OPENTUNITY asset and planning developments (v1)

4 November 2024

opentunity

OPENTUNITYproject.eu

Deliverable details

Title	WP	Version
OPENTUNITY asset and planning 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
Dimitris Lagos	ICCS
Lucas Pons	ETRA I+D

Version	Date	Person	Action	Status***
0.1	06.08.2024	Dimitris Lagos	Table of Content	Draft
0.5	25.10.2024	Dimitris Lagos Lucas Pons	1st version of the deliverable for peer review	Draft
0.6	29.10.2024	Janez Gregor Golja, Edin Lakić, Jan Jeriha, Tomi Medved, Jurij Jurše, Georgia Eirini Lazaridou, Ilias Palaiologou	Peer review	Draft
1.0	31.10.2024	Dimitris Lagos	Version ready to submit	Final

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

Authors (organization)

Dimitris Lagos (ICCS); Lucas Pons (ETRA I+D); Fernando López Laso (ETRA I+D); Janez Gregor Golja (IRI UL), Edin Lakić (IRI UL), Jan Jeriha (UL), Tomi Medved (UL), Jurij Jurše (EPR); Georgia Eirini Lazaridou (HEDNO); Ilias Palaiologou (HEDNO)

Keywords

DSO, TSO, long-term asset management, short term asset management, non-technical losses detection, network planning optimization, investment deferral

Executive Summary

This deliverable was developed under WP5, which targets the design of software modules to provide innovations for grid operators. Specifically, it addresses innovations related to T5.4 “Advanced Asset Management” and T5.5 “Network Planning and Investment deferral from optimal use of flexibility”. This is the first version of the deliverable with the next and final version expected to be submitted in M30 of the project duration.

To bring this task into a successful completion, several innovations are being developed by the technical partners of the project. In this deliverable, the modules for short- and long-term asset management and non-technical losses detection are presented, which are linked to T5.4. In addition, the modules related to the grid planning tool that is developed under T5.5 are also presented.

This document centers on the technical details of the modules developed under Task 5.4, and Task 5.5. It includes a description of each module's design and implementation, along with initial mock-ups of the user interface. In the subsequent deliverable, D5.4 "OPENTUNITY Asset and Planning Developments (v2)," the focus will shift to showcasing the final functionalities and user interface of the modules, without revisiting the underlying design and implementation details.

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 5.4 "Advanced Asset Management" and Task 5.5 "Network Planning and Investment Deferral from optimal use of flexibility". Firstly, an overview of the different modules is provided and then, the detailed individual explanation is shown for every innovation developed.

However, the information included in this deliverable may change and could be enhanced later during the refinement phase and those improvements will be reported in the deliverable D5.4 "OPENTUNITY asset and planning 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 5.4 "Advanced Asset Management" and Task 5.5 "Network Planning and Investment Deferral from optimal use of flexibility". In this deliverable the description of the design and implementation of the modules will be provided, together preliminary mock-ups of the User Interface. D5.4 "OPENTUNITY asset and planning developments (v2)" will focus on showing the functionalities of the final version of the modules and their User Interface, without 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 and Transmission 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

Under the framework of T5.4 and T5.5 a range of modules have been developed for power system operators, focusing on advanced asset management maintenance and distribution network planning.

For advanced asset management three modules are being developed. The first one is the Long-Term Asset Management module which targets the end-of-life (EoL) prediction of equipment, specifically smart meters, by using data from Advanced Metering Infrastructures (AMI) and applying machine learning (ML) models. The Short-Term Asset Management module focuses on critical power system equipment like high-power transformers. Through real-time monitoring and data analysis, the module detects early fault indicators, such as gas levels and temperature anomalies, using common standard guidelines and machine learning models. Machine learning models are also applied in the third tool under Advanced Asset Management umbrella which is the non-technical losses detection. These machine learning tools are trained on historical patterns collected from smart meters that energy theft was detected to predict which of the operating smart meters might exhibit similar patterns and identified as candidate for inspection to find cases of non-technical losses.

Additionally, the Network Planning tool facilitates strategic investment and operational planning through efficient load-flow algorithms and time series clustering, offering insights on optimal distribution system upgrades with user defined goals like cost reduction, maximum RES (Renewable Energy Sources) integration, and investment deferral using flexibility. Below it is possible to find the summary of the purpose of each of the modules:

Long term asset management:

This module aims to provide end of life prediction on equipment of system operators. Unlike traditional maintenance, which focuses on grid core equipment, this module focuses on digital equipment, such as smart meters—a field with minimal existing research. Smart meters faults are influenced by environmental and operational factors like temperature, humidity, and power system fluctuations. Routine inspections help mitigate failures but are labour-intensive and costly due to the large scale of smart meter deployments.

To reduce manual inspections, the module integrates predictive tools, using data from sources like Advanced Metering Infrastructures (AMI) and failure logs. The module has two main functions: generating EoL curves to predict failures and applying a machine learning model to identify high-risk smart meters. EoL curves are calculated based on the age of both failed and operational meters, with different statistical models like Weibull and Normal distributions, which are commonly used for reliability analysis. The model then forecasts the number of failures for each smart meter model within a specified time frame.

A machine learning submodule, using XGBoost, analyses pre-fault data trends to detect critical meters likely to fail soon. It calculates failure probabilities, incorporating the brand/model-specific EoL probability.

Short term asset management:

Short Term Asset Management for critical power system equipment, i.e. high-power transformers, aims to prevent equipment failures by predicting potential critical issues over the next few hours. Given that transformers are vital yet vulnerable elements in power systems, any failure could lead to widespread outages and substantial repair costs. The tool, therefore, leverage real-time monitoring

and data analysis, using methods like Dissolved Gas Analysis (DGA) and oil temperature anomaly detection, to detect early signs of faults. When unexpected conditions—such as abnormal levels of gases (like H_2, CH_4, or C_2H_2) or temperature fluctuations—are identified, alarms can be triggered, helping operators assess the condition of the asset and maybe take proactive measures before the situation escalates.

In practice, the module operates by continuously analysing sensor data and applying machine learning models to perform anomaly detection. The system generates immediate alerts for transformer components like bushings, dissolved gas levels, and oil temperature, based on IEEE guidelines for fault thresholds. For future failure probabilities, the tool also uses real-time and forecasted data (such as weather conditions) to predict the likelihood of issues occurring within a six-hour timeframe. With its predictive and alert capabilities, short term asset management provides operators with two main interfaces: one for real-time alerts and another for future predictions, supporting proactive maintenance decisions and reducing the likelihood of costly, unplanned outages.

Non-technical losses detection:

Non-Technical Losses (NTL) in energy systems refer to electricity consumed but not billed, typically due to fraud or energy theft. ETRA's ETER software, employs a hybrid approach using Deep Neural Networks and Power Flow Analysis to identify fraudulent users and illegal connections effectively.

The preferred ML approach for NTL detection is data-oriented, particularly with neural networks like autoencoders and Convolutional Neural Networks (CNNs), which achieve high accuracy. CNNs analyse energy data grouped by week to detect periodic consumption patterns, with anomalous patterns often indicating fraud. For unregistered connections, network-oriented methods estimate thermal losses, apply power flow analysis, and compare estimated and computed losses to identify discrepancies. The implementation leverages Python with libraries such as Keras, PyTorch, and TensorFlow for ML and Panda Power for network analysis. The deployment utilizes Docker for consistent application environments, with InfluxDB and MongoDB handling data storage, supporting efficient and scalable fraud detection in energy networks.

Network Planning tool:

The distribution network planning module, currently in development, provides to DSOs tools for comprehensive planning. This module is designed to optimize investment and operation strategies efficiently using fast load-flow algorithms and time series clustering, which enables quick, adaptable responses to user-defined goals like cost reduction, investment deferral and RES maximization. The process leverages mathematical programming, including mixed-integer linear and second-order cone programming models, solved through the GUROBI optimizer. The system's interactive interface accepts topology data and demand curves to generate results rapidly, allowing scenario testing and enabling DSOs to make informed, strategic planning decisions.

4 MODULES' DEVELOPMENT

The modules developed consist of four different tools, Long Term Asset Management, Short Term Asset Management, Non-Technical losses detection and Distribution Network Planning, which are analysed in the following subsections.

4.1 Long Term Asset Management

The ambition of this module is to utilize the data coming from various sources (historical data, failure logs) and develop risk assessment methodologies (in the form of expected failure prediction and critical equipment detection) which can help the maintenance planning.

Usually, both system operators modules & research activities are focusing on the core equipment of the grids, namely condition monitoring of transformers etc., with very limited research concentrating on digital equipment, like smart meters. ETIP SNET [1] identified almost zero research activity in the research topic of Asset Management and Maintenance. To this end, the Long Asset Management module of OPENTUNITY will focus on smart meter equipment, aiming to contribute to the research in this field and providing new methodologies for DSO to perform asset management on smart meters.

Initially, it is important to discuss which are the failure modes for smart meters. Smart electricity meters are seriously affected by various factors, such as temperature, humidity, lightning events, power system fluctuations, and electromagnetic interference, and the failure risk increases with long-term operation. With the increased number of installed electricity meters, failures also grew gradually. Failure number prediction is necessary for decision-making on spare parts inventory [2].

Common faults of smart meter are battery fault, clock fault, display fault, burning fault, communication fault and so on. The details are shown in Table 1. Common breakdown for smart meter can be the abnormal display, inaccurate measurement and inter transformer burnout and so on.

Table 1. Smart Meter Fault types and causes

Fault type	Fault cause
Battery fault	Meter failure, battery under-voltage, Poor contact of battery connector
Clock Fault	Chip failure, battery under-voltage, Poor communication signal
Display fault	High temperature and high humidity
Burning fault	Improper installation of smart meter, overload, poor wiring contact
Communication fault	Wrong parameter setting, component virtual welding

Routine inspection of smart meters can reduce the number of failure occurrences. These inspections include accuracy validation of the power factor, frequency, voltage and current measurements, whether there is an indicator for voltage loss, current loss and so on. Next, it is also necessary to inspect whether there are any loose parts, strike fires, even burnout mark, etc.

Implementing such routine inspection measures requires significant personnel involvement due to the large-scale deployment of smart meters. With a vast number of meters in operation, the manual inspection of each device for issues like power factor abnormalities, voltage imbalances, or signs of physical damage could result in substantial operational costs for a DSO. This increased demand for human resources not only raises labor costs but also reduces efficiency, as frequent inspections across large geographical areas can be time-consuming. Therefore, DSOs need predictive tools to identify potential meter failures in advance. By utilizing advanced analytics and machine learning models, they can detect abnormal behavior in specific meters, allowing them to target inspections more effectively. This would help minimize unnecessary inspections and reduce operational expenses, while ensuring timely maintenance of at-risk meters.

A limited amount of research work has been focused on this. For example, in [3], a life prediction model of smart meters is proposed that uses a random forest and tested on the data of more than 150,000 meters provided by the Ningxia Metrology Center. [4] proposes a method for health analysis of electric energy meters based on the KNN (K-Nearest Neighbor) algorithm and structural equation model. Key indicators are selected and the evaluation system is initially constructed through the KNN algorithm. Then, the influence relationship and weight value between each key index are analyzed and determined by utilizing the structural equation model. Finally, the health of the smart meter is analyzed and predicted based on the weighted average method.

Overall, the application of machine learning methodologies relies on the available data. Our module aims to leverage existing data sources from DSOs, such as AMI and failure or installation logs, to extract valuable information on smart meter failure dates. By analyzing this data, the module assesses which meters exhibit critical behavior, helping to identify those at higher risk of failure. This enables more targeted maintenance and timely interventions.

4.1.1 Design

Figure 1 presents the process of the module responsible for performing long term asset management, i.e. predicting the end of life of equipment. In the figure the case of smart meters is presented but this can be extended to other assets of system operators too. The whole process takes as inputs historical data of equipment measurements and failure data.

The flow begins with the user providing historical data about smart meter (or other asset) installations based on which the model and the age of the smart meter are computed. This version of the tool requires, information on installation point ID, brand and model of smart meter, and secondary substation that the meter belongs to. This information is then used to update a repository that stores smart meter data and compute the age of operating assets and the age that failed assets have reached. The age of both functional and failed assets (e.g. smart meters) serves as an important input for the subsequent steps and it is mandatory for deploying any tool that performs long term asset management.

Using the age of healthy and failed assets, per brand and model of equipment, EoL curves for the smart meters are computed using statistical models, such as Weibull or normal distribution. The purpose of generating these curves is to have an estimate of expected failures within the total population of operating smart meters within a specific brand and model of a smart meter.

When the EoL curves are fit, the user can select a value that represents the months in the future horizon where the prediction of the number of failures per equipment type is performed. The module

will generate a list of expected number of failures that will occur per smart meter model within the chosen time horizon.

The process continues by accessing a repository of smart meter measurement data, which is then used to train a machine learning model—specifically, an XGBoost model. The raw measured data that could be provided are active/reactive power consumption/generation at hourly resolution and voltage and current measurements if they are available. The average values of these measurements can be used or additionally maximum and minimum values collected by the smart meters for every hour in these respective measurements can be also used, The model internally preprocess the data and finds out the importance of the features.

This model is trained to detect critical smart meters that are at high risk of failure. Finally, the trained model is used to predict which smart meters are likely to fail based on the latest data, allowing for proactive inspection, maintenance and replacement planning.

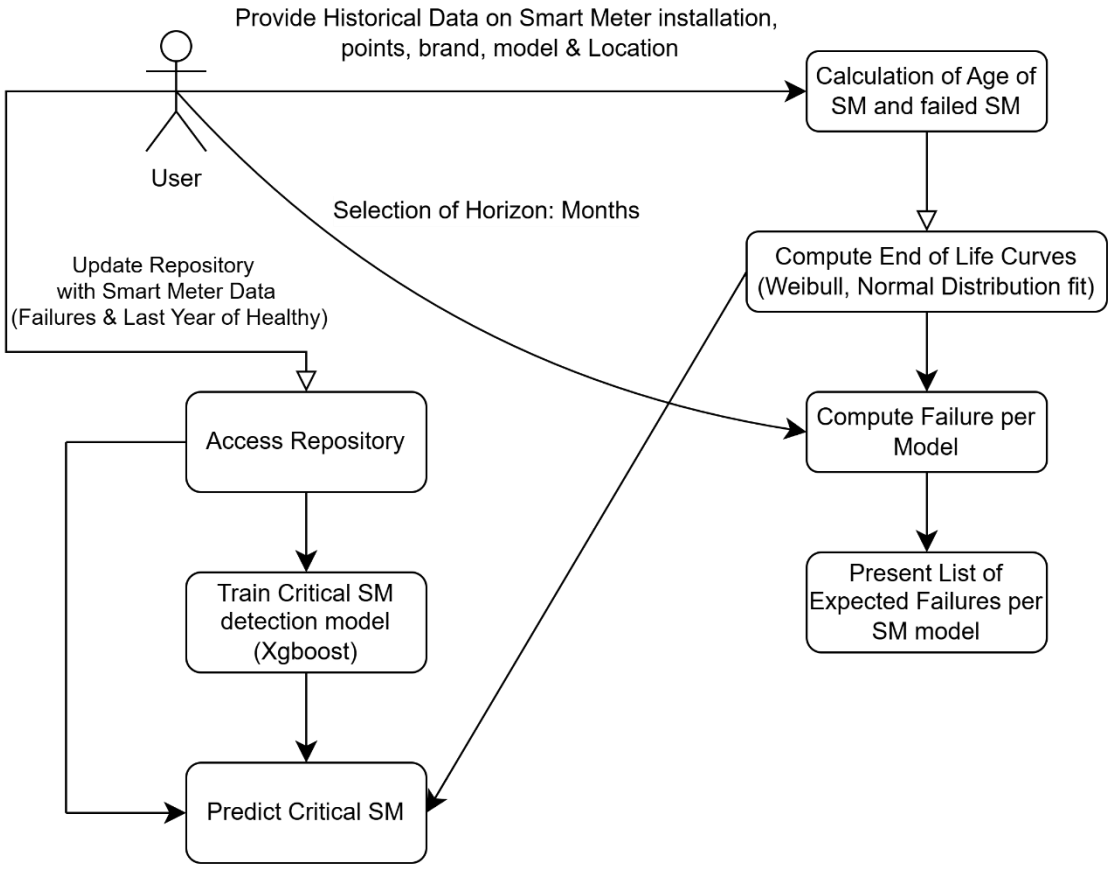

Figure 1: Logic Flow Diagram of Smart Meter Long Term Asset Management Module

4.1.2 Implementation

As mentioned in the previous section, the long-term asset management for smart meters mainly contains two submodules. The calculation of EoL curves and predictions of failures based on them and the machine learning model for the identification of critical smart meters.

The EoL curves are calculated using the age of healthy and failed assets, per brand and model of equipment, based on the data provided by the user. The EoL curves for smart meters are computed

using statistical models, such as Weibull or normal distribution, which are commonly used for reliability analysis.

To produce the CDF (cumulative distribution function) of these curves their parameters (i.e.. the mean and standard deviation of normal distribution, scale and shape for Weibull) have to be calculated. The curve fit is performed by solving a non-linear least squares problem.

The input data are used to calculate the percentages of the population of a specific model that have failed after their operating life has reached a specific value. Thus, distinctive points on a CDF are generated with these two attributes, the operating time (in months - (t_i)) and the percentage of the total population that by that month have failed (FR_i). The mathematic formulation presented in (1) is used to calculate the parameters of the probability distribution curves, using the SciPy library in python [5]. The Weibull distribution is selected if less than 20% of the smart meters have failed otherwise the normal distribution is selected. If less than 1% of the population of smart meters in a specific brand / model have failed, then no EoL curve is calculated.

$$\max \sum_{i \epsilon N} (cdf(t_i) - FR_i)^2$$
$$\text{s.t.} \quad cdf(t_i) = cdf(parameters, ti) \; \forall \epsilon N \tag{1}$$

When the EoL curves are fit, the user can select a value that represents the months in the future horizon where the prediction of the number of failures per equipment type is performed. The module will system a list of expected failures number for each smart meter model within the chosen time horizon.

To generate population failure estimations per smart meter model the user has to define the prediction horizon in months. Based on that the age of the smart meters at the end of this horizon is calculated. The increase in Total Failure probability difference between the time that was used as reference for the computation of EoL curves and the probability that occurs in the horizon defined by the user is multiplied by the total population number to generate the expected failures in this period.

The result of this computation is organized using pandas library [6], into a pandas DataFrame where each row corresponds to a smart meter model and the number of failures expected for that model. The DataFrame is then sorted in descending order, so models with the highest expected failures are listed at the top. After sorting, the function filters out models with no expected failures, returning a list of models that are predicted to have at least one failure in the selected time horizon. This filtered and sorted DataFrame is what is finally presented to the user.

The second submodule is the critical meter identification. For the training of this module the user has to upload (in a repository) historical data of electrical measurement of smart meters. The failure dates of smart meters are then used to collect historical data of pre-fault trends in the measurements of failed smart meters (e.g. 1 month). In addition, historical data of healthy smart meters are also collected.

A preprocessing procedure then takes place to generate important metrics on these data like maximum, average and standard deviation of active and reactive power consumption, accumulated active power & reactive power consumption, occurrences of communication failures, etc, which are the attributes of the training data. The Flag of the training data dictate if the meter is healthy or has failed.

These training data is then used to train a classification model (XGboost - eXtreme Gradient Boosting), with the library in [7], which serves the purpose of identifying critical smart meters that could fail soon, using the latest measurements available of these smart meters. XGboost is a more advanced method compared to both Decision Trees and Random Forests. It belongs to the gradient boosting algorithms, which build models in a sequential manner, compared to Random Forests for example, which build trees independently. With every new tree XGBoost corrects the errors made by the previous one. This boosting approach reduces errors incrementally, leading to more accurate models. XGBoost also implements several optimizations, such as regularization, handling of missing data, and efficient use of memory, which make it faster and more powerful compared to Random Forests and traditional gradient boosting algorithms.

To avoid overfitting and to perform feature selection, XGBoost classifiers are built iteratively. After the training the built trees are collected and the feature importance is calculated. Features with importance less than 5% are dropped and the rest are kept to re-train the model until all the remaining features have importance over 5%. Finally, the trained model is used to predict which assets are at high risk, allowing for proactive maintenance or replacement strategies.

XGBoost calculates the probability of each class by combining the outputs of all its decision trees. These trees output "scores" instead of class labels directly. For classification tasks, XGBoost needs to convert these raw scores (called logits) into probabilities. For binary classification, which is the aforementioned case, the sigmoid function is used to turn the score into a probability between 0 and 1. The sigmoid formula ensures that the sum of probabilities across two classes equals 1. The score is fed into this function to generate a probability for failure.

This probability of failure that is presented to the user uses both the XGBoost and EoL curves. Initially, the last month data are collected for the smart meter and the criticality probability (XGBoost model) is computed. Then, if this smart meter has an EoL probability greater than 10% based on is brand/model EoL then the criticality score is multiplied by factor 1, otherwise by 0.5. This is performed to take into consideration the brand/model characteristics. Since similar behavior between smart meters of different models might not necessarily yield failure in both of them. The smart meters with criticality score (%) over 70% is presented then to the user.

4.1.3 Mock-Ups

The preliminary mock-up of the GUI for the long-term asset management tool is presented in this section. It consists of four tabs. The “Historical Data Input”, the “End of Life Curves”, the “Critical Meters Calculation” and the “Map Representation” tab.

Figure 2 shows the tab “Historical Data Input”. In this tab, the user (DSO) has to upload historical data about the smart meters in a “.csv” format. This file should include a row per smart meter that include information about the metering point, the secondary substation that it belongs to, the installation date, the brand and the model, etc. Finding common metering points between smart meters the replacement of a smart meter by a new one is found and used to flag smart meters as failed. The page includes a file upload feature where users can upload a “.csv” file containing this information

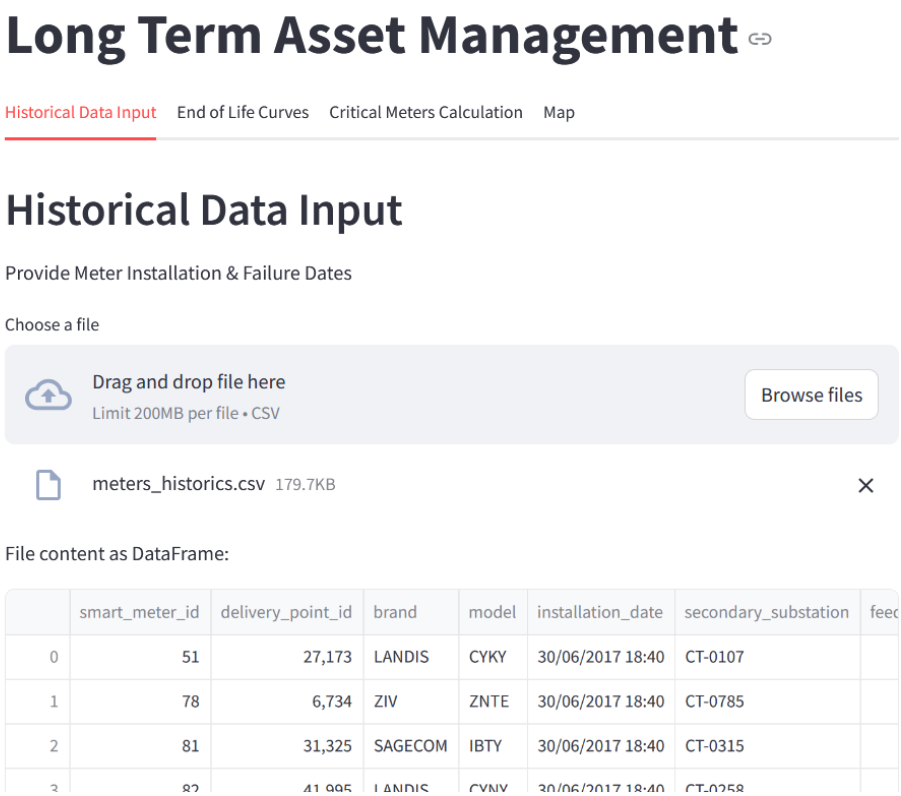

Figure 2: Historical Data Input on Smart Meter Failure tab

Figure 3 presents the "End of Life Curves" tab. Using the historical data provided in the previous tab, the EoL curves are calculated according to the procedure described in the previous section. Using the drop-down options the user can select the specific brand and model of a smart meter and see its EoL curve. If not enough failures have occurred, i.e. less than 1% of the total population, in a specific brand/model then the message "Limited Failure Numbers recorded" is printed in the interface. If adequate number of failures have occurred to calculate a trustworthy EoL curve, then a EoL curve will be generated and presented as depicted in Figure 4. Finally, in this tab the user can select the future horizon (in months) and through the EoL curves the expected failures per smart meter model are computed. The models in which failures are expected are then presented via the GUI (Figure 4).

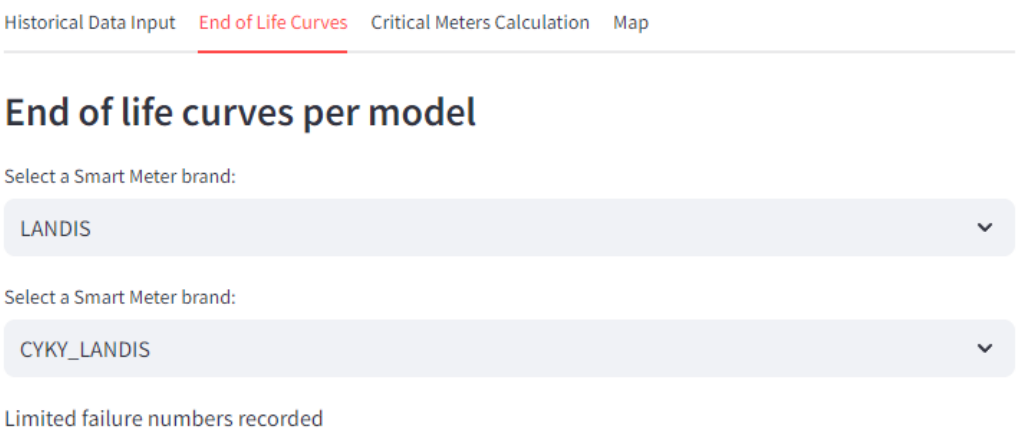

Figure 3: End of Life Curves tab

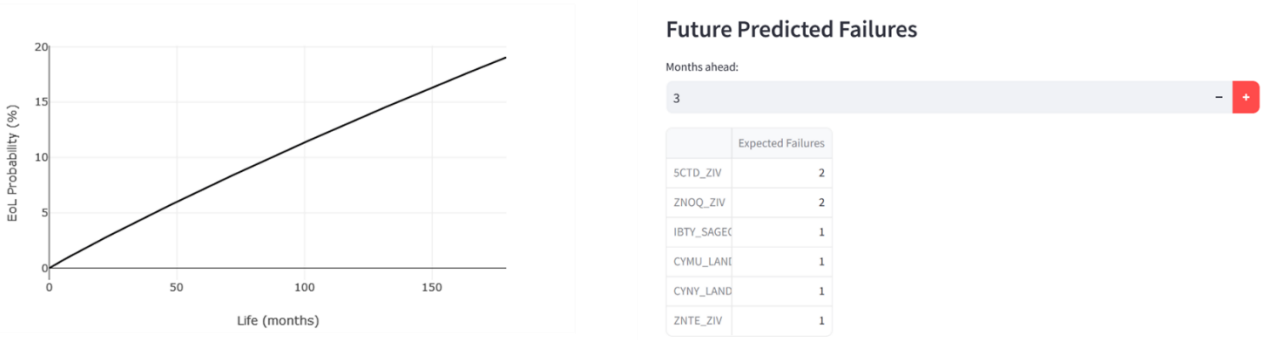

Figure 4: End of Life Curves tab (EoL curve presentation & future predicted failures)

In the "Critical Meters Calculation" tab (Figure 5), the user has to provide the information of the repository in which the smart meter measurement data are collected. With this data and the failure data provided by the user in the first tab, the XGBoost classifier is trained according to the procedure that is described in the previous section. The critical smart meters are calculated and presented in a list to the user. In addition, the importance of the different attributes in the XGboost classifier is presented to make the tool more interpretable for the system operator.

Critical Smart Meters

point	Failure Probability	model	Substation
24,193	86.6706	ZOXV_ZIV	CT-0593
49,730	75.3027	ZOXV_ZIV	CT-0593
11,809	75.3027	ZOXV_ZIV	CT-0593

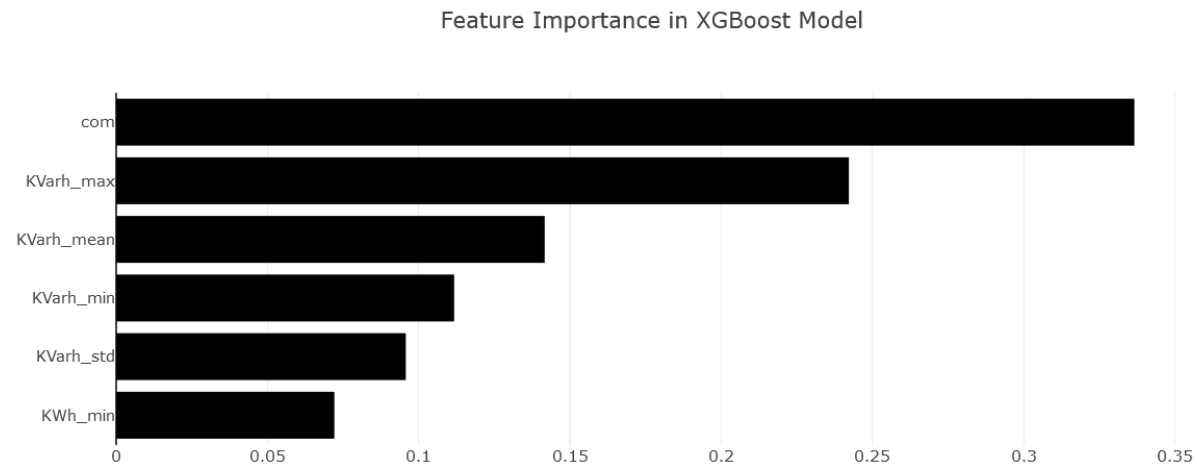

Figure 5: Critical Meters Calculation tab

Finally, in the "Map" tab (Figure 6), the user can upload a ".csv" file containing the geodata (latitude, longitude) of the secondary substations and get a map representation of the substations. If critical smart meters belong to a specific secondary substation then the pin of the substation in the map turns red. By clicking in the pin, the user can see the number of critical smart meters in that secondary substation.

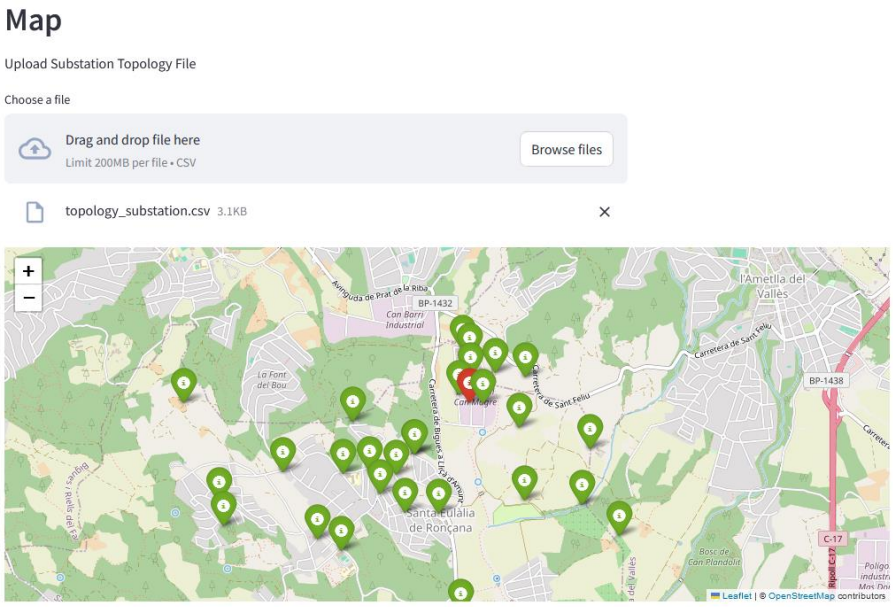

Figure 6: Map visualization tab

4.1.3.1 Preliminary Results

The tool has been tested initially with data from the Spanish demo for almost 2900 smart meters, with 2531 healthy meters and almost 370 failed smart meters. The installations of smart meters in the dataset start at June of 2016 up to May of 2024. The status of the smart meters up to the end of 2023 was used to have a dataset and check with the failed smart meters of 2024 the accuracy of the model. The actual smart meters that failed between January and May 2024 are presented in Table 2.

Table 2. Smart Meter Failures at the first five months of 2024

Meter	model	Failure Date
1	IBTY_SAGECOM	2024-01-02T13:34:00.000
2	IBTY_SAGECOM	2024-01-16T09:36:00.000
3	ZNOQ_ZIV	2024-01-16T16:43:00.000
4	ZNOQ_ZIV	2024-01-23T15:49:00.000
5	IBTY_SAGECOM	2024-02-21T16:39:00.000
6	ZNOQ_ZIV	2024-02-21T16:43:00.000
7	ZNOQ_ZIV	2024-03-05T11:26:00.000
8	ZOXV_ZIV	2024-04-03T11:31:00.000
9	ZNPQ_ZIV	2024-04-16T15:23:00.000
10	ZNOQ_ZIV	2024-04-23T09:19:00.000
11	CYNY_LANDIS	2024-05-01T13:14:00.000
12	IBTY_SAGECOM	2024-05-04T12:50:00.000

Based on the EoL curves with data up to the end of 2023, the calculated total number of smart meter failures for the next five months was 17 compared to the actual 12 failures. For specific models it identified a total number of 2 failures for IBTY_SAGECOM (4 occurred), 3 for ZNOQ_ZIV (6 occurred) and correctly 1 for CYNY_LANDIS and ZOXV_ZIV. A total number of 10 smart meters was identified as critical. Two of the failed smart meters (highlighted with red in Table 2) was identified among the 5 most critical smart meters by the data driven classifier.

4.2 Short Term Asset Management

Short Term Asset Management purpose is the estimation in the operating time scale (i.e. for the next hours) if there is critical equipment that might experience failure, in order to mitigate the consequences of the failure. To this end, such functionality is more important for critical power system assets. The equipment that is currently considered for short term asset management in OPENTUNITY are HV/MV transformers for DSOs and UHV/HV transformers for the Greek TSO.

High-power electrical equipment plays a vital role in the operation of power systems, making fault detection and prediction techniques critically important. Power transformers, as key elements in power transmission networks, are particularly essential. The failure of a transformer can lead to a chain of cascading failures, resulting in widespread power outages across the system. In addition, power transformers are costly assets within power systems. They are vulnerable to catastrophic failure and irreversible internal damage due to various electrical, mechanical, or thermal stresses.

Transformer failures can be categorized as electrical, mechanical, environmental, or thermal. Studies conducted by CIGRE on power transformer failures show that 41% of the failures occur with tap changers, 19% on windings, 13% due to leakage, 12% for faults in bushings, 3% for faults in core and 12% for other reasons [8].

For most utilities, a strict maintenance schedule is followed, in which maintenance activities are performed at specific times or when an asset fails. This practice can lead both to improper maintenance, by not performing maintenance on equipment that is in need, or to unnecessary maintenance to healthy equipment that leads to an unnecessary increase in maintenance costs. Tools that perform short-term asset management focus on preserving the operational functionality of equipment. To achieve this, various types of sensors and information systems are utilized to gather data and through the use of machine learning techniques, detect quickly or predict abnormal conditions, which then finally can lead to predictive maintenance actions.

Most diagnostic techniques are based on the analysis of the gases dissolved in the oil of the transformer, and through them identify the faults that have occurred. When abnormal situations occur within a transformer flammable gases such as $H2$, CO and hydrocarbons such as $CH4$, $C2H2$, $C2H4$, $C2H6$, are produced. By analyzing the gases produced during this reaction, it is possible to determine whether a transformer is working properly and, if not, what type of fault it is. The IEEE Standard C57.104 [9] presents the different analysis (duval triangle, roger ratios, etc.) that are used to categorize the faults in transformers and a procedure to detect if the transformer is in normal condition or not.

Most of the intelligent fault diagnosis methods in power transformers are based on a variety of data-driven machine-learning technologies and DGA data. In [10] techniques that use Multilayer Artificial

Neural Network (ANN), Support Vector Machines (SVM) and genetic algorithms are mentioned for classification tasks on transformer fault diagnosis.

However, condition monitoring is often applied towards the end of life when critical faults are expected in transformers and there are already sufficient faults or abnormal behavior data by the equipment. By that stage, years of operation of the asset have led to a particular trend presenting the normal behavior for one unit, which could be considered faulty behavior if it suddenly manifests in another. In practice, recognizing changes in behavior of the equipment can be more indicative of faults developing in assets than absolute values of measurements. Thus, detecting the operating condition of new equipment that has no records of faults yet can only be made via anomaly detection techniques.

Anomaly detection techniques are a way of recognizing changes in the equipment's behavior. Rather than simply matching patterns of expected faults, a model of behavior specific to each transformer under study can be trained to represent the normal operation of that particular asset. New measurements can be compared against the model to quantify how likely or anomalous they are. This allows for the natural differences between normal behavior in different transformers, and low-level fault behavior can be trained into the model as normal for that unit. The combination of anomaly detection can enhance the diagnosis of faults, by reserving fault classification for situations where an anomaly is detected, thus reducing the volume of data engineers must examine [11].

4.2.1 Design

A data driven condition assessment tool is based on the available data. Thus, the tool submodules will be implemented in each pilot only if there are specific data available. At the moment, the tool supports Dissolved Gas Analysis, Oil Temperature Anomaly detection & prediction, bushings insulation assessment and failure probability estimation based on weather data.

For DGA analysis, oil temperature prediction & anomaly detection and bushings insulation assessment, data from on-line monitoring systems are required, which provide various daily measurements of the data relevant to each submodule. as presented in the implementation subsection.

For the failure probability estimation based on weather data. dates that failures have occurred due to extreme weather conditions are provided as well as the location of the equipment and the collected electrical data available (active/reactive power, current, voltage) with hourly resolution for at least 24 hours prior to the fault. In addition, the same parameters should be provided for the last year with hourly resolution.

If none of the aforementioned approaches can be implemented in a pilot due to absence of relevant data sources, or failure data a prediction of oil temperature will be made according to the characteristics of the equipment and relevant thermal models to predict the oil temperature for the next hours as an indicator of transformer condition.

The designed short term asset management module performs two tasks, a) use the latest data to raise alarms for specific transformer' components (according to the available measurements) and b) predict possible critical conditions in the future and link them to a failure probability.

Figure 7 presents a system designed for short term condition assessment, specifically focusing on an On Line Monitoring System (OLMS). The process begins with the user providing historical data from the OLMS, which includes measurements. Apart from these data dates of failures can be also added.

This data is collected either in ".csv" format or via a repository. The historical data is then used to train predictive models using machine learning. Due to the absence of failure in the historical data mainly regression models for transformer oil temperature levels will be built to estimate whether abnormal conditions have occurred and predict the evolution of the temperature. If failure data exists, e.g. due to weather conditions, additional classification models will be built to assess short term probability of failure. These models are stored in a database for future use.

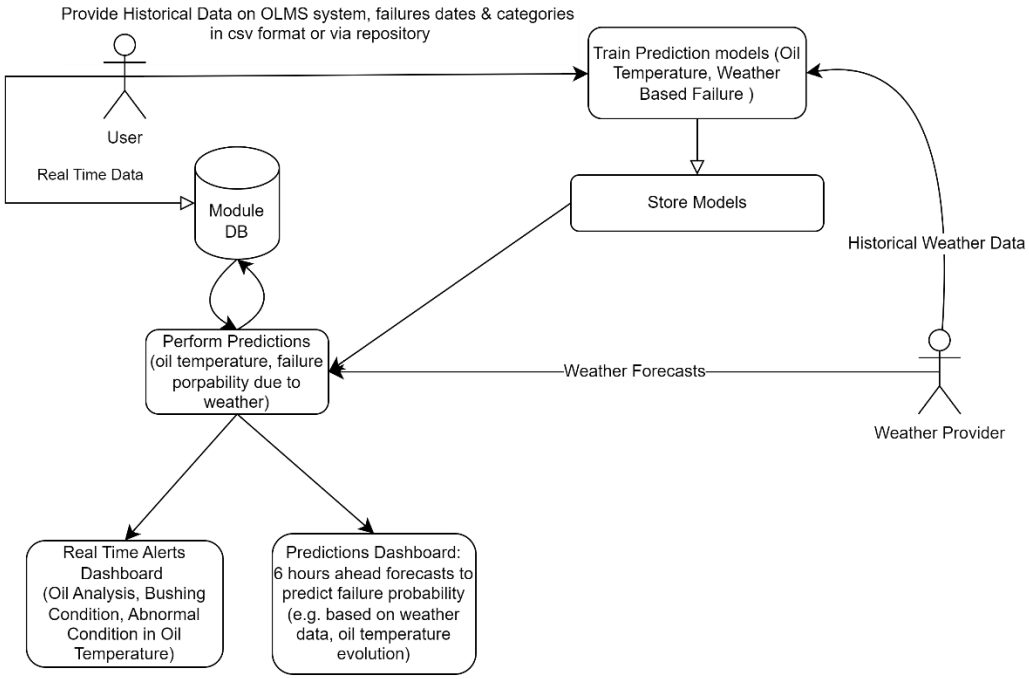

Figure 7: Logic Diagram of Short-Term Asset Management

Using the trained models, real-time measurements and weather data, predictions are made regarding the probability of failure due to factors like oil temperature or external weather conditions. Apart from the predictions the real-time measurements will be used to generate alarms, e.g. according to gas levels on transformer oil or for bushings insulation, according to the available measurements.

The results will be mainly presented in two different dashboards. The first is a real-time alerts dashboard that monitors e.g. oil analysis based on anomaly detection, bushing conditions, and any abnormalities in gas analysis, providing immediate alerts when necessary. The second dashboard focuses on predictions, offering six-hour forecasts that estimate the likelihood of failure based on weather data and the evolution of oil temperature. These predictions help operators take preventive actions ahead of potential issues. The data from the monitoring process is stored in a module database, ensuring that the system can continually update and refine its predictions.

4.2.2 Implementation

The tool developed serves two functionalities: raise alarms if abnormal measurements are collected and predict the behavior of critical values in the short-term horizon, like the top oil temperature.

4.2.2.1 Alarms

In this section the developed rational for alarms on the real-time condition assessment are presented.

4.2.2.1.1. Bushings

The condition of the insulation is essential for secure and reliable operation of your transformer. Measuring capacitance and dissipation (or power factor or tand) help you determine insulation condition in bushings or between windings.

Changes in capacitance can, for example, indicate mechanical displacements of windings or partial breakdown in bushings. Aging and degradation of the insulation, coupled with the ingress of water, increase the amount of energy that is converted to heat in the insulation. The rate of these losses is measured as dissipation factor [12].

To assess the condition of bushings the real-time measurements of the last 24 hours are used. There are two mechanisms that perform the condition assessment on bushings. The first one is based on the IEEE C7.19.01-2017 [13],and the limits are presented in Figure 8. The mean of the measurement of the last day is used for the limit in power factor (tand) limit. The change is computed from the mean of the last week excluding the last day and the mean of the last day.

IEEE Std C57.19.01-2017
IEEE Standard for Performance Characteristics and Dimensions for Power Transformer and Reactor Bushings

Table 6—C1 or C power factor and capacitance limits for new bushings

Type of construction	C1 or C power factor and capacitance		
	Power factor[a]		Capacitance
	Limit (%)	Acceptable change [b]	Acceptable change (%)[c]
Col. 1	Col. 2	Col. 3	Col. 4
Oil-impregnated, paper-insulated	0.50	+0.02/–0.04	±1.0
Resin-impregnated, paper-insulated	0.85	±0.04	±1.0
Resin-bonded, paper-insulated	2.00	±0.08	±1.0
Cast insulation	1.00	±0.04	±1.0
Solid	N/A[d]	—	—

Figure 8: IEEE limits on power factor (tand) and capacitance

The second mechanism collects the measurements of the bushings of the same voltage level of the equipment and checks if any of them have considerable different measurements than the other two. Again, the values of the last day are used to compute the mean per bushing. If any of the measurement (tand, capacitance) has an absolute error higher than 3% with both the other bushings of the same voltage level then an alarm is issued.

4.2.2.1.2. Dissolved Gas Analysis

Dissolved gas analysis (DGA) is used to diagnose power transformer faults based on the concentration of dissolved gases. The IEEE C57.104-2019 - IEEE Guide for the Interpretation of Gases Generated in Mineral Oil-Immersed Transformers [9] provides some guidelines for assessment of the condition of the transformer based on the concentration of gases in the oil. For example, the tables depicted presented in Figure 9 and Figure 10 present the limits that flag the condition as normal or critical if these levels are respected or passed. Additional tables are also for the deviation of the gases

concentration between successive samples and the rate in ppm/year. This procedure is followed to assess the condition and raises an alarm if the thresholds are passed.

Table 1—90th percentile gas concentrations as a function of O_2/N_2 ratio and age in µL/L (ppm)

		O_2/N_2 Ratio ≤ 0.2				O_2/N_2 Ratio > 0.2			
		Transformer Age in Years				Transformer Age in Years			
		Unknown	1 – 9	10 – 30	>30	Unknown	1 – 9	10 – 30	>30
Gas	Hydrogen (H_2)	80	75		100	40	40		
	Methane (CH_4)	90	45	90	110	20	20		
	Ethane (C_2H_6)	90	30	90	150	15	15		
	Ethylene (C_2H_4)	50	20	50	90	50	25	60	
	Acetylene (C_2H_2)	1	1			2	2		
	Carbon monoxide (CO)	900	900			500	500		
	Carbon dioxide (CO_2)	9000	5000	10000		5000	3500	5500	
NOTE—During the data analysis, it was determined that voltage class, MVA, and volume of mineral oil in the unit did not contribute in significant way to the determination of values provided in Table 1.									

Figure 9: IEEE limits for normal Gas Levels

Table 2—95th percentile gas concentrations as a function of O_2/N_2 and age in µL/L (ppm)

		O_2/N_2 Ratio ≤ 0.2				O_2/N_2 Ratio > 0.2			
		Transformer Age in Years				Transformer Age in Years			
		Unknown	1 – 9	10 – 30	>30	Unknown	1 – 9	10 – 30	>30
Gas	Hydrogen (H_2)	200	200			90	90		
	Methane (CH_4)	150	100	150	200	50	60		30
	Ethane (C_2H_6)	175	70	175	250	40	30	40	
	Ethylene (C_2H_4)	100	40	95	175	100	80	125	
	Acetylene (C_2H_2)	2	2		4	7	7		
	Carbon monoxide (CO)	1100	1100			600	600		
	Carbon dioxide (CO_2)	12500	7000	14000		7000	5000	8000	
NOTE—During the data analysis, it was determined that voltage class, MVA, and volume of mineral oil in the unit did not contribute in significant way to the determination of values provided in Table 2									

Figure 10: IEEE limits for faulty Gas Levels

The deviations are computed by calculating the daily average and comparing it with the daily average of the measurements collected seven days (a week) ago. The rate of increase (ppm/year) is calculated with samples of daily averages a month apart. For the limits the daily averages are used to compare them.

To generate a general score for the DGA and assess its overall condition, the approach recommended by OFGEM is followed [14]. The approach sets a score for each gas in the dataset (C2H2, C2H6, C2H4, CH4, and H2). Next, a total score is calculated based on the weighted sum of these gas condition states, where H2 is weighted at 50, CH4, C2H4, and C2H6 each have a weight of 30, and C2H2 has the highest weight of 120. This formula emphasizes the importance of certain gases in assessing transformer health, with acetylene (C2H2) carrying the greatest significance in the

scoring. The score is normalized with a division of 120 and if the score is less than three the condition is considered normal, otherwise an alarm is triggered.

4.2.2.1.1. Oil Temperature Anomaly detection

The purpose of this model is to detect whether an abnormal condition occurs in the oil of the transformer by checking whether the oil temperature experiences expected temperatures, based on the ambient conditions and the loading of the equipment. To do that, a machine learning model is built to predict the oil temperature for the next hour and compare it with the actual measurement that is received at that hour.

In order to implement this, the machine learning model have to be built on normal operating conditions of the transformer. Thus, a data cleaning process is performed to keep only the data on timestamps that normal operation is detected. The data cleaning uses the processes described for DGA and bushings conditions to flag an operating procedure as normal or abnormal. Thus, if in a period a DGA or bushing alarm would have raised, the data of this period are removed from the training set. If failure dates are also available, they could have been used to remove also the data of that period from the training set. Finally, only the relevant data to the oil temperature are kept to form the training set of the oil temperature predictor.

The oil temperature predictor is based on a Deep Neural Network which is built in python using the tensorflow library [15]. The training data contain the oil temperature measurement of the previous hour (mean value of the measurements), the current of the previous hour, the ambient temperature of previous hour. The targets are the top oil temperature of the next hour. When, the model is trained the mean error and the standard deviation are calculated for the training set.

To evaluate the deviations between each instance of oil temperature reading (measured sensor value and predicted values by the model), we used statistical process control to identify anomalies. The Shewhart control chart is used to evaluate the deviations (errors) as they evolve. The fault threshold is defined by two control limits used to evaluate abnormal behaviors: Upper Control Limit (UCL) Lower Control Limit (LCL). The two control limits describe the sensitivity of the control chart, which is expressed as multiples of the standard deviation of the error distribution. The following equations (2) -(4) describe the control input MR_i definition and the Upper Control Limit (UCL) and Lower Control Limit (LCL). The $\overline{MR}$ and σ are computed on the training data.

$$\boldsymbol{MR_i = |error_i - error_{i-1}|} \quad (2)$$

$$\mathbf{UCL} = \boldsymbol{\overline{MR} + 3\sigma} \quad (3)$$

$$\mathbf{UCL} = \boldsymbol{\overline{MR} - 3\sigma} \quad (4)$$

Anomaly is defined by data points beyond the fault threshold/control limits, for at least four consecutive hours. In the Figure 11the anomaly detection prediction graph is presented, with the MR presented for the last 24 hours with the thresholds.

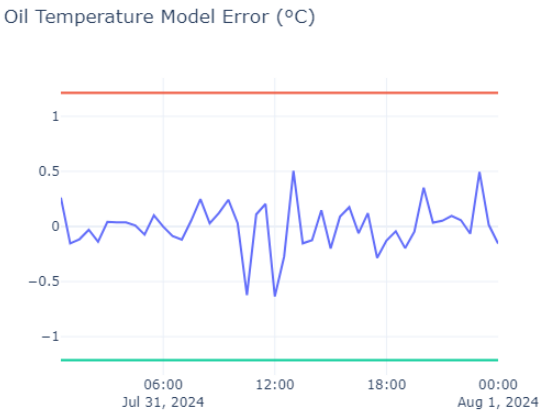

Figure 11: Oil Temperature Anomaly detection graph

4.2.2.2 Predictions

The future predictions for transformer are based on the prediction of transformer of the oil temperature. In the next version of the deliverable an additional module will be described that will be built on a classifier for failures that are based on weather conditions, if the system operators have adequate number of failures that are attributed to extreme weather conditions. This model will have as training set the weather conditions that occurred prior and during the fault, as well as the equipment loading, to build a classification model that estimates the probability of failure according to the weather conditions. In real-time operation weather forecasts will be used to estimate whether a risk of failure is imminent or not.

The developed fault prediction model, at this point, is based on the oil temperature prediction of the next six hours. The probability of exceeding the maximum oil temperature is used as an indicator of failure probability.

A probabilistic current prediction model is built using Random Forests as regressor, to get the probability distribution of current for the next six hours. To train the model that performs the probabilistic current the loading uses training data that contain the current values of the last 6 h (hourly average) and the month, day of the week and hour of the prediction time. As targets are the actual current measurements of that period.

The oil temperature model is used to predict the oil temperature for the next hour. The oil temperature of the next hour, the ambient temperature as received from weather forecasts and the estimated current for the next hour (mean, and 1% and 99% quantiles) are used as inputs in the oil prediction model to generate the oil temperature probability two hours ahead and so on.

Thus, the oil prediction is calculated as presented in the Figure 12. The limit in temperature (light blue line) is presented alongside the mean predicted temperature (red line), the maximum predicted oil temperature (blue line) and minimum predicted oil temperature (green line).

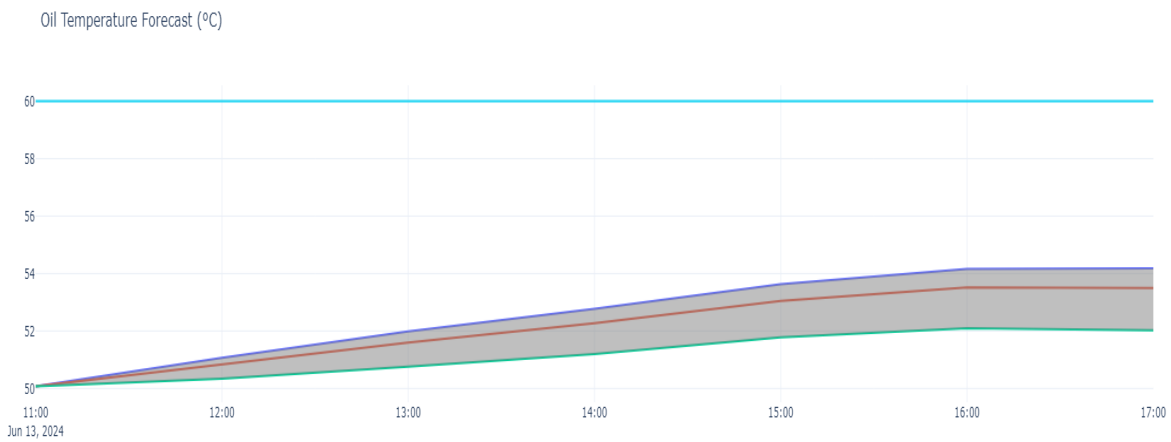

Figure 12: Oil Temperature prediction

4.2.3 Mock-Ups

Overall, the tool contains three different tabs, Historical Data Input, Alarms, and Predictions. Figure 13 shows the GUI for the Short-Term Asset Management module and specifically the Historical Data Input tab. This interface allows users to easily upload and preview historical data, which will likely be used in subsequent processes such as generating alarms or making predictive forecasts in other sections of the asset management system.

In this tab, the user initially uploads a ".csv" file containing the historical OLMS data. Users can either drag and drop a file or browse to select a file from their system. In the current state of the interface, a file named "QTMS_Data-2024-09-30...ATF8.csv" has already been uploaded, with OLMS data from a UHV/HV transformer in the Greek demo, with a file size of 51.5MB. Below the file name, there is a preview of the data from the uploaded file displayed in a table format.

The file contains various columns, including a measurement ID, a timestamp indicating when the data was recorded, the type of measurement taken (such as oil condition parameters like "TM8 O2 in Oil" and "TM8 C24 H in Oil"), the value of the measurement, and the units in which the data was recorded.

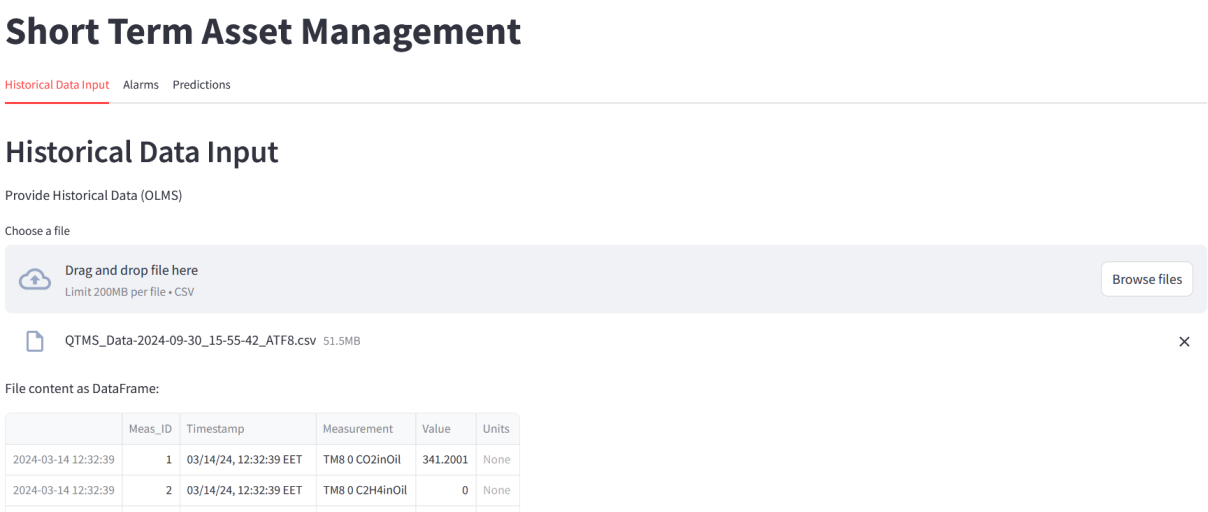

Figure 13: Histrorical Data Input dashboard

Figure 14 displays the Real Time Alarms tab. The interface has three primary categories of monitoring: Bushings, Dissolved Gas Analysis, and Oil Anomaly Detection. This interface is designed to provide operators with a real-time overview of critical asset conditions, focusing on the health of bushings, dissolved gases, and oil temperature. Each section uses both visual indicators and numerical data to show the current operational status, ensuring that any potential issues are flagged in real time.

Under the Bushings section, the system reports that there are no warnings related to the bushings, and the status is shown as "ok." The light would be red if a warning has been issued according to the rational presented in the previous sections for bushing condition. The table below further details the condition of all the different bushing components labeled as Cap H1, Cap H2, Cap Y1, and so on. Each of these components is marked with an "OK" status, indicating that no issues have been detected. This table is useful to see specifically which bushing has abnormal behavior in case of an alarm issued in bushings.

In the Dissolved Gas Analysis (DGA) section, the system again shows a "no action suggested" status, meaning that the dissolved gas levels do not indicate any need for maintenance or concern. Similarly, to the bushings the light would be red if an alarm was issued according to the methodology described in the previous section. The table below the alarm presents the latest data (mean values of last 24 hours measurements) on different gases like H2, CH4, C2H2, and others, with corresponding values for each. For instance, H2 shows a value of 3.416, while CH4 is at 5.4865. The overall score is marked as "ok," reinforcing that there are no abnormalities in the gas levels.

The third section, Oil Anomaly Detection, shows that no anomalies have been detected in the oil temperature and is based on the oil temperature anomaly detection submodule presented in the previous section. The system provides a simple status indicator, which confirms that oil temperature readings are within acceptable limits. A graph is also presented in this section that shows the Oil Temperature Model Error over the past 24 hours, with the temperature difference (in degrees Celsius) plotted against time. The blue line indicates the model's output, which fluctuates but remains within an acceptable range, as defined by horizontal boundaries on the graph.

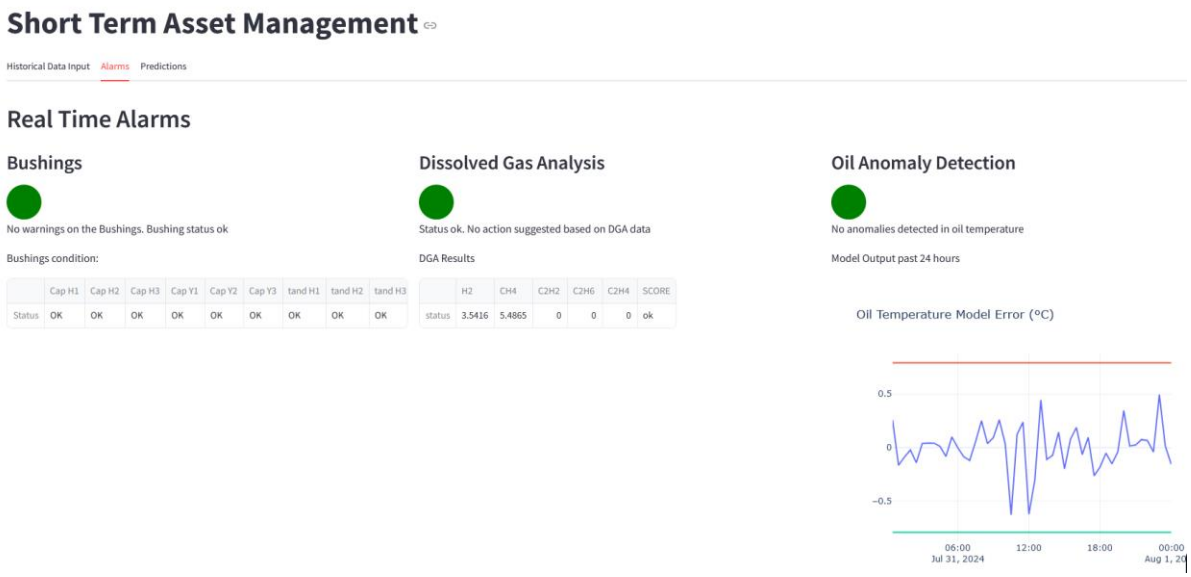

Figure 14: Alarms Dashboard

Figure 15 shows the Predictions tab of the Short-Term Asset Management GUI. At this moment, it presents the Oil Temperature Prediction but it might be updated for the next version of the deliverable.

At the top of the page, the heading "Oil Temperature Prediction" is displayed, followed by a smaller description noting that the system is assessing the failure probability based on predicted oil temperatures. This probability is calculated based on the probability of oil temperature surpassing the temperature limit.

After the probability table, a graph presents the Oil Temperature Forecast (°C) over time, starting from around 11:00 on June 13, 2024 and extending through 17:00 on the same day. The graph features a curved line representing the forecasted oil temperature. The shaded area around the line suggests a confidence interval, showing the possible range within which the actual temperature may fall. The temperatures gradually rise from around 50°C to a peak of nearly 60°C by the end of the forecast period. The graph provides a clear visualization of how the oil temperature is expected to evolve over the next few hours, allowing operators to monitor and take preemptive action if necessary. The prediction shows a steady increase in oil temperature but no immediate risk, as indicated by the zero failure probability in the table.

Figure 15: Predictions Dashboard

4.2.3.1 Preliminary Results

Historical data from a 400/150kV transformer in the Greek pilot was used for an initial assessment of the module. The data contained values for DGA, bushing capacitance and tand and oil temperature from March 2024 to the end of September 2024. The data of September was used for evaluation and the rest for training. At this period the transformer operated without any issues.

The oil prediction model resulted in mean absolute percentage error of around 1% on the training set, and the threshold for anomaly detection was computed at 1.5 C°. No alarm was raised for abnormal behaviour on for the top-oil temperature. The limit was only passed at 2024-09-10 19:00:00 but only for only one hour thus no alarm was raised. At the same time interval a change in capacitance of

Bushing H1 (raising an alarm) was also observed indicating that a possible incident might have occurred in that period. No alarm was raised for DGA since the values and the changes in this data was below the limits used to detect an issue.

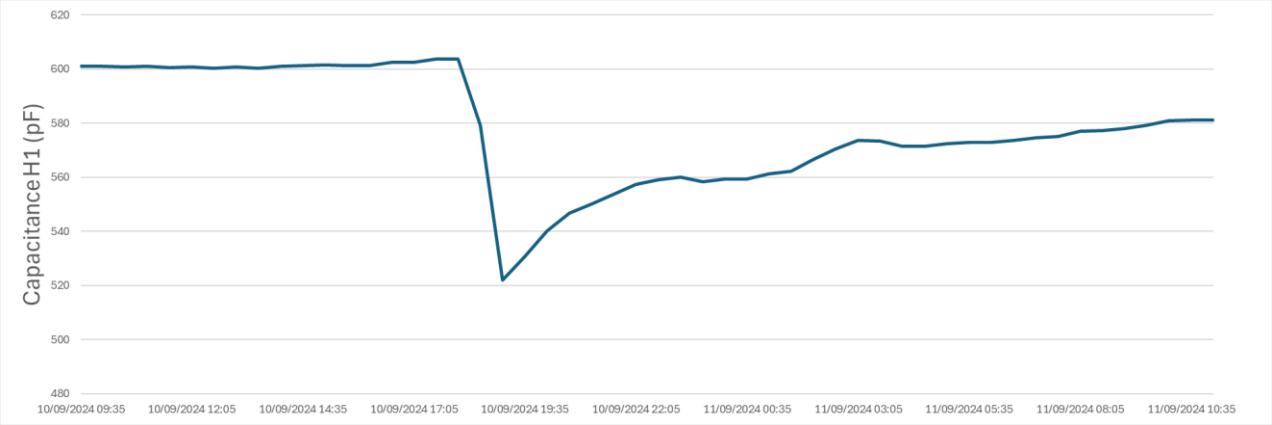

Figure 16: Bushing H1 capacitance at 10/09/2024

The historical data was few to build a predictor with good accuracy on the future currents. Based on the existing historical data the mean error in current (A) of current prediction submodule is presented on Table 3. This submodule will be updated as more historical data is collected. Nevertheless, even with this error the oil temperature prediction always lied within the calculated confidence interval in the testing period and no top-oil temperatures above the threshold were estimated, which was what was observed in the historical data.

Table 3. Accuracy of Current Prediction Submodule

Hour Ahead	Accuracy
	Mean (A)
1	4.3
2	7.7
3	12.3
4	15.5
5	18.5
6	19

4.3 Non-Technical Losses detection

Non-Technical Losses (NTL) is any electrical energy consumed and not invoiced and can therefore be considered fraud or energy theft. Several solutions exist in commercial products and in the literature for detecting fraud in energy delivery. This interest is mainly because it is something that impacts the economic revenues of the energy actors.

On the other hand, Technical Losses (TL) are inherent to the transmission and consist mainly of the dissipation of electricity in transportation (lines, shunts, etc.), transformation, distribution, and energy measurement.

The relation between them is direct. NTL are equal to total energy injected in the network minus TL and the legal and measured consumptions.

$$NTL = P_{injected} - TL - \sum P_{consumer}$$

Many methods based on different machine learning techniques have been sketched to detect NTL, using SVM, KNN, decision trees or other ML algorithms. These methods can be classified as *data-oriented*, which use user related data, mainly energy consumption. With the rise of ML and Big Data techniques, those data-oriented techniques have emerged as the most promising to predict and classify if a user has a fraudulent consumption.

As shown in the Figure 17, data-oriented methods are divided into supervised and unsupervised. Methods that make use of both labels (positive/negative fraud classes) are *supervised*, while methods that make no use of labels are *unsupervised*. Because of the scarcity of fraud data, for supervised methods data-augmentation and synthetic data generation must be used to ensure the correctness of the solution if a supervised method is chosen.

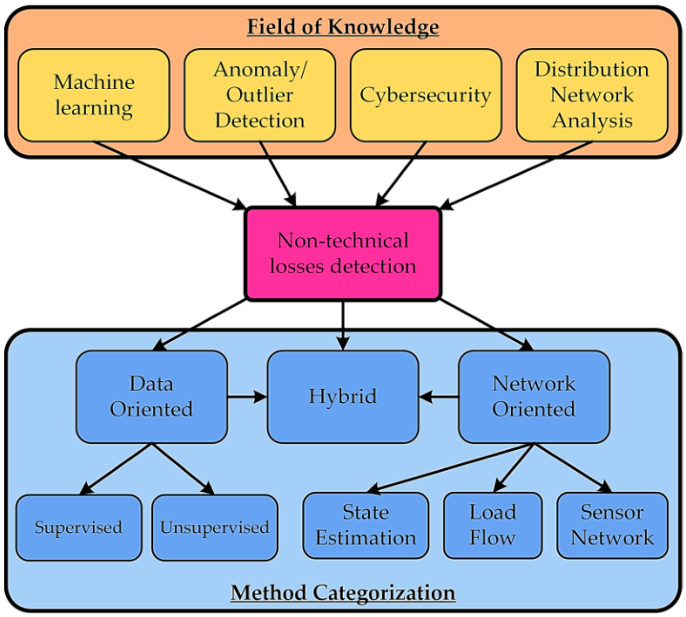

Figure 17: Classification of methods for detecting NTL [16]

Also *Network-Oriented* and *Hybrid methods* can be used to detect subsystems inside the network where there are probably energy thefts, either because of fraud consumption or unregistered connections. The former are based on network analysis and the physical rules that describe such systems. The latter are a combination of network and data-oriented methods. These methods are categorized according to the main concept/algorithm used, i.e. state estimation, load flow, or special sensors for fraud detection.

As part of the ETER software, ETRA has a model based on a mix of data and network-oriented techniques (hybrid method). Deep Neural Network and Power Flow Analysis are both used to detect fraudulent users and illegal unregistered connections.

4.3.1 Design

As explained in the introduction, a data-oriented method technique will be used to detect fraud consumption.

Data Oriented Methods

With the raise of Big Data, methods based on Neural Networks have emerged as the choice-to-go for NTL detection. Among the different architectures and other ML techniques, auto-encoders and CNN are those with best performance.

As can be seen in Figure 18, both Auto-encoders and CNN architectures have the best performance. Different models have been tested and finally an architecture based on CNN has been chosen.

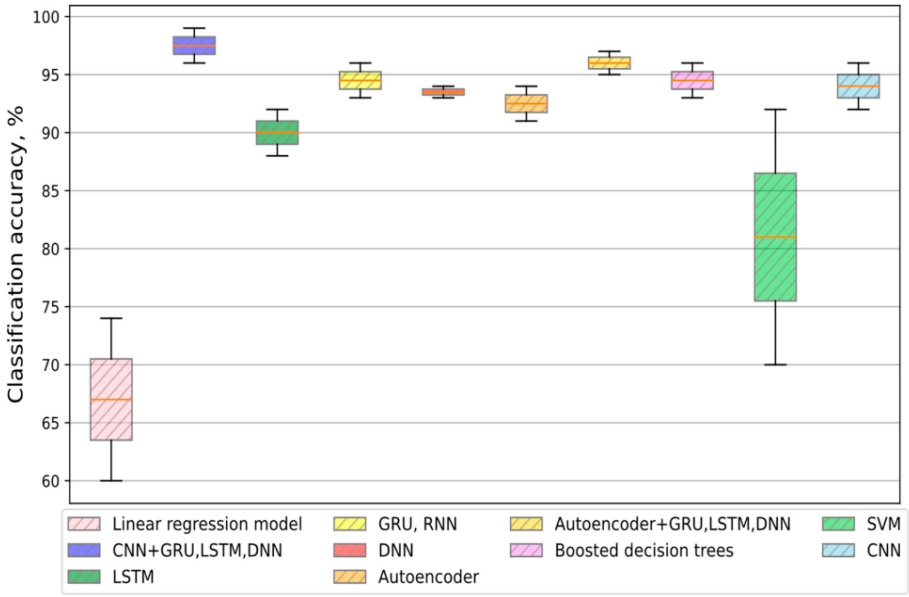

Figure 18: Accuracy of different NTL detection ML methods [17]

CNN works usually over 2D data. Because of that, data is grouped weekly. Also, because it is hard to capture key features and main differences between daily data (1D), grouping data allows to observe a periodicity on the electricity consumption [18]. There is a periodicity for most normal customers if data is aligned, but no for electricity fraud (Figure 19-Figure 20).

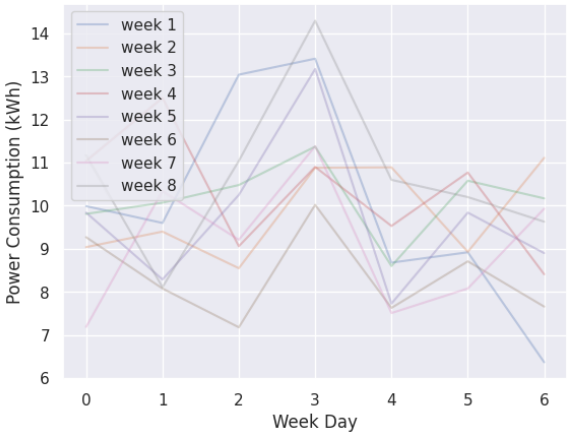

Figure 19: Alignment of regular consumption

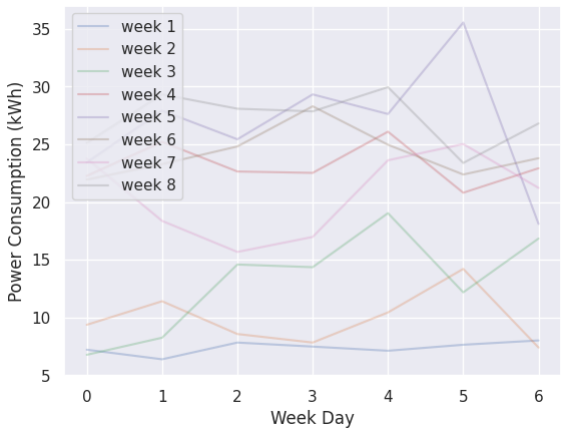

Figure 20: Unalignment for fraud consumption

Before processing data, it must be prepared. Nonexistent data will be interpolated with PCHIP algorithm. Also, normalization, data augmentation and synthetic data creation will be run to avoid bias to the regular user class. This is mandatory because of the lack of many fraud data correctly classified that allow a good training phase.

As an output of this phase, each user will be classified as fraud/non-fraud consumer.

Network Oriented Methods

The data-oriented fits perfectly to classify a user as legal/fraud. But it does not allow to detect unregistered connections. To carry on this computation, network analysis based on global energy supply to the network, topology, morphology and power supply line physical features will be run.

This process can be simplified as follows [19]:

- With the network definition, thermal losses to the outside can be computed/estimated.
- The TL can be approximated with this thermal loss.
- We can then estimate the NTL with total energy supply minus this TL estimation minus consume registers.
- A power flow analysis is run from terminal network nodes to the parent root of the subsystem using registered real powers. Voltages are initialized with real root voltage for every bus.
- This power flow analysis is done many times, updating losses and voltages.
- When network analysis converges, $TL_{estimated}$ versus $TL_{computed}$ are compared, and those lines where the difference between them is bigger than an epsilon may have an unregistered connection.

4.3.2 Implementation

The model implementation is based on the Python programming language and used several open-source libraries.

For the ML techniques, NumPy, pandas, Scikit-Learn and Keras are the base of the deep learning algorithm. Keras has been chosen because of the flexibility it provides, allowing to use almost the same code while the backend code can be changed amongst JAX, PyTorch and TensorFlow.

- PyTorch is a Meta library widely used in CNN and transformer architectures.
- TensorFlow and JAX are google development for high-performance numerical computations.
- Numpy is a library for fast numerical python computation that uses underlying C libraries like BLAS or MKL, allowing parallel computations with the flexibility provided with python over raw C code.

The preprocessing is based on pandas and scikit-learn. These are used to clean up the dataset, prepared it and split it between train, validation and test data, while Keras is used to the real deep learning algorithm implementation.

For the network analysis, data from smart meters pre-processed in the last step will be used. Also, the network topology and analysis will be done with Panda Power library. Itself, Panda Power relies on PyPower and NetworkX libraries to create the topology and analyse it. The main features of Panda Power are:

- It makes easy to parametrize electric model of lines, transformers and switches.
- It is designed to automate evaluations
- Can be easily adapted to our needs

This network analysis module will be deployed with an http and NATS interface. In Figure 21 and Figure 22 a power flow performed with Panda Power is shown.

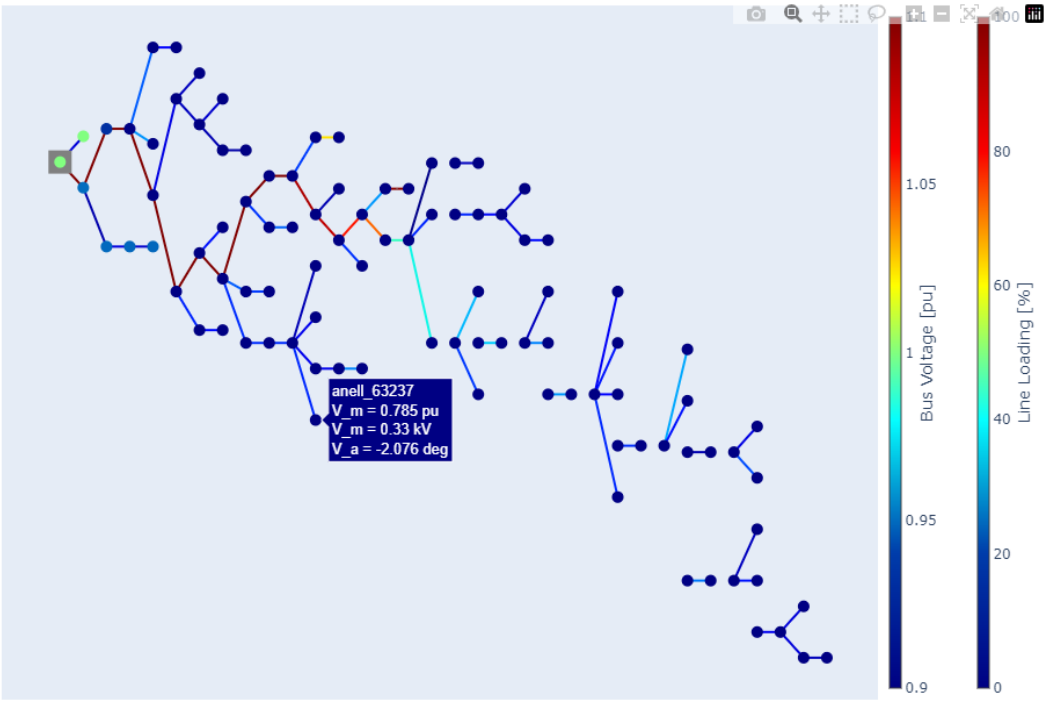

Figure 21: Powerflow analysis result

	vm_pu	va_degree	p_mw	q_mvar
0	0.695971	-4.380847	0.000000	0.000000
1	1.000000	0.000000	-269.238504	-52.939987
2	0.950005	-0.209304	0.000000	0.000000
3	0.948527	-0.207757	4.000000	0.843436
4	0.948557	-0.208037	0.000000	0.000000

	p_from_mw	q_from_mvar	p_to_mw	q_to_mvar	pl_mw	ql_mvar	i_from_ka	i_to_ka	i_ka	vm_from_pu	va_from_degree	vm_to_pu	va_to_degree	loading_percent
0	0.010002	0.001944	-0.010000	-0.001944	2.113685e-06	1.197262e-07	0.019778	0.019778	0.019778	0.708188	-3.911060	0.708042	-3.909453	98.891017
1	0.010016	0.001946	-0.010002	-0.001944	1.377097e-05	1.802483e-06	0.019778	0.019778	0.019778	0.709151	-3.915880	0.708188	-3.911060	27.469727
2	0.093911	0.012320	-0.092839	-0.011784	1.071500e-03	5.357502e-04	0.183600	0.183600	0.183600	0.709151	-3.915880	0.700681	-4.155773	79.826116
3	0.007001	0.001700	-0.007000	-0.001700	1.059966e-06	6.004006e-08	0.013407	0.013407	0.013407	0.738707	-3.135631	0.738599	-3.134105	67.034514
4	0.005001	0.000999	-0.005000	-0.000999	8.866231e-07	3.359013e-08	0.010258	0.010258	0.010258	0.683427	-4.388246	0.683310	-4.386664	13.321507

Figure 22: Results after performing power flow with Panda Power

The deployment will use Docker, a containerization platform that allows for easy packaging and deployment of applications in a consistent and isolated environment. Docker ensures that the

application runs reliably across different environments, making it an essential tool for efficient deployment strategies.

Implementation will make also of other components and databases:

- For the storage, time-series and document-oriented database are needed. The data feed into the database is either coming from the smart meters or from other sources that allow to complete the historical series. The time-series will be Influx DB, a very efficient database used to store huge amounts of time data. The document-oriented database used for projections and power flow storage will be Mongo DB.

4.3.3 Mock-Ups

The application will have a section that will include information (Figure 23) about how likely a meter is tampered with:

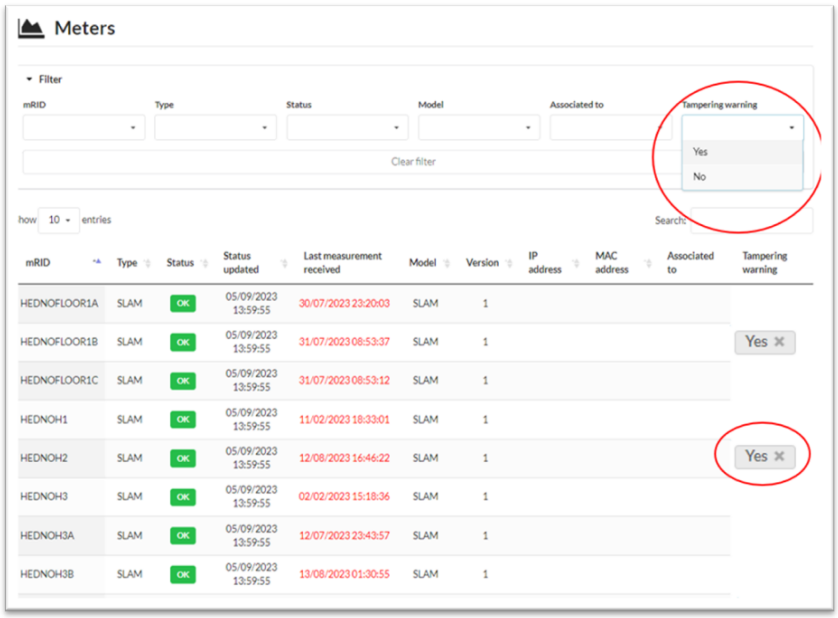

Figure 23: Fraud consumers labeled

For each meter/user, the corresponding row will either include or not the tampering tag. There will be also the possibility to filter the table to show only the meters potentially affected by NTL.

The section in the application that presents the topology (Figure 24) will also include information on the substation/feeder affected by NTL, and the possible sections or subsystems in the network with possible illegal unregistered consumption.

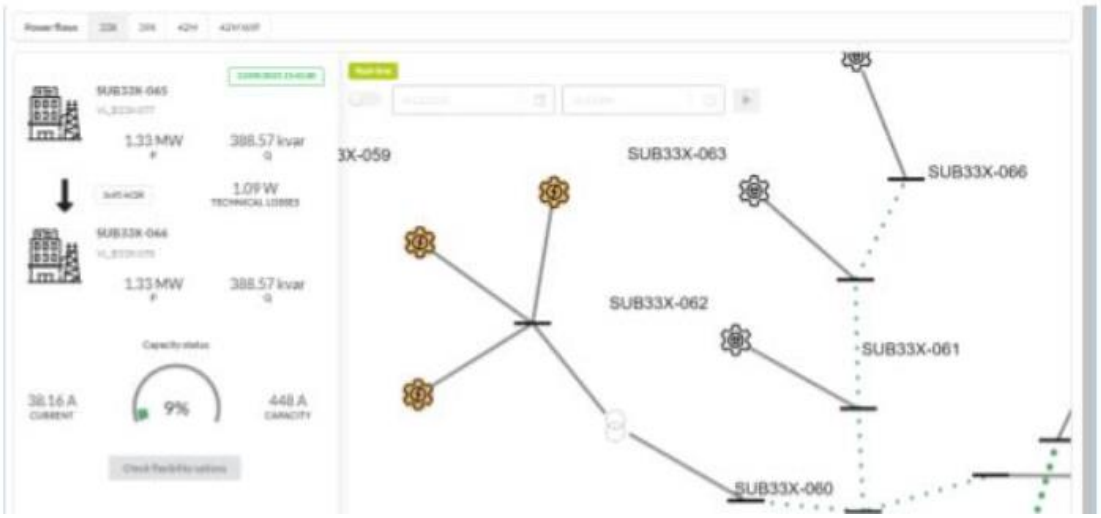

Figure 24: Network Topology

4.4 Network Planning tool

In recent years, substantial research has focused on optimizing the design of modern power distribution systems. The goal of distribution system planning is to ensure that the system can meet increasing demand and renewable energy source (RES) growth in a timely, economical, reliable, and safe manner [20].

During the decarbonization process that takes place in power systems, new technologies, operational patterns, and players have emerged (e.g., flexibility resources, flexibility services, transactive energy markets, and prosumers). Flexibility resources include technologies like distributed generation (DG), electric vehicles (EVs), energy storage systems (ESSs), and demand response (DR). These resources can provide flexibility services in distribution systems, where end-users have traditionally been passive participants. However, with the growing integration of distributed energy resources (DERs) and the shift from passive to active distribution networks, end-users are now playing a more active role, offering flexibility services to distribution system operators (DSOs) and enhancing their energy security (e.g., using photovoltaic (PV) systems paired with ESSs) [21].

Local energy markets are emerging as an efficient way to maximize the use of RES within specific areas, enabling smaller consumers to take a more active role in the energy market. These new market mechanisms can significantly impact distribution system planning, requiring new approaches that account for market characteristics and promote the development of sustainable power systems. While there is extensive literature on the impact of local energy markets on distribution system operations, less attention has been given to how these markets affect distribution system planning.

Traditionally, distribution system planning focused on reinforcing or installing new substations and circuits. The increasing penetration of DERs and the decentralization of power systems have added complexity to this problem. Other expansion options, like capacitor banks (CBs) and voltage regulators (VRs), are often included in short-term planning strategies. Additionally, with the growing adoption of DERs, decision variables like the allocation of DG units, electric vehicle charging stations (EVCSs), and ESSs have been integrated into the planning problem.

Modern distribution system planning must account for the impact of flexibility to minimize the need for grid upgrades by maximizing the use of existing resources. The objective of the traditional

distribution system planning problem is to determine, in the most economical way, the necessary investments to meet demand growth, ensure system safety, and maintain high-quality electricity supply. A complete formulation would result in a mixed-integer non-linear programming (MINLP) model, involving continuous and integer variables under a set of non-linear equations representing the distribution network's operation. Solving this problem is complex due to its combinatorial nature and the large number of decision variables.

Furthermore, the distribution system planning problem can be approached through single-period or multi-period planning. In single-period planning, the entire planning horizon is considered at once, with demand forecasts for the end of the period. In contrast, multi-period planning divides the horizon into several stages, with expansion actions taken at different times based on demand forecasts. The solution techniques often involve either mathematical programming or heuristic and metaheuristic approaches.

Mathematical programming is one of the most commonly used approaches for solving the planning problem. It offers the advantage of guaranteeing optimal solutions and is generally easier to implement than heuristic or metaheuristic methods. Commercial solvers, such as CPLEX and GUROBI, are often used to solve mixed-integer linear programming (MILP) formulations of the problem. Another popular formulation is the mixed-integer second-order cone programming (MISOCP) model [21].

However, exact solution methods may struggle to solve large-scale planning problems. In such cases, heuristic and metaheuristic techniques become advantageous. These methods have been widely applied in multi-objective optimization to minimize costs and maximize reliability. For example, Tabu Search (TS) and Particle Swarm Optimization (PSO) are frequently used due to their flexibility in modelling various aspects of the distribution system planning problem. The branch-exchange heuristic has also been applied to long-term planning, and genetic and evolutionary algorithms have been employed in many cases.

Bi-level optimization has gained traction in planning problems involving energy markets. In this approach, distribution system operators, DER owners, and load aggregators participate in local energy markets (LEMs). Investment decisions regarding network assets and DERs are made at the upper level, while the lower level optimizes DER operation by clearing the LEM. However, this approach can be computationally demanding, often requiring several hours to reach a solution [21].

In OPENTUNITY, the planning module that is being developed is focused on the DSO perspective of the planning problem and is formulated either as a single year MILP or a multi-year MISOCP according to the objective selected by the user. The GUROBI solver will be utilized to solve the problem. Since, it is designed to be an interactive tool operated by system operators to run multiple scenarios, its computational complexity must be maintained limited to have the ability to provide solutions in minute scale. Therefore the impact of flexibility will be considered as part of a single stage optimization model with a fixed price selected by the user. The detailed planning too design is described in the next section.

4.4.1 Design

The aim of the developed module is to offer a complete solution with all the necessary submodules for system planning. The tool is being designed to support to an extent the handling of the data provided by DSOs in the most raw form possible. That would include to extract topology from CIM

files and generate demand curves that are used for the planning tool values from smart meter data, that contain consumption in hourly intervals. The three main core tools consist of a fast load flow algorithm, a clustering sub module that is used to generate the optimization scenarios and the optimization module.

Figure 25 presents the logic diagram that illustrates a process for distribution system planning tool. Initially, the user provides the system topology which is modified from the user provided format to the data model that the tool uses. Thus, distinctive sub modules are being built that support this format change. The topology input format that is supported are ".csv" files and CIM files. The necessary load curves per secondary substation are then uploaded by the system operator to compute the demand curves and power factors that will be considered in the tool. A second option will be also developed, where the user can upload smart meter data in a repository and the aforementioned curves can be generated by the tool automatically. Apart from these data, the user has to also define specific settings for the analysis that will be performed, define location and capacity of future PV installations and provide equipment data that should be considered in future upgrades. These operations will be facilitated by a graphical user interface (GUI) and specific checks will be performed on the data provided if their format is not correct, e.g. the topology data check presented in the Figure 25.

If the aforementioned input files are correct, the tool proceeds to generate the PV power curves with hourly resolution, according to the location of the PV installation. Then, both the PV and demand curves are used to run load flows for every hour in the number of years selected by the user. The load flows are based on Fixed-Point Iteration algorithm, which is an algorithm that can perform load flow calculations in radial networks, which is the typical case of distribution networks. The FPI algorithm structure allows parallel processing achieving fast computational times that over 100 times faster than conventional load flow tools that use newton Raphson algorithm. The results of the load flows are presented to the user in the forms that are presented in section 4.4.2.

The next step utilizes a submodule that uses the PV generation curves, the demand profiles and the load flow results to perform time series clustering in order to generate the scenario profiles that will be used by the optimization algorithm. Finally, the user selects an optimization goal that could be either multi-year cost reduction, multi-year investment deferral or maximizing renewable energy sources for a single year. Then, the optimization problem is automatically designed and solved. If a solution is reached and it is feasible, the process concludes with a presentation of the costs and necessary system upgrades.

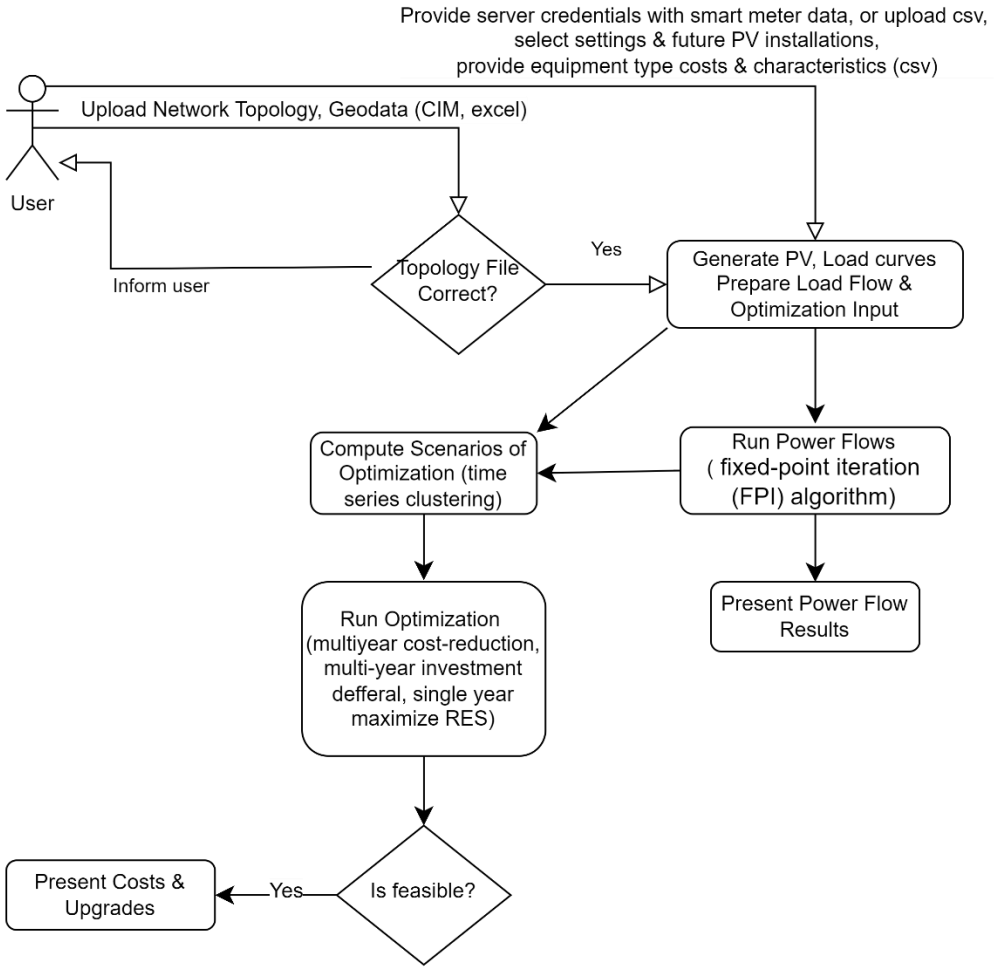

Figure 25: Logic Diagram of Planning toolbox

The next sections present a more detailed description of the module design, presenting in detail the methodologies followed to build the planning tool in python, mentioning also what libraries have been developed or utilized.

4.4.2 Implementation

4.4.2.1 Load Flow Module

Initially the PV and demand curves are generated by the module. The demand curves are generated by the user data, the PV curves are generated via the PVGIS (Photovoltaic Geographical Information System) tool.

The Photovoltaic Geographical Information System (PVGIS) [22] is a tool developed by the European Commission's Joint Research Centre to assess solar energy potential in various regions worldwide. It provides solar radiation data, which helps users estimate how much solar energy is available in specific locations by using satellite imagery and ground measurements. This data includes daily, monthly, and hourly solar radiation levels and allows for the analysis of solar patterns over time.

PVGIS also enables users to estimate the energy output of different photovoltaic systems, including grid-connected and off-grid systems, as well as systems with sun-tracking technology. It can calculate how efficiently solar panels will perform based on the geographical location, type of solar panels, and their positioning. The tool even takes into account the inclination of panels and how the angle affects solar irradiance.

Additionally, PVGIS considers terrain elevation and shadows caused by hills or mountains, which may reduce the availability of solar energy in certain areas. Users can input their location, either manually or by selecting it on a map, and receive downloadable data in charts or spreadsheets for further analysis. This makes PVGIS highly useful for conducting feasibility studies and planning solar energy projects in different regions. PVGIS APIs can be called directly using different languages like Python, NodeJS, Perl, Java and many others. The user selects in the GUI the secondary substation that the PV will be installed and the nominal power and uploads also the geographical coordinates of each substation, which are all used to request from the API to generate a typical production year curve for that location.

Having collected the PV curves, demand curves and the user defined annual load growth rate the tool runs a multi year load flow analysis to identify critical lines and buses in the distribution system under test. Typical load flow tools would require several minutes to run these simulations if hourly resolution is considered.

Advances in computer hardware, such as the increase in the number of cores in CPUs and the evolution of GPU designs, transitioning from simple graphics processors to highly parallel multiprocessors of many cores, opened a new paradigm of programming and rethinking PF algorithms.

Fixed-point iteration (FPI) algorithms have been proposed to solve the load flow problem, showing accurate results and numerical stability. As described in [23], FPI methods can guarantee convergence to the power flow solution via the Banach fixed-point theorem. The fixed point algorithm has shown robust performance; it has a simple formulation for the case of radial distribution system analyses, is suitable and scalable for multidimensional applications, and can also benefit from multicore CPU and GPU parallelization.

The FPI algorithm has the following rationale. Considering a distribution system feeder with one HV/MV on bus $\boldsymbol{b_s}$ and $\boldsymbol{b_d}$ demand nodes results in an admittance matrix of the following form (5), where V are complex components of the respective nodal voltages and I the complex injections of nodal current at the substation and at the demand nodes.

$$\begin{bmatrix} \boldsymbol{Y_{ss}} & \boldsymbol{Y_{sd}} \\ \boldsymbol{Y_{ds}} & \boldsymbol{Y_{dd}} \end{bmatrix} \begin{bmatrix} \boldsymbol{V_s} \\ \boldsymbol{V_d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I_s} \\ -\boldsymbol{I_d} \end{bmatrix} \tag{5}$$

The FPI algorithm is based on the notation of the equivalent version of the successive approximation method (SAM) presented in [11] as in (6)-(8):

$$\boldsymbol{V_{n+1}} = (-\boldsymbol{Y_{dd}} \cdot \boldsymbol{A^{-1}}) \cdot \boldsymbol{V_n} + -\boldsymbol{Y_{dd}} \cdot \boldsymbol{c} \tag{6}$$

$$\boldsymbol{c} = \boldsymbol{Y_{ds}} \cdot \boldsymbol{V_s} \tag{7}$$

$$\boldsymbol{A} = \boldsymbol{diag}(\boldsymbol{S}^*) \quad (8)$$

where S^* is the conjugate nominal complex power at demand nodes. The iterative procedure stops when the computed voltages deviation between iterations are below a specific threshold.

This module is built on Numba [24] library in python. Numba is designed to be used with NumPy arrays and functions. Numba generates specialized code for different array data types and layouts to optimize performance. Numba-compiled numerical algorithms in Python can approach the speeds of C or FORTRAN and supports parallel processing.

For a selected distribution system feeder in the Spanish demo the designed approach runs ten years with hourly resolution (87600 scenarios) in around a second, which was at least 100 faster when the Panda Power library and Newton Raphson algorithm was used together with parallel processing.

The results of load flow analysis are also passed to the scenario clustering module.

4.4.2.2 Scenario Clustering Module

The scenario clustering module requires as input the load curves per substation, the PV production curves, the number of years in the analysis, the load flow results and the load growth rate. The objective of this code is to generate the daily scenarios for the optimization module by clustering the daily profiles every year based on the PV and load curves and the load growth rate. The other objective is to compute weights that are used in the optimization problem to evaluate more appropriately the trade-off between flexibility and system upgrades.

Initially, the total net load is computed by adding the demand of the substations and the production of PVs for every year, generating an aggregated net load curve of 8760 points (hours of the year). Then, using the pandas library in python the 365 daily profiles are generated. The 365 daily profiles are then clustered in three distinctive clusters. The clustering is performed using the time series clustering, utilizing the tslearn toolbox [25]. Compared to typical k-means algorithm the uses the Euclidean distance as a similarity measure between the examples the tslearn toolbox uses the Dynamic Time Warping metric for clustering the daily profiles

Dynamic Time Warping (DTW) is a method for measuring the similarity between two time series that may vary in timing. Unlike traditional distance metrics like Euclidean distance, DTW allows for non-linear alignments between the time points of two series, making it particularly useful for time series data where patterns can shift or stretch over time. DTW aims to "warp" the time axis of one or both time series so that similar shapes or patterns in the data are aligned, even if they occur at different time points. It calculates the optimal alignment between two time series by considering various possible paths and selecting the one that minimizes the cumulative distance between the points. On the other hand, Euclidean distance, which is the typical distance measure in K-means clustering algorithms, is the distance calculated by summing the squared differences between values at similar times (hours). This strict one-to-one alignment means that even slight shifts in time between similar patterns can lead to large distance values, thus if the same pattern occurs at different times in the two timeseries, Euclidean distance will not recognize them as similar because it does not accommodate for time shifts. The equation (9) with the conditions presented in (10) is the formulation of DTW for two timeseries with 24 samples (daily power curves with hourly resolution).

$$\boldsymbol{DTW}(\boldsymbol{x},\boldsymbol{y}) = \min_{\boldsymbol{\pi}} \sqrt{\sum_{(\boldsymbol{i},\boldsymbol{j})\in\boldsymbol{\pi}} \boldsymbol{d}\left(\boldsymbol{x}_{\boldsymbol{i}},\boldsymbol{y}_{\boldsymbol{j}}\right)^2} \tag{9}$$

where $\boldsymbol{\pi} = [\boldsymbol{\pi_0}..\boldsymbol{\pi_K}]$ is a path that satisfies the following properties:

It is a list of pair of indexes $\boldsymbol{i},\boldsymbol{j}$ with $\boldsymbol{0} \leq \boldsymbol{i},\boldsymbol{j} \leq \boldsymbol{23}$

$\boldsymbol{\pi_0} = (\boldsymbol{0},\boldsymbol{0})$ and $\boldsymbol{\pi_K} = (\boldsymbol{23},\boldsymbol{23})$ (10)

For all $\boldsymbol{k} \geq \boldsymbol{0}$, $\boldsymbol{\pi_k} = (\boldsymbol{i_k},\boldsymbol{j_k})$ is related to $\boldsymbol{\pi_{k-1}} = (\boldsymbol{i_{k-1}},\boldsymbol{j_{k-1}})$ as follows:

$$\boldsymbol{i_{k-1}} \leq \boldsymbol{i_k} \leq \boldsymbol{i_{k-1}} + \boldsymbol{1}$$

$$\boldsymbol{j_{k-1}} \leq \boldsymbol{j_k} \leq \boldsymbol{j_{k-1}} + \boldsymbol{1}$$

Among the curves in each cluster the curve that has the highest amount of power among its hours is selected as the representative curve of the cluster, and the PV and demand curves are collected for that specific day to be included in the optimization problem. How many days are in each cluster are counted and this number is also passed to the optimization problem.

Finally, having the worst-case scenario might lead to an overestimation of the flexibility required for the whole cluster of scenarios or an overestimation of power losses in the network. Thus, two factors are calculated, one for the power losses and one for the flexibility usage, which are then sent to the optimization module to update the objective function. The factor of the power losses is computed as the mean power losses of the scenarios in the cluster as computed by the load flow results divided by the power losses of the worst-case scenario. The factor that is used as a weight for the flexibility cost is computed as the ratio of the scenarios that have loading over 90%, as computed by the load flow analysis to the total scenarios of the cluster. If no scenarios have loading >90% then the factor value is selected as 1.

If the user selects as an optimization goal to maximize RES capacity then a different approach is followed. The current year data is selected and the demand is normalized to the peak demand of the year. A normal PV generation profile is also computed for a mean latitude and longitude in the demo, using the geodata of the substations. The hour where the difference between the normalized PV profile and demand profile is selected as the worst-case scenario for RES maximization and based on that scenario the optimization problem of maximizing RES capacity is performed.

4.4.2.3 Optimization Module

The optimization module encompasses the formulation via the GUROBI solver of the three different options that the DSOs can select for their system planning. All of the formulations are built in python using the gurobipy library [26]. The variables and the sets used in the optimization are presented in Table 4 and Table 5. For the maximization of the RES which is a single year (and scenario) problem the indices of year, days and hours are obviously are not used.

Table 4. List of Variables of Optimization Problems

Variable	Description	Indices
$L_{y,d,h,l}$	Current Square	years, days, hours, lines
$P_{y,d,h,l}$	Line Active Power Flow	years, days, hours, lines
$Q_{y,d,h,l}$	Line Reactive Power Flow	years, days, hours, lines
$U_{y,d,h,b}$	Bus Voltage Square	years, days, hours, buses
$LS_{y,d,h,b}$	Involuntary Load Shedding	years, days, hours, buses
$P_{y,d,h,r}$	RES production	years, days, hours, res units
$CP_{y,d,h,r}$	Involuntary RES curtailment	years, days, hours, res units
$Qr_{y,d,h,r}$	RES reactive power	years, days, hours, res units
$Fr_{y,d,h,r}$	RES flexibility	years, days, hours, res units
$Fd_{y,d,h,b}$	Demand flexibility	years, days, hours, secondary substation
$z_{y,l,t}$	Binary variable indicating the upgrade distribution line l at year y with type t	Years, lines, types of new lines
$x_{y,l,t}$	Binary variable indicating that an upgrade has been done already at line l at year y with type t	Years, lines, types of new lines
$Ploss_{y,d,h,l}$	Auxiliary Variable calculating active power loss on every line.	years, days, hours, lines
$Qloss_{y,d,h,l}$	Auxiliary Variable calculating reactive power loss on every line.	years, days, hours, lines
$Udrop_{y,d,h,l}$	Auxiliary Variable calculating voltage drop on every line.	years, days, hours, lines
$BESloc_{y,b}$	Auxiliary Binary Variable indicating the selection of	years, candidate buses

	BES (Battery Energy Storage) installation at bus b	
$E_{y,b}$	BES system energy capacity	years, candidate buses
$Pb_{y,b}$	BES system power capacity	years, candidate buses
$Pch_{y,d,h,b}$	BES charge power	years, days, hours, candidate buses
$Pdch_{y,d,hb}$	BES discharge power	years, days, hours, candidate buses
$SoC_{y,d,hb}$	BES State of Charge	years, days, hours, candidate buses
$Qb_{y,d,h,b}$	BES Reactive Power	years, days, hours, candidate buses

Table 5. List of Sets of Optimization Problems

Set	Description	Set	Description
Y	Number of years in the horizon. User defined	B	Set of Busses in the feeder. Computed by topology file.
D	Number of different days scenarios per year. Computed by the clustering sub module	B_b	Set of Candidate Busses for BES installation in the feeder. User defined.
H	24. Hours of a day	R	RES units. Provided by the user.
Li	Set of Lines in the feeder. Computed by topology file.	Ti	Set of line types to be considered in upgrades. Provided by the users.
B	Set of Busses in the feeder. Computed by topology file.	Lc	Set of Lines connected to the end bus of a line. Computed by topology file.

The detailed formulation of the optimization problems is presented in detail in the following sections.

4.4.2.3.1. Investment Deferral & Cost Optimization formulation

The objective function of GUROBU multi-year optimization function for either problem formulation is (11):

$$\min \sum_{y \in Y} \left(\frac{1+if}{1+i}\right)^{y} \cdot \Big(\Big(\sum_{d \in D} \sum_{h \in H} t_d \cdot \Big(w_{loss} \sum_{l \in Li} c_e \cdot Ploss_{y,d,h,l} + \sum_{b \in B} c_{LS} \cdot LS_{y,d,h,b} + \sum_{b \in Bb} c_e \cdot (Pdch_{y,d,hb} - Pch_{y,d,h,b}) + \sum_{r \in R} c_{PC} \cdot CP_{y,d,h,r} + w_{flex} \sum_{r \in R} c_F \cdot Fr_{y,d,h,r} + w_{flex} \sum_{b \in B} c_F \cdot Fd_{y,d,h,r}\Big)\Big) + \sum_{l \in Li} \sum_{t \in Ti} c_t \cdot lenght_l \cdot z_{y,l,t} + \sum_{b \in Bb} c_{E_b} \cdot E_{y,b} + c_{P_b} \cdot Pb_{y,b}\Big) \tag{11}$$

It contains two general terms, the first (three first rows of equations) represent the operating costs and the second (last term) the upgrade costs. The operating costs contain the cost of energy losses, the cost of involuntary load shedding and RES curtailment, the charge & discharge cost of BES and the cost of flexibility (RES or Demand flexibility). The second term contains the line upgrade costs, associating each line upgrade with the line length and the cost per km of specific line type that can be selected, and the battery system installations costs that contain the cost of energy and power capacity.

The optimization problem has the various constraints to respect. The formulation of the power flow is performed via the equations of the DistFlow model [27]. The DistFlow model allows to have convex constraints for the power flow for radial systems, which allow the solver (GUROBI) to find the global optimum solution. The DistFlow equations are presented in (12) – (15) four for every line:

$$P_l^2 + Q_l^2 = L_l \cdot U_{b_s} \tag{12}$$

$$P_l = L_l \cdot r + \sum_{l \in Lc} P_l - P_{G_e} + P_{D_e} \tag{13}$$

$$Q_l = L_l \cdot x + \sum_{l \in Lc} Q_l - Q_{G_e} + Q_{D_e} \tag{14}$$

$$U_{b_e} = -2 \cdot (r \cdot P_l + x \cdot Q_l) + L_l \cdot (x^2 + r^2) + U_{b_s} \tag{15}$$

The equation (12) is used as an inequality to make the optimization problem convex. Equation (13) and (14) are used for active and reactive power in every line. The power and the starting bus (b_s) flowing to the line (P_l, Q_l) are equal to the power loss in the line ($L_l \cdot r$, $L_l \cdot x$) the power flowing to other lines (Lc) connected and the end bus (b_e), the demand at end bus (P_{D_e}, Q_{D_e}) minus the production on that bus (P_{G_e}, Q_{G_e}). (15) links the voltage square of the terminal busses on the line according to the current and power flowing to the line.

The formulation of the constraints in the optimization problem have to take into account the possibility of a line upgrade in that line, thus these constraints cannot be used directly. To this end, the active, reactive power loss and voltage drop on the lines is computed the constraints (16)-(18).

$$-\boldsymbol{M}\cdot\sum_{t\in Ti}\boldsymbol{x_{y,l,t}}\leq \boldsymbol{Ploss_{y,d,h,l}}-\boldsymbol{L_{y,d,h,l}}\cdot\boldsymbol{r}\leq \boldsymbol{M}\cdot\sum_{t\in Ti}\boldsymbol{x_{y,l,t}}, \quad \forall\ \boldsymbol{y}\in\boldsymbol{Y},\boldsymbol{d}\in\boldsymbol{D},\boldsymbol{h}\in\boldsymbol{H},\boldsymbol{l}\in\boldsymbol{Li} \tag{16}$$

$$-\boldsymbol{M}\cdot\sum_{t\in Ti}\boldsymbol{x_{y,l,t}}\leq \boldsymbol{Qloss_{y,d,h,l}}-\boldsymbol{L_{y,d,h,l}}\cdot\boldsymbol{x}\leq \boldsymbol{M}\cdot\sum_{t\in Ti}\boldsymbol{x_{y,l,t}}, \quad \forall\ \boldsymbol{y}\in\boldsymbol{Y},\boldsymbol{d}\in\boldsymbol{D},\boldsymbol{h}\in\boldsymbol{H},\boldsymbol{l}\in\boldsymbol{Li} \tag{17}$$

$$\boldsymbol{M}\cdot\sum_{t\in Ti}\boldsymbol{x_{y,l,t}}\leq \boldsymbol{Udrop_{y,d,h,l}}-\boldsymbol{L_{y,d,h,l}}\cdot(\boldsymbol{x^2}+\boldsymbol{r^2})+\boldsymbol{2}\cdot\left(\boldsymbol{r}\cdot\boldsymbol{P_{y,d,h,l}}+\boldsymbol{x}\cdot\boldsymbol{Q_{y,d,h,l}}\right)\leq \boldsymbol{M}\cdot\sum_{t\in Ti}\boldsymbol{x_{y,l,t}}, \quad \forall\ \boldsymbol{y}\in\boldsymbol{Y},\boldsymbol{d}\in\boldsymbol{D},\boldsymbol{h}\in\boldsymbol{H},\boldsymbol{l}\in\boldsymbol{Li} \tag{18}$$

Constraints (16)-(18) are only valid if no upgrade is selected for the line (l) at year (y), day (d) and hour (h). The term $\sum_{t\in Ti}\boldsymbol{x_{y,l,t}}$ becomes zero and the DistFlow equations are computed through the r, x of the currently installed conductor or cable characteristics. If an upgrade is selected, $\sum_{t\in Ti}\boldsymbol{x_{y,l,t}}$ equals 1 the big M number makes the constraints always feasible. To compute the losses and the voltage drop if an upgrade is made the constraints (19)-(21) are added. These constraints are generated for every possible line upgrade, if the upgrade increases the current limit of the line, and force with the same logic as above to use the characteristics of the new line (r_t, x_t) if an upgrade with that type is selected ($\boldsymbol{x_{y,l,t}}$=1). If the candidate type for upgrade has lower limit than the existing then a constraint is also added to force $\boldsymbol{x_{y,l,t}}=0$ to reduce the number of binary variables in the problem.

$$-\boldsymbol{M}\cdot\left(\boldsymbol{1}-\boldsymbol{x_{y,l,t}}\right)\leq \boldsymbol{Ploss_{y,d,h,l}}-\boldsymbol{L_{y,d,h,l}}\cdot\boldsymbol{r_t}\leq \boldsymbol{M}\cdot\left(\boldsymbol{1}-\boldsymbol{x_{y,l,t}}\right), \quad \forall\ \boldsymbol{t}\in\boldsymbol{T_i},\boldsymbol{y}\in\boldsymbol{Y},\boldsymbol{d}\in\boldsymbol{D},\boldsymbol{h}\in\boldsymbol{H},\boldsymbol{l}\in\boldsymbol{Li} \tag{19}$$

$$-\boldsymbol{M}\cdot\left(\boldsymbol{1}-\boldsymbol{x_{y,l,t}}\right)\leq \boldsymbol{Qloss_{y,d,h,l}}-\boldsymbol{L_{y,d,h,l}}\cdot\boldsymbol{x_t}\leq \boldsymbol{M}\cdot\left(\boldsymbol{1}-\boldsymbol{x_{y,l,t}}\right), \quad \forall\ \boldsymbol{t}\in\boldsymbol{T_i},\boldsymbol{y}\in\boldsymbol{Y},\boldsymbol{d}\in\boldsymbol{D},\boldsymbol{h}\in\boldsymbol{H},\boldsymbol{l}\in\boldsymbol{Li} \tag{20}$$

$$-\boldsymbol{M}\cdot\left(\boldsymbol{1}-\boldsymbol{x_{y,l,t}}\right)\leq \boldsymbol{Udrop_{y,d,h,l}}-\boldsymbol{L_{y,d,h,l}}\cdot(\boldsymbol{x_t}^{\boldsymbol{2}}+\boldsymbol{r_t^2})+\boldsymbol{2}\cdot\left(\boldsymbol{r_t}\cdot\boldsymbol{P_{y,d,h,l}}+\boldsymbol{x_t}\cdot\boldsymbol{Q_{y,d,h,l}}\right)\leq \boldsymbol{M}\cdot\left(\boldsymbol{1}-\boldsymbol{x_{y,l,t}}\right), \quad \forall\ \boldsymbol{t}\in\boldsymbol{T_i},\boldsymbol{y}\in\boldsymbol{Y},\boldsymbol{d}\in\boldsymbol{D},\boldsymbol{h}\in\boldsymbol{H},\boldsymbol{l}\in\boldsymbol{Li} \tag{21}$$

Through the (16)-(21) constraints the auxiliary variables for active and reactive power losses and the voltage drop are computed. Through them, the equations for active and reactive power flow and voltage drop are computed, through the constraints (22)-(25):

$$\boldsymbol{P^2_{y,d,h,l}}+\boldsymbol{Q^2_{y,d,h,l}}\leq \boldsymbol{L_{y,d,h,l}}\cdot\boldsymbol{U_{y,d,h,b_s}},\ \forall\ \boldsymbol{y}\in\boldsymbol{Y},\boldsymbol{d}\in\boldsymbol{D},\boldsymbol{h}\in\boldsymbol{H},\boldsymbol{l}\in\boldsymbol{Li} \tag{22}$$

$$\boldsymbol{P_{y,d,h,l} = Ploss_{y,d,h,l} + \sum_{l \in Lc} P_{y,d,h,l} - P_{G_e} + P_{D_e},\ \forall\, y \in Y, d \in D, h \in H, l \in Li} \quad (23)$$

$$\boldsymbol{Q_{y,d,h,l} = Qloss_{y,d,h,l} + \sum_{l \in Lc} Q_{y,d,h,l} - Q_{G_e} + Q_{D_e},\quad \forall\, y \in Y, d \in D, h \in H, l \in Li} \quad (24)$$

$$\boldsymbol{U_{y,d,h,b_e} = Udrop_{y,d,h,l} + U_{y,d,h,b_s},\ \forall\, y \in Y, d \in D, h \in H, l \in Li} \quad (25)$$

For the active power balance, the term P_{G_e} resembles the production of active power at the receiving end of the line, which is the flexibility of demand and involuntary load shedding activated on that bus (decrease in load), the production of RES and the discharge power of the BES if it is selected for that bus. The demand term (P_{D_e}) contains the demand of the secondary substation on the receiving end, the flexibility of RES and RES involuntary curtailment activated on that bus (decrease in RES generation) and the charge of the BES if it is selected for that bus.

For the reactive power balance the same holds true. A fixed power factor is used for demand provided by the module user, while RES and BES reactive power are selected via the optimization problem. The constraint (26) is used to ensure that the current limit in every line is respected. If no upgrade in the line is selected $\left(\sum_{t \in Ti} \boldsymbol{x_{y,l,t}}\right) = 0$ and the existing current limit I_{lim} is used. If an upgrade is computed, $\left(\sum_{t \in Ti} \boldsymbol{x_{y,l,t}}\right) = 1$, and the $\boldsymbol{x_{y,l,t}} = 1$ for the selected type of line selected for upgrade, which forces the new current limit ($I_{t,lim}$) as a threshold for the current.

$$\boldsymbol{L_{y,d,h,l} \leq (1 - \sum_{t \in Ti} x_{y,l,t}) \cdot I^2_{lim} + \sum_{t \in Ti} x_{y,l,t} \cdot I^2_{t,lim},\ \forall\, y \in Y, d \in D, h \in H, l \in Li} \quad (26)$$

The bus voltage thresholds are set via constraint (27), for all the busses apart from the slack (main substation bus) where the upper and lower threshold force the voltage to be 1 p.u.

$$\boldsymbol{0.9^2 \leq U_{y,d,h,b} \leq 1.1^2,\quad \forall\, y \in Y, d \in D, h \in H, b \in B - \{slack\}} \quad (27)$$

The flexibility of the assets is also subject to various constraints. The flexibility of the demand ($\boldsymbol{Fd_{y,d,h,b}}$) is constrained via a factor (c_{fd}), which represents the amount of the demand in that secondary substation ($Demand_{y,d,h,b}$) that is flexible, as presented in (28).

$$\boldsymbol{Fd_{y,d,h,b} \leq c_{fd} \cdot Demand_{y,d,h,b},\quad \forall\, y \in Y, d \in D, h \in H, b \in B - \{slack\}} \quad (28)$$

The flexibility of the RES units is subject to constraints (29)-(30). The first correlates the unit's production, the involuntary RES curtailment and the RES downward flexibility to the available power computed via the scenario generation sub module. The second constraint constraints the reactive

power generation or consumption according to the user defined maximum power factor (PF_r^{max}), through which the factor w_r is computed via $w_r = \tan(\text{acos}(PF_r^{max}))$.

$$\boldsymbol{CP_{y,d,h,r} + P_{y,d,h,r} + Fr_{y,d,h,r} = P_{y,d,h,r}^{available}, \quad \forall\, y \in Y, d \in D, h \in H, r \in R} \tag{29}$$

$$\boldsymbol{-w_r \cdot P_{y,d,h,r} \leq Q_{y,d,h,r} \leq w_r \cdot P_{y,d,h,r}, \quad \forall\, y \in Y, d \in D, h \in H, r \in R} \tag{30}$$

The constraints (31) – (37) are related to BES units. (31)-(32) constraint the reactive power of BES units similar to RES units. (33) and (34) constraint the charge and discharge power to the according to the installed power capacity in that bus selected for the BES unit. (35) sets the initial SoC as selected by the user at the beginning and end of the day. The energy limits based on the limits of SoC ($\boldsymbol{SoC_{min}, SoC_{max}}$) provided by the user are respected by (37) according to the installed energy capacity in that bus. (36) is used to calculate the SoC evolution.

$$\boldsymbol{-w_b \cdot Pch_{y,d,h,b} \leq Qb_{y,d,h,b} \leq w_b \cdot Pch_{y,d,h,b}, \forall\, y \in Y, d \in D, h \in H, b \in B} \tag{31}$$

$$\boldsymbol{-w_b \cdot Pdch_{y,d,h,b} \leq Qb_{y,d,h,b} \leq w_b \cdot Pdch_{y,d,h,b}, \quad \forall\, y \in Y, d \in D, h \in H, b \in B} \tag{32}$$

$$\boldsymbol{Pch_{y,d,h,b} \leq \sum_{yi \in \{1,y\}} Pb_{yi,b}, \quad \forall\, y \in Y, d \in D, h \in H, b \in B_b} \tag{33}$$

$$\boldsymbol{Pdch_{y,d,h,b} \leq \sum_{yi \in \{1,y\}} Pb_{yi,b}, \quad \forall\, y \in Y, d \in D, h \in H, b \in B_b} \tag{34}$$

$$\boldsymbol{E_{y,d,hb} = \frac{SoC_{init}}{100} \cdot \sum_{yi \in \{1,y\}} E_{yi,b}, \forall\, y \in Y, d \in D, h \in \{0,23\}, b \in B_b} \tag{35}$$

$$\boldsymbol{E_{y,d,hb} = E_{y,d,h-1b} + Pch_{y,d,h,b} * n_{ch} - Pdch_{y,d,h,b} * n_{dch}, \quad \forall\, y \in Y, d \in D, h \in H - \{0,23\}, b \in B_b} \tag{36}$$

$$\boldsymbol{\frac{SoC_{min}}{100} \cdot \sum_{yi \in \{1,y\}} E_{yi,b} \leq E_{y,d,hb} \leq \frac{SoC_{max}}{100} \cdot \sum_{yi \in \{1,y\}} E_{yi,b}, \forall\, y \in Y, d \in D, h \in H, b \in B_b} \tag{37}$$

Finally, constraints (38)-(41) have to do with the binary variables of the problem. (38)-(39) dictate the lower and upper limits of power and energy in installed BES units as they dictated by the user. Constraints (40)-(41) force the logic on the binary variables of the line upgrades.

$$\boldsymbol{BESloc_{y,b} \cdot P_b^{min} \leq Pb_{y,b} \leq BESloc_{y,b} \cdot P_b^{max}, \quad \forall\, y \in Y, b \in B_b} \tag{38}$$

$$\boldsymbol{BESloc_{y,b}} \cdot \boldsymbol{E_b^{min}} \leq \boldsymbol{E_{y,b}} \leq \boldsymbol{BESloc_{y,b}} \cdot \boldsymbol{E_b^{max}}, \quad \forall\, \boldsymbol{y} \in \boldsymbol{Y}, \boldsymbol{b} \in \boldsymbol{B_b} \tag{39}$$

$$\boldsymbol{x_{y,l,t}} - \boldsymbol{x_{y-1,l,t}} = \boldsymbol{z_{y,l,t}}, \forall\, \boldsymbol{t} \in \boldsymbol{T_i}, \boldsymbol{y} \in \boldsymbol{Y}, \boldsymbol{l} \in \boldsymbol{Li} \tag{40}$$

$$\boldsymbol{x_{y,l,t}} \geq \boldsymbol{x_{y-1,l,t}}, \forall\, \boldsymbol{t} \in \boldsymbol{T_i}, \boldsymbol{y} \in \boldsymbol{Y}, \boldsymbol{l} \in \boldsymbol{Li} \tag{41}$$

To run the aforementioned optimization problem, the user has to define also the costs (c_{LS} : involuntary load shedding costs in €/MWh, c_{PC} : involuntary RES curtailment costs in €/MWh, c_F: flexibility purchase costs in €/MWh, c_t: cost per km per line type in €/km, $\boldsymbol{c_{P_b}}$: battery installation costs on battery power capacity cost in €/MW and c_{E_b} battery installation costs on battery energy capacity cost in €/MWh). The cost of energy (c_e) is automatically calculated by historical data on energy market per demo. The inflation and interest rate have to be also defined by the user.

The optimization runs on daily scenarios that are computed by the clustering submodule. Based on the days that exist in each cluster every year the parameter t_d is calculated as the number of days in that cluster. The weights w_{loss} and w_{flex} are selected according to the rational presented in the clustering sub module and according to the optimization goal selected by the user (cost-optimization or investment deferral). If investment deferral is selected the w_{loss} becomes small and the objective function compares the flexibility, load shedding & RES curtailment scenarios with the upgrades' costs. Thus, the use of flexibility will be selected in that option to postpone investments, until the investments costs are comparable to the flexibility costs or if the flexibility can not resolve the issues in the grid and RES curtailment or load shedding must occur.

Furthermore, the user has to upload the grid topology, which should include the characteristics of the line (r,x), the line rating and the connectivity information of the feeder. For the candidate lines that can be used to replace existing equipment the same information (r, x and current rating) has to be also provided apart from the costs.

Regarding the flexibility, the user also has to select specific parameters. For RES units, the user can select to consider control of reactive power via setting the maximum allowed power factor parameter (PF_r^{max}) to a value different than 1.

If the user wants to consider in the planning scenario the use of a centralized storage device, then additional parameters must be set. Initially, the set of candidate buses (B_b) for possible installation of central BES have to be set, the initial and end of day State of Charge ($\boldsymbol{SoC_{init}}$), the minimum and maximum SoC limits ($\boldsymbol{SoC_{min}}$, $\boldsymbol{SoC_{max}}$), the efficiency for charge and discharge (n_{ch}, n_{dch}) and the maximum power factor for reactive power control. Finally, regarding the demand flexibility, the user must set the percentage of total demand that is flexible via the parameter c_{fd}.

The available RES power per unit ($P_{\boldsymbol{y,d,h,r}}^{available}$) and the demand per secondary substation ($Demand_{y,d,h,b}$) are provided by the scenario clustering sub module.

4.4.2.3.1. Maximize RES capacity Optimization formulation

This formulation aims to find the best upgrades in the system, given a constraint in budget, that maximize the RES capacity considering also the flexibility available, A subset of the variables that are

presented in the previous section is utilized in this formulation as well as only 1 scenario generated by the clustering sub module. A relaxed linearized DistFlow model is used in this section to have a convex problem formulation, since the DistFlow model is not convex for this maximization problem. The objective function of this formulation is (42):

$$\max \sum_{b \in B} PV_b^{nom} \tag{42}$$

The term in the objective function aims to maximize the PV capacity The general relaxed DistFlow equations are (43)-(45):

$$BESloc_{y,b} \cdot P_b^{min} \leq Pb_{y,b} \leq BESloc_{y,b} \cdot P_b^{max}, \quad \forall\, y \in Y, b \in B_b \tag{43}$$

$$BESloc_{y,b} \cdot E_b^{min} \leq E_{y,b} \leq BESloc_{y,b} \cdot E_b^{max}, \quad \forall\, y \in Y, b \in B_b \tag{44}$$

$$x_{y,l,t} - x_{y-1,l,t} = z_{y,l,t}\,, \forall\, t \in T_i, y \in Y, l \in Li \tag{45}$$

The voltage drop term $(-2 \cdot (r \cdot P_{,l} + x \cdot Q_{,l})$ is computed via similar constraints to the ones presented in the previous section that use the binary $x_{l,t}$ that indicates whether an upgrade is selected to use the r,x of the upgraded line in the equation or the characteristic of the existing line. The demand and production in every bus are computed via the scenario input, the RES production which is the selected nominal power of PV in that substation and a factor computed by the clustering submodule (RES power = PV_b^{nom} ·wf) and have to do with the percentage of nominal power of RES in the selected worst case scenario. Flexibility is also considered in this scenario both in active and reactive power similar to the previous implementation. The voltage limits constraints are set similar to the prior formulation. The same holds true for the reactive power control of RES (PV) units.

The formulation is completed via the (46)-(49) constraints. (46) set the power limit in each line according to the existing or the rating of the upgraded line type. The maximum power limit is computed by multiplying the maximum power with the nominal voltage. Constraint (47) forces that up to one upgrade is considered per line. (48) forces the budget constraint limit and the last one set the flexibility limit per PV unit as a percent of its installed power which is selected by the user.

$$P^2{}_l + Q^2{}_l \leq S_{max,l}^2 \cdot \left(1 - \sum_{t \in Ti} x_{l,t}\right) + \sum_{t \in Ti} S_{max,t}^2 \cdot x_{l,t}\,, \forall\,, l \in Li \tag{46}$$

$$\sum_{t \in Ti} x_{l,t} \leq 1, \in Li \tag{47}$$

$$\sum_{l \in Li} \sum_{t \in Ti} c_t \cdot lenght_l \cdot x_{l,t} \leq Budget_{limit} \tag{48}$$

$$Fr_b \leq c_{fd} \cdot PV_b^{nom}, \quad b \in B - \{slack\} \tag{49}$$

Since this formulation do not contain the exact equations of power flow, i.e. it could lead to current limit violations, the resulted rated powers of PVs are checked by running power flows for the whole year and gradually decreased if any limit is violated.

4.4.3 Mock-Ups

The GUI of the planning module is under development and will be fully presented in the v2 of this deliverable. Some of its functionalities will be presented in this section to provide a first view of the interface. After the log in and the start of a new project, the user has to define the general settings in the scenario. This is performed in the scenario configuration tab.

Figure 26 shows a section of tab titled “Scenario configuration”. Under this tab, several sub-options are displayed: "Network Topology," "Future PV Installations," "Geodata," "Load Curves," and "Equipment Costs & Data."

In “Scenario configuration”, the user selects the horizon in years to run the analysis of power flow (and a multi-year optimization) scenario and the load growth rate to be considered. In the “Network Topology” tab the user has to upload an excel file containing the topology data of the feeder under test. In future versions other types of files, i.e. CIM files will be also included. The topology file has to contain the data of the lines which are r(Ohm/km), x(Ohm/km), maximum current (kA), line length (km), the name of the line and the busses that the line is connected to. In addition, the buses data are also included mentioning their name, the secondary substation name or a virtual name if it no substation is there and the nominal voltage.

When the topology is uploaded the “Future PV installations” tab appears, which is presented in Figure 27. In there the user selects among the substation the location of future PV installations, the year that it will be installed and the nominal power in kW.

In the “Geodata” tab the user uploads a “.csv” file contain the latitude and longitude of each bus presented in the topology file, which are used for the map representation in the load flow results and to generate the PV production curves using the tools mentioned in the previous section. In the “Load curves” tab information of the demand per secondary substation have to be included. At this moment a “.csv” file is supported, expecting per secondary substation the demand data for a year with hourly resolution and a “.csv” file for the power factor to be considered per secondary substation. In future version, it is likely to include an interface to a repository where the DSOs can upload smart meter data to generate these files automatically.

Finally, the user has to upload a “.csv” of equipment data, i.e. lines, that will be considered for possible upgrades. The “.csv” file should contain a name for the type, r in Ohm/km, x in Ohm/km, maximum current (kA), cost per km in €/km. At the bottom, there is a Save Changes button, suggesting that users can modify these settings and save them for future use. The overall design uses a dark theme with white text for clear visibility.

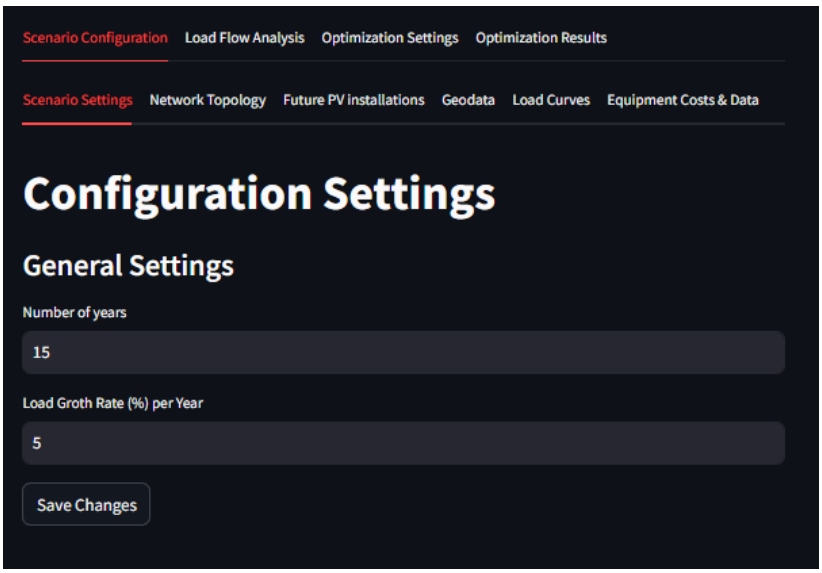

Figure 26: GUI of scenario configuration tab

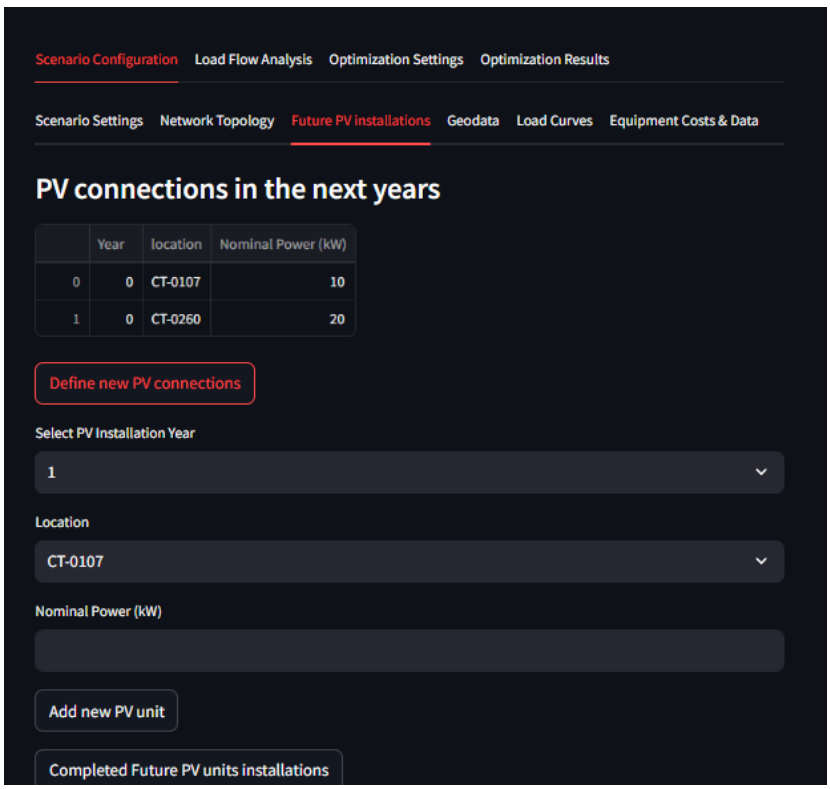

Figure 27: GUI of Future PV installations tab

Figure 28 and Figure 29 present the second general tab called "Load Flow Analysis". In this tab the user can execute the power flow, by clicking the respective button and the check for every year in the analysis the results in a map or select buses and lines and see voltage and loading boxplots. In Figure 28 it is shown that the user can select the year and see the results on map. Critical lines and buses, based on the loading and voltage levels respectively will have red or yellow colours to be more easily distinguishable. By clicking on a line or a bus the user can see the name of the asset and

same metrics, i.e. maximum, average and minimum loading in the lines (%) and maximum, average and minimum voltages in the buses in per unit.

In Figure 29 it is shown how the user can select a specific bus or a specific line and see the results for all the years in the same graph, in boxplots to identify at which year a threshold might be surpassed and either the use of flexibility or upgrades are needed. The results presented in these graphs are based on a feeder on the Spanish demo considering 15 years in the horizon and a load growth rate of 6% per year. This growth rate was selected in order to cause an issue in this feeder in the respective horizon.

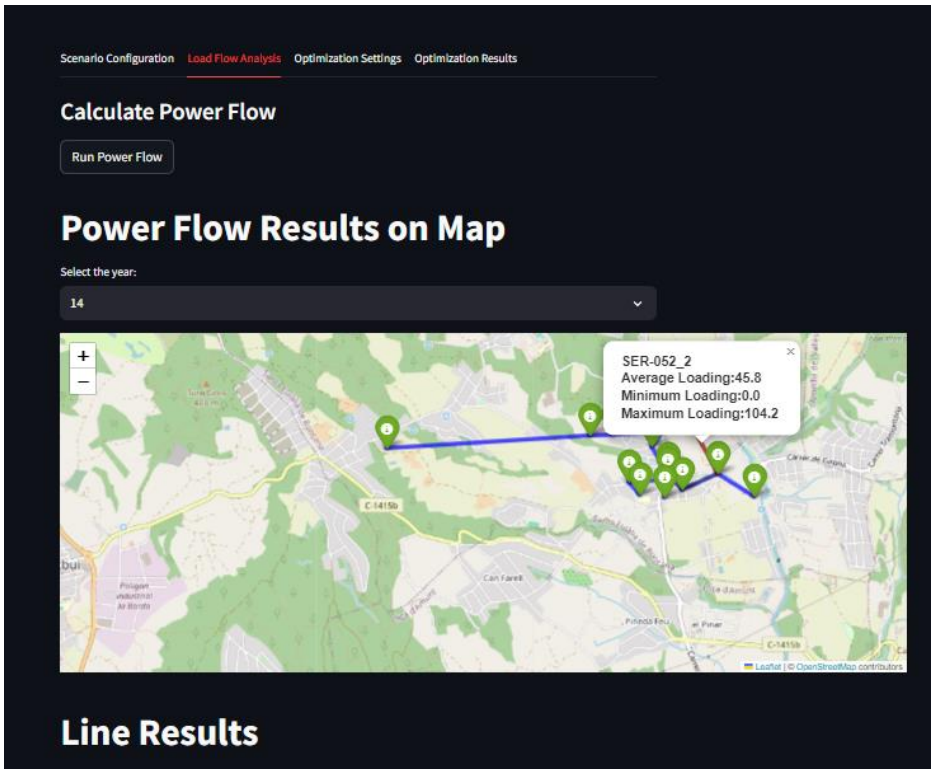

Figure 28: Load Flow analysis tab and results representation on map

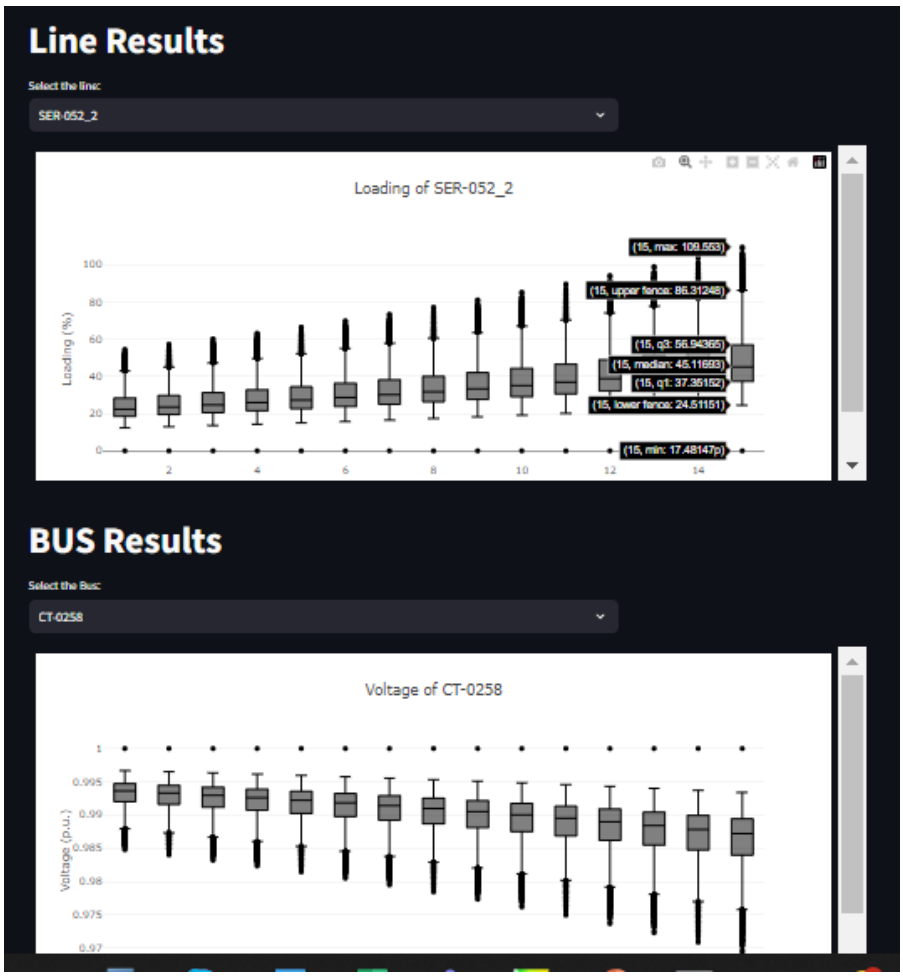

Figure 29: Load Flow analysis pre bus and line presentation via boxplots

The next tab that is presented after the execution of the load flow analysis is called "Optimization Settings" and is presented in Figure 30 and Figure 31. In optimization economic settings (Figure 30) the user sets the costs to be considered by the optimization formulation presented in Section 4.4.2.3 and in the Flexibility settings section (Figure 31) the user can select the whether to consider storage and possible locations from a drop down menu that contains the existing secondary substation and the storage system associated parameters, maximum flexibility by RES units as percentage of available power and whether to consider reactive power control and for demand the percentage of demand to be considered flexible.

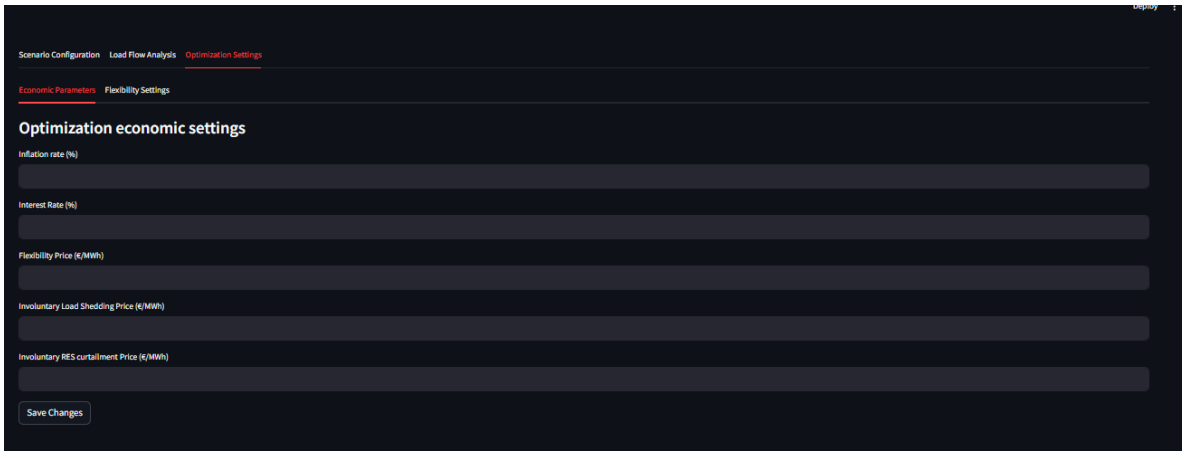

Figure 30: Economic parameters selection

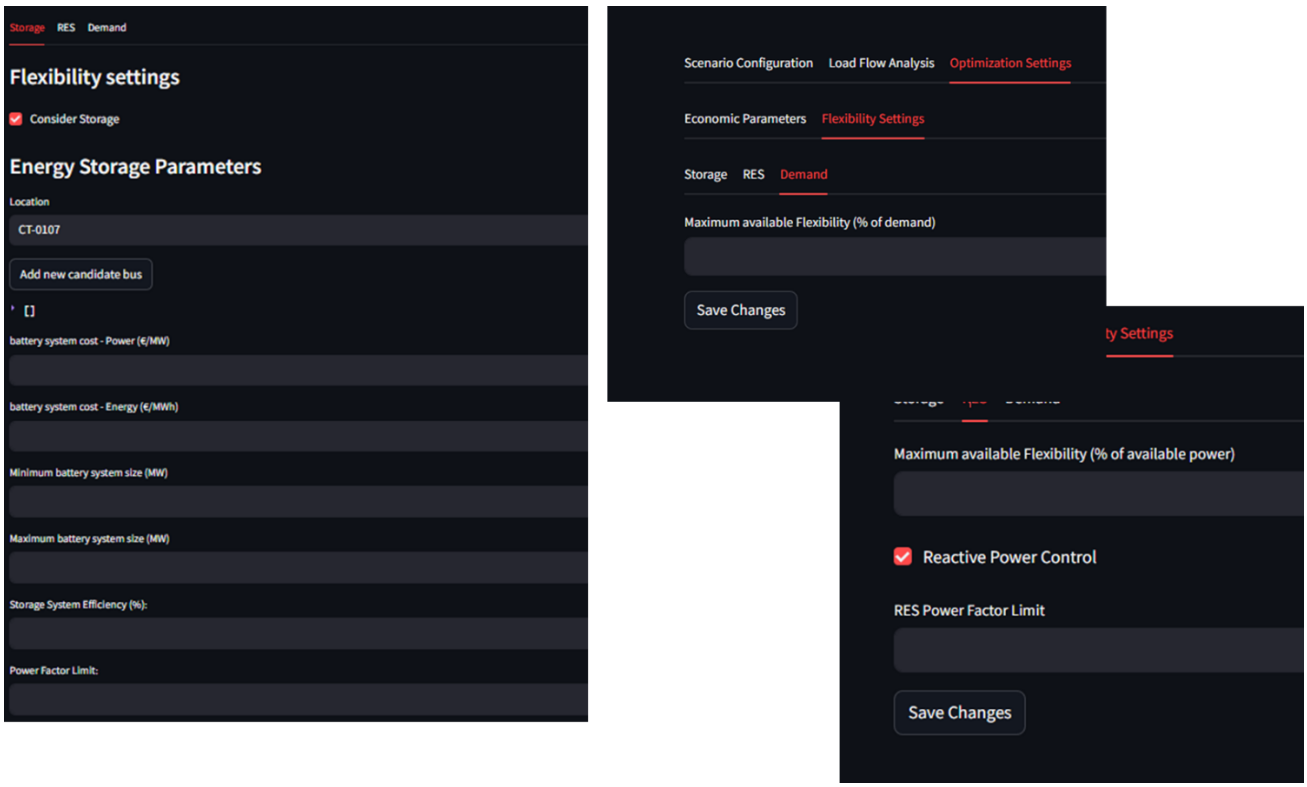

Figure 31: Flexibility settings selection

The last tab, Optimization Results (Figure 32-Figure 35), is presented after the optimization settings are set and the optimization scenarios are calculated. As shown in the following figures the user can select the optimization goal among the following (cost reduction, investment deferral and maximize RES). By clicking on the "run optimization" button the GUROBI solver solves the specific optimization problem and presents in a tabular format the upgrades selected (which line, what type should be installed and at what year). It can be seen from Figure 32 and Figure 33 for example that the investment deferral option selects to postpone by two years the update of the same line with the same type, compared to the cost reduction scenario in the Spanish pilot.

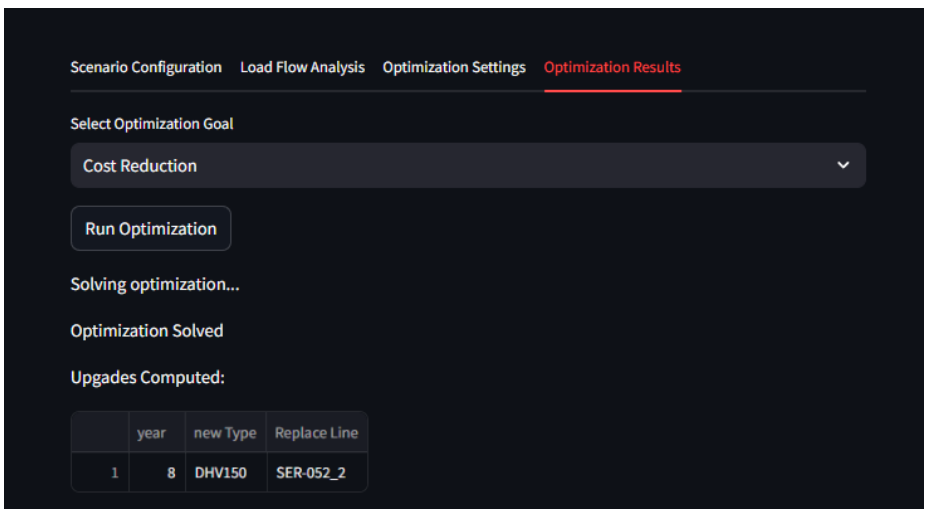

Figure 32: Optimization results tab with cost reduction selection

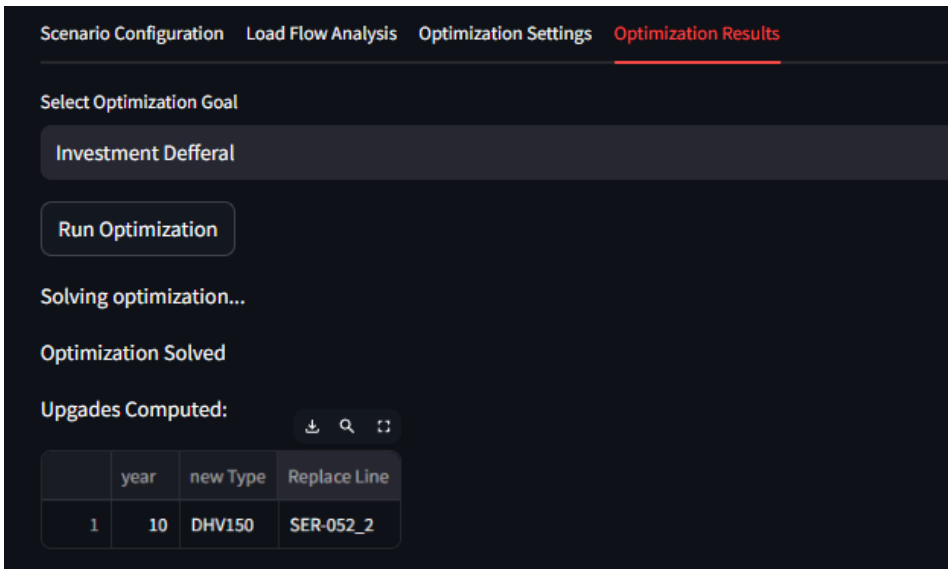

Figure 33: Optimization results tab with investment deferral selection

Apart from the tabular data of the upgrades, a bar plot is presented with the costs of the analysis. It contains the different categories of costs and the net present value (NPV) calculated regardless the optimization goal. For example, in Figure 34 the analysis in the cost reduction is presented showing when the upgrade in the line takes place and the impact it has on the power losses cost. The user can also click to the cost type on the right to disable the view of specific costs (e.g. in Figure 35) where only the flexibility costs are presented which take place in year 9 to postpone the investment for year 10. By presenting the NPV in each scenario, the user can investigate if how much are the worst-case costs (if the worst case load growth rate is selected for example by the user) could be in a 'wait and see" policy, i.e. to wait to perform the investment and utilize flexibility.

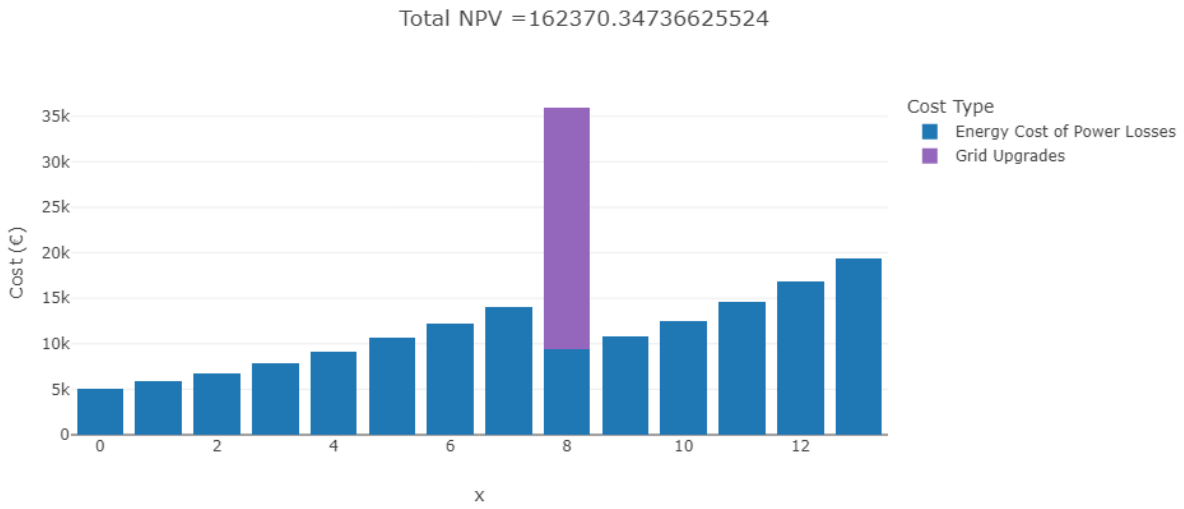

Figure 34: Cost analysis bar plot with cost reduction selection

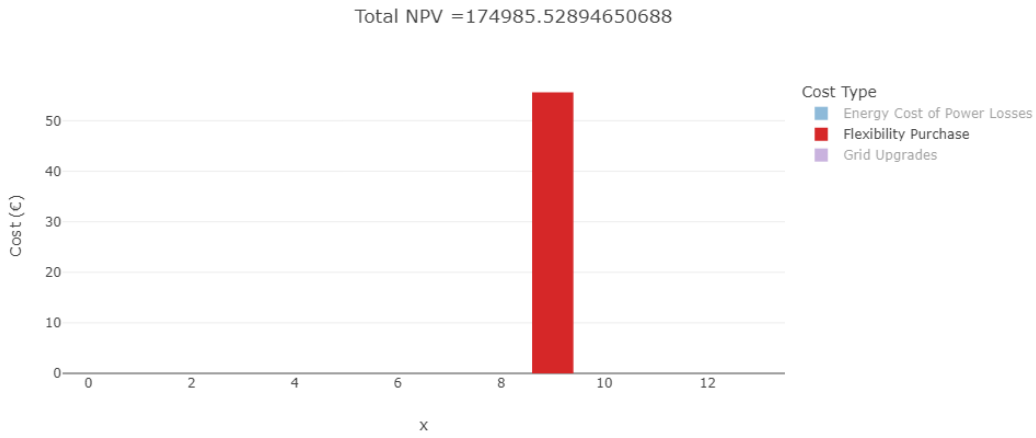

Figure 35: Cost analysis bar plot with investment deferral selection

Finally, if the user selects the "maximize RES" option in optimization goals the worst scenario is computed in the current year and a budget limit is requested from the user to constraint the number of upgrades that can be considered. Based on these the optimal upgrades will be presented as well as the maximum RES capacity according to the current limits in the lines and voltages in the bases. In addition, the whole year is executed using the RES capacity computed in every secondary substation and presents the results in map form as presented in Figure 36.

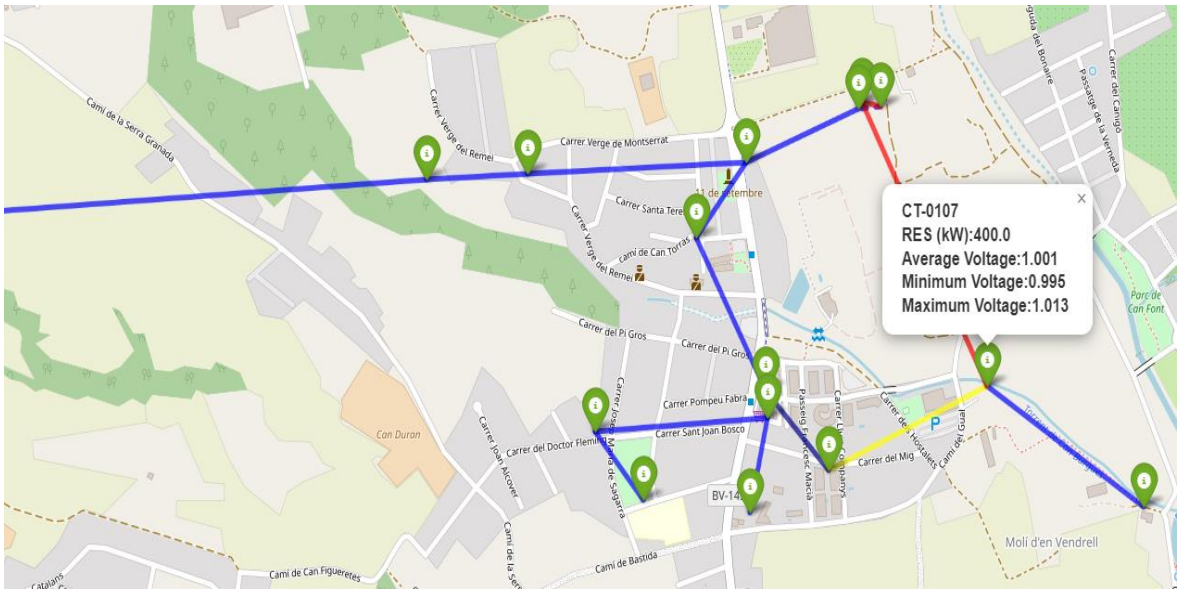

Figure 36: Power Flow results on Maximum RES scenario.

5 CONCLUSIONS

This document emphasizes on the technical details of the modules developed under Task 5.4, and Task 5.5. It includes a description of each module's design and implementation, along with initial mock-ups of the user interface. In the subsequent deliverable, D5.4 "OPENTUNITY Asset and Planning Developments (v2)," the focus will shift to showcasing the final functionalities and user interface of the modules, without revisiting the underlying design and implementation details.

The modules described are short term asset management, long term asset management, non-technical losses detection and network planning. The first three are linked with Task 5.4 and the fourth with Task 5.5.

For every module a description of the tool importance is presented, as well as the specified module design, a description of the modules developed and the tools and libraries used and preliminary mock-ups of the user interface. In the next version of the deliverable the user interface will be further highlighted.

The next steps, that will lead to the final tool development and its reporting in the next version of the deliverable, will mainly focus on the development of the necessary interfaces that will allow more seamless integration of system operators data to the modules, the re-evaluation of the modules, if needed, with additional historical data from the system operators and the development of a user interface that allows seamless use of the tools.

6 REFERENCES AND ACRONYMS

6.1 References

[1] «1st Report on mapping progress in energy systems research and innovation ETIP SNET»

[2] Dong X, Jing Z, Dai Y, Wang P, Chen Z. Failure Prediction and Replacement Strategies for Smart Electricity Meters Based on Field Failure Observation. Sensors (Basel). 2022 Dec 14;22(24):9804. doi: 10.3390/s22249804.

[3] Wang, J. *et al.* (2022) 'Smart meter life prediction method based on Health Index and Random Forest', *Journal of Physics: Conference Series*, 2384(1), p. 012058. doi:10.1088/1742-6596/2384/1/012058.

[4] L. Kang, B. Li, J. Liu, E. Wang, X. Fang and D. Huang, "Health Analysis Method of Batch Smart Electricity Meter Based on K-Nearest Neighbor Algorithm," *2021 CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS)*, Chengdu, China, 2021, pp. 1-5, doi: 10.1109/SAFEPROCESS52771.2021.9693646

[5] https://scipy.org/

[6] https://pandas.pydata.org/

[7] https://xgboost.readthedocs.io/en/stable/#

[8] Mateus BC, Farinha JT, Mendes M. Fault Detection and Prediction for Power Transformers Using Fuzzy Logic and Neural Networks. *Energies*. 2024; 17(2):296. https://doi.org/10.3390/en17020296

[9] IEEE Guide for the Interpretation of Gases Generated in Mineral Oil-Immersed Transformers," in *IEEE Std C57.104-2019 (Revision of IEEE Std C57.104-2008)* , vol., no., pp.1-98, 1 Nov. 2019, doi: 10.1109/IEEESTD.2019.8890040.

[10] Jiao, F. et al. (2024) 'A causal reasoning approach for power transformer failure diagnosis', *Frontiers in Energy Research*, doi:10.3389/fenrg.2024.1340421.

[11] A. J. Brown, V. M. Catterson, M. Fox, D. Long, and S. D. J. McArthur, "Learning Models of Plant Behavior for Anomaly Detection and Condition Monitoring," Engineering Intelligent Systems, vol. 15, Jun. 2007.

[12] https://www.omicronenergy.com/en/solution/dissipation-power-factor-capacitance-measurement-on-power-transformers/

[13] IEEE Standard for Performance Characteristics and Dimensions for Power Transformer and Reactor Bushings," in *IEEE Std C57.19.01-2017 (Revision of IEEE Std C57.19.01-2000)* , vol., no., pp.1-30, 12 July 2018, doi: 10.1109/IEEESTD.2018.8410922.

[14] https://www.ofgem.gov.uk/sites/default/files/docs/2021/04/dno_common_network_asset_indices_methodology_v2.1_final_01-04-2021.pdf

[15] https://www.tensorflow.org/

[16] Messinis, G.M.; Hatziargyriou, N.D. Review of non-technical loss detection methods. Electr. Power Syst. Res. 2018, 158, 250–266

[17] Andrey Pazderin, Firuz Kamalov, Pavel Y. Gubin, Murodbek Safaraliev, Vladislav Samoylenko, Nikita Mukhlynin, Ismoil Odinaev and Inga Zicmane, Data-Driven Machine Learning Methods for Nontechnical Losses of Electrical Energy Detection: A State-of-the-Art Review, Energies 2023, 16, 7460

[18] Z. Zheng, Y. Yang, X. Niu, H. -N. Dai and Y. Zhou, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," in IEEE Transactions on Industrial Informatics, vol. 14, no. 4, pp. 1606-1615, April 2018, doi: 10.1109/TII.2017.2785963.

[19] Henriques, H.O. et al. (2020) 'Monitoring technical losses to improve non-technical losses estimation and detection in LV Distribution Systems', Measurement, 161, p. 107840. doi:10.1016/j.measurement.2020.107840.
[20] Georgilakis, P.S. and Hatziargyriou, N.D. (2015) 'A review of Power Distribution Planning in the modern power systems era: Models, methods and future research', Electric Power Systems Research, 121, pp. 89–100. doi:10.1016/j.epsr.2014.12.010.
[21] de Lima, T.D. et al. (2024) 'Modern distribution system expansion planning considering New Market Designs: Review and Future Directions', Renewable and Sustainable Energy Reviews, 202, p. 114709. doi:10.1016/j.rser.2024.114709.
[22] Photovoltaic Geographical Information System (PVGIS). https://joint-research-centre.ec.europa.eu/photovoltaic-geographical-information-system-pvgis_en
[23] Duque, E.M. et al. (2024) 'Tensor power flow formulations for multidimensional analyses in distribution systems', International Journal of Electrical Power & Energy Systems, 162, p. 110275. doi:10.1016/j.ijepes.2024.110275.
[24] https://numba.pydata.org/
[25] Tavenard, R. et al. Tslearn, a machine learning toolkit for Time Series Data, Journal of Machine Learning Research. Available at: https://jmlr.org/papers/v21/20-091.html
[26] https://pypi.org/project/gurobipy/
[27] T. Ding, R. Lu, Y. Yang and F. Blaabjerg, "A Condition of Equivalence Between Bus Injection and Branch Flow Models in Radial Networks," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 3, pp. 536-540, March 2020, doi: 10.1109/TCSII.2019.2916208

6.2 Acronyms

Table 6. Acronyms

Acronym	Explanation
AMI	Advanced Metering Infrastructure
DER	Distributed Energy Resource
OPF	Optimal Power Flow
SCADA	Supervisory Control And Data Acquisition
DSO	Distribution System Operator
RES	Renewable Energy Sources
ML	Machine Learning
EoL	End of Life
GUI	Graphical User Interface
SoC	State of Charge
LEM	Local Energy Market
CIM	Common Information Model