipment failure.

Protection devices in Low Voltage (LV) networks are selected and configured using **short circuit current calculations**. These calculations determine the magnitude of fault currents that can occur during a short circuit, which is crucial for selecting and setting protection devices. The calculation results are used for:

1. **Protecting device selection:** Short circuit current calculations help in choosing appropriate protection devices (circuit breakers, fuses, RCDs) that can manage and interrupt the maximum fault current without damage.

2. **Setting trip parameters:** The calculated short circuit currents inform the settings for trip thresholds and response times of protection devices, ensuring they operate correctly during faults.

Distributed Energy Resources (DERs) connected to distribution systems affects the fault current and power flow direction. The most significant impact of DERs on distribution systems relates to increasing the short-circuit current and contributing to the fault current for downstream faults. As a result of the fault current increase in distribution networks, DERs units might reduce the contribution of the system's fault current which in turn causes the protection system to be blinded and cause malfunctioning of protection systems during faults. For instance, if a fault occurs at one of the feeders adjacent to DG units, an undesired tripping command by protection relays may be triggered. As such, it should be noted that the location of DERs in distribution systems (as well as their number and penetration level), highly impacts protection systems.

The aim of this tool is to evaluate the sections of the LV network that are prone to errors (or in other words more fragile or critical) given the current network topology and considering different power flow scenarios, like peak load, PV generation surplus, line tripping, fuse burn, etc. The tool will allow to define for such scenarios and will make the required calculations to identify for the following problems:

- **Congestion in cable sections**. Each cable has an inherent ampacity and over currents (congestions) could happen if the network is not properly structured and dimensioned.
- **Voltage problems**. The voltage profile at the LV network buses could be affected by the location of DERs.
- **Incorrect protection settings**. The short circuit current could be affected by the presence of DERs in the LV. The short circuit current will be calculated for all LV buses in the selected scenario and compared to the nominal characteristics of the protection systems. The simulation could help identify situations where the protection scheme is not appropriate.

4.6.1 Design

The tool will be composed of two main parts:

The first part is the ETER tool section that allows for defining LV network situations or scenarios. This will have the form of a graphical user interface that will allow to, starting from a given network topology, define the value of the power consumptions and injections. There will be a mechanism to let user manage these scenarios. The actions available for the scenario definition will be:

- Creating empty scenario for a given LV network.
- Creating scenario for a given LV network based on the measurements for a given date,
- In a scenario, changing instantaneous active and reactive power consumption of each load.
- In a scenario, changing instantaneous active and reactive power generation of each DER.
- Opening or closing the switches
- Tripping some of the lines

When the scenario is defined, the DSO operator could make use of the second part of the tool: a piece of software will receive the scenario details and make the required calculations to identify for the criticalities mentioned before: Congestions in cable sections, Voltage problems and Incorrect protection settings.

The results of the calculations will be presented in the ETER tool for the operator to analyse and take informed decisions. This is the design of the whole solution:

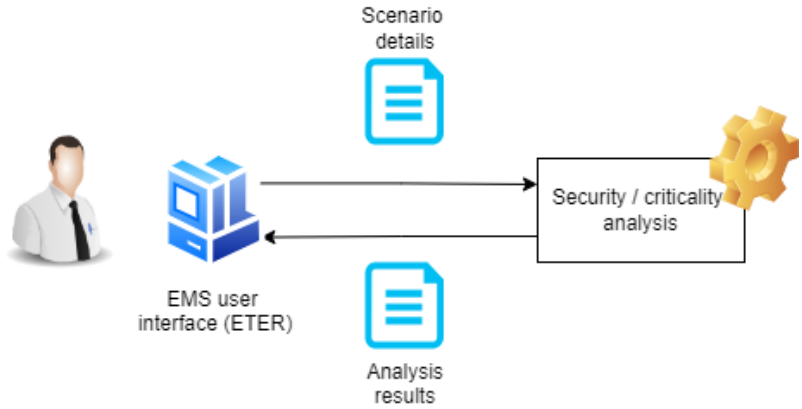


Figure 25 security process integration with ETER

The communication among the different components will make use of NATS messaging. NATS (Neural Autonomic Transport System) is a high-performance messaging system designed for cloud-native applications, IoT messaging, and microservices architectures. It provides lightweight, reliable, and scalable communication between distributed systems. NATS supports various messaging patterns, including publish-subscribe, request-reply, and queuing. Key features include low latency, high throughput, and ease of deployment. It is suitable for scenarios requiring real-time data streaming and efficient message distribution. NATS is open-source and can be used as a standalone server or embedded within applications.

4.6.2 Implementation

The ETER tool GUI has the form of a web application and is developed in Node.js using the Meteor and REACT frameworks. The new scenario-definition functionalities of the tool will be added to ETER using the same technology.

For the security analysis of the scenario defined, a microservice will be developed using Python and PandaPower. The microservice will perform the corresponding load flow and short circuit calculations from the inputs received and will compile and return results for being displayed In the ETER tool.

4.6.3 Mock-Ups

In this section some preliminary version of screenshots of the critical points detection process are presented. The GUI is not finalized but the current version is presented here to illustrate about the entire process.

The network visualization section of the ETER tool is as follows:

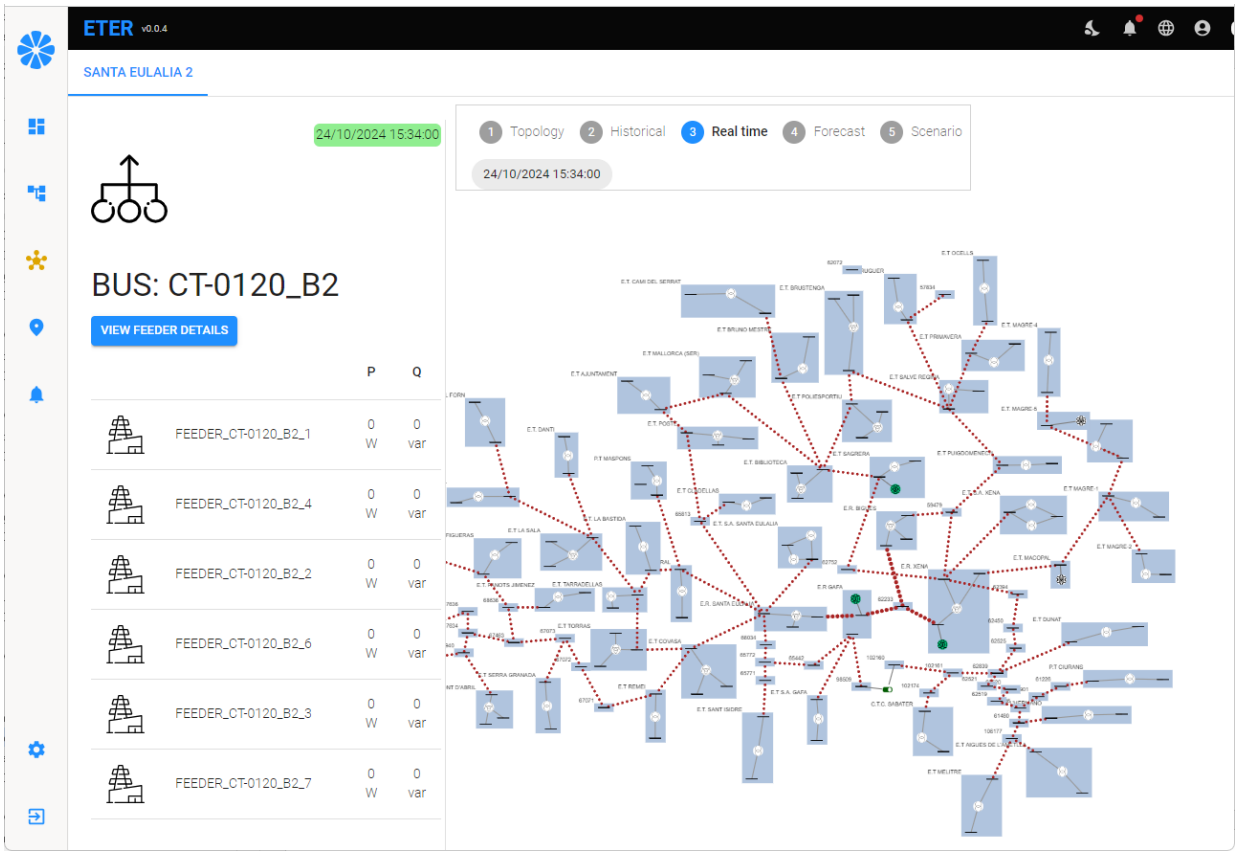


Figure 26 ETER network visualization

There are different sections in this interface. In the centre there is the single line diagram (SLD) of the selected MV network, in this case, Santa Eulalia, the Spanish Pilot. For all the elements in the SLD when selected, the details are presented in a left-hand side panel. In the example, a bus has been selected, but other elements like substations, loads, generators, switches, lines or transformers can also be selected. For each element type, the appropriate information will be presented in the panel upon selected.

When a bus is selected and it contains feeders of LV, there is the option of displaying the related LV network by clicking the appropriate button VIEW FEEDER DETAILS. This adds a new tab at the top of the screen for accessing the details about this specific LV network. This is an example of the visualization of the LV network connected as a feeder on the selected bus:

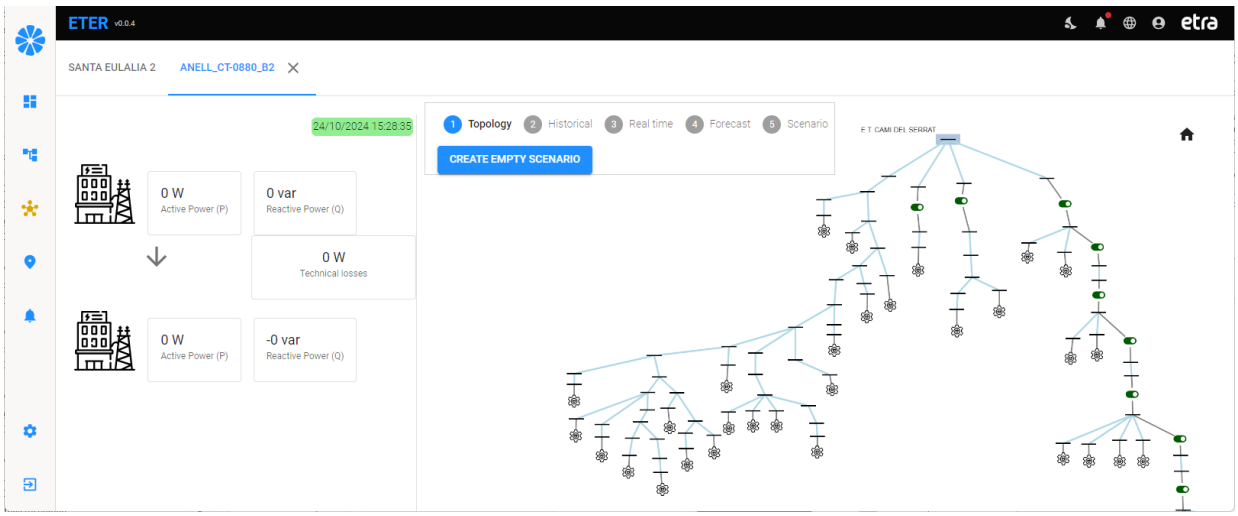


Figure 27 LV network connected as a feeder on the selected bus

Opposite to the MV network, the LV network usually has a radial shape starting from the secondary bus of the transformer. The visualization for the elements is the same as in the MV network.

In the SLD part there is controls to select the type of data to visualize in the SLD and the side panel. It allows selecting among these options:

- Topological data. With no measurements nor states
- Historical data. Allows selecting a past timestamp. The corresponding measurements will be presented, including power flow and state estimation results
- Real time data. Presents the most up-to-date data with the latest power flow calculations
- Forecasted data. Allows selection a future timestamp for whom a power flow and estate estimation exist.
- Scenarios. Allows selecting and managing pre-defined scenarios to run what-if scenarios and simulations. This will be used to run the calculation linked to the critical point detection tool.

These are examples of how the visualization controls are used to conFigure the data to visualize:

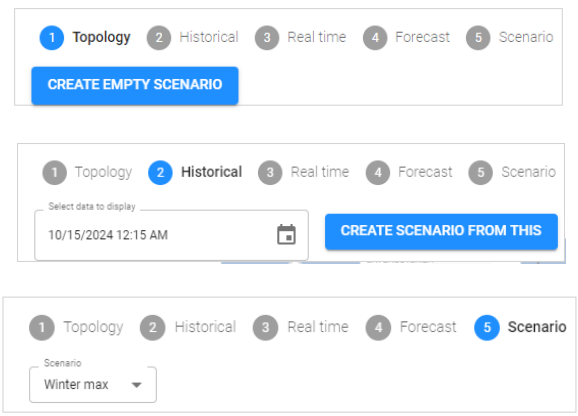


Figure 28 Different visualization types in ETER

The process starts by defining a scenario. This can be done by picking a network topology in the ETER tool and selecting the option for creating an empty scenario in the 'topology' view mode. The name of the scenario must be provided:

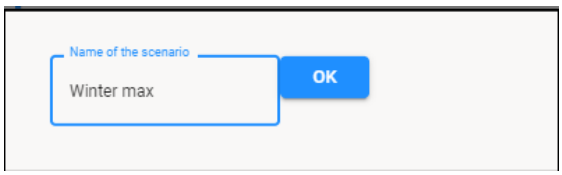


Figure 29 Scenario creation dialog

Other option is to define the scenario from some historical values. This option used to be more convenient because it initializes the scenario with active and reactive power measurements for all load and generators, using the selected timestamp. The first option requires a lot of manual configurations, as active and reactive values are mandatory for the analysis.

The scenario created can be visualized by selecting it in the drop-down list of the scenario visualization control. When selected, the scenario data is presented in the screen:

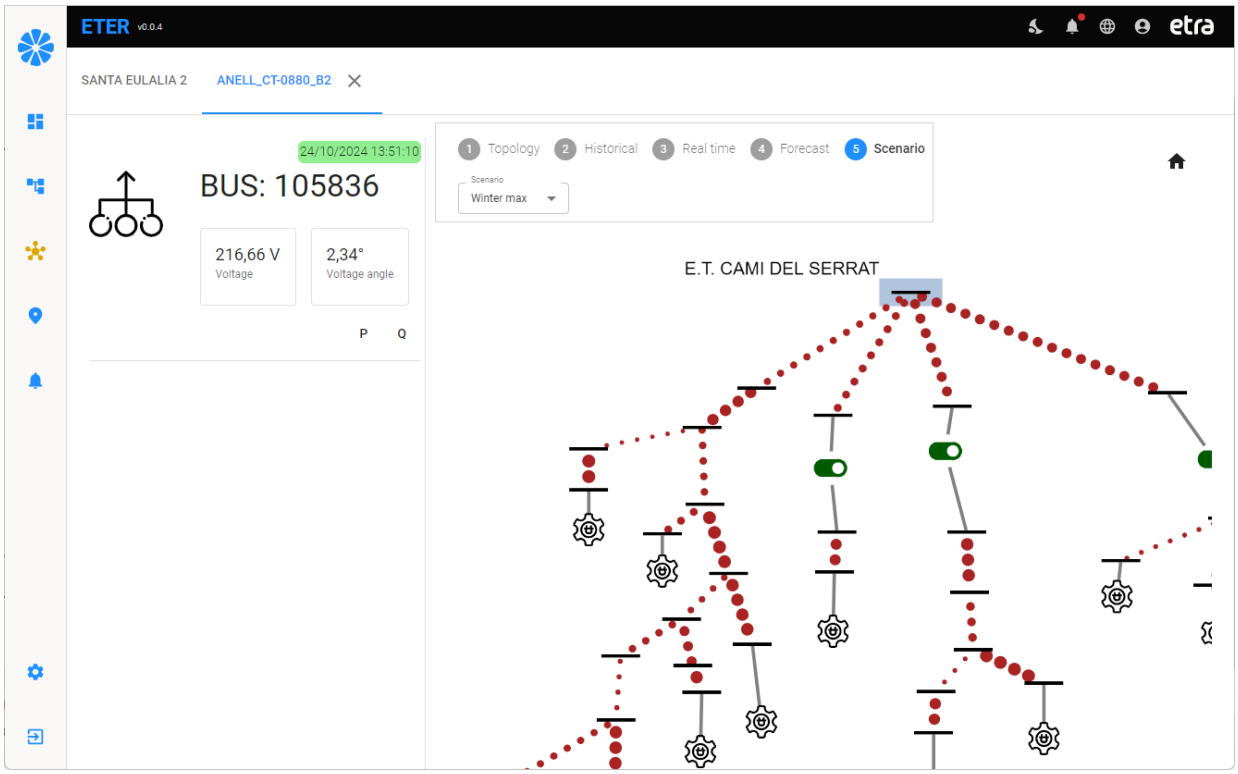


Figure 30 ETER GUI presenting scenario data

Since power flow is already calculated for this scenario, sizes of lines represent the amount of energy flowing through the conductors, and the buses contains voltages and phase angles. By clicking on a load, the corresponding data is displayed:

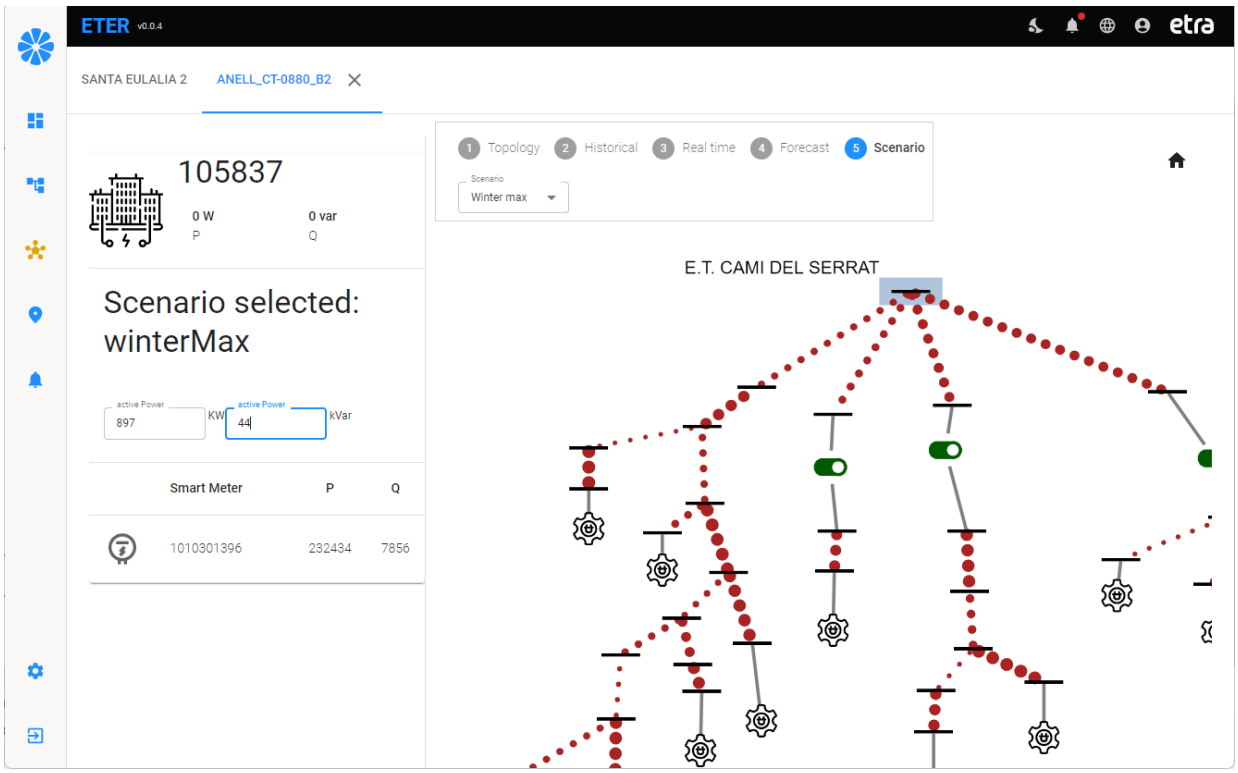


Figure 31 Load details visualization and edition interface

In this case, besides monitoring the load, the interface allows to modify the active or reactive power. This is done by typing the corresponding value in the appropriate text box. This can be done in all loads and generators to simulate specific situations.

The switches status could also be changed by selecting the element and activating the control:

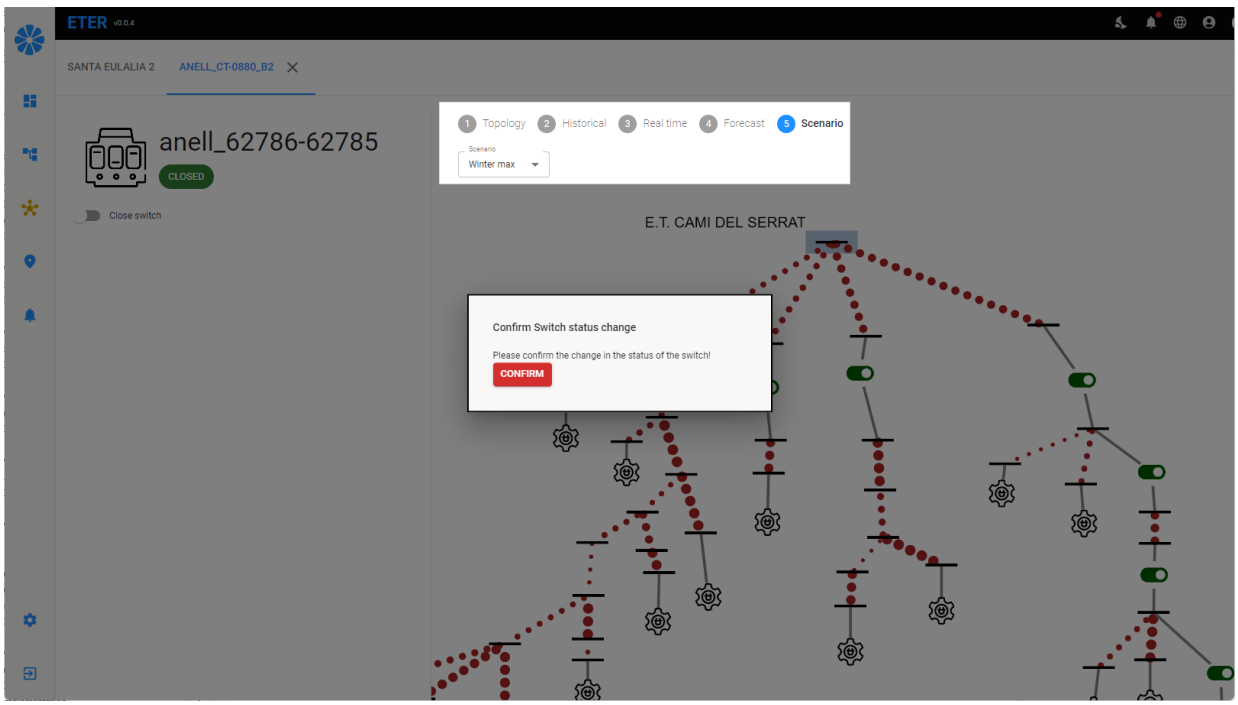


Figure 32 Switch status change confirmation

All the changes applied on the scenario elements are stored and can be reviewed by clicking on the background of the SLD diagram:

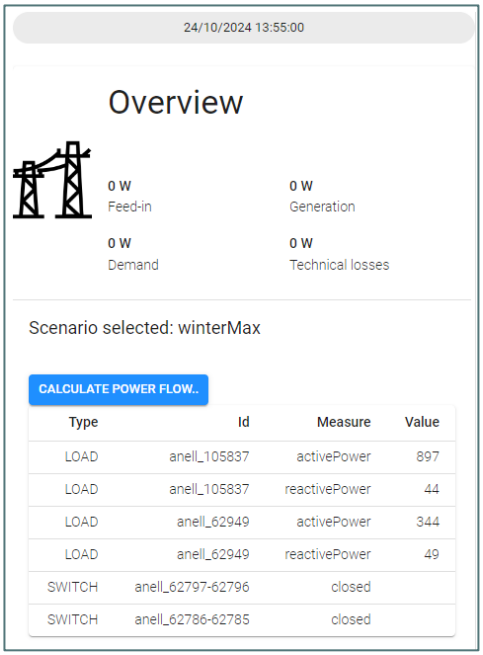


Figure 33 List of changes applied on the scenario

Here the summary of applied changes is presented. When the introduction of scenario parameter finalizes, the resulting power flow can be calculated. This calculation will include the latest updates and will identify potential areas and points that might be affected by the problems considered by this tool. The results are presented as follows:

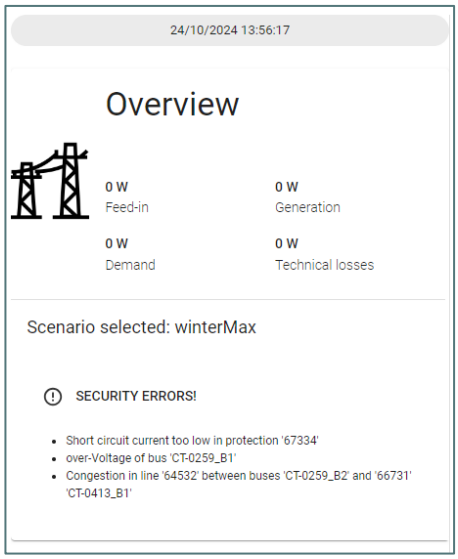


Figure 34 Results of power flow on the scenario defined

4.7 Short term analysis of the impact of DER in the Distribution grid

Distributed energy generation (DERs) has been introduced to power systems, particularly at the Low Voltage level, to make the existing systems more reliable, secure, and efficient. Simultaneously, DER brings different challenges to the system as existing systems are not yet ready to accommodate high DER penetration levels. In this respect. since the current and future trend of electric power systems is set towards increased integration of DERs, a discussion of the impacts of those generation technologies on distribution networks is needed.

One of the problems brought by the adoption of DERs in LV is related to the dimensioning of the networks. Low voltage networks are designed to distribute sufficient energy to all customers. To manage uncertainties in load consumption, each supply point is capped with a maximum active power limit. If consumption exceeds this limit, the customer experiences curtailment. The dimensioning of the network considers the sum of the limits of all the customers in a feeder.

The simultaneity factor is a coefficient used in electrical engineering to estimate the peak load on a distribution network. It reflects the probability that multiple consumers will not use their maximum active power demand at the same time. By applying this factor, DSOs can design more efficient and cost-effective low voltage networks, ensuring adequate capacity without over-dimensioning. This factor typically ranges from 0.6 to 0.8, indicating that only 60-80% of the maximum possible load is considered for network planning by the DSO. This factor is legally regulated in all countries.

The simultaneity factor can be impacted by Distributed Energy Resources (DERs). DERs, such as solar panels and battery storage, can introduce variability and unpredictability into the grid. This can affect the traditional load patterns and potentially reduce the accuracy of the simultaneity factor, necessitating adjustments in network planning and dimensioning.

There are different approached to tackle for this issue, but we will focus on enhanced forecasting models that will identify problems in the network during peak distribution generation periods, The idea is to identify these problems beforehand by using advanced forecasting techniques and apply corrective actions to alleviate the problem. This could be considered as **curtailment for peak-shaving** and is a technique that allows saving money avoiding investments in the grid that will only be necessary on very rare extreme conditions. Additionally, this will let the DSO safely use the simultaneity factor for network modelling.

4.7.1 Design

The core of the short-term analysis of the impact of DERs in the distribution grid is the forecasting of the DERs in the LV network and the calculation of the state estimation for some future periods.

This is very similar to the pseudo-measurements calculation described in section 4.2 – 'Enhanced state estimation tool', Actually the AI models for calculating the active and reactive power injections of the DERs will be the same, but instead of using the current weather state conditions for the estimation, it will consider the weather forecast obtained from an external source.

This functionality will be calculated for different future instants:

- Every hour at minute 0, the DER generation forecast for the next hour will be generated, considering the most up-to date weather forecast.
- Every hour at minute 0, the DER generation forecast for 12 hours ahead will be generated, considering the most up-to date weather forecast.

The idea of having two-time windows is for because the weather forecast could change and thus it is required to recalculate it periodically. The 12 hours ahead calculation will let DSO time enough to react and apply corrective actions, while the 1 hour ahead calculation will trigger the last and most accurate signal danger signal in case of potential errors predicted.

4.7.2 Implementation

This tool will be developed as a microservice linked to ETER. It will have the form of a python script that will internally feature different tasks periodically scheduled to calculate for forecasts. This task will select a future timestamp, compile the necessary data from the weather forecast provider and request the state estimation microservice to calculate the power flow and state estimation of the forecasted values. The results will be stored and presented in the ETER tool.

4.7.3 Mock-Ups

As presented in the previous section, ETER features controls to select the information to present in the topological representation. For this tool, the forecast visualization is selected:

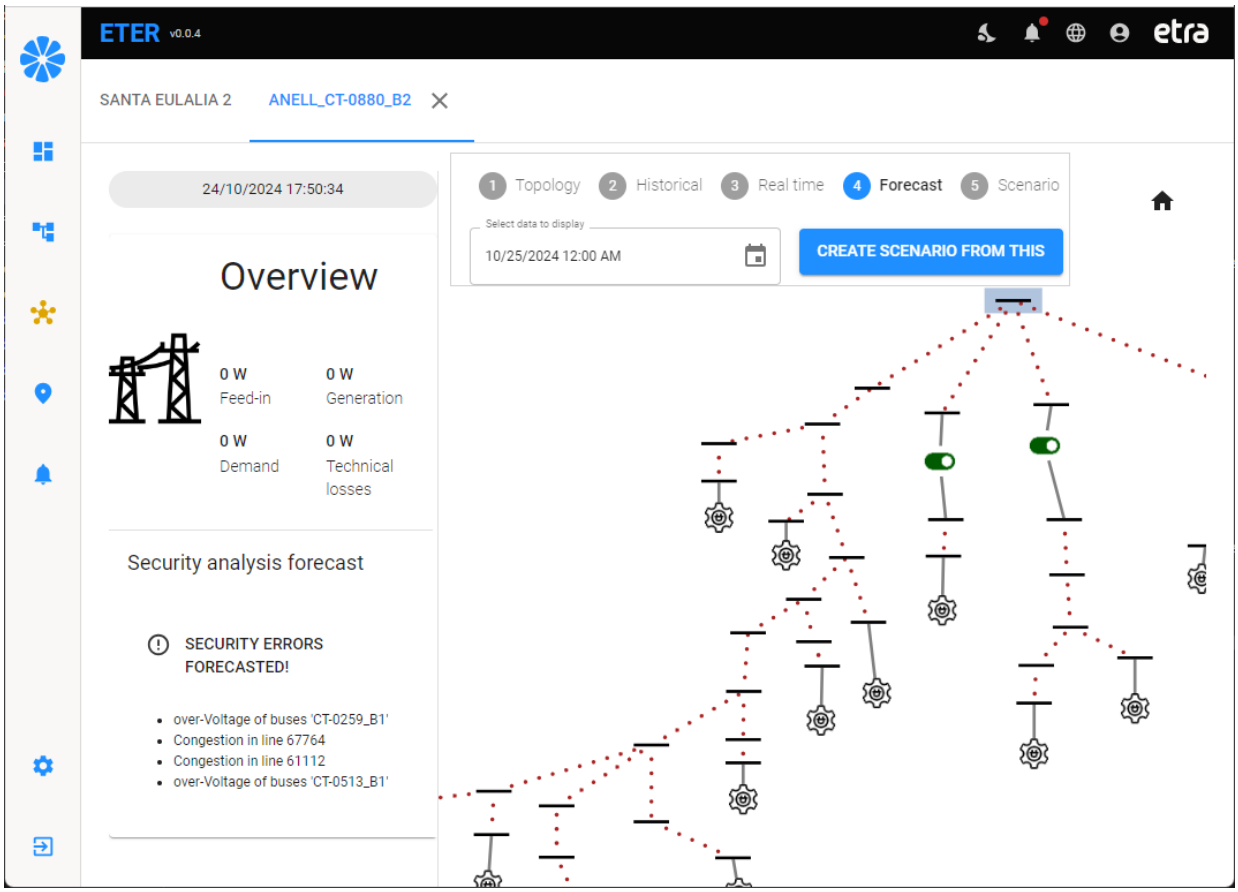


Figure 35 forecast visualization selected

As can be seen, for the selected period of time, the network presents problems associated with the voltages at buses with DERs and line congestion.

A scenario can be created from this data, for testing different solutions by curtailing specific DERs. This is described in section '4.6 - Critical point detection tool'

4.8 Real-Time Thermal Rating Module

The static limits for overhead lines used by system operators are calculated usually based on the worst case of expected environmental conditions. The major benefit of Dynamic Line Rating (DLR) or RTTR systems lie in the calculation of the maximum current, that leads to the temperature limit in

the overhead conductor, taking into consideration the existing environmental conditions, solar radiation, wind speed and direction and ambient temperature and not the worst-case conditions. Thus, additional capacity can be unlocked in the transmission/distribution lines, the flexibility of the system can be increased, future investments decisions can be made with more solid knowledge on the actual line loading conditions and higher RES penetration levels can be allowed.

Industrial practices are mainly focussed on transmission lines and direct methods to compute the RTTR. Direct methods measure either conductor sag, conductor ground clearance, line tension or conductor temperature. In indirect methods of RTTR, weather data at specific locations along a transmission line are analyzed to calculate the current carrying capacity of the line.

To calculate the steady-state current-carrying capacity of the transmission line conductor under given weather conditions, the heat balance equation in IEEE std [35] is followed:

$$I = \sqrt{\frac{q_c + q_r - q_s}{R(T_c)}}$$

where q_c and q_r are the heat removed by convection and radiation to surrounding air, respectively, while q_s and $I^2 * R(T_c)$ are the heat gained from solar radiation and the heat generated by the current flowing through the conductor, respectively. *I* represents the line current and $R(T_c)$ is the conductor resistance at temperature T_c (the conductor core temperature).

Both indirect and direct approaches require measurements of current in the line (and temperature or voltage sag or conductor ground clearance or line tension in direct methods), as well as weather measurements on many sections of the line. Based on these measurements the current rating on various sections of the line is computed and the minimum among them is specified as the dynamic current rating. Thus, there is a need for multiple sensors along the line for the implementation of an industrial RTTR system.

RTTR applications have been predominantly developed for transmission lines, with limited research or industrial focus on their implementation for distribution feeders. Compared to a transmission line that has the same current flowing in all its sections, a distribution feeder might have different current in various sections due to MV/LV substations to supply loads or distributed generation. RTTR in a distribution system could offer the same benefits but might not be a cost-effective solution if an extensive roll-out of sensors is required (since the lower voltage levels would result in less power allowed if the current limit is increased, compared to a transmission line). Taking into account that an industrial approach might require an adequate number of sensors to cover the different current values among the feeder sections, gathering conductor temperature and weather measurements, a sensor based RTTR approach might not be cost effective.

The proposed solution in OPENTUNITY focusses on distribution systems providing a low-cost solution that uses existing measurements of a distribution network to estimate the current values on different feeder sections as well as to estimate, via machine learning the future current measurements in a probabilistic manner, using also high resolution weather forecasts.

4.8.1 Design

This module is basically composed of two distinctive functionalities, one offline and one online. The offline functionality purpose is to collect the necessary data to train the machine learning models and define the real-time communication settings. The online operation collects the latest measurements,

store them in a local database, then use them to prepare the input of the machine learning models, collect the required weather data from a weather provider and compute the RTTR of the line, and finally publish the results in the GUI and via a communication protocol to the DSO.

4.8.1.1 Offline sub-module

The first step in offline operation is for the user to access the GUI and upload the files that provide the characteristics of the feeder. The feeder static data consist of the data of each line section: a name of line section, whether that section is overhead or underground, the cross section of the conductor, the conductor material (e.g. Al or CU), the resistance and reactance in Ohm/km, the section length, the static current rating, and the latitude and longitude of the terminals of the section. The larger the resolution on the different points that divide the line in sections the better, since the actual line direction in each section can be computed which is required in a later stage to compute the angle of attack of the wind on the conductor. Additional data that are required are: section terminal names, meter tag names, terminal location and measurement types.

The second step is to upload the historical measurements collected on the feeder. These measurements either have historical current measurements that can be used, or measurements that can be used to estimate the current on sections that no current measurements are available (via state estimation).

When the historical current data on all the feeder sections are collected or generated, a machine learning regression model is trained. The predictor aims to calculate the probabilistic prediction of the future current values, in a horizon of 6 hours.

Finally, the settings of communication between the on-line submodule and the system operator infrastructure are also part of the sub-model, where the user can define in the GUI the communication settings.

4.8.1.2 Online sub-module

In the on-line submodule the measurements are collected and stored in the device that performs the RTTR. Every hour, the latest historical data are then accessed to perform the probabilistic forecast of current in every section for the next 6 hours. At the same intervals the weather forecasts for the different sections are collected from a weather provider. The weather forecasts consist of estimation with hourly resolution of the ambient temperature, wind speed and direction. The wind direction in every section is then used alongside with the latitude and longitude of the terminals of the section to calculate the angle of attack of the wind to the overhead conductor in that section.

Finally, the wind angle of attack, the wind speed, the ambient temperature and the current probabilistic forecasts are used to compute a probabilistic real time thermal rating and a temperature estimation according to IEEE std [36] . The impact of solar radiation is computed via the IEEE method that uses the hour of day as input to compute the solar radiation on the conductor based on the sun location in the sky.

The probabilistic loading based on the calculated RTTR and conductor temperature forecast are then sent via the defined communication to the system operator and presented in the GUI results section.

4.8.2 Implementation

The flowchart of Figure 36 describes the process of the code implemented for the off-line submodule. The process begins with the user uploading an Excel file containing topology data and measurement location and type data. Once the file is uploaded, the system checks if the file is in the correct format (e.g. if data is missing in any section, etc.). If the format is incorrect, errors are printed for the user to address. However, if the format is correct, the system proceeds to upload the topological data into a local MySQL database.

Following this, the user uploads a CSV file containing historical measurements. The system then retrieves the topological data from the MySQL database and proceeds with a measurement assessment to ensure the data's validity, i.e. checking if the measurement tag is assigned in correctly. If the measurement format is incorrect, the system prints errors to inform the user. If both the topology and measurement files are in the correct format, the process is successfully completed. The GUI is built using the streamlit [37] library in python and pandas library is used to process the input file and load them to the MySQL database.

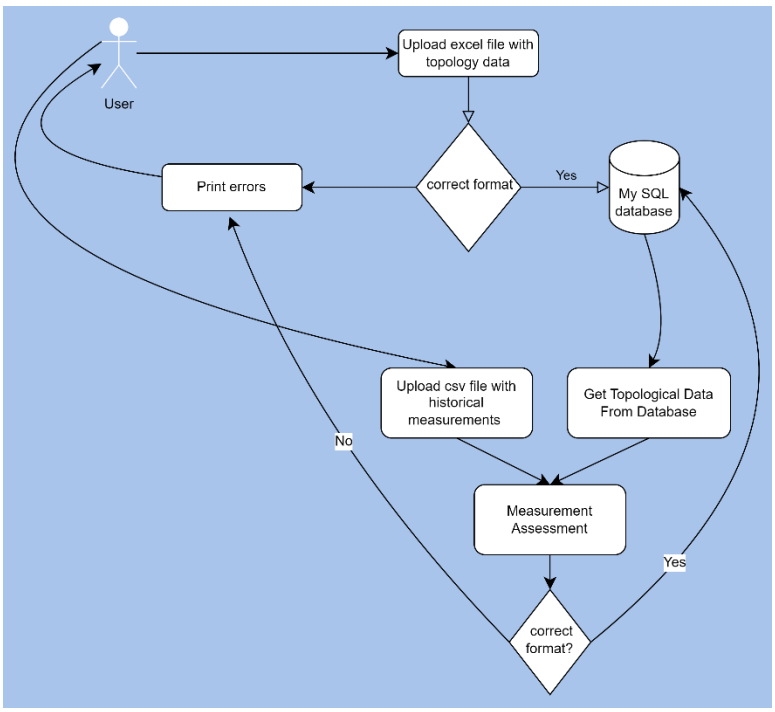


Figure 36: Topology & Historical data assessment

The flowchart of Figure 37 illustrates the process of training probabilistic forecast models for the current in different feeder sections, using historical and topological data. Initially, the user triggers the model training via the GUI. The programs searches the local database to get the topology data and the measurements that have been stored there, based on the data provided by the user.

Then, it searches to see if there are sections' currents that are missing. If there are no such sections, it uses the measurement data to generate the training data for the machine learning models. If currents are missing in a section, then the measurements and the topology data are used to build the

feeder as a pandapower network using the pandapower [38] library in python. With the available measurements, the state estimation module of pandapower library is used to estimate the currents that are not directly available.

Using the available (or generated historical data), the training data for the machine learning models are built. Specifically, the moving average of current for the last six and three hours, the differences of the latest hour to the moving averages, the last hour current, the hour and the month of the prediction horizon are used as attributes on the training data. The currents of the next six hours are used as output data. A different model is built for every hour in the horizon.

The algorithm of quantile regression forests [39], is used to train a model for each hour in the prediction horizon that computes 0.1, 0.5 and 0.9 quantiles in each forecast. The key difference between quantile regression forests and random forests is that for each node in each tree, random forests keeps only the mean of the observations that fall into this node and neglects all other information. In contrast, quantile regression forests keeps the value of all observations in this node, not just their mean, and assesses the conditional distribution based on this information [38]. Once the models are trained, they are stored for use with real-time data, to predict conductor temperature or line loading predictions.

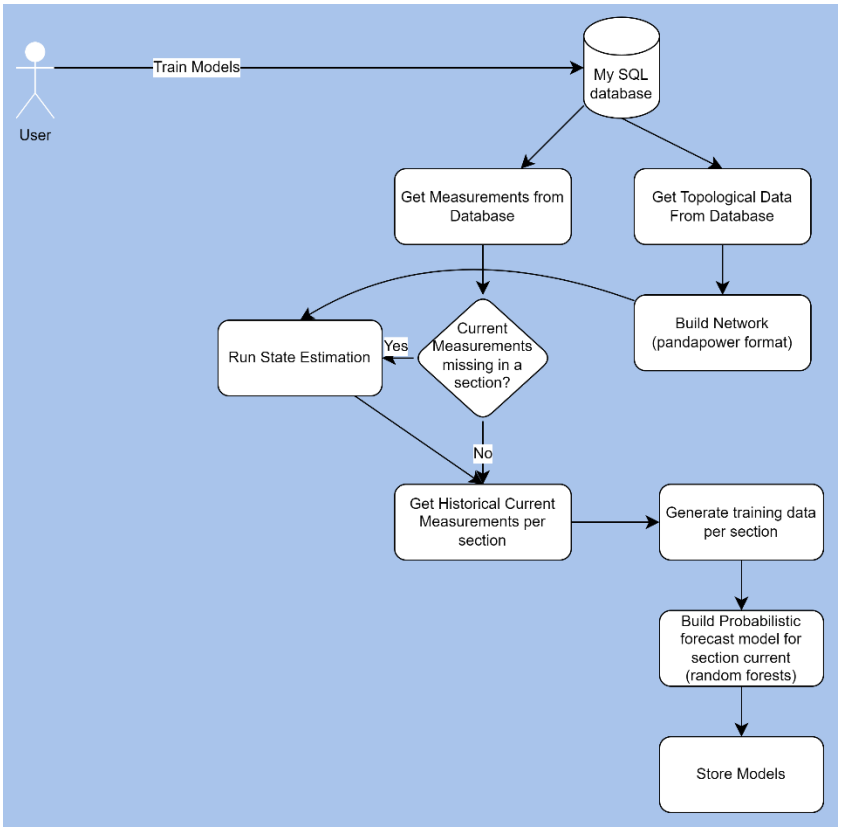


Figure 37: Prediction Models training procedure

This flowchart of Figure 38 outlines the process of online submodule, i.e. generating and publishing conductor temperature and loading forecasts using a combination of real-time measurements, network topology, weather data, and machine learning models.

The real-time measurements will be provided via MQTT protocol, using the paho-mqtt library [40] , which is then stored in a MySQL database. The system retrieves the last six hours of these measurements and checks if any conductor sections have missing current. If some sections are missing measurements, in a similar manner to what is described previously, a state estimation algorithm to fill in the missing data. Finally, the data are formed in the format required by the machine learning models to produce the probabilistic current forecasts, using the locally stored models.

The next step involves fetching topology data, which is used to build the network in a format suitable for the pandapower framework, in case state estimation is needed. In addition the geodata, cross sections and resistance in Ohm/km of the line sections are also pulled. Simultaneously, the system gathers weather data, wind speed, direction, and ambient temperature, from a weather provider (meteomatics [41]. This specific provider allows high resolution weather forecasts of 1km that can be downscaled to 90m. Forecasts are pulled for every section based on the mean latitude and longitude, computed by the geodata of the section terminals. The weather forecasts are pulled every hour. Based on the geodata of every section and the wind direction forecasts the angle of attack to the overhead conductor of each section is calculated.

With all the required inputs ready (probabilistic current for the next six hours, wind speed, angle of attack and ambient temperature, conductor section static data), the system performs calculations according to IEEE standard to compute both conductor temperature forecasts and conductor loading forecasts. Based on the probability of the current forecasts, a probability is assigned in every forecast calculation. These forecasts are then published using MQTT for the DSOs to consume. Additionally, the system generates graphical plots for the graphical user interface (GUI), making the results visually accessible.

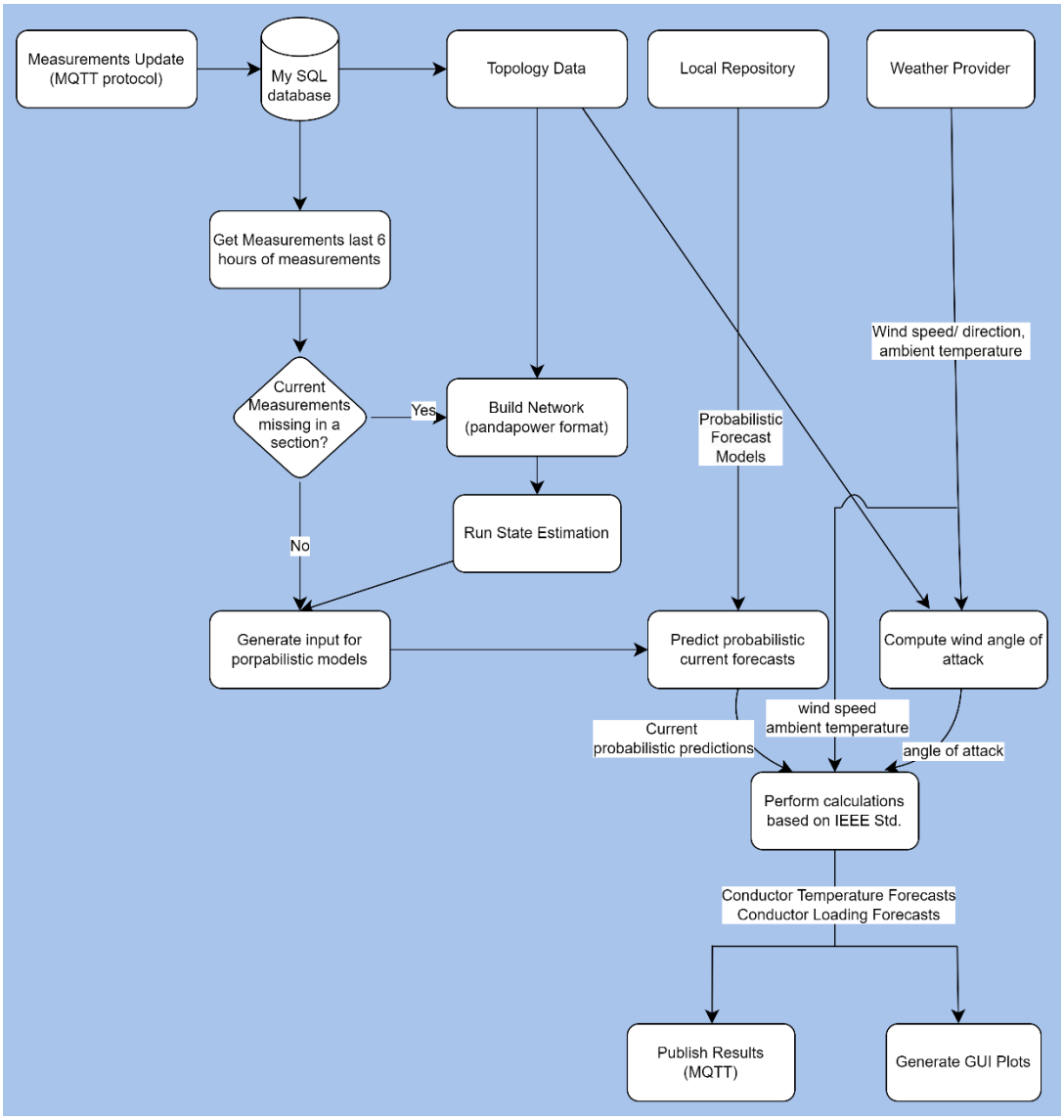


Figure 38: Real-time operation of RTTR Module

4.8.3 Mock-Ups

The Graphical User Interface will be designed to manage the machine learning models for forecasting line loading and conductor temperature based on uploaded topology and historical data, as well as to establish connection for reading the latest measurements and write the latest results. Finally, it has a section where the results are presented graphically. It consists of four tabs (Topology, Historical Data & Model Training, Server Configuration, Results) each serving a specific purpose. The GUI presented here is the first version and it will be updated and presented further in detail in the second version of the deliverable.

In the topology tab the user has to upload an Excel file (.xlsx or .xls), with the topological data of the feeder. These data contain information of the sections of the feeder that are underground or

overhead, the cross section and the material of overhead conductor, the section length, the resistance and reactance in (ohm/km), the maximum current and the latitude and longitude of the feeder section end points. In addition, the different measurements across the feeder that will be provided in historical data and the real -time interface are also described, linking the different measurement names that will be uploaded in the historical data tabs with locations in the feeder and type of measurement (Voltage, Current, Active Power, Reactive Power).

The second tab serves two purposes, 1) to gather the necessary historical data for training the current prediction models through a csv file that the user uploads, 2) train the models or delete existing models and re-train them with new data. The tab is presented in the following Figure, where the section to select the csv file is first, on overview of the data is presented next and information about the machine learning models is shown at the bottom giving the option to the user in that specific case to delete the existing models and re-train new ones.

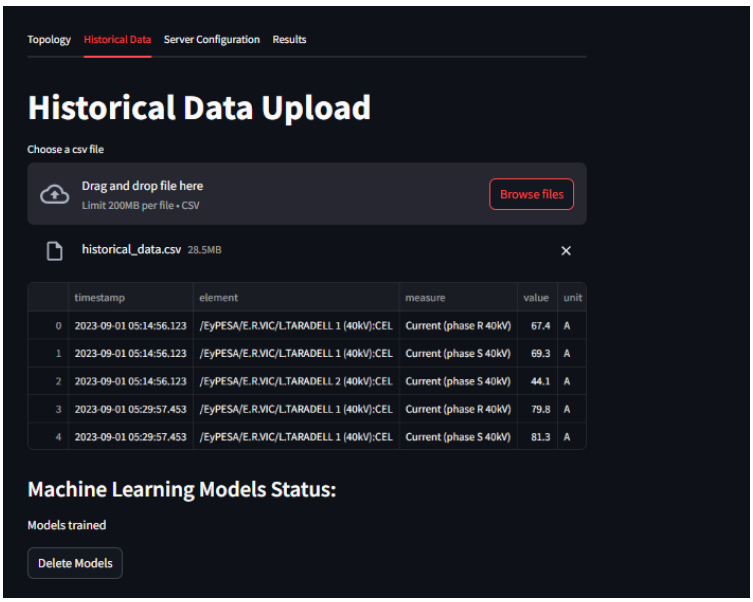


Figure 39: Historical Data upload tab

The third tab (Server Configuration) requests the MQTT server credentials from the user in order to get the latest measurements and store them locally in the MySQL database. Both Spanish and Greek pilots that will use this innovation will use MQTT protocol to provide real-time measurements and get the results from the module. Specifically, it requests the credentials (username and password), HOST IP, Port, the topics to subscribe to get the measurements and the topic to publish the results. The following Figure presents the overview of this tab in the GUI.

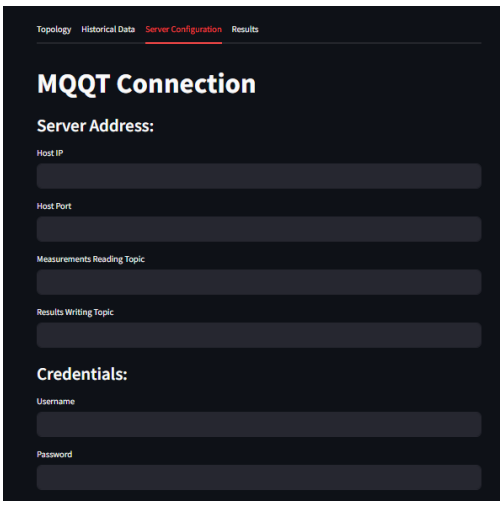


Figure 40: Server Configuration tab

Finally, during real-time operation, the user would be able to access through the GUI of the module the results of the probabilistic forecast on conductor loading and temperature. The predictions will be updated every hour using the measurements provided through the MQTT interface. The results, conductor loading (probabilistic RTTR, static) and probabilistic conductor temperature forecasts, will be available in two ways:

- Through two plots in the tab results of the app. These two plots are presented in the following Figures.
- Via an mqtt message that publish in a specific topic the timestamps and the respective forecasts.

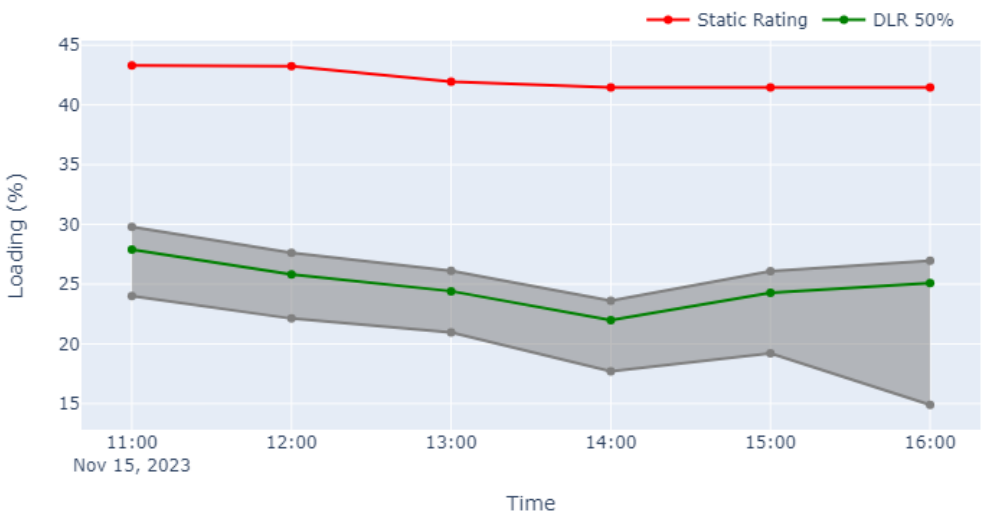


Figure 41: Conductor loading with static rating and Probabilistic Real Time Thermal Rating

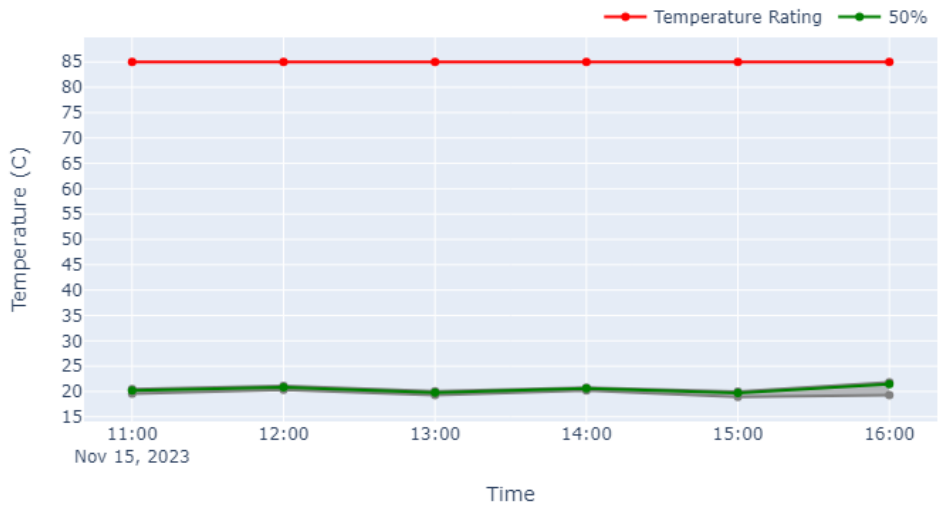


Figure 42: Conductor temperature limit and probabilistic forecast of conductor temperature

4.8.3.1 Numerical studies

A candidate feeder have been selected for the Spanish demo that operates on 40kV, has 116 square mm cross section and 315A static current rating and total length of 8.6 km on its overhead part. The current is measured in one of its ends and no secondary substations are across the line length.

Histrorical data on current from the whole 2023 and up to February of 2024 was used for training and data from March 2024 to September 2024 for validation of current estimation algorithm. The following table presents the results on current probabilistic prediction presenting the mean and standard deviation in error estimation, between the 50% quantile and the actual value, and the percentage of scenarios that lay above the upper limit of the current.

In the evaluation period, assuming that the system operator makes the most conservative choice for the RTTR, selecting the 90% quantile, an average increase of 10 MW in the maximum power transfer has been observed and an increase in the current rating at 98.5% of the time.

Table 3. Accuracy of Current Prediction Submodule

Hour Ahead	Results on Accuracy		
	Mean (A)	Std (A)	% of scenarios over the 90% quantile
1	3	4.53	8
2	**4.12**	**5.93**	**6.5**
3	5.93	4.9	6.4
4	**5.39**	**5.66**	**5.3**
5	5.77	6.13	5.0
6	**6.2**	**6.62**	**5.3**

5 CONCLUSIONS

The modules described in this document make significant advancements in technologies for Distribution System Operators (DSOs). The developed tools aim to enhance the accuracy, reliability, and efficiency of power distribution networks.

The Topology identification tool focus on an innovative task, the semi-automatic building of the LV network topology by making use of the smart meter measurements. This could facilitate the DSO tasks of deploying and start managing low voltage networks.

Two of the modules, namely the **Topology detection** and the **Fuse burn detection for early outage and islanding recovery** focus on detecting errors in the topological database that might lead to security problems and suboptimal operation of the grid.

The last module linked to the topology is the **real-time thermal rating module**, that is focusing on determining the real ampacity of power lines given actual the climate conditions.

The two **state estimation** versions defined could also assist DSO in the process of safely manage the grid, by proving enhanced observability based on new measurements coming from PMUs and pseudo-measurements calculation.

Finally, two processes are designed to complement the DSO duties: the **critical point detection tool** can be seen as a grid planification tool that helps identifying weakness in **the grid. The Short term analysis of the impact of DER in the Distribution grid**, will warn DSO upon the detection of security problems linked to DER assets.

It is noteworthy that the modular approach ensures that the modules can be adapted and scaled according to the specific needs of different DSOs and network configurations.

The information included may evolve and be enhanced in future deliverables, particularly in the upcoming D5.2 "OPENTUNITY power flow developments (v2)", which will focus on the final functionalities and user interfaces of the developed modules.

6 REFERENCES AND ACRONYMS

6.1 References

1. Control and automation systems for electricity distribution networks (EDN) of the future, Paris: Cigré, 2017.

2. Power System State Estimation: Theory and Implementation, New York, NY: Marcel Dekker, 2004.

3. «State estimation concepts and terminology,» 2016.

4. «Fundamental research challenges for distribution state estimation to enable high-performing grids,» 2018.

5. «Power system static state estimation – Part I, II, III,» *IEEE Trans. Power App. Syst.,* Vols. %1 de %2PAS-89, nº 1, p. 120–135, Jan. 1970.

6. «A survey of power system state estimation using multiple data sources: PMUs, SCADA, AMI, and beyond,» *IEEE Trans. Smart Grid,* vol. 15, nº 1, p. 1129–1151, Jan. 2024.

7. G. W. G. G. Liang Zhang, «Distribution System State Estimation Via Data-Driven and Physics-Aware Deep Neural Networks,» [En línea]. Available: https://www.researchgate.net/publication/334238222_Distribution_System_State_Estimation_Via_Data-Driven_and_Physics-Aware_Deep_Neural_Networks.

8. G. W. a. G. B. G. L. Zhang, «Real-time power system state estimation via deep unrolled neural networks,» *Proc. Global Conf. on Signal and Info. Process..*

9. «Open Weather,» [En línea]. Available: https://openweathermap.org/.

10. «MLFlow,» [En línea]. Available: https://mlflow.org/.

11. *. M. Joshi, H.K. Verma, "Synchrophasor measurement applications and optimal PMU placement: A review," Electric Power Syst. Res., vol. 199, no. 2, Oct. 2021..*

12. *G. Cheng, Y. Lin, A. Abur, A. Gómez-Expósito, and W. Wu, "A survey of power system state estimation using multiple data sources: PMUs, SCADA, AMI, and beyond," IEEE Trans. Smart Grid, vol. 15, no. 1, pp. 1129–1151, Jan. 2024..*

13. *P. M. Joshi, H.K. Verma, "Synchrophasor measurement applications and optimal PMU placement: A review," Electric Power Syst. Res., vol. 199, no. 2, Oct. 2021..*

14. *Y. Liu, L. Wu, and J. Li, "D-PMU based applications for emerging active distribution systems: A review," Electr. Power Syst. Res., vol. 179, Art no. 106063, Feb. 2020..*

15. *M. Netto, V. Krishnan, Y. Zhang, and L. Mili, "Measurement placement in electric power transmission and distribution grids: Review of concepts, methods, and research needs," IET Gener. Transm. Distrib., vol. 16, no. 5, pp. 805–838, Mar. 2022..*

16. *"Fundamental research challenges for distribution state estimation to enable high-performing grids," NYSERDA, Report Number 18–37, May 2018..*

17. *K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan, and F. Bu, "A survey on state estimation techniques and challenges in smart distribution systems," IEEE Trans. Smart Grid, vol. 10, no. 2, pp. 2312–2322, Mar. 2019..*

18. *"State estimation concepts and terminology," Power and Energy Society (PES-TR20), Tech. Rep., 2016..*

19. *G. Cheng, Y. Lin, A. Abur, A. Gómez-Expósito, and W. Wu, "A survey of power system state estimation using multiple data sources: PMUs, SCADA, AMI, and beyond," IEEE Trans. Smart Grid, vol. 15, no. 1, pp. 1129–1151, Jan. 2024..*

20. *G. Cheng, Y. Lin, A. Abur, A. Gómez-Expósito, and W. Wu, "A survey of power system state estimation using multiple data sources: PMUs, SCADA, AMI, and beyond," IEEE Trans. Smart Grid, vol. 15, no. 1, pp. 1129–1151, Jan. 2024..*

21. *"Distribution system state estimation via data-driven and physics-aware deep neural networks," 2019 IEEE Data Science Workshop (DSW), Minneapolis, MN, USA, 2019, pp. 258–262..*

22. *Y. Weng, R. Negi, C. Faloutsos and M. D. Ilić, "Robust Data-Driven State Estimation for Smart Grid," in IEEE Transactions on Smart Grid, vol. 8, no. 4, pp. 1956–1967, July 2017,.*

23. *S. Radhoush, B. Maryam, N. Hashem, and S. Zagros Shahooei, "A review on state estimation techniques in active distribution networks: Existing practices and their challenges," Sustainability, vol. 14, no. 5, Feb. 2022..*

24. *R. Madbhavi, B. Natarajan and B. Srinivasan, "Graph Neural Network-Based Distribution System State Estimators," in IEEE Trans. Ind. Inform., vol. 19, no. 12, pp. 11630–11639, Dec. 2023..*

25. *B. Azimian, R. S. Biswas, S. Moshtagh, A. Pal, L. Tong, and G. Dasarathy, "State and topology estimation for unobservable distribution systems using deep neural networks," IEEE Trans. Instrum. Meas., vol. 71, Art no. 9003514, pp. 1–14, Apr. 2022..*

26. *H. Sapountzakis, T. Xygkis, K. Andresakis, A. Dimeas, and G. Korres, "Machine learning for distribution grid topology identification and state estimation," PAC World Conf. 2024, Athens, Greece, 2024, pp. 1–14..*

27. *K. R. Mestav, J. Luengo-Rozas, and L. Tong, "Bayesian state estimation for unobservable distribution systems via deep learning," IEEE Trans. Power Syst., vol. 34, no. 6, pp. 4910–4920, Nov. 2019..*

28. *B. Azimian, R. S. Biswas, S. Moshtagh, A. Pal, L. Tong, and G. Dasarathy, "State and topology estimation for unobservable distribution systems using deep neural networks," IEEE Trans. Instrum. Meas., vol. 71, Art no. 9003514, pp. 1–14, Apr. 2022..*

29. *B. Azimian, R. S. Biswas, S. Moshtagh, A. Pal, L. Tong, and G. Dasarathy, "State and topology estimation for unobservable distribution systems using deep neural networks," IEEE Trans. Instrum. Meas., vol. 71, Art no. 9003514, pp. 1–14, Apr. 2022..*

30. *S. Radhoush, B. Maryam, N. Hashem, and S. Zagros Shahooei, "A review on state estimation techniques in active distribution networks: Existing practices and their challenges," Sustainability, vol. 14, no. 5, Feb. 2022..*

31. *R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education, " IEEE Trans. Power Syst., vol. 26, no. 1, pp. 12–19, Feb. 2011..*

32. *S. H. Dolatabadi, M. Ghorbanian, P. Siano, and N. D. Hatziargyriou, "An enhanced IEEE 33 bus benchmark test system for distribution system studies," IEEE Trans. Power Syst., vol. 36, no. 3, pp. 2565–2572, May 2021..*

33. *R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education, " IEEE Trans. Power Syst., vol. 26, no. 1, pp. 12–19, Feb. 2011..*

34. *"IEEE/IEC International Standard - Measuring relays and protection equipment - Part 118 -1: Synchrophasor for power systems - Measurements," IEC/IEEE 60255 -118 -1:2018, pp. 1–78, Dec. 2018..*

35. *IEEE Standard for Calculating the Current-Temperature Relationship of Bare Overhead Conductors," in IEEE Std 738-2012 (Revision of IEEE Std 738-2006 - Incorporates IEEE Std 738-2012 Cor 1-2013) , vol., no., pp.1-72, 23 Dec. 2013, doi: 10.1109/IEEESTD.2013.*

36. *EEE Standard for Calculating the Current-Temperature Relationship of Bare Overhead Conductors," in IEEE Std 738-2012 (Revision of IEEE Std 738-2006 - Incorporates IEEE Std 738-2012 Cor 1-2013) , vol., no., pp.1-72, 23 Dec. 2013, doi: 10.1109/IEEESTD.2013..*

37. «https://streamlit.io/,» [En línea].

38. *L. Thurner, A. Scheidler, F. Schäfer et al, pandapower - an Open Source Python Tool for Convenient Modeling, Analysis and Optimization of Electric Power Systems, in IEEE Transactions on Power Systems, vol. 33, no. 6, pp. 6510-6521, Nov. 2018.*

39. *N. Meinshausen, "Quantile Regression Forests", Journal of Machine Learning Research, 7(Jun), 983-999, 2006. http://www.jmlr.org/papers/volume7/meinshausen06a/meinshausen06a.pdf.*

40. «https://pypi.org/project/paho-mqtt/,» [En línea].

41. «https://www.meteomatics.com/en/weather-api/weather-api-free/,» [En línea].

42. «Synchrophasor measurement applications and optimal PMU placement: A review,» *Electric Power Syst. Res.,* vol. 199, nº 2, Oct. 2021.

43. «State and topology estimation for unobservable distribution systems using deep neural networks,» *IEEE Trans. Instrum. Meas.,* vol. 71, pp. Art no. 9003514, pp. 1–14, Apr. 2022.

44. «Robust Data-Driven State Estimation for Smart Grid,» *IEEE Transactions on Smart Grid,* vol. 8, nº 4, p. 1956–1967, July 2017.

45. «Measurement placement in electric power transmission and distribution grids: Review of concepts, methods, and research needs,» *IET Gener. Transm. Distrib.,* vol. 16, nº 5, p. 805–838, Mar. 2022.

46. «MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education,» *IEEE Trans. Power Syst.,* vol. 26, nº 1, p. 12–19, Feb. 2011.

47. «Machine learning for distribution grid topology identification and state estimation,» de *PAC World Conf. 2024*, Athens, Greece, 2024.

48. «Graph Neural Network-Based Distribution System State Estimators,» *IEEE Trans. Ind. Inform.,* vol. 19, nº 12, p. 11630–11639, Dec. 2023.

49. «D-PMU based applications for emerging active distribution systems: A review,» *Electr. Power Syst. Res.,* vol. 179, p. Art no. 106063, Feb. 2020.

50. «Distribution system state estimation via data-driven and physics-aware deep neural networks,» de *2019 IEEE Data Science Workshop (DSW)*, Minneapolis, MN, USA, 2019.

51. «Bayesian state estimation for unobservable distribution systems via deep learning,» *IEEE Trans. Power Syst.,* vol. 34, nº 6, p. 4910–4920, Nov. 2019.

52. «Bayesian framework for multi-timescale state estimation in low-observable distribution systems,» *IEEE Trans. Power Syst.,* vol. 37, nº 6, p. 4340–4351, Nov. 2022.

53. «An enhanced IEEE 33 bus benchmark test system for distribution system studies,» *IEEE Trans. Power Syst.,* vol. 36, nº 3, p. 2565–2572, May 2021.

54. «A survey on state estimation techniques and challenges in smart distribution systems,» *IEEE Trans. Smart Grid,* vol. 10, nº 2, p. 2312–2322, Mar. 2019.

55. «A review on state estimation techniques in active distribution networks: Existing practices and their challenges,» *Sustainability,* vol. 14, nº 5, Feb. 2022.

56. M. S. J. A. E. N. L. Seyed-Ehsan Razavi. Ehsan Rahimi, «Impact of distributed generation on protection and voltage regulation of,» *ELSEVIER - Renewable and Sustainable Energy Reviews,* vol. 105, pp. 157-167, 2019.

6.2 Acronyms

Table 4. Acronyms

Acronym	Explanation
AMI	Advanced Metering Infrastructure
DER	Distributed Energy Resource
RTTR	Real-Time Thermal Rating
OPF	Optimal Power Flow
SCADA	Supervisory Control And Data Acquisition

DSSE	Distribution system state estimation
DSO	Distribution System Operator
TSO	Transmission system operator
PSSE	power system state estimation
ReLU	Rectified Linear Unit
RNN	Deep recurrent neural networks
PMU	Phasor measurement unit
LV	Low voltage
MV	Medium voltage
HV	High voltage
NN	Neural network
DNN	Deep neural network
ADN	Active distribution network