MASTER'S DEGREE IN INDUSTRIAL ENGINEERING

FINAL MASTER THESIS

FAULT ANALYSIS OF MESHED NETWORKS USING ARTIFICIAL INTELLIGENCE

Author: Alberto Santiago Salazar

Director: Adam Dysko

Co-Director: Ciaran Higgins

Madrid

August 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Fault Analysis of Meshed Networks using Artificial Intelligence

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Alberto Santiago Salazar Fecha: 14/08/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Adam Dysko Fecha: 14/08/2025

This dissertation was typeset with Word 2016. The font families used are Times New roman and Book Antiqua. Unless otherwise noted, all figures were created by the author using Microsoft PowerPoint® and Python®.
AI tools were used to enhance the clarity and quality of the writing throughout this document.

Acknowledgements

I would like to express my gratitude to my project supervisors, Adam Dysko and Ciaran Higgins, for their guidance and support throughout this work, and to my family for their encouragement and belief in me.

ANÁLISIS DE FALLOS ELECTRICOS EN REDES MALLADAS UTILIZANDO INTELIGENCIA ARTIFICIAL

Autor: Santiago Salazar, Alberto.
Director: Dysko, Adam.
Director: Higgins, Ciaran.
Entidad Colaboradora: Scottish Power Energy Networks (SPEN)

RESUMEN DEL PROYECTO

Este proyecto evalúa los árboles de decisión potenciados por gradientes (GBDT), las máquinas de vectores de soporte (SVM) y las redes neuronales gráficas (GNN) para la clasificación de fallos en redes de distribución de baja tensión (LV) malladas, y aplica un modelo GNN para la localización de fallos. Utilizando modelos de red reales de SP Energy Networks (SPEN) simulados en OpenDSS, se generó un dataset de mediciones de tensión para múltiples escenarios de fallos con dos niveles diferentes de penetración de tecnologías bajas en carbono (LCT). Los tres modelos de clasificación alcanzaron una precisión similar, superior al 99%, con SVM ofreciendo un mejor rendimiento computacional. En cuanto al modelo de clasificación, la precisión de la predicción top-1 (con mayor probabilidad) fue de alrededor del 65 %, aumentando hasta aproximadamente el 99 % para las top-10 predicciones con mayor probabilidad. Los modelos se validaron con datos reales de eventos de fallos de SPEN, confirmando su potencial para mejorar la gestión de fallos y la resiliencia operativa.

Palabras clave: Machine Learning (Aprendizaje automático), Graph Neural Networks (Redes Neuronales Gráficas), Gradient Boosting Decision Trees (Arboles de decisión con potenciación de gradiente), Support Vector Machines (máquinas de soporte vectorial) Low Carbon Technologies (Tecnologías Bajas en Carbono).

*Se utilizarán las siglas en inglés para referirse a las palabras clave en el texto.

1. Introducción

La digitalización del sector energético está impulsando la necesidad de herramientas avanzadas para mejorar la detección, clasificación y localización de fallos en las redes de distribución. Esto es especialmente relevante para los operadores de sistemas de distribución (DSO), como SP Energy Networks (SPEN), que buscan estrategias para gestionar los fallos de maneras más inteligentes, rápidas y fiables. La motivación de este proyecto surge de la transformación en curso de la red de distribución, impulsada por la creciente integración de tecnologías bajas en carbono (LCT), como la energía fotovoltaica (PV), los vehículos eléctricos (EV) y las bombas de calor, así como la adopción de configuraciones de redes malladas.

Las LCT se caracterizan por su papel en los esfuerzos de descarbonización y electrificación. Los sistemas fotovoltaicos generan energía renovable a partir de la luz solar, los vehículos eléctricos proporcionan un transporte alimentado por baterías con capacidad de recarga que puede actuar como recurso energético distribuido (DER), y las bombas de calor ofrecen una calefacción eléctrica eficiente. Estas tecnologías se están adoptando debido a los cambios normativos, los objetivos medioambientales y el

impulso de la independencia energética, lo que está dando lugar a altos niveles de penetración en las redes modernas.

Sin embargo, las LCT plantean retos importantes, como patrones de flujos de potencia menos predecibles [1], desequilibrios de fase debidos a cargas desiguales, perturbaciones armónicas procedentes de interfaces basadas en inversores y perfiles irregulares de corriente de fallo. Estos factores pueden reducir la sensibilidad y la selectividad de los esquemas de protección tradicionales, lo que puede dar lugar a fallos no detectados o disparos innecesarios.

Paralelamente, se están utilizando cada vez más en redes de baja tensión (BT) configuraciones malladas para mejorar la flexibilidad, la redundancia y la resiliencia de la red. A diferencia de las redes radiales tradicionales con flujo de potencia unidireccional desde la fuente a la carga, las topologías malladas cuentan con múltiples rutas interconectadas, lo que permite que la potencia fluya en ambos sentidos y proporciona rutas alternativas durante las interrupciones o el mantenimiento. Esta configuración mejora la continuidad del servicio, pero complica el análisis de fallas. Entre los principales retos se encuentran los flujos de energía bidireccionales que suponen un desafío para los relés de protección direccionales, los niveles más altos de cortocircuito debido a la reducción de la impedancia equivalente (según la ley de Ohm, donde una impedancia más baja da lugar a corrientes más altas) y el posible estrés térmico en los conductores y dispositivos. A medida que las redes BT malladas se vuelven más comunes en las zonas urbanas, sus impactos requieren métodos de diagnóstico avanzados para mantener la fiabilidad y permitir una respuesta rápida ante fallos.

2. Definición del proyecto

El objetivo del proyecto es desarrollar modelos de IA/ML para el análisis de fallos, con el fin de evaluar la viabilidad para su implementación por parte de los DSO. Los tres modelos que han sido desarrollados para la detección y clasificación de fallos son los árboles de decisión potenciados por gradientes (GBDT), las máquinas de vectores de soporte (SVM) y las redes neuronales gráficas (GNN), mientras que el modelo desarrollado para la localización de fallos es otro modelo GNN. Estos algoritmos se han analizado teniendo en cuenta la precisión del modelo, la facilidad de implementación, la complejidad computacional y la escalabilidad.

Los modelos desarrollados se han entrenado y evaluado en dos escenarios diferentes: uno similar al estado actual de la red y otro con una alta integración de LCT, ambos basados en modelos reales de redes de baja tensión malladas proporcionados por SPEN, lo que garantiza unas características eléctricas y topológicas realistas.

Para validar los modelos más allá de los datos simulados, también se han utilizado datos reales de eventos de fallos proporcionados por SPEN para probar los modelos GBDT y SVM. Esta comparación permite evaluar la utilidad práctica de los modelos, su capacidad de generalización a condiciones reales y su potencial para su futura integración en los sistemas de gestión de fallos de los DSO.

El análisis de todos los modelos permitirá determinar si las técnicas de IA/ML pueden mejorar los marcos de detección de fallos existentes y respaldar estrategias de funcionamiento de la red más eficientes y automatizadas en el contexto de las redes inteligentes.

3. Metodología

La metodología del proyecto tuvo la siguiente estructura: primero se modelaron los escenarios LCT, luego se realizaron las diferentes simulaciones para obtener los datasets en OpenDSS, que es un software que resuelve flujos de potencia desarrollado por EPRI [2], y finalmente se entrenaron los modelos ML en Python.

Modelización de los dos escenarios LCT

Dos escenarios fueron modelizados para evaluar la robustez de los modelos en las condiciones actuales y futuras de la red. El escenario de referencia (escenario 1) representa la red existente sin LCT adicionales. A todas las cargas se les asignó un perfil de consumo residencial típico, con picos durante el mediodía y por la tarde, y una demanda nocturna baja. Esta configuración evalúa la precisión del modelo en redes tradicionales. El escenario de alta penetración de LCT (escenario 2) proyecta operaciones futuras con una adopción significativa de LCT. Los sistemas fotovoltaicos fueron modelados como dos grandes plantas fotovoltaicas, mientras que para modelar el efecto de los vehículos eléctricos y las bombas de calor se modificaron los perfiles de carga. Las bombas de calor, con una penetración prevista del 40 % en los hogares para 2030 en Europa y Reino Unido [3], añaden una carga base continua de 0,4 por unidad (pu) para mantener una temperatura interior constante, mientras que los vehículos eléctricos, con una penetración del 30 %, modifican la demanda nocturna mediante estrategias de recarga inteligente, como las tarifas por tiempo de uso, que aplanan la curva de demanda. La forma de la carga resultante muestra picos y valles alterados, lo que permite analizar la resiliencia del modelo en redes en evolución.

Simulación para obtener los datos

Las simulaciones se realizaron utilizando OpenDSS en modo diario, con 48 intervalos de tiempo cada 24 horas para capturar las variaciones horarias. La única magnitud que fue guardada fue el voltaje RMS. Las fallas modeladas incluyeron faltas fase-tierra (SLG), faltas fase-fase (LL) y cortocircuitos trifásicos. Para realizar las simulaciones se utilizó un enfoque de Monte Carlo, en el que las fallas se colocaron aleatoriamente en los 341 buses de la red y además se aplicaron resistencias de cortocircuito aleatorias que oscilaban entre 0,01 y 1 Ω para simular faltas con distinta impedancia, generando conjuntos de datasets etiquetados para el entrenamiento del modelo.

Modelización de los modelos de clasificación y localización de fallos

Todos los modelos se implementaron en Python utilizando bibliotecas como scikit-learn para GBDT y SVM, y TensorFlow con Spektral para GNN. Los datasets se dividieron en conjuntos de entrenamiento y prueba con una proporción de 80/20 para evaluar el rendimiento del modelo. Para cada algoritmo (GBDT, SVM y GNN), se entrenaron dos modelos: uno para el escenario LCT 1 (red actual) y otro para el escenario LCT 2 (alta integración de LCTs). Los dos modelos GNN (uno para la clasificación de fallos y otro para la localización de fallos), fueron modelados de manera que aprovecharan la topología gráfica de la red. Tras la simulación se calculó la precisión de todos los

modelos utilizando el conjunto de datos de prueba. Además, los modelos GBDT y SVM fueron probados con datos reales de eventos de fallos proporcionados por SPEN para evaluar su rendimiento en condiciones reales.

4. Resultados

Todos los modelos de clasificación (GBDT, SVM y GNN) alcanzaron índices de precisión superiores al 99 % en ambos escenarios de LCT, lo que demuestra la solidez de estos métodos para el análisis de fallos en las redes actuales y su adaptabilidad a las futuras transformaciones de la red impulsadas por la integración de LCT. La precisión exacta de los modelos de clasificación se puede ver en la *Tabla 1*. En cuanto al rendimiento computacional, El modelo de SVM demostró ser el más eficiente, completando el entrenamiento en aproximadamente 5 minutos, seguido de GNN con 31 minutos y GBDT con 41 minutos.

Classification Model	Baseline Scenario	High LCT penetration Scenario
GBDT	99.82	99.76
SVM	99.76	99.73
GNN	99.71	99.74

Tabla 1 – Precisión de los modelos de clasificación (%)

Las pruebas con los datos reales de SPEN demostraron que el modelo SVM clasificó eficazmente la mayoría de las muestras con alta precisión, mientras que GBDT no funcionó adecuadamente. Esto sugiere que SVM podría ser la mejor opción para la implementación práctica, aunque los datos reales eran bastante limitados y solo se disponía de un conjunto de lecturas de Smart meters muy reducido, lo que podría explicar por qué GBDT no funcionó bien.

El modelo de localización de fallos GNN, también obtuvo resultados prometedores, ya que en ambos escenarios LCT, la precisión de la predicción *top-1* superó el 60 %, y esta cifra aumentó a más del 98 % para las predicciones *top-10*, lo que indica una gran capacidad para localizar con precisión las ubicaciones de los fallos dentro de un rango reducido de candidatos. La siguiente figura muestra las predicciones *top-10*, donde se observa cómo todas las ubicaciones predichas están concentradas en un área muy cercana al lugar real de la falla. Esta agrupación demuestra claramente la eficiencia del modelo a la hora de aprovechar la estructura gráfica de la red para comprender la propagación de las fallas y las relaciones topológicas.

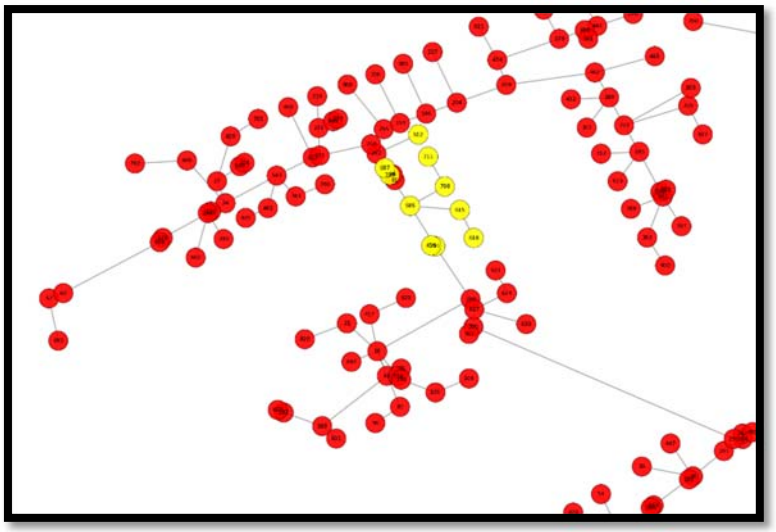

Figura 1 – Predicciones Top-10

5. Conclusiones

La alta precisión de los modelos desarrollados confirma su validez como herramientas eficaces para la detección y clasificación de fallos, adecuadas tanto para las condiciones actuales como futuras de la red. Entre los modelos de clasificación, destaca el SVM por su precisión comparable a la del GBDT y el GNN cuando se prueba con datos simulados, combinada con un coste computacional significativamente menor, lo que lo posiciona como una opción práctica para su implementación operativa. Los modelos de localización de fallos basados en GNN también ofrecieron resultados prometedores, ya que brindan a los DSO la posibilidad de mejorar la fiabilidad de la red al permitir una eliminación más rápida de los fallos mediante la identificación precisa del bus defectuoso, al tiempo que utilizan eficazmente la estructura gráfica de la red para mejorar la precisión de la localización.

En general, la integración de estos enfoques de aprendizaje automático en los sistemas de gestión de fallas de las redes de distribución podría aportar mejoras significativas, como la aceleración de la detección de fallas, la mejora de la clasificación para intervenciones específicas y una mayor precisión en la localización, especialmente en redes de baja tensión mallada, donde los métodos convencionales tienen limitaciones. Los próximos pasos deberían centrarse en ampliar el conjunto de características utilizadas en los modelos (por ejemplo, incorporando datos de corrientes y meteorológicos), ampliar la gama de tipos de fallos y perfeccionar los procesos de recopilación de datos para facilitar la implementación práctica.

6. Referencias

[1] N. Damianakis, G. R. C. Mouli and P. Bauer, “Grid impact of photovoltaics, electric vehicles and heat pumps on distribution grids — An overview,” Applied Energy, vol. 380, no. 125000, 2025.

[2] Electric Power Research Institute (EPRI), “OpenDSS Documentation,” 2024. [Online]. Available: https://opendss.epri.com/OpenDSSCustomScripting.html.

[3] National Infrastructure Commission (NIC), “National Infrastructure Assessment,” 2018.

FAULT ANALYSIS OF MESHED NETWORKS USING ARTIFICIAL INTELLIGENCE

Author: Santiago Salazar, Alberto.
Supervisor: Dysko, Adam.
Supervisor: Higgins, Ciaran.

Collaborating Entity: Scottish Power Energy Networks (SPEN)

ABSTRACT

This project evaluates Gradient Boosting Decision Trees (GBDT), Support Vector Machines (SVM), and Graph Neural Networks (GNN) for fault classification in meshed Low Voltage (LV) distribution networks, and applies a GNN model for fault location. Using real SP Energy Networks (SPEN) network models simulated in OpenDSS, a comprehensive dataset of voltage measurements was generated for multiple fault scenarios under two different Low Carbon Technology (LCT) penetration levels. The three classification models achieved similar accuracy, with SVM offering superior computational performance. For classification, the accuracy for the top-1 prediction was around 65%, increasing to approximately 99% for the top-10 predictions. The methodology was validated with real SPEN fault event data, confirming its potential to enhance fault management and operational resilience.

Keywords: Machine Learning, Fault location, Fault Classification, Graph Neural Networks, Distribution Networks.

1. Introduction

The digitalization of the power sector is driving the need for advanced tools to enhance fault detection, classification, and location in distribution networks. This is particularly relevant for Distribution System Operators (DSOs) like SP Energy Networks (SPEN), who seek smarter, faster, and more reliable fault management strategies. The motivation for this project arises from the ongoing transformation of the distribution grid, driven by the increasing integration of Low Carbon Technologies (LCTs) such as photovoltaics (PV), electric vehicles (EVs), and heat pumps, as well as the adoption of meshed network configurations.

LCTs are characterized by their role in decarbonization and electrification efforts. PV systems generate renewable energy from sunlight, EVs provide battery-powered transportation with charging capabilities that can act as distributed energy resources, and heat pumps offer efficient electric heating. These technologies are being adopted due to regulatory shifts, environmental goals, and the push for energy independence, leading to high penetration levels in modern grids.

However, LCTs introduce significant challenges, including less predictable power flow patterns [1], phase imbalances from uneven loading, harmonic disturbances from inverter-based interfaces, and irregular fault current profiles. These factors can reduce the sensitivity and selectivity of traditional protection schemes, potentially leading to undetected faults or unnecessary trips.

In parallel, meshed low-voltage (LV) networks are being used to enhance grid flexibility, redundancy, and resilience. Unlike traditional radial networks with unidirectional power flow from source to load, meshed topologies feature multiple interconnected paths, allowing power to flow bidirectionally and providing alternative routes during outages or maintenance. This configuration improves service continuity but complicates fault analysis. Key challenges include bidirectional power flows that challenge directional protection relays, higher short-circuit levels due to reduced equivalent impedance (following Ohm's Law, where lower impedance results in higher currents), and potential thermal stress on conductors and devices. As meshed LV networks become more common in urban areas and smart grid pilots, these impacts necessitate advanced diagnostic methods to maintain reliability and enable rapid fault response.

2. Project Definition

The objective of the project is to develop AI/ML models for fault analysis, with the goal of evaluating their feasibility for real-world deployment by DSOs. The three models have been developed for fault detection and classification are Gradient Boosted Decision Trees (GBDT), Support Vector Machines (SVM), and Graph Neural Networks (GNN), while the model that has been developed for fault location is another GNN model. These algorithms have been analysed considering model accuracy, ease of implementation, computational complexity, interpretability, scalability, and compatibility with real-time processing environments.

The developed models have been trained and evaluated under two different scenarios: one resembling the current state of the network and another with high LCT integration, both based on real meshed LV network models provided by SPEN, ensuring realistic electrical and topological characteristics.

To validate the models beyond simulated data, real fault event data provided by SPEN has also been used to test the GBDT and SVM models. This comparison enables the assessment of the models' practical utility, their capacity to generalize to real-world conditions, and their potential for future integration into DSO fault management systems.

The analysis of all the models will allow to determine whether AI/ML techniques can enhance existing fault detection frameworks and support more efficient and automated network operation strategies in the context of smart grids.

3. Methodology

The project methodology had the following structure, first the LCT scenarios were modeled, then the different simulations to obtain the datasets were done in OpenDSS, which is a power flow solver developed by EPRI [2] and finally the ML models were trained in Python.

Modeling of the Two LCT Scenarios

Two scenarios were modeled to evaluate model robustness under current and future grid conditions. The baseline scenario (Scenario 1) represents the existing network without additional LCTs. All loads were assigned a typical residential consumption profile, featuring morning and evening peaks corresponding to startup activities and home usage, with low nighttime demand. This configuration assesses model accuracy in traditional networks. The high LCT penetration scenario (Scenario 2) projects future operations

with significant LCT adoption. PV systems were modeled as two large plants, while EVs and heat pumps modified load profiles to create bidirectional and complex power flows. Heat pumps, projected at 40% household penetration by 2030 in Europe and the UK [3], add a continuous 0.4 per unit (pu) base load for consistent indoor temperatures. EVs, at 30% penetration, concentrate demand during nighttime via smart charging strategies like time-of-use tariffs, flattening the curve. The resulting load-shape shows altered peaks and valleys, enabling analysis of model resilience in evolving grids.

Simulation to Obtain the Data

Simulations were conducted using OpenDSS interfaced with Python in daily mode, with 48 timesteps to capture hourly variations. Only the RMS voltage was recorded as the primary data feature. The modeled faults included single-line-to-ground (SLG), line-to-line (LL) and three-phase short-circuits. In order to do the simulations a Monte Carlo approach was used, where faults were placed randomly across the network's 341 buses. Random fault resistances ranging from 0.01 to 1 Ω were applied to simulate a range of impedance conditions, generating labeled datasets for model training.

Modeling of the Classification and Fault Location Models

All models were implemented in Python using libraries such as scikit-learn for GBDT and SVM, and TensorFlow with Spektral for GNN. Datasets were split into training and test sets with an 80/20 ratio to evaluate model performance. For each algorithm (GBDT, SVM, and GNN), two models were trained: one for LCT Scenario 1 (baseline) and another for LCT Scenario 2 (high LCT integration). The GNN models included one for fault classification and a separate one for fault location, leveraging the network's graph topology. Accuracy was calculated for all models using the test dataset. Additionally, the GBDT and SVM models were tested with real fault event data provided by SPEN to assess their performance on actual network conditions.

4. Results

All classification models—Gradient Boosted Decision Trees (GBDT), Support Vector Machines (SVM), and Graph Neural Networks (GNN)—achieved accuracy rates above 99% across both LCT scenarios, proving the robustness of these methods for fault analysis in current networks and their adaptability to future grid transformations driven by LCT integration. The exact accuracy of the classification models can be seen in the table below. Regarding computational performance, SVM proved to be the most efficient, completing training in approximately 5 minutes, followed by GNN at 31 minutes, and GBDT at 41 minutes.

Classification Model	Baseline Scenario	High LCT penetration Scenario
GBDT	99.82	99.76
SVM	99.76	99.73
GNN	99.71	99.74

Table 1 – Classification Model Accuracy (%)

Testing with real SPEN data showed that SVM effectively classified the majority of samples with high accuracy, whereas GBDT failed to perform adequately. This suggests SVM might be the better for practical implementation, though the real data was pretty

limited and only a subset of smart meter readings were available, which might explain why GBDT didn't perform well.

The GNN fault location model, had promising results. For both LCT scenarios, the top-1 prediction accuracy surpassed 60%, with this figure rising to over 98% for top-10 predictions, indicating a strong capability to pinpoint fault locations within a narrow range of candidates. Visualization of the top-10 predictions in the following figure revealed a tightly clustered pattern, with all predicted locations concentrated in a very close area around the actual fault site. This clustering strongly demonstrates the model's efficiency in leveraging the network's graph structure to understand fault propagation and topological relationships.

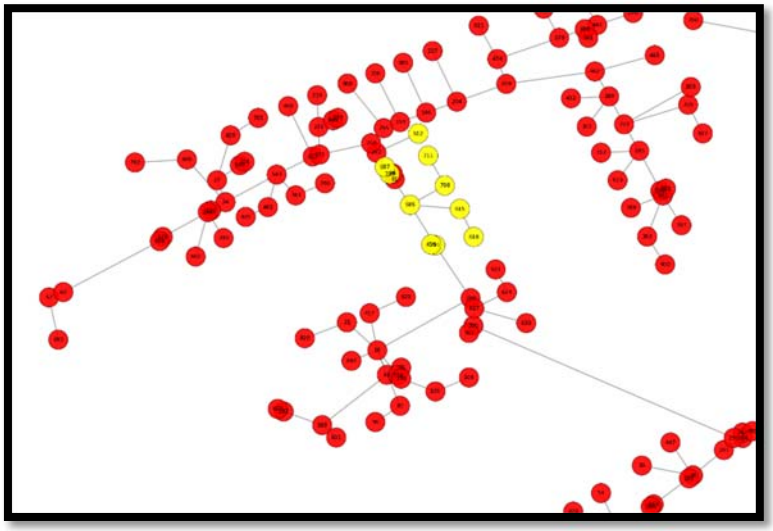

Figure 1 – Top-10 Location Model Predictions

5. Conclusions

The high accuracy of the developed models confirms their reliability as effective tools for fault detection and classification, suitable for both present and future network conditions. Among the classification models, SVM stands out due to its comparable accuracy to GBDT and GNN when tested with simulated data, combined with a significantly lower computational cost, positioning it as a practical choice for operational deployment. The GNN-based fault location models also delivered impressive results, offering DSOs the potential to enhance reliability by enabling faster fault clearance through precise identification of the faulted bus, while effectively utilizing the network's graph structure to improve location accuracy.

Overall, integrating these machine learning approaches into distribution network fault management systems could yield significant improvements, including accelerated fault detection, enhanced classification for targeted interventions, and greater location precision, particularly in meshed LV networks where conventional methods encounter limitations. Future efforts should focus on expanding the feature set (e.g., incorporating currents and weather data), broadening the range of fault types, and refining data collection processes to facilitate practical implementation.

6. References

[1] N. Damianakis, G. R. C. Mouli and P. Bauer, “Grid impact of photovoltaics, electric vehicles and heat pumps on distribution grids — An overview,” Applied Energy, vol. 380, no. 125000, 2025.

[2] Electric Power Research Institute (EPRI), “OpenDSS Documentation,” 2024. [Online]. Available: https://opendss.epri.com/OpenDSSCustomScripting.html.

[3] National Infrastructure Commission (NIC), “National Infrastructure Assessment,” 2018.

Project Index

Figure Index

Table Index

1 INTRODUCTION

This Master's thesis, focuses on the design, development, and analysis of artificial intelligence (AI) and machine learning (ML) tools to help detect and diagnose electrical faults in distribution networks. The project is aligned with the ongoing digitalization of power sector and meet the needs of Distribution System Operators (DSOs) such as SP Energy Networks (SPEN), who are trying to develop smarter, faster and more reliable approaches to manage electrical faults.

The motivation of this project comes from the ongoing transformation of the distribution grid, which due to the increasing integration of Low Carbon Technologies (LCTs) such as photovoltaics (PV), electric vehicles (EVs), and heat pumps, are being reshaped and need new ways of managing their effects in electrical faults. The high penetration and adoption of these technologies is caused by regulatory shifts, decarbonization goals and the electrification of heat and transport. Their implementation brings many advantages, like the reduction of greenhouse gas emissions and the support to energy independence, however, these new technologies introduce technical challenges that affect traditional protection and monitoring systems by introducing complexities such as fluctuating fault levels, phase imbalances, harmonic disturbances, and irregular fault current profiles. All of these factors, can affect the reliability of conventional protection schemes by reducing their sensitivity and selectivity. At the same time, meshed low-voltage network configurations are being explored by DSOs to improve flexibility, redundancy, and resilience. Unlike radial networks, meshed systems have multiple current paths for a same load and can introduce bidirectional flows, which complicates fault analysis. The detection, classification, and localization of faults in these networks is more challenging due variable current directions and higher short-circuit levels. As meshed LV networks become more common in urban areas and smart grid pilots, developing advanced diagnostic methods becomes crucial for maintaining reliability and fast fault response.

The objective of the project is to develop AI/ML models for fault analysis, with the goal of evaluating their feasibility for real-world deployment by DSOs. The three models have been developed for fault detection and classification are Gradient Boosted Decision Trees (GBDT), Support Vector Machines (SVM), and Graph Neural Networks (GNN), while the model that has been developed for fault location is another GNN model. These algorithms have been analysed taking into account model accuracy, ease of implementation, computational complexity, interpretability and scalability.

The developed models have been trained and evaluated under two different scenarios: one which resembles the current state of the network and another with high LCT integration, both based on meshed topologies. The training data for these models has been generated using OpenDSS, an open-source distribution system simulator that allows to adapt the load behaviour, to integrate DER, like PV systems, and model different electrical faults. The simulations have been done in real meshed LV network models provided by SPEN, allowing the project to have realistic electrical and topological characteristics. The fault simulation has been done following a Monte Carlo approach, by injecting various types of fault in different locations of the network to generate labelled datasets for model training.

To validate the models beyond simulated data, real fault event data provided by SPEN has also been used to test the models. This comparison enables the assessment of the models' practical utility, their capacity to generalize to real-world conditions and their potential for future integration into DSO fault management systems. The analysis of all the model will allow to determine whether AI/ML techniques can enhance existing fault detection frameworks and support more efficient and automated network operation strategies in the context of smart grids.

2 THEORETICAL BACKGROUND

This section introduces and explains the main electrical concepts and machine learning techniques that have been used throughout the project. It provides a background on meshed distribution networks, the electrical faults that have been analysed, the main Low Carbon Technologies that have been included in the different scenarios and the developed ML models.

2.1 ELECTRICAL CHARACTERISTICS OF MESHED DISTRIBUTION NETWORKS

The growing demand for higher supply reliability and the increased integration of distributed energy resources is transforming distribution networks, making DSOs consider their operation in a meshed configuration. This topology allows the power to flow from the supply to the demand through different paths, allowing for redundancy and improved service continuity when a failure happens or when there is planned maintenance.

One downside of this topology is that it introduces bidirectional power flows, which complicates traditional protection schemes because the direction of the current can vary depending on the configuration if the network. As a result, overcurrent protection can lose selectivity to detect certain faults.

In addition, the short circuit levels of meshed networks tend to be higher than radial ones, since the multiple paths for the current create parallel circuits that reduce the line impedance, which causes the current to be higher following Ohm's Law (Voltage = Current * Impedance). This higher current can stress the thermal limits of conductors and protection devices, making short-circuit analysis essential in these configuration for controlling and operating the network in a secure and efficient way.

2.1.1 ANALYSED FAULTS

The faults that have been analysed in this project are short-circuits (shunt faults), which are faults that happen when electricity takes an unintended path with low resistance, causing undesired high currents that lead to equipment damage, power outages or safety hazards. This kind of faults account for 80% of power outages in distribution networks [1], which is why this project focuses in analysing these specific kind of faults.

Short circuits are typically categorized based in the number of phases that are involved and the value of the fault impedance, which determinates their behaviour and detection challenges. Depending on the number of phases, short-circuits can be classified as symmetric or asymmetric faults. Symmetric faults, which are also known as three-phase short circuits, involve all three phases of a line and are balanced, which means that the fault current has a similar value across the three phases. Asymmetric fault can be phase-to-ground, phase-to-phase or phase-to-phase-to-ground fault [2]. Depending on the value of the fault impedance, short circuits can be low-impedance or high-impedance type. Low-impedance short circuits occur when the fault path has a negligible impedance, which results in a high fault current making this kind of faults easy to detect but also causing thermal stress on equipment. Meanwhile, high-impedance short circuits occur when a conductor contacts a high resistance surface like a tree branch, concrete or dry ground, causing the fault currents to be low and difficult to detect due to the current not being high enough to trigger the protections, potentially creating safety risks and fire hazards due to prolonged overheating [3].

Each of these faults create different electrical parameters in term of voltage drops and overcurrent, whose accurate recognition can be used for detection, classification and location. Furthermore, understanding these electrical parameters helps the design of adaptive protection schemes and improves the reliability and safety of modern power systems, especially in complex grids with distributed generation where fault characteristics are less predictable.

2.2 LOW CARBON TECHNOLOGIES (LCT) AND THEIR OPERATIONAL DYNAMICS

LCT are transforming how distribution networks behave due to their effect on demand and power flows. Their deployment is a fundamental driver in the energy transition and can help reduce carbon emissions, achieving a more sustainable future. LCTs include distributed generation units like PV systems, flexible consumption devices like EVs and heat pumps and energy storage systems. These technologies change the way that networks are operated since they introduce variability in both supply and demand due to their decentralized nature, making the prediction of power flows and voltage control a more complicated task. In addition, LCTs are connected to the grid through inverters, which can introduce harmonics and phase imbalances if not connected in an efficient way. Because of all these characteristics, data driven systems are needed in distribution grids to adapt to these changing operating conditions introduced by LCTs.

In the following subsections, the operational characteristics and specific impacts of three representative LCTs (photovoltaic systems (PV), electric vehicles (EVs), and heat pumps (HPs)) will be analysed in more detail, as they are the focus of this study's simulation scenarios.

2.2.1 PHOTOVOLTAIC (PV) SYSTEMS

Photovoltaic (PV) systems are installations that consist of solar panels, which contain photovoltaic cells that convert sunlight into direct current (DC) electricity, inverters that transform DC into alternating current (AC) suitable for grid use and mounting structures to position panels for optimal sunlight exposure. As a key low-carbon technology, PV integration supports sustainable energy goals but introduces challenges like reverse power flow, voltage fluctuations, and protection coordination, requiring innovative grid management strategies.

The widespread adoption of PV systems at the consumer level introduces distributed generation at the far end of distribution feeders. This can lead to reverse power flows during high generation and low load scenarios. This can cause the feeder voltage to rise, due to the current flowing backwards causing a voltage rise through the line impedance as seen in the formula $V_{end} = V_{substation} + I * Z$. The intermittent nature of PV output is another concern, since cloud movement can cause the power output of the installations to vary rapidly and the voltage of the network to fluctuate, complicating voltage regulation. Protection devices, such as feeder relays, must adapt to reverse current flows and distinguish normal operation from faults, often requiring directional elements. In addition, in high PV penetration scenarios, during a short-circuit the fault current can be feed by the inverters of the PV plant, which may sustain faults or delay protection timing. From a planning perspective, PV systems reduce the net demand of a network, reducing the losses due to a lower power flow, however this benefit holds only up to a penetration threshold, which is around 30-40% of households. Beyond this percentage, over-voltage and reverse flow issues increase notably [4]. In order to reduce the negative effects and increase hosting capacity, solutions like meshed feeder operation, which interconnects radial feeders to redistribute PV output in a more efficient way, and energy storage, which can absorb energy when there is an excess of PV generation, are being studied by DSOs. In summary, while PV systems are a fundamental energy source to support the energy transition, they need advanced voltage control, protection coordination, and operational strategies to manage two-way power flows and ensure grid reliability.

2.2.2 ELECTRIC VEHICLES (EVS)

Electric vehicles are transforming the transportation industry by replacing fossil fuel-powered vehicles with battery powered ones, which require charging stations to restore the battery. These stations are formed by an on-board charger that connects through a connector to a charging station that can supply single-phase or three-phase power. The power that these chargers deliver depends on if it is a home charger or if it is a public fast charger, the power that they deliver goes from 7-22 kW to 50 kW respectively. These chargers also have the ability to inject power to the network, so they can act as a load or as a DER. EVs represent

a rapidly growing load on distribution networks and it is estimated that the number of EVs will grow six times from 2021 levels by 2030 [5]. The growth of this industry can be seen in Figure 1, which shows the global sales of EV throughout the years. This growth introduces new challenges to the distribution network, like peak demand growth and voltage issues, since if many EVs charge during the evening, the traditional evening peak in residential areas can spike dramatically. Projections by distribution utilities indicate that uncontrolled EV charging could increase feeder peak loads by a large factor, which will have to be addressed with advanced charging strategies and Vehicle-to-X (V2X) technologies to ensure grid reliability.

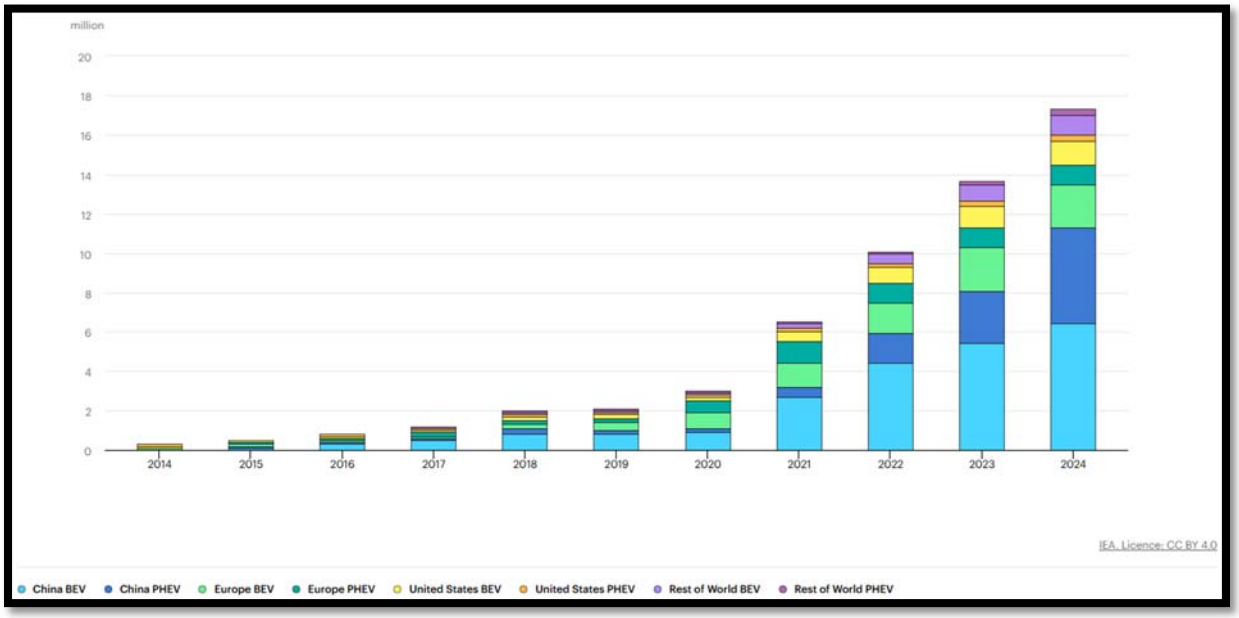

Figure 1. Global Electric Car Sales, 2014-2024 [6]

Smart charging strategies consist of different initiatives that try to coordinate the charging of EV in an efficient way, for instance, smart chargers that delay charging to off-peak overnight hours, or utility programs that signal EVs to slow/stop charging during grid stress. In fact, new regulations in the UK mandate that home EV chargers come with a default random delay function to avoid synchronized starts at the top of the hour [7]. Another initiative is (TOU) pricing, which offers lower electricity rates at night or during low demand

periods in order to encourage EV owners to charge during off-peak hours, and real-time pricing (RTP), which dynamically adjusts electricity costs based on current grid conditions, enabling more responsive load management. By using these strategies, EVs can be implemented in a more efficient way and instead of increasing the demand peaks they would increase the demand during the time where there is currently low demand, flattening the curve [8].

EVs also bring the potential of Vehicle-to-X (V2X) support, which consist of leveraging the EV battery as an energy storage system, and use bi-directional chargers to feed energy from the batteries to a load (Vehicle-to-Building V2B, Vehicle-to-Home V2H, Vehicle-to-Load V2L) or even back into the grid (Vehicle-to-Grid, V2G). If widely adopted, V2X could act as a distributed battery network, providing peak shaving or emergency power support. Some pilot projects have shown EVs stabilizing local voltage or frequency by discharging at critical moments, like for example the Parker Project, which showed that EVs successfully stabilized grid frequency by responding to automatic generation control (AGC) signals, reducing frequency deviations by up to 10% in test scenarios and also demonstrated voltage support through reactive power injection, maintaining voltage within standard limits [9]. However, V2G also means that EVs turn from just loads to distributed generators when they are injecting power to the grid, which means that protection settings, and the traditional operation of the network must adapt to these changes.

In conclusion, the integration of EVs in the distribution grid can overload the grid and increase the peak demand if it is not well managed. In order to tackle this problem, different solutions like smart charging coordination, which leverages EVs as flexible resources through V2X and infrastructure upgrades (larger transformers, additional feeders) will be needed.

2.2.3 ELECTRIC HEAT PUMPS

Electric heat pumps are LCTs that transfer thermal energy using electricity in order to provide heating and cooling for homes. They are formed by a compressor, an evaporator, a

condenser and a refrigerant cycle, with a typical electrical consumption is typically around 3.5-7 kW [10], which is higher than most of household appliances. Their deployment, driven by policy incentives and their high efficiency, is key for decarbonizing buildings that otherwise would use fossil fuel based heating. As heat pump adoption grows, distribution networks will need to find ways to manage the demand increase, necessitating strategies like demand response, capacity upgrades and innovative energy management to maintain reliability.

The demand increase of heat pumps has a seasonal pattern, since it happens in winter due to heating applications and in summer due to cooling applications, and is a demand that can last the whole day, unlike PV (which affects midday voltage) or EVs (which can be flexibly charged) so it is added to the base demand curve as an almost constant load. Studies show that a penetration of heat pumps of around 30% to 50% could lead in an increase of peak demand of 50%-100% in residential areas, which could cause conductor and transformer overloads and increased voltage drops, this was stated by Britain's National Infrastructure Commission, which forecast that electrification of heat could double peak electricity demand by 2050 [11].

From a network perspective, the introduction of heat pumps will lead to different strategies to cope with the demand. Demand response strategies, such as time-of-use (TOU) pricing or smart thermostats, can reduce the net demand by shifting the operation to off peak periods, however, customer tolerance for temperature changes is limited, so these strategies must be implemented in a way that ensures comfort for the users.

In summary, heat pump integration will significantly raise base and peak loads on distribution networks, forcing the network to handle a much higher load. This proves the need for network upgrades and intelligent load management. Combined with EVs, the electrification of heat and transport is reshaping demand curves and require distribution networks to be more robust and smarter in balancing supply, demand, and power flow.

2.3 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING TECHNIQUES

The AI/ML techniques that have been explored in this project for fault classification and localization are Gradient Boosted Decision Tree (GBDT), Support Vector Machines (SVM) and Graph Neural Network (GNN). These are all supervised learning algorithms, meaning that they use labelled dataset to learn the relationship between the input features and the target outputs [2]. Each technique has different characteristics that make it suitable for different aspects of fault analysis.

2.3.1 GRADIENT BOOSTED DECISION TREE

Gradient Boosted Decision Tree is a machine learning technique that combines multiple decision trees into an ensemble model and are typically used for regression, classification, and ranking tasks. These models are able to find complex, non-linear patterns in data and thanks to the gradient bosting, which improves predictions iteratively in order to get high accuracy. This makes them a versatile tool to model complex systems like distribution networks.

A decision tree is a ML classification algorithm that represent decisions and their possible consequences by using a tree-like structure. It is formed by internal nodes, which represent a decision based on a feature, branches, which connect the nodes and represent the possible outcomes of that decision, and leaf nodes, which represent the final output [12]. Decision trees work by splitting the input data into subsets that are based on feature thresholds which are chosen to minimize a loss function, like for example mean squared error.

GBDTs build an ensemble of decision trees sequentially and use gradient boosting, which is an optimization technique that minimizes the loss function, to improve predictions by correcting errors from the previous trees. The algorithm computes residuals of the previous trees, which are the differences between predicted and actual values, and then trains a new decision tree that reduces these residuals. This process fits each new tree in the direction of

the negative gradient of the loss function in order to ensure that each update reduces the overall error in the most effective way while maintaining model stability. By doing this, gradient boosting ensures that each tree contributes a small, targeted improvement, allowing GBDTs to capture complex, non-linear relationships and handle unbalanced datasets effectively.

GBDTs have different hyper parameters that control the models complexity and the training method:

- Number of Trees. Determines how many decision trees are included in the ensemble. Increasing the number of trees generally improves accuracy by enabling more iterations to refine predictions but it increases computational cost and can lead to overfitting.
- Tree Depth. Controls the maximum number of levels in each decision tree. Shallow trees only capture general patterns in the data and reduce overfitting, while deeper trees model finer details but have a higher risk of overfitting. The optimal depth depends on the dataset's complexity and size.
- Learning Rate. It is the parameter that scales the contribution of each tree's predictions to the ensemble. If the model has a lower learning rate, it will require more trees to train but it will improve its robustness by making smaller and more cautious updates, reducing overfitting risk. On the other side, a model with a higher rate will train faster but can be unstable. Because of this, tuning the learning rate and the number of trees is critical for optimal performance.

GBDTs combine the simplicity of decision trees with the power of gradient boosting and are very useful for complex tasks in energy systems, but they require careful tuning to avoid overfitting and ensure reliable performance.

2.3.2 SUPPORT VECTOR MACHINES

Support Vector Machines are supervised ML algorithms which are used for classification and regression, and are very effective for complex datasets. SVMs divide data in different classes with hyperplanes that are used as boundaries, being the objective to maximize the margin between hyperplanes. In addition, when data can't be separated with a hyperplane, SVMs use a kernel function which transforms the data to a higher dimensional space, allowing the classification to be done in a more precise way. By leveraging these techniques, SVMs can achieve a high accuracy and generalization and has a big potential in electrical fault detection.

Hyperplanes are defined as linear equations and depending on the dimensions of the data, the hyperplane can be a line (for 2D spaces) or a plane (for n-dimensional spaces, being *n* the number of features that the dataset has) [13]. The objective of the SVM algorithm is to find the hyperplane that has the biggest margin between the nearest data points that it is separating, which are called support vectors, and that same hyperplane, while also minimizing the loss function. By doing this the risk of misclassification data is reduced, since the margins are as wide as possible. Figure 2 shows how there is a wide area where any parallel hyperplane would separate the data in a correct way. However, the final hyperplane chosen by the SMV algorithm is the one which maximizes the margins between the support vectors of each class.

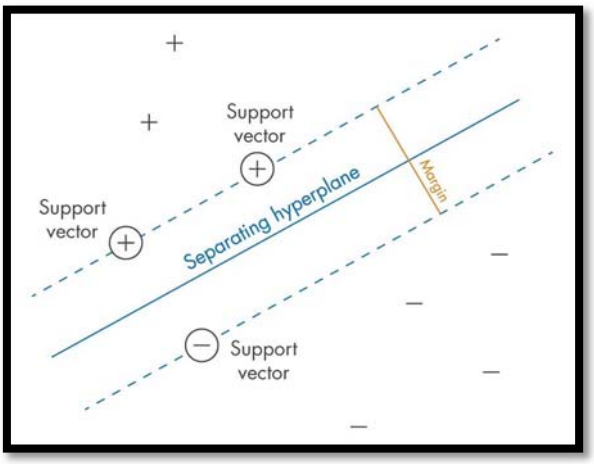

Figure 2. SVM example [13]

For non-linearly separable data, SVMs use kernels, which are mathematical functions that transform data into a higher-dimensional space where a linear boundary can be found. The most common types of kernels are linear, which is suitable when data is approximately linear, polynomial, which increases the dimensions of the feature space, Gaussian, which is widely used for its ability to handle highly non-linear data, and sigmoid, which mimics the activation function used in neural networks.

In order to improve the performance of SVM models, the following hyperparameters are used:

- Regularization Parameter (C). This parameter controls the trade-off between maximizing the margin and minimizing the error. If this parameter is high, the predictions will have better accuracy and the margins will be smaller, this can lead to overfitting, especially with noisy data. On the other hand, if C is smaller, the model will allow more misclassification, improving generalization and regularization.
- Kernel Parameters (Gamma for RBF Kernel). These parameters define the transformation of data to a higher dimensional space. Gamma is used in the RBF kernel to control the shape of the decision boundary, being a more localized and tight one if Gamma is high, and a smoother one for low gammas.

Support Vector Machines combine margin maximization with kernel functions to create robust models for classification and regression, effectively handling non-linear and complex datasets. In applications like grid fault detection, SVMs offer high accuracy and adaptability, making them a valuable tool for managing the challenges of modern distribution networks.

2.3.3 GRAPH NEURAL NETWORKS

Graph Neural Networks are a specialized class of neural networks that are able to analyse data structured in a graph, where every data point from a sample is assigned to a node which interconnects with all the other nodes based on the relation with them, represented by the links. Thanks to the use of a graph structure, these NN are able to capture dependencies and patterns in interconnected systems, which makes them a great tool for fault classification and location.

Artificial neural networks are machine learning models that are formed by nodes called artificial neurons, which are inspired in the human brain. These neurons are organized into three different layers: input layers, which receive the initial data, hidden layers, which process the data, and the output layer, which gives the predictions. When a neuron receives an input it applies a weight to determine its importance and ads a bias to shift the result. These two parameters (weight and bias) are adjustable parameters that change throughout the training of the model. In addition, in order to introduce non-linearity to the process, each neuron applies an activation function which enables the network to model complex relationships. The most common activation ones are: sigmoid, which maps any real-valued input to a value between 0 and 1; tanh, used for centered data with outputs from -1 to 1; ReLU (Rectified Linear Unit), which promotes sparsity and a faster training; and linear, which is used for regression tasks. Figure 3 shows the previously mentioned activation functions.

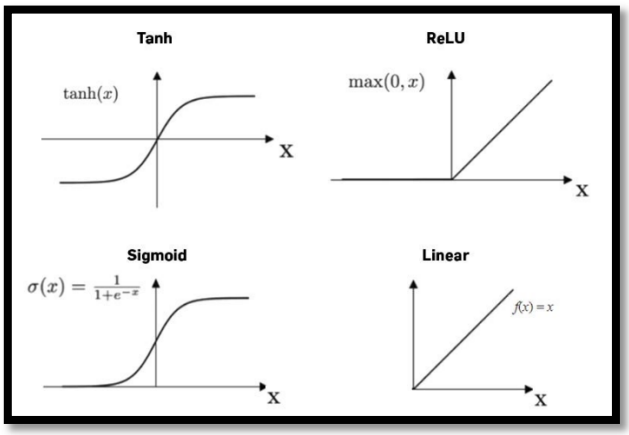

Figure 3. Activation functions [14]

Each neuron of a layer interacts with the neurons of the next layer by weighted connections, where the output of a neuron is scaled by a weight and becomes the input of networks of the next layer. The training process involves forward propagation, where the data is passed through the neurons to produce outputs, which will be compared to the real value in order to quantify the error. This error is then propagated backward through the network to update the weights and biases. NN use Stochastic Gradient Descent (SGD) to optimize the network by calculating the gradients for a small batch of data and updating the parameters in the direction that reduces the loss the most.

NN need a great volume of data to train the model, that is why, in order to manage computation, large datasets are split into smaller subsets known as batches. These batches pass through the NN sequentially in multiple iterations and a complete pass through the entire dataset constitutes an *epoch*. In order to improve the performance of the model NN train over multiple epochs.

Graph Neural Networks extend neural networks to handle graph structured data. They aggregate information from the different nodes and capture local and global graph structures. The process begins by initializing node features as input vectors, then each node aggregates information from its neighbors and combines it with its own ones and then an activation

function produces updated node representations. This is done many times, allowing nodes to get information from distant neighbors, capturing complex dependencies.

The hyperparameters that are used in GNN are the following:

- Learning rate. Controls how much the weights update during training. A higher learning rate speeds up training but can overshoot optimal weight, while a low one ensures a stable convergence but requires more epochs.
- Number of Layers. Determines how many message-passing iterations occur, affecting how far information propagates. If the number of layers is small, only local patterns will be captured, while a higher number can model global structures.
- Batch Size. Defines the number of samples that will be processed per iteration. Smaller batches improve generalization but slow training, while larger batches speed up computation but may reduce accuracy on small datasets.

The use of GNN for fault detection allows to leverage graph structure data to capture complex dependencies and patterns. By optimizing the hyperparameters, they can achieve a robust performance, being a powerful tool for improving grid reliability and efficiency.

3 STATE OF THE ART

3.1 FAULT DETECTION IN RADIAL VS. MESHED DISTRIBUTION NETWORKS

Electrical distribution networks have traditionally been operated in a radial configuration, meaning that the power flows go in one direction from distribution substations to customers following a tree-like structure in which electrical lines are separated into branches without interconnection between them. Because of that, low impedance fault detection and isolation in radial networks are relatively straightforward tasks, since, when a fault occurs and the current increases, the traditional protection devices employed by distribution companies are overcurrent relays, which can be electromechanical or numerical and can detect currents exceeding a predefined threshold initiating circuit tripping; fuses, which melt to interrupt fault currents and reclosers, which reconnect the circuit after temporary faults [15]. These devices are coordinated in cascade, which ensures that the device that trips is the closest one to the fault, minimizing the impact of the fault and maintaining service continuity for as many clients as possible. This scheme has proven effective and simple, leveraging the fact that any fault current will originate from the upstream source and flow downstream [16], however, this topology has limitations in terms of reliability and redundancy, since a fault anywhere along a feeder or a line can isolate all downstream customers until the fault is cleared. In order to tackle this problems, distribution companies are moving towards meshed or looped network configurations, especially in urban areas. In these topologies the feeders are interconnected at multiple points, allowing power to be delivered to the loads through different paths, some advantages of these systems are the reduction of power losses, improved voltage profiles due to a reduced voltage drop, more flexibility, the enhancement of power quality and the capability to deal with high electricity demand growth postponing costly investments to increase line capacity [17]. As stated in *2.1*, meshed networks also introduce significant challenges for protection design. The most immediate issue is the increase in short-circuit current levels. The interconnection of parallel lines reduces the total

short-circuit impedance compared to radial configurations, causing a current that can potentially exceed the interrupting capacity of circuit breakers and switchgear originally sized for radial conditions. If fault levels are not controlled, equipment may be damaged or fail to clear faults in time. Another major challenge is coordination and selectivity. In a radial feeder, the direction of fault current is predictable and fixed, allowing protective relays to be graded in a clear upstream–downstream sequence. In a meshed system, however, the fault current can flow in either direction, making it difficult to determine the "upstream" relay. Without directional sensitivity, overcurrent protection may trip not only the faulted section but also adjacent healthy feeders, causing widespread outages. Because of this, traditional protection schemes like overcurrent relays and fuses face big limitations.

To address the aforementioned challenges, utilities are moving towards Active Network Management (ANM) techniques. ANM may include real-time monitoring of many elements of the network such as feeder flows and voltages, electric vehicles charging stations or inverters of DERs; automated control of switches, inverters, and other actuators to maintain stability and also integrates information with the marketing control system, production management system and each substation system, allowing the acquisition and sensing, optimization management, risk control, and fault handling of all the grid [18]. For example, if a section of network experiences high reverse flow from PV and high voltage, an ANM system could command some inverters to absorb reactive power (droop control) or could reconfigure the network (close a link to route power to a heavier loaded feeder). Similarly, during high load periods, ANM can prioritize certain loads or tap available local generation/storage (including possibly discharging EVs in V2G mode) to support the network. Essentially, the presence of bi-directional flow pushes the distribution grid toward a smart grid paradigm, wherein continuous adjustments are made to accommodate the dynamic flow patterns. The goal is to ensure that even with bidirectional and unpredictable flows, the network remains within safe operating limits (voltage, thermal, fault clearance) at all times. In addition, utilities rely on specialized protection schemes to ensure reliability and selectivity like directional overcurrent relays (DOCRs) and Multi-Agent Systems (MAS)

have been introduced. DOCRs improve selectivity by considering both the magnitude and direction of fault currents, making them more suitable for networks with multiple power sources and closed loops, while MAS introduces a hierarchical protection architecture formed by relays, local controllers, and central coordination units which collaborate to achieve coordinated protection. This structure enables faster decision-making, scalability, and improved fault isolation. Another solution is differential protection, since it offers high sensitivity by comparing the current at both ends of a protected zone, however it is costly because it requires a communication infrastructure [15].

In parallel, machine learning and artificial intelligence techniques are increasingly being explored to enhance fault detection capabilities, particularly in dynamic and data-rich environments. These developments are especially relevant given the growing penetration of distributed generation and the increasing demand for reliability, which are driving interest in the 'smart meshing' of feeders. Such an evolution in network topology will inevitably require more advanced, adaptive, and data-driven fault detection and protection methods capable of handling bidirectional flows and rapidly changing system conditions.

3.2 INTEGRATION OF LOW CARBON TECHNOLOGIES (LCTS) IN DISTRIBUTION NETWORKS

Electrical distribution networks worldwide are experiencing the rapid integration of Low Carbon Technologies, which include distributed renewable generation and new electric loads that support decarbonization. Notable LCTs impacting distribution grids are photovoltaic (PV) solar panels, electric vehicles (EVs), electric heat pumps, micro-CHP (Combined Heat and Power) units, and energy storage systems (ESS). The penetration of these technologies has been increasing through the years and will keep increasing due to policy targets for carbon reduction and consumer adoption of green energy solutions. This can be proved by the fact that many countries have set ambitious goals for decarbonization, EV adoption and electrified heating. Spain's National Integrated Energy and Climate Plan (PNIEC) focuses on reducing greenhouse emissions by 32% relative to 1990 levels, achieve an 81% share of

renewable energy in electricity generation by 2030 and transition to a 100% renewable generation by 2050. Additionally, among other initiatives, the plan sets a target of 22.5 GW of energy storage capacity to support the integration of renewable energy sources [19]. In the same way, The UK's Electricity North West (ENWL) reports that penetrations of technologies like PV, EVs, and heat pumps are likely to increase significantly in the near future, affecting LV networks. According to ENWL's most recent Distribution Future Electricity Scenarios report, published in January 2025, it is expected that by 2040 the energy demand will double driven by the adoption of 3 million electric vehicles and 1.2 million heat pumps [20]. This transition means that the once passive distribution network (characterized by one-way power flow from substations to loads) is evolving into an active network with generation and new types of demand. The evolution and penetration of LCTs bring new opportunities and challenges for distribution companies. On one hand, LCTs like rooftop PV and local wind generation can supply clean energy locally, potentially reducing peak power drawn from the grid and losses. EVs and batteries can provide flexible demand or storage that might be leveraged for grid support in the future (vehicle-to-grid services, load shifting, etc.) [21]. On the other hand, these technologies were not taken into account in the original design of most distribution systems. Historically, LV feeders were built assuming all customers are consumers only, and network capacity was calculated for certain peak demand per household – with no generation or high-power new loads like EV chargers in mind. As a result, substantial LCT uptake can drive the network beyond its limits unless changes are made. The following sections detail the impacts of key LCTs – PV, EVs, and heat pumps – and the consequent technical challenges introduced by the bidirectional flows and new load patterns they create.

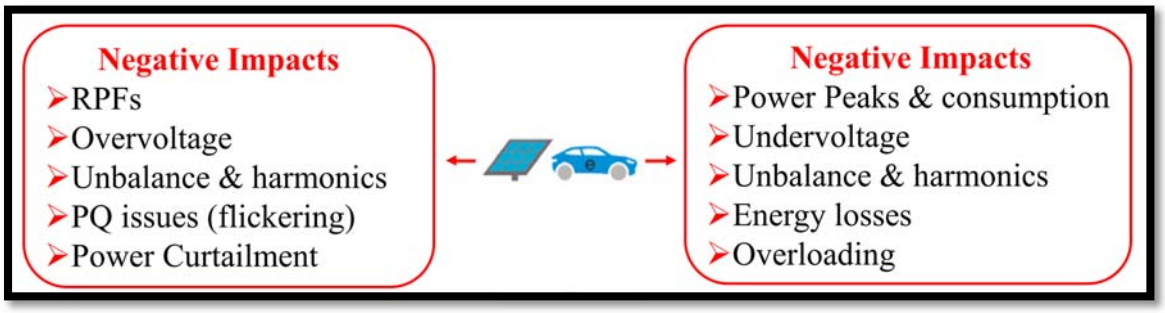

Figure 4. Summary of negative impacts of PVs & EVs [22]

3.2.1 IMPACTS OF HIGH LCT PENETRATION

The increasing adoption of LCTs can create adverse effects on distribution networks. The most common issues that have been identified are voltage regulation, thermal overloading, increased fault levels, power quality degradation and challenges to protection schemes.

- Voltage Regulation: Distributed generation like PV can cause the voltage to rise in feeders especially at times of low load and high generation, whereas concentrated new loads like EV charging clusters or heat pumps in winter can cause deeper voltage drops. Moreover, if the deployment of LCT is uneven among the different phases of the network, which is something that can happen specially with rooftop PV systems and with home EV chargers, the voltage valance can be even more compromised. That is why maintaining all customers within the required voltage range becomes more challenging and may require advanced voltage control equipment, like on-load tap changers, voltage regulators, capacitor banks or power electronic components like STATCOMS or FACTS. These technologies require advanced monitoring and control strategies. In this context, data driven approaches are crucial for maintaining voltage stability.
- Thermal Overloads: The addition of multiple LCTs demands can push line and transformer loading beyond their rated capacities particularly during evening peaks or seasonal cold periods. Likewise, cables and overhead lines may run hotter due to both increased peak currents and longer durations of high load (e.g., an EV charging

for hours), potentially leading to asset aging or failure if not addressed through network reinforcement [22].

- Fault Levels and Protection: With the deployment of many LCTs, the short-circuit current profile of the network changes, this can difficult the protection coordination, especially in meshed or bidirectional flow conditions and can lead to the development of new protection strategies. Even if inverter-based technologies like PV and HPs usually have a limited fault current contribution, a big amount of penetration in a network can still cause this effects.
- Power Quality and Losses: The high penetration of power electronic devices can degrade power quality through the injection of harmonics, voltage flicker, or rapid fluctuations in load and generation. While individual devices meet standards (IEEE 519, etc.), the aggregate effect of dozens of inverters can raise background harmonic distortion. On the losses side, a modest level of local generation can reduce net current flow and losses, but as penetration rises, network losses can actually increase due to circulating currents and periods of reverse flow that force power through more stages of transformation. Essentially, distribution networks face a more dynamic and less predictable power flow pattern as LCTs proliferate, which affects everything from component aging to the efficiency of operation [22].

Distribution operators are addressing these challenges through a combination of measures, which are the reinforcement of existing infrastructure, the deployment of smart grid elements like sensors and voltage controllers, and the implementation of demand response or time-of-use pricing in order to flatten demand peaks with the correct price signals. Additionally, the concept of hosting capacity, which is maximum amount of DERs that a given distribution network can accommodate without causing technical problems or without requiring infrastructure modifications [23], is becoming fundamental when planning new investments.

3.3 AI AND MACHINE LEARNING IN POWER SYSTEM FAULT DIAGNOSIS

The complexity that has been introduced in the networks by the meshed configurations and distributed resources has increased the interest in more intelligent fault diagnosis techniques. Better data availability due to the widespread deployment of smart meters and digital relays alongside improved computing capabilities have made possible that during the last decades Knowledge based approaches like Artificial Intelligence (AI) and Machine Learning (ML) have been increasingly applied in power system protection and fault management to complement or enhance conventional methods. These techniques have been used to achieve faster and more accurate fault detection, classification, and location, even under the challenging conditions of modern grids.

Several AI/ML approaches have been developed throughout the years, starting by expert systems in the 1980s-90s [24], which followed a set of rules derived from human experts and mimicked human operator actions for fault diagnosis and clearance. In order to manage uncertainty, fuzzy logic systems emerged and have been used for fault classification and even adaptive relay setting; these systems model partial truths rather than binary logic and handle uncertainties in fault patterns.

The most widely used AI-based algorithms for fault location are Artificial Neural Networks (ANNs) due to their ability to learn complex relationships between inputs and outputs, their flexibility and their high precision [25]. Researchers have successfully trained neural network models to recognize different fault types on transmission lines and distribution feeders, achieving high accuracy in simulations.

Beyond ANNs, many other ML techniques have been applied to electrical fault analysis. The main ones are Support Vector Machines (SVMs), which separate data by searching for a linear optimal hyperplane that acts as boundary among classes; decision tree algorithms (DTs), which have a hierarchical shape structure and classify data by asking questions about

the data features; and clustering algorithms like k-means, which determine the class of new data points based on their similarity to known training data [26]. Each of these techniques offer unique strengths in handling the dynamic conditions of the modern grids. Another common theme in AI/ML fault diagnosis is the use of signal feature extraction. Voltage and current waveforms that are captured during fault transients are typically processed through methods such as the Fast Fourier Transform (FFT), wavelet transform (WT), or more advanced time-frequency analyses (S-transform, Hilbert-Huang transform) to extract characteristic features that distinguish different fault conditions. For instance, the wavelet transform is extremely popular for analysing fault transients because it can isolate the high-frequency components induced by faults.

3.3.1 FAULT DETECTION AND CLASSIFICATION

Two key factors of fault analysis are fault detection and classification. A rapid fault detection will lead to a fast actuation of the protection systems and the clearance of the fault, protecting the system from a much worse impact in the network while a correct classification of the fault will lead to an adequate actuation to clear each specific fault. Many AI models have demonstrated promising performance in identifying and classifying different fault types:

- ANNs are particularly prevalent due to their adaptability to approximate nonlinear mappings between features and fault categories. ANN architectures have been combined with WT or FFT feature extraction in many studies in order to classify symmetrical and asymmetrical faults, achieving high accuracy even under noisy conditions. Recent studies in fault classification using NN seem to be focusing on Convolutional Neural Networks (CNN), Probabilistic Neural Network (PNN) and Feedforward Neural Network (FNN) [27].
- Support Vector Machines (SVMs), known for their robustness in high-dimensional spaces, have also been effectively used for multi-class classification of fault types. This type of models often outperforms simple classifiers in terms of generalization and their ability to find the optimal hyperplane makes them especially useful for

separating closely clustered fault classes. Hybrid model that combine wavelet transform for pre-processing features and SVM to classify faults show good performance in real-time fault detection [27].

- Decision Trees (DTs), Random Forests (RFs), and Gradient Boosting Trees (GBTs) are models that offer different trade-offs in terms of interpretability, computational cost, and performance. Decision Trees are valued for their simplicity and low computational cost, however, their performance is often susceptible to overfitting. Random Forests address these limitations by constructing an ensemble of decision trees using random subsets of both data samples and features, this increases the robustness and accuracy of the model. RFs are also less sensitive to outliers in the training data, but the trade-off is that the model has a higher computational complexity. Gradient Boosting Trees, on the other hand, build trees sequentially, and every sequence is focused on correcting the errors of the previous ones. This step-by-step optimization allows GBTs to achieve high accuracy on complex datasets. Nonetheless, they are more prone to overfitting, and require longer training times compared to DTs and RFs [26].

3.3.2 FAULT LOCATION

AI and ML-based fault location methods aim to estimate the exact distance or section of the line where a fault has occurred, which is particularly challenging in distribution networks due to bidirectional power flows and dynamic configurations. To address this challenges, advanced ML methods have been developed using inputs such as current and voltage magnitudes or features that have been extracted from WT/FFT, and output the distance to the fault from the substation or the node where the fault has occurred in:

- Graph Convolutional Networks (GCNs): this fault location model was trained on the IEEE-123 bus systems, obtaining good accuracies. It aggregates measurements from different nodes of the network and is robust to noise, missing data, and topology changes, outperforming conventional ML classifiers [28].

- Physics-Preserved Graph Networks (PPGN): This two-stage GNN method use sparse observations to train the model and then uses limited labelled data to predicts fault location at node-level. It was tested on IEEE-37 and IEEE-123 feeder models and it maintained high accuracy in different load and topology scenarios [29].
- Multi-Head GATs with PMU Data: Recent work proposes a multi-head Graph Attention Network that uses Phasor Measure Units PMU data to learn topological and electrical relationships of the features for fault localization. Compared to traditional methods, this approach significantly improves accuracy and speed [30].
- Spatial-Temporal Recurrent GNNs: these models incorporate temporal data sequences and spatial graph structure, this allows fault detection, classification, and location at with improved generalization across network scenarios and a reduced dependency on having sensors in all nodes of the network.

The development of these machine learning techniques show that data-driven fault diagnosis and location is becoming feasible for modern smart distribution networks. Thanks to the increasing deployment of smart meters and the improvement of computational capacity, accurate and adaptable models will be key for future self-healing, resilient grids.

As distribution grids continue to modernize, we can expect these intelligent fault location techniques to move from research into field implementations, which is one of the main focus of this project.

3.3.3 DISTRIBUTION COMPANIES USING ML/AI

Currently, some distribution companies are beginning to implement ML/AI models and techniques to manage electrical faults.

Pacific Gas and Electric (PG&E) has developed an ensemble model based on decision trees which uses smart meter data, asset allocation, weather conditions and load data to predict transformers failures. Between April 2021 to February 2022, over 270 predictions were reviewed, with 64% confirming relevant transformer anomalies [31]. In parallel, PG&E is

developing a project aimed to predict sustained outages using meter data, historical outage records and weather data, but it is still in the Minimum Viable Product stage.

State Grid Corporation of China (SGCC) has implemented AI-driven strategies to enable the development of self-healing grids by having different sensors in the grid equipped with AI capabilities which allows them to independently route power and address faults. This strategy has reduced fault resolution time from hours to less than 5 seconds, with the grid being able to automatically locate faults and change its topology [32]. Additionally, SGCC applies machine learning algorithms to identify patterns in historical outage data, which allows them to predict and mitigate faults.

Other leading utilities, like Korea Electric Power (KEPCO), Electricite de France (EDF) and China Southern Power Grid (CSPG) hold patents related with the use of AI techniques in distribution grids. KEPCO focuses on fault diagnosis and management, while CSPG is developing AI-assisted fault detection tools to enhance response times and network resilience [33].

3.4 DATA ACQUISITION AND UTILIZATION

Throughout the past years the availability of data readings from the distribution network has increased thanks to the deployment of Advanced Metering Infrastructure (AMI). Smart meters, which are installed between the customer loads and the network, capture parameters like voltage, current, power or power factor in intervals that can go from second to minutes. This data can provide help in scheduling power plants, operation of subsystems or maintenance for power equipment [34]. These meters also permit two-way communication between the utility and the meter, allowing utilities not only to collect the data centrally but also to control the functioning of the smart meter remotely.

As the speed at which new data is generated increases, the volume of measurements becomes too large to be stored and analysed using traditional database technology, that is why several initiatives on how to use this data are being studied by DSOs to leverage the immense amount

of data to better understand the functioning of the network and improve its reliability. One example is state estimation, in which smart meter data is fed into models that estimate voltage profiles across distribution grids, improving system reliability by enhancing network visibility. Another data application is power flow and load forecasting by using past energy consumptions from each smart meter and weather forecast data, allowing to prioritize grid investments by prediction the load at a desired point in time [35].

This project uses smart meters RMS voltage readings as inputs for the machine learning models, enabling the classification and location of electrical faults, every time than an alarm is detected. This approach demonstrates how AMI data can be transformed from raw measurements into useful data for improving fault management and overall network reliability.

4 METHODOLOGY

This project can be divided in two different parts, the first one is the simulation of the electrical faults and the different LCT scenarios using the software OpenDSS in order to get the data needed to train the ML model. The second part is the training and testing of the ML models using specific Python libraries and the respective analysis of results. Figure 5 shows a block diagram with the main tasks that have been done.

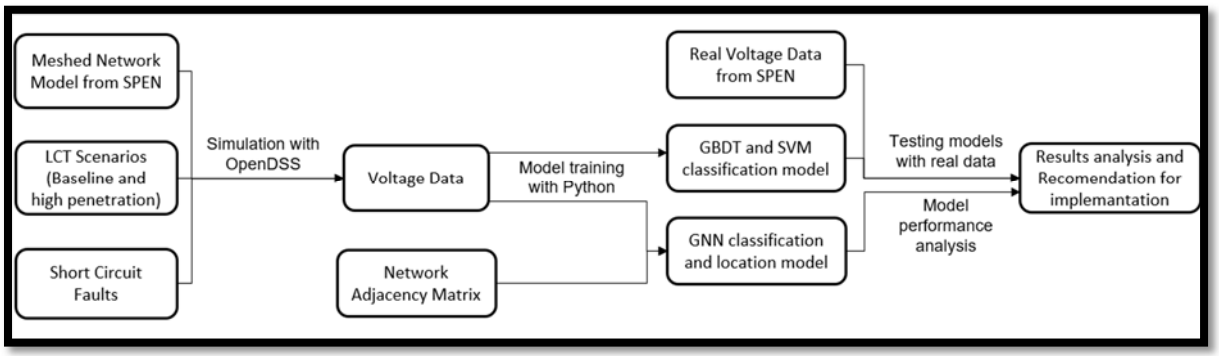

Figure 5. Project Block Diagram

4.1 SOFTWARE

Two main software have been used in the project: OpenDSS and Python

4.1.1 OPENDSS

The software used in this project to simulate electrical faults and network operating conditions is OpenDSS (Open Distribution System Simulator), an open-source simulation tool developed by the Electric Power Research Institute (EPRI). OpenDSS was chosen for this project since it is designed for comprehensive analysis of electric power distribution systems, it is very useful for studying unbalanced, three-phase networks under a wide range of operating conditions and because all SPEN network models were available in this format.

OpenDSS solves power flow problems using the fixed-point iterative method, also known as the current injection method [34], which enables the accurate modeling of unbalanced and

non-linear systems. The software operates using a text-based input system through .dss script files, where each file defines various electrical components in the system such as transformers, lines and loads (resistive, inductive, or capacitive). Each element is assigned to a bus, which typically contains three electrical nodes corresponding to the three phases. OpenDSS can also simulate electrical faults. Faults can be defined at any bus within the network by specifying parameters such as the phases involved (e.g., A-B, B-C, A-Ground) and the fault resistance in ohms. This allows the modeling of single-phase, two-phase, and three-phase short-circuit events, providing flexibility in fault characterization.

Figure 6 shows the declaration of a transformer, a line, a load and a three-phase fault, all interconnected through designated buses.

Code 1

```
New transformer.Trafo_1 windings=2 phases=3 buses=(sourcebus 1) conns=(delta wye)
           kvs=(11 0.415) kvas=(500 500) XHL=12

New line.Line_1 phases=3 bus1=1 bus2=2 linecode=Cable_230V_0.1_AL length=10 units=m

New load.Load_1 bus1=2 kv=0.2396003617136947 phases=3 kW=5 pf=1

New Fault.F_1 phases3 bus=2 r=0.01
```

Figure 6. OpenDSS Declaration of Elements

The modular and script-based nature of the platform allows the modeling of different network topologies and configurations, including the incorporation of distributed generation resources. For instance, PV plants can be integrated into the model through the *PVSystem* object, which allows the representation of grid-connected photovoltaic plants and their behaviour under different irradiance and temperature scenarios.

OpenDSS has several simulation modes to analyse network behaviour over time [35], these are:

- Snapshot: Solves a single steady-state power flow representing a fixed point in time.

- Daily: Simulates a 24-hour period with time-varying load and generation profiles.
- Yearly: Extends the simulation over a full year, using long-term profiles for seasonal variation.
- Direct: Allows the user to control the solution process manually, solving each time step individually.
- DutyCycle: Simulates short-term repetitive behaviour (e.g., inverter switching cycles).
- Dynamic: Performs transient simulations including control actions and dynamic system responses.

Simulation outputs in OpenDSS include detailed electrical quantities such as voltage magnitudes and angles, current flows, real and reactive power flows, and network losses. These results are stored in output files or can be exported through COM interfaces, enabling further analysis using external tools such as Python, as done in this project.

4.1.2 PYTHON

Python is the programming language used in the project for its extensive number of scientific libraries and its use in machine learning applications. It was used to manage data preprocessing, to interact with OpenDSSS and to implement the ML models. The main libraries used were:

- **py_dss_interface**. This library was used as a wrapper for the OpenDSS interface, enabling simulations of different networks, applying faults, extracting data and automating workflows directly from the Python scripts.
- **ScikitLearn**. This library is widely used in ML applications and was used to implement the GBDT and SVM models. The library also allows hyperparameter tuning and performance evaluation metrics such as confusion matrices or classification reports.
- **Tensorflow and Keras**. This library, which was developed by google, was used to develop and train the GNN models. Along with its application programming

interface Keras, Tensorflow allows graph-based computation, making this library suitable for modeling meshed LV networks

4.2 DATA GENERATION

For the purposes of this study, the Daily simulation was selected in OpenDSS. The simulations were conducted using 30-minute time intervals, corresponding to 48 discrete steps per day. A custom load shape object with 48 entries was defined to simulate the varying demand profile typical of a residential or urban distribution feeder.

4.2.1 NETWORK MODEL

The project focused on the study of a specific SPEN network, which was selected due to its meshed topology with three different feeders and its amount of fault alarms that were recorded by smart meters of that network.

The model was obtained in dss format, which had all the electrical components along with their parameters. A Json file which contained the exact coordinates of every node and the id names of the smart meters of the network was also obtained.

The network used in this project is a low-voltage distribution network located in an urban area close to Liverpool, it operates at 415 V and has 341 different buses. The network is structured around a central main loop, which gives it its meshed topology and provides multiple paths for the energy to reach the loads. In addition to the main loop, the upper-right section of the network is a radial branch that connects to different loads and to a feeder. Figure 7 shows the location of all the buses of the network, where there are 125 buses highlighted in yellow which represent the consumers or electrical loads with smart meters in the system. These costumers have an average maximum consumption of 5 kW, which means that the grid has a peak demand of 625 kW.

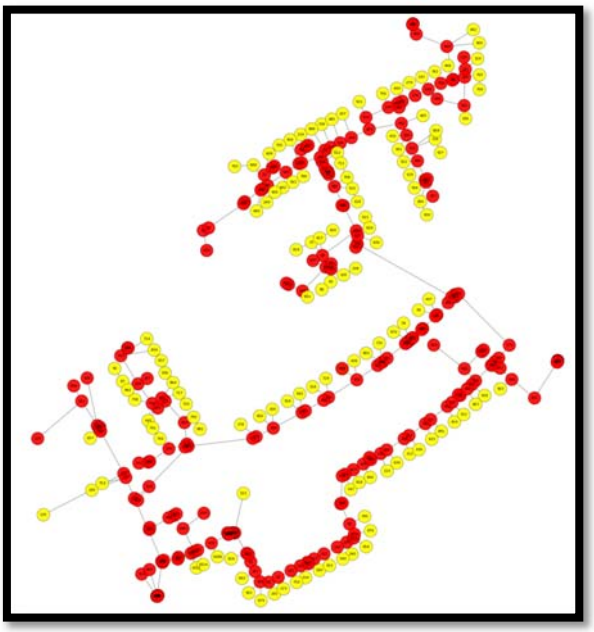

Figure 7. Network Loads

In addition, the network has three feeders that represent step-down transformers that convert voltage from 11 kV to 415 V. They are represented by the yellow buses in Figure 8.

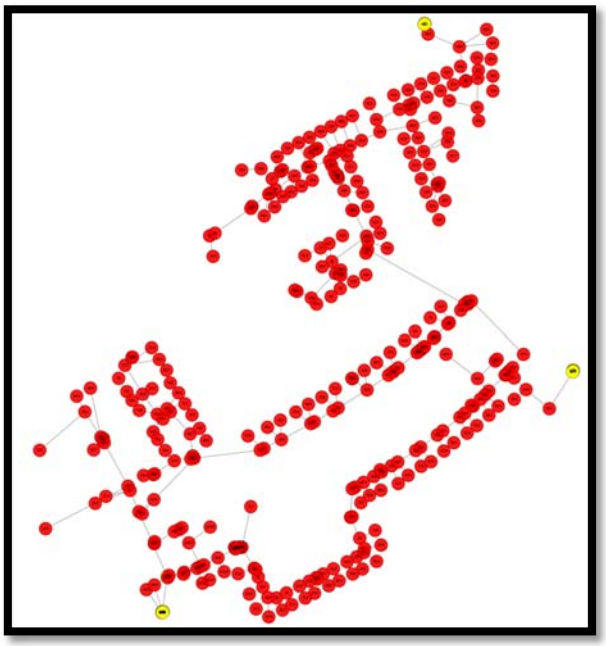

Figure 8. Network with Feeders

4.2.2 LCT SCENARIOS

In order to evaluate the performance of the models under evolving grid conditions two different LCT penetration scenarios were modelled.

4.2.2.1 Baseline Scenario (Scenario 1)

This scenario is a representation of the current state of the network, without extra LCTs beyond the ones that are already present. In order to simulate this scenario, all the loads of the simulation were assigned a load-shape that corresponded with a typical residential consumption profile with no influence of DG. The load-shape has two peaks, one in the morning that corresponds to the residential and commercial start up activities and other one in the evening when customers are at home, and during night-time the demand is relatively low. The load-shape can be seen in Figure 9 represented as the grey curve. This baseline configuration serves to assess the models accuracy in traditional networks.

4.2.2.2 High LCT Penetration Scenario (Scenario 2)

This scenario acts as a projection of how the distribution network will operate under a context of high LCTs adoption. In the simulations, three key LCTs were be implemented across the network to reflect their growing presence in LV grids. PV systems were included in the model as two large photovoltaic plants, while the effect of EVs and heat pumps was modelled by modifying the demand profile of the loads. By doing these modifications, the power flows were bidirectional and more complex. These changes allowed to analyse the resilience and adaptability of the developed models, and if their performance remained within accepted values, it would demonstrate their suitability for supporting the integration of these emerging technologies in a secure and controlled way.

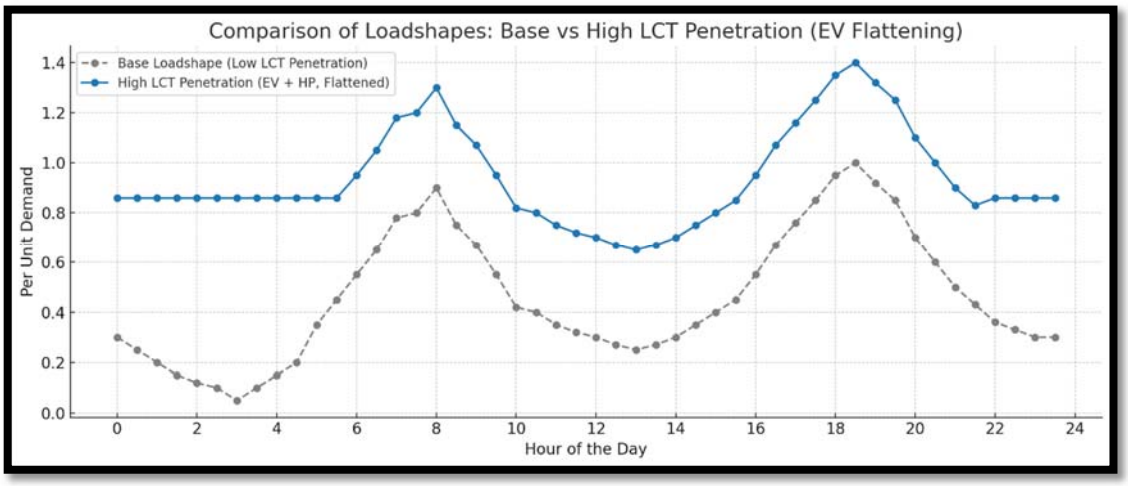

Figure 9. LCT Scenarios Load-shape

The new load shape was configured following the following parameters:

In the case of heat pumps, projections and policies estimate that by 2030 in Europe and the UK approximately 40% of households will have electric heat pumps [36]. These systems will have low power continuous modes that will operate constantly throughout the day to maintain consistent indoor temperatures. The average power that heat pumps will consume will be similar to the base household load. Taking everything into account, the effect in the load shape will result in an additive load equal to 0.4 per unit (pu) across all of the base load.

For electric vehicles, the adoption forecast estimates that there will be a 30% penetration among households by 2030. To reflect the expected charging behaviour, the demand was concentrated during night time hours, since smart charging strategies like time-of-use (ToU) tariffs will be adopted. These strategies will flatten the overall demand curve by fitting this new demand during times when prices are low or when the network capacity is not being used. As a result, the overnight valley in the base load shape will evolve into a constant curve with a higher average demand.

All of these changes can be seen in Figure 9, where both load shapes (baseline and high penetration) are compared.

The two PV plants were implemented in the network by using a specific OpenDSS command shown in Figure 10. This command allowed defining parameters such as the number of phases of the system, the bus where the plant was going to be connected, the nominal voltage of the plant, the peak power generation under maximum solar irradiance, the installation's power factor and a solar generation profile. Both plants were configured as three-phase systems, each generating 75 kW at unity power factor (pf = 1). Since the total network consumption was approximately 625 kW, the combined PV output represented about 25% of the total demand. The daily generation profile used for both plants is a generic solar curve, which as seen in Figure 11, peaks at midday when solar irradiance is highest.

Code 2

```
New PVSystem.PV1 phases=3 bus1=16680183 kv=0.415 kVA=75 daily= solar_loadshape
                 irradiance=1.0 pf=1.0
```

Figure 10. PVSystem Command in OpenDSS

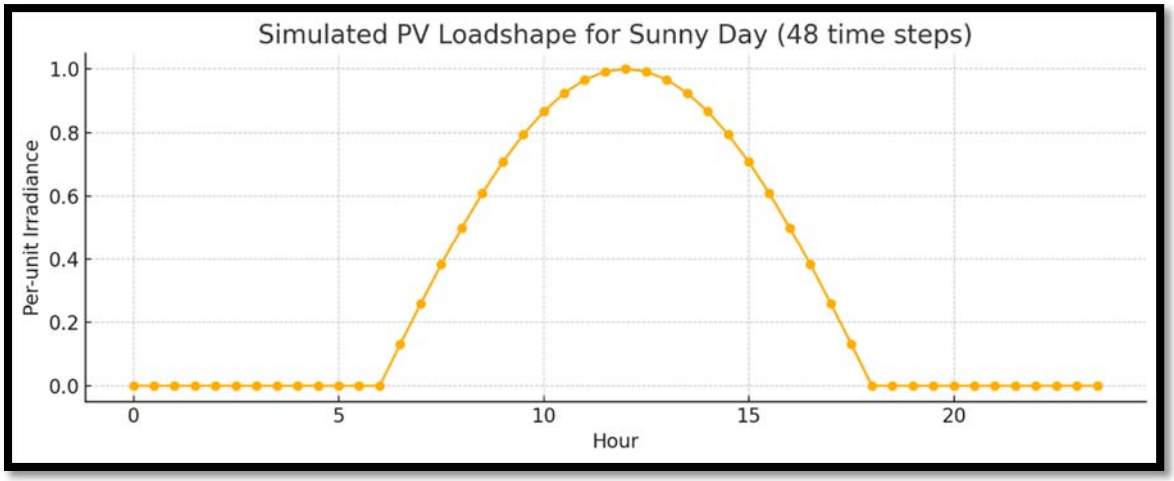

Figure 11. Solar Generation Curve

4.3 DATA RECORDING

The only electrical variable that was recorded in the project was rms voltage magnitude. This was decided based on the real fault data that was provided by SPEN, which consisted of rms voltage readings and no current measurements, due to customer privacy considerations. Because of this, the simulations were designed to only capture voltage measures, ensuring that the developed models could be applied directly to SPENs data to predict scenarios. The daily mode was used in the OpenDSS simulations to record the voltage magnitudes for every step of the daily load shape used.

Since the ML models used for the project require two main inputs, which are *X* (date features) and *y* (targets), three arrays were generated for each simulation to match this requirement. The feature set (*X*) consisted of a *Voltage_Magnitude_Readings* Array, and two separate targets arrays (*y1* and *y2*) were created, the *Fault_Type_One-Hot* Array for the classification model and the *Fault_Location_One-Hot* Array for the fault location one. The structure of these arrays is the following:

- *Voltage_Magnitude_Readings:* This array is the primary input for the ML models. Its shape varies depending on the model, for the Gradient Boost Decision Tree and for the Support Vector Machine the shape is (number of nodes with smart meters, 1) and for the Graph Neural Network the shape is (number of buses, number of voltages per bus), where the number of voltages is 3, since each bus has three nodes with a different voltage. Each column represents a different bus of the network and each row contains the steady state voltage magnitudes for that bus.
- *Fault_Type_One-Hot:* This array is the label for the fault classification model. Its shape is (number of fault types, 1), where each column represents all the possible scenarios (no fault, SLG on each of the three phases, LL for each phase pair (A-B, A-C, B-C), and three-phase faults). All entries are zeros except for a 1 in the columns corresponding to the fault type that is being simulated.

- *Fault_Location_One-Hot:* This array is the label for the GNN fault location model. Its shape is (number of buses in the network, 1). All entries are zeros except for a '1' in the position corresponding to the bus where the fault is located.

The data collection process is done with a Python function *collect_data*, which simulates the scenarios and saves each array into their respective dataset (*X_data, Y_fault_type, Y_fault_location*).

4.4 FAULT SIMULATION USING OPENDSS

The next step after having all the scenarios ready is to simulate all the faults. This process employed the Monte Carlo method, which consists of repeatedly doing random samples of an event to create a large number of scenarios, to ensure that the data set had a wide variety of possible fault conditions.

The faults that were simulated included the most common type of faults in distribution systems: single line-to-ground faults on each phase; line-to-line faults between phases A-B, A-C and B-C; and three-phase faults, which makes a total of 7 different fault scenarios. Since the objective of the project was to evaluate model performance for both low and high impedance short circuits, the fault impedance values ranged between 0.01 to 1 Ω.

The fault generation process was done in Python, which was used as the interface to interact with OpenDSS, following the next sequence:

1. Scenario definition: A *fault_scenarios* array was created, containing the name of each fault type, along with the phases that were involved.
2. Bus selection: A *fault_bus* array was created containing a predefined number of buses, which were randomly chosen from a list of all the network buses.
3. Automation loop: A Python loop was used to automate the execution of all fault scenarios. The loop iterated over every bus in *fault_scenarios*, simulating all the defined scenarios in *fault_bus*. For each iteration, a random fault resistance was chosen form the range 0.01 to 1 Ω. Then, each parameter (location, fault type, and

impedance) was passed to OpenDSS via the *py-dss-interface* library for them to be simulated.

4. Data recording: after the execution the resulting electrical measurements were recorded using the *collect_data* function.

The pseudo code used in Python is shown in Figure 12.

Code 3

```
fault_scenarios = '7 fault scenarios'

buses = random.sample(bus_names, n) # n decides the number of buses that are
chosen randomly

for fault_name in fault_scenarios:

   for bus in buses:

         R = random.uniform(0.01,1) # Choose a random R

         dss.text(f"New Fault.F{fault_count} phases=3 bus={bus} r={R}")# fault
in  OpenDSS

         collect_data(fault_type, fault_location) # Simulate and collect data

         dss.text(f"edit Fault.F{fault_count} enabled=no") # Remove fault

         fault_count += 1
```

Figure 12. Fault Simulation Code

4.5 TRAINING THE MACHINE LEARNING ALGORITHMS

This section describes the design, configuration and training of the ML models. The specific architecture, code and hyper parameters of each model is explained along with its results and achieved accuracy.

4.5.1 GRADIENT BOOSTED DECISION TREES (GBDT)

The GBDT model was implemented using the Scikit-learn library's *GradientBoosingClassifier*. The models objective was to classify each sample into one of

eight possible scenarios using *Voltage_Magnitude_Readings* as input features (X) and *Fault_Type_One-Hot* as target (y).

The code used to implement the model in Python had the following procedure:

- The target array was modified from a one-hot encoding to an integer label, which means that in the original array every sample had all the columns as zero except the one which represented the target with a one but the modified array consisted of only one number which corresponded to the fault class (e.g. y'=0,0,0,1,0,0,0 becomes y=4).
- The dataset was split into training and testing sets with an 80/20 ratio for training/test. Each set had a different purpose, the train set was used to fit the model, enabling it to learn the relationships between inputs and outputs, while the test set was used to evaluate the models accuracy using unseen data.
- The ML model was initialized and the hyper parameters were adjusted. The hyperparameters used in the model were *number_of_trees*, which sets the number of boosting stages that the model has, *learning_rate*, which controls the size of the steps taken by the optimizer, *max_depth*, which sets the maximum branches of the individual trees and *random_state* which controls reproducibility.
- The model was trained using the training data set.
- The accuracy was calculated with the test data set.

In order to set the optimal hyperparameters a grid search with cross-validation was performed, giving the following results:

- number_of_trees =100
- learning_rate=0.1
- max_depth=3
- random_state=42
- verbose=1

The pseudocode used to develop the model can be seen in Figure 13.

Code 4

```
# --- 1. Load data ---
X = load_data["X"]
Y_fault_type = load_data["Y_fault_type"]

# --- 2. Convert labels from one-hot encoding to integers ---
Y_labels = np.argmax(Y_fault_type, axis=1)

# --- 3. Split into training and testing sets ---
X_train, X_test, y_train, y_test = train_test_split(
    X,
    Y_labels,
    test_size=0.2,)
# --- 4. Initialize Gradient Boosting model ---
gb_classifier = GradientBoostingClassifier(
    number_of_trees =100,
    learning_rate=0.1,
    max_depth=3,
    random_state=42
)
# --- 5. Train model, predict on test set and evaluate accuracy ---
gb_classifier.fit(X_train, y_train)
y_pred = gb_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

Figure 13. GBDT pseudocode

After training the models with datasets from the LCT scenarios 1 and 2 defined in section 4.2.2, the following results were obtained:

For the model from scenario 1 (baseline) the accuracy was of 99.82% with a training time of 40 minutes and 50 seconds. The confusion matrix can be visualized in Figure 14, where it can be seen predicted class with the highest misclassification rate was class 0, which is no fault.

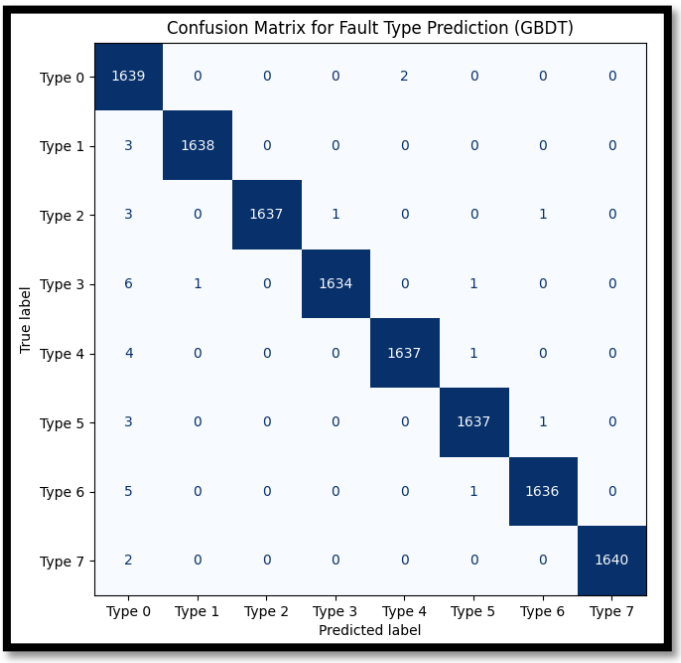

Figure 14. GBDT Scenario 1 Confusion Matrix

For the model from scenario 2 (high LCT penetration), the models accuracy is 99.76% with a similar training time to model 1 of 42 minutes and 14 seconds. Its decision matrix can be visualised in Figure 15.

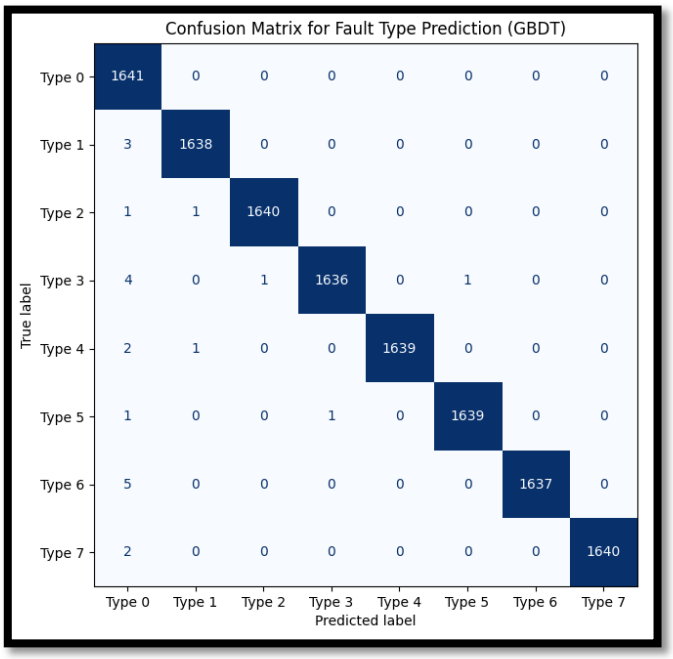

Figure 15. GBDT Scenario 2 Confusion Matrix

One of individual trees that is part of the final ensemble of trees can be visualized in Figure 16, where it can be seen the typical tree structure. Each internal node of the tree includes:

- The decision rule for the node that decides the split.
- The *friedman_mse* value, which computes the mean squared error that is produced by the split [37].
- The number of samples that fall to that node.
- The predicted output value for the node, which in a GBDT represents the "pseudo-residuals" that are added to the overall prediction. The model's final output for a sample is the sum of the values from all the trees it passes through.

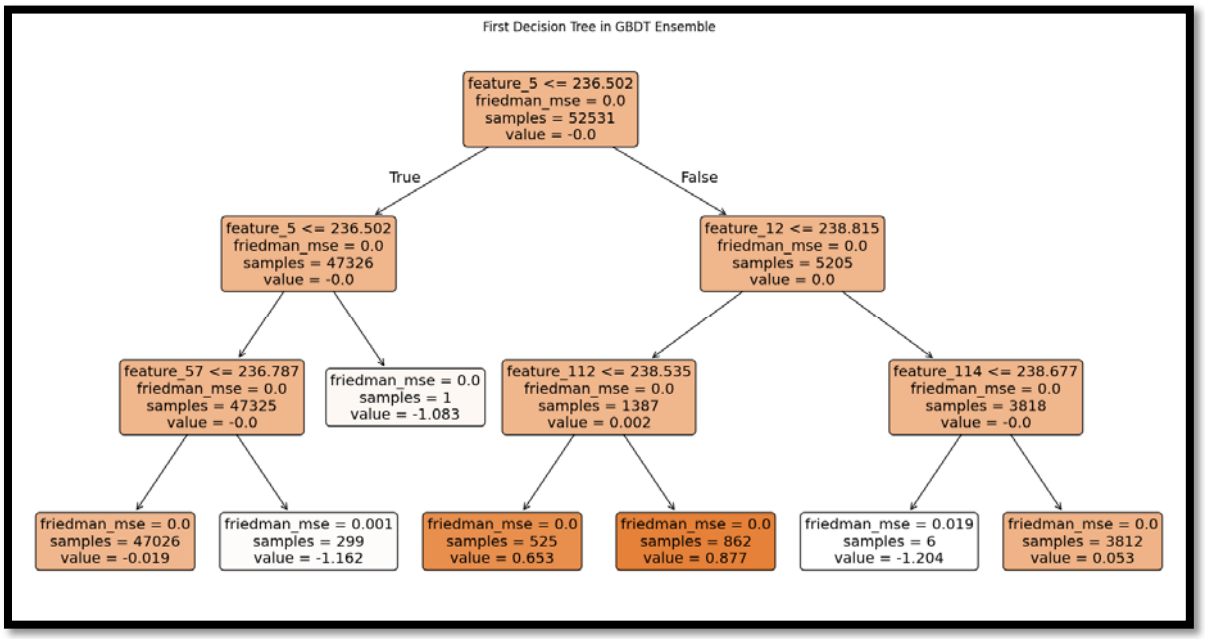

Figure 16. Decision Tree in GBDT Ensemble

4.5.2 SUPPORT VECTOR MACHINE

The SVM model was implemented using the Scikit-learn library's SVC class. Its objective was also to classify each sample into one of eight possible scenarios, using the Voltage Magnitude Array as input features (X) and the Fault Type One-Hot Array as the target (y).

The code used to implement the model in Python followed the same general process as the GBDT model: transforming the target from one-hot encoding to integer labels, splitting the dataset into training and testing sets, initializing the model, training it on the training data, and finally evaluating accuracy on the test set. The only difference was the hyperparameters settings, which were: the kernel type; *gamma 'σ'*, which is a kernel coefficient that controls the shape of the decision boundary; the regularization parameter; and *random_state*.

In order to set the most optimal hyperparameters a grid search with cross validation was used, resulting in:

- Radial basis function (RBF) kernel, which is a common kernel for SVM whose formula is $K(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$, where x are two feature vectors [38].
- σ='scale', which calculates automatically a suitable gamma value based on the training data.
- C=1.0
- *random_state*=42

The pseudocode used in Python to develop the SVM model can be seen in Figure 17.

Code 5

```
# --- 1. Load data ---
X = load_data["X"]
Y_fault_type = load_data["Y_fault_type"]

# --- 2. Convert labels from one-hot encoding to integers ---
Y_labels = np.argmax(Y_fault_type, axis=1)

# --- 3. Split into training and testing sets ---
X_train, X_test, y_train, y_test = train_test_split(
    X,
    Y_labels,
    test_size=0.2,)

# --- 4. Initialize Support Vector Machine model ---
svm_classifier = SVM(
    kernel='rbf',
    C=1.0,
    gamma='scale',
    random_state=42
)
# --- 5. Train model, predict on test set and evaluate accuracy ---
svm_classifier.fit(X_train, y_train)
y_pred = svm_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

Figure 17. SVM pseudocode

After training two models, with datasets from the LCT scenarios 1 and 2 defined in section 4.2.2, the results obtained were the following:

The first SVM model achieved an accuracy of 99.76% and a training time of 5 minute and 7 seconds. As seen in Figure 18, the class with the most misclassifications is the no-fault class (class 0).

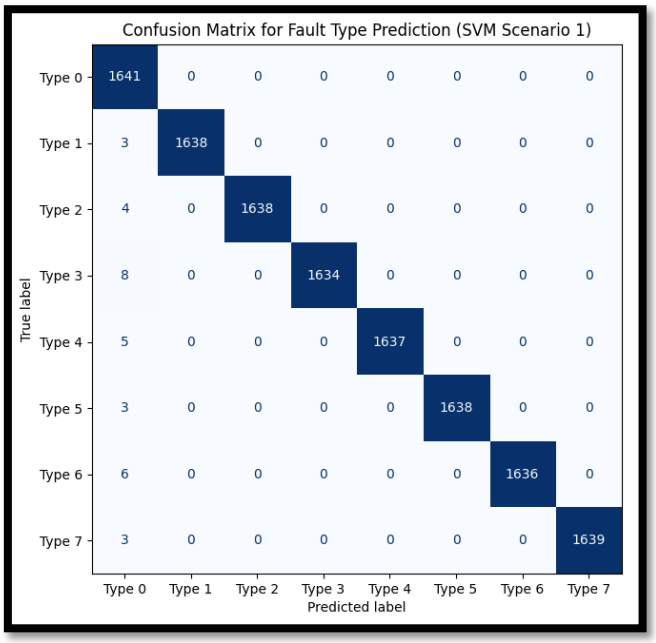

Figure 18. SVM Scenario 1 Confusion Matrix

For the scenario 2 model, an accuracy of 99.73% was achieved, which shows that the model maintains its performance despite the increased complexity of high LCT penetration. The model's confusion matrix can be visualized in Figure 19.

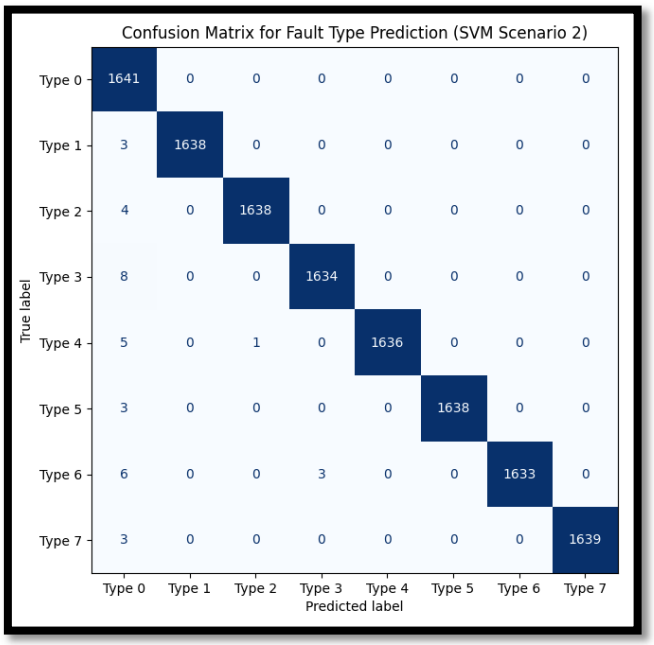

Figure 19. SVM Scenario 2 Confusion Matrix

4.5.3 GRAPH NEURAL NETWORK

Graph Neural Networks were selected among all the possible Neural Network architectures to take advantage of the inherent graph topology of electrical distribution networks, since all buses have to be interconnected by lines, and each bus electrical state is influenced by that of its neighbors. By representing the network as a graph it is possible to explicitly model these relationships.

Two different GNN models were developed, one dedicated to fault classification, like with the GBDT and SVM models, and other one for fault location, which was able to localize the bus were the fault had occurred. Both models share the same input structure, which consists of:

- A graph where each node corresponds to a bus in the network with three voltage magnitudes that represent each phase of the bus.
- The network adjacency matrix, which was obtained by analyzing the *dss* file and seing the connection among all buses.

This models, unlike the previous GBDT and SVM models, use the voltages measurements from all the nodes in the network. This is crucial, since if only the voltages from smart meters were used, the resulting adjacency matrix for the measured buses would result in a matrix were all nodes would be connected to each other, since all the nodes that previously represented different bifurcations and intersections would collapse into a single node.

4.5.3.1 Fault Classification Model

The GNN model was implemented in Python using the Spektral library for handling graph-structured data within the Tensorflow framework. The methodology has the following stages:

- The adjacency matrix is pre-processed using symmetric normalization, which is a standard procedure for NNs that ensures that the information is passed between connected nodes. The normalized matrix and the *Voltage_Magnitude_Readings* dataset are then encapsulated within a custom Spektral dataset class, which transforms the data in a format that can be processed by the GNN, where each sample is treated as an independent graph.
- The data is split into training and testing sets, and then a BatchLoader, which is in charge of feeding batches of data to the model during training, assigns every sample to a batch.
- The model is defined with the following parameters:
    - Three graph convolutional layers with 64 neurons each with a *ReLU* activation function.
    - A graph-level pooling layer which aggregates all the learned features of all the nodes in to a single vector.
    - A final dense layer with a *softmax* activation function. This activation function converts a vector of real numbers into a probability distribution, where each element represents the probability of belonging to a specific class, and the sum of all elements equals 1, which is why this layer takes the vector

created by the graph-level pooling and outputs a probability distribution over the number of possible fault types.

- The model is trained using the *adam* optimizer, which is typical optimization algorithm in NNs, and *categorical_crossentropy* as the loss function that is being minimized. The training process has an early stopping function which prevents overfitting by stopping the training when the model's performance on the validation set stops improving.

The pseudocode used in Python can be seen in Figure 20.

Code 6

```
# --- 1. Load Data ---
X = load_node_features()
A = load_adjacency_matrix()
Y_fault_type = load_one_hot_labels()

# --- 2. Preprocess Adjacency Matrix and Data in Spektral Dataset ---
A_norm = gcn_filter(A)
dataset = create_spektral_dataset(X, A_norm, Y_fault_type)

# --- 3. Split Data into Train and Test Sets and create Batch Loaders -
--
train_dataset, test_dataset = train_test_split(dataset, test_size=0.2)
train_loader = create_batch_loader(train_dataset, batch_size=32)
test_loader = create_batch_loader(test_dataset, batch_size=32)

# --- 4. Initialize GNN Model Architecture ---
model = initialize_model_with_layers(
    GCNConv_1(64, activation="relu"),
    GCNConv_2(64, activation="relu"),
    GCNConv_3(64, activation="relu"),
    GlobalAveragePooling1D(),
    Dense_output(NUM_FAULT_TYPES, activation="softmax")
)
# --- 5. Compile the Model ---
model.compile(
    optimizer="adam",
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)
# --- 6. Train and evaluate the Model ---
model.fit(
```

```
    training_data=train_loader,
    epochs=40,
    validation_data=test_loader,
    callbacks=[EarlyStopping(monitor='val_accuracy', patience=10)]
)
evaluation_metrics = model.evaluate(test_loader)
print_results(evaluation_metrics)
```

Figure 20. GNN Fault Classification pseudocode

After training two identical fault classification GNN models with datasets from the LCT scenarios 1 and 2, the models' performances were the following:

For the model trained with data from scenario 1 (baseline), the accuracy was a 99.71% and the model's training time was 30 minutes and 2 seconds. The model's confusion matrix can be visualized in Figure 21.

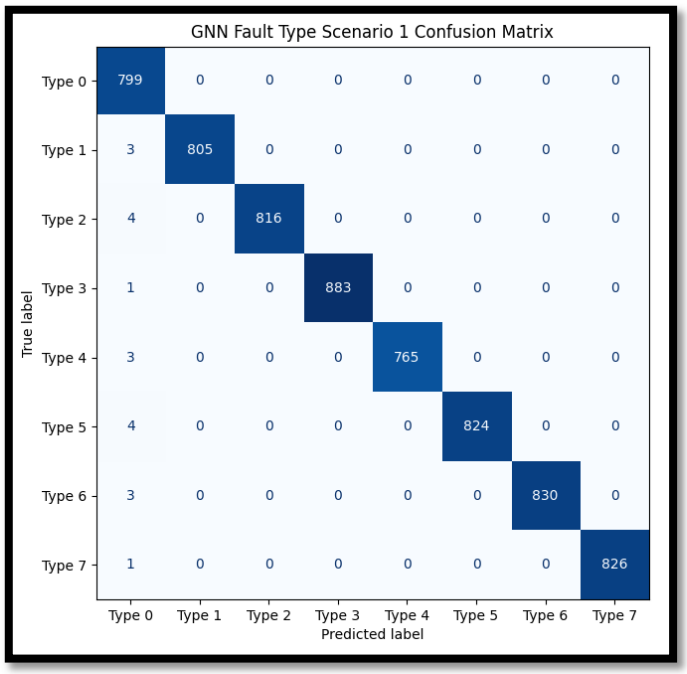

Figure 21. GNN Fault Classification Scenario 1 Confusion Matrix.

For the model trained with data from scenario 2 (high LCT penetration), the accuracy was a 99.74% and the model's training time was 32 minutes and 27 seconds. This model's confusion matrix can be visualized in Figure 22.

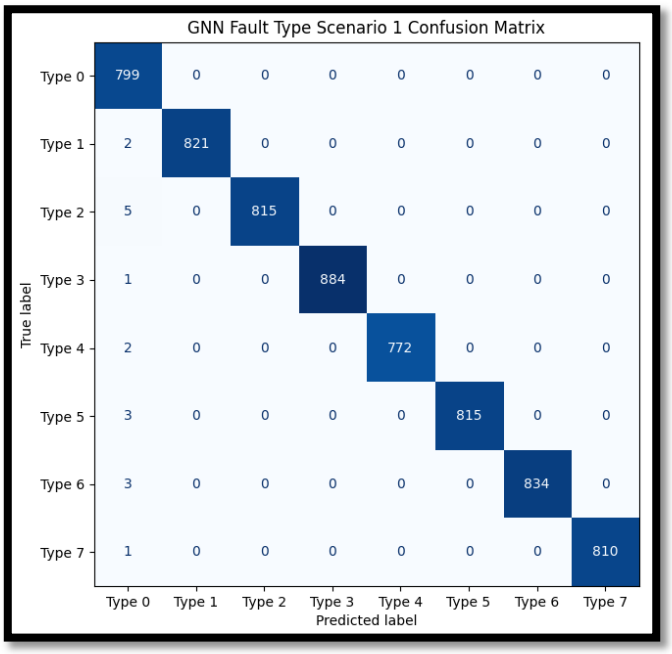

Figure 22. GNN Fault Classification Scenario 2 Confusion Matrix

Figure 23 shows a heat map that represents the kernel weights of the first graph convolutional layer of the model. In y-axis of the heat map represents the three input features, while the x-axis represents the 64 output features. The colour at any point indicates the magnitude of the kernel weights, which represent how much a specific input feature contributes to a particular output feature.

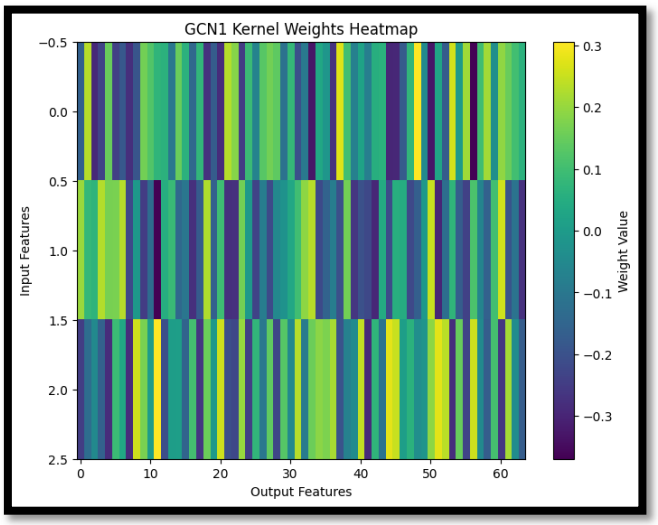

Figure 23. GNN Fault Classification kernel Weights Heat map

4.5.3.2 Fault Location Model

The methodology that has been used to implement the GNN fault location model is similar to the one used for the GNN fault classification model, including the data preprocessing and splitting to training and testing sets. Therefore, only the distinct model architecture will be described:

The fault location model is designed for a node level classification task to identify which bus is faulted. The architecture of the model is defined with the following parameters:

- Three graph convolutional layers with 60 neurons each with a *ReLU* activation function.
- A final dense layer is applied, which processes each node's feature vector independently. This is followed by a *softmax* activation function, which produces a probability distribution across all buses. The predicted node where the fault has occurred is the one with the highest probability.

The changes in the code respect the classification model can be seen in Figure 24.

Code 7

```
# --- 6. Initialize GNN Model Architecture ---
location_model = initialize_model_with_layers(
    GCNConv_1(60, activation="relu"),
    GCNConv_2(60, activation="relu"),
    GCNConv_3(60, activation="relu"),
    TimeDistributed_Dense_output(1, activation="linear"),
    Softmax_activation())
# --- 7. Compile the Model ---
Location_model.compile(
    optimizer="adam",
    loss="categorical_crossentropy",
    metrics=["accuracy"])
```

Figure 24. GNN Fault Location pseudocode

Two location models were trained with datasets from the LCT scenarios 1 and 2 defined in section 4.2.2, to evaluate their performance. Both models trained with 20 epochs and took around 80 minutes to train.

The model trained with data from Scenario 1 (baseline) achieved a top-1 accuracy of 65.83%, which means that the model correctly identified the faulted bus as its single highest-probability prediction in over 65% of cases. This accuracy improves if we take the top ranked nodes that have the highest probability. If we take the top-3 buses the probability of the faulted bus being among the three predictions increases to 87.73%, if we take the top-5 buses the probability increases to 94.10% and if we take the top-10 buses the probability increases to 98.36%.

The model trained with data from Scenario 2 (high LCT penetration) achieved a slightly lower accuracy of 62.80%, which may be because the power flows from scenario 2 are more complex. Similar to the first model, the accuracy increased with a higher number of top predictions. The specific increase in accuracy for both models can be seen in Figure 25.

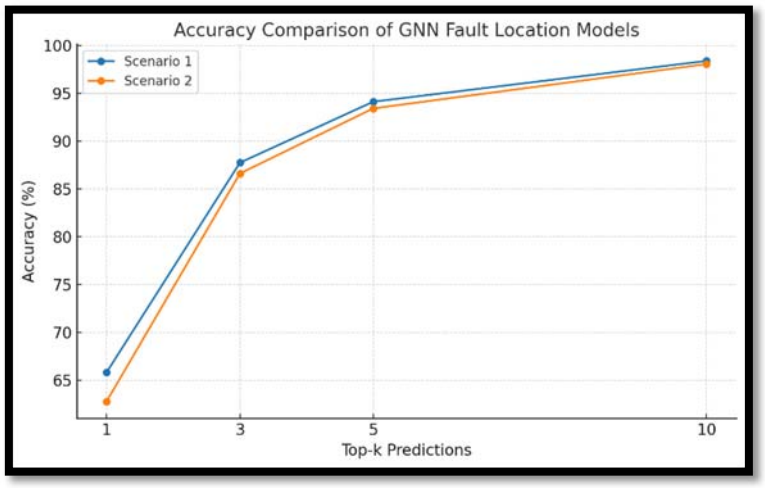

Figure 25. Accuracy Comparison of GNN Fault Location Models

The model predictions were visualized on the network. Figure shows in yellow the top-10 buses with the highest probability of being the faulted bus locations for a sample, with the correct faulted bus being among the top-10 set, and in red the other buses in the network. The image shows how all the top-10 predicted nodes are clustered in a single localized area.

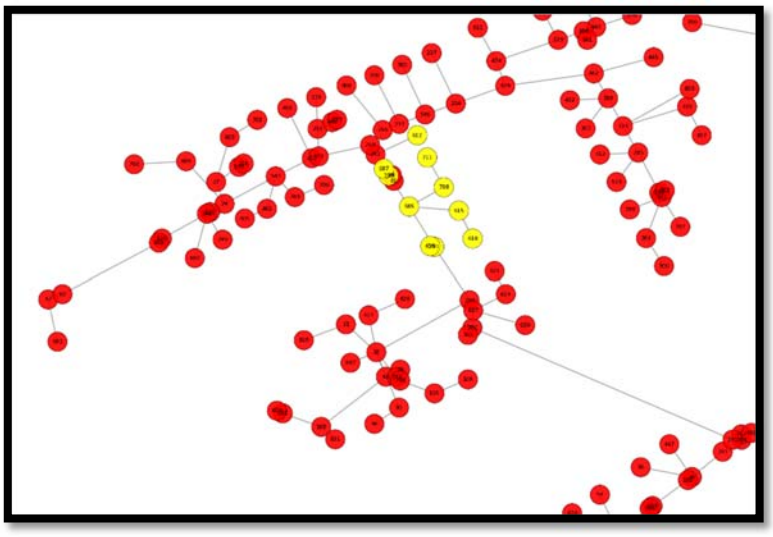

Figure 26. Top-10 Location Model Predictions

This clustering proves that the GNN is not simply learning to classify the correct node in isolation, but is successfully leveraging the graph structure to understand how the fault

propagates. With this result, even if the model doesn't predict the exact faulted bus, it would allow operators to narrow down their search to a specific area of the grid.

4.5.4 TESTING WITH REAL FAULT DATA FROM SPEN

To evaluate the real world applicability of the ML models, real fault data was used to test the GBDT and SVM models. This data was provided by SPEN and consisted of alarm records from the network, along with smart meter rms voltage readings taken every 30 minutes for the day on which each alarm occurred.

In order to be able to use the data in the model a preprocessing was done to the raw dataset for it to be in the same format as the model's input:

The raw dataset contained various types of alarms, which included overvoltage, frequency deviation, under voltage and power outage. Since this study focuses on short circuits, which can lead to low voltages or even service interruption, only under voltage and power outage data was considered.

- The voltage readings were recorded in 30 minute intervals, starting at 00:00 from that day. Due to that, only the reading immediately following the alarm timestamp was selected. For example, if an alarm occurred at 02:18 the voltage readings used would be from 02:30. This can lead to readings that do not correspond to the fault, since if these faults are detected by the network they can be cleared in a very short time.
- Erroneous or faulty meter data was removed. For example, one smart meter consistently reported a voltage of 0V for all measures so it was removed from the dataset.
- The location of the meters that had available readings was identified in the network model, the network under analysis contains 125 smart meters, but only 37 had readings for the relevant events. Figure 27 shows the location of the buses with available readings on the network, represented in yellow .

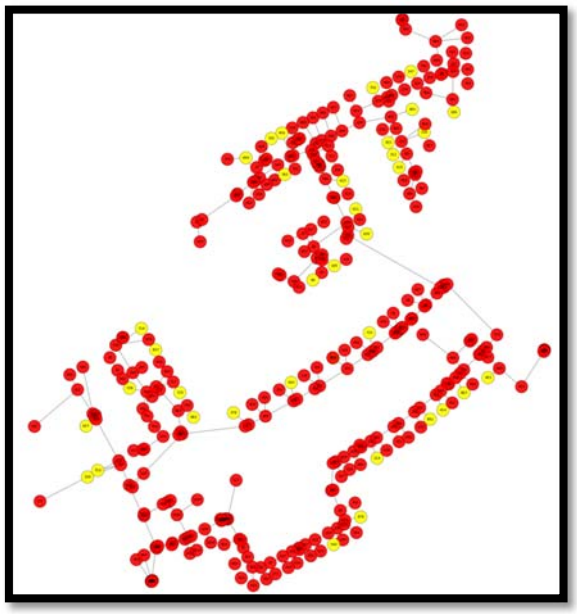

Figure 27. Smart Meters with Data Readings

After preprocessing the data, the final dataset contained 70 samples, each containing voltage readings from 36 smart meters. Figure 28 shows a heatmap of the final dataset, where it can be seen how the majority of the samples do not have any voltage variation and all the voltages are around the 240 V range. However, there are 7 samples that could suggest a possible fault event, since some of their nodes have notable voltage drops.

This dataset was not used to test the GNN models, since these models require readings from all nodes of the network, which were not available, but for future implementations, machine learning techniques for imputing missing data could be used to implement this model in real-world scenarios.

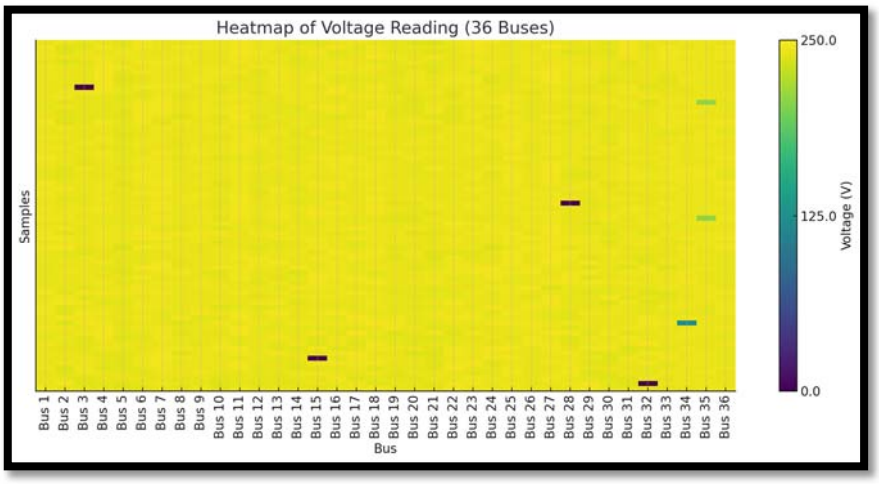

Figure 28. Real Fault Data Heatmap

The SVM and GBDT models were retrained using a simulated dataset, using only the readings from the nodes that had smart meters in the real dataset. After training these models, the real data was analysed to see the predictions.

As seen in Figure 29, the SVM model classified 64 of the samples as ‘no fault’ and the remaining 4 samples were classified as ‘single phase fault’ two of them in phase 1, one in phase 2 and one in phase 3. The samples that were predicted as faults correspond to the cases with the lowest voltage recorded in one of their nodes, which as seen in Figure 28, are sample 10, 33, 64 and 69. These results are considered reasonable since the fault predictions align with the expected scenario in the presence of single nodes with low voltages and the samples where voltage remained constant throughout all nodes were correctly classified as ‘no fault’, with the exception of the three cases that had voltage readings bellow the typical values but not significant enough to be clearly classified as a fault.

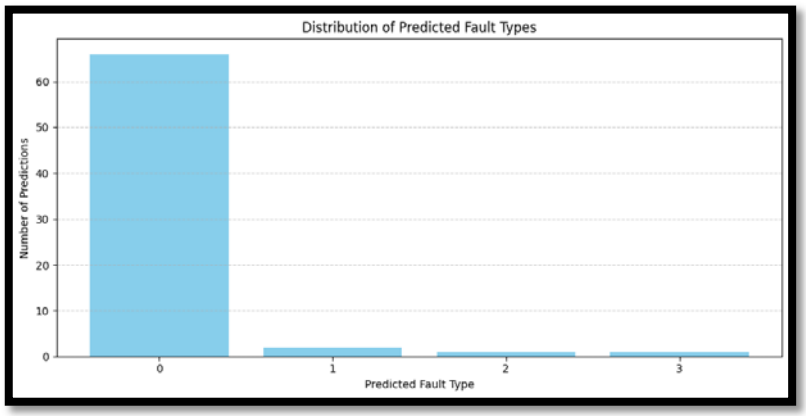

Figure 29. SVM Predictions for Real Data

The GBDT classified the majority of the samples as 'phase to phase faults', being 53 of them B-C faults and 16 A-C faults, as seen in Figure 30. This results do nt seem to be correct predictions, since they are not consistent with the observed voltage profiles and they do nt resemble the predictions made by the SVM model. Because of this, the predictions of the GBDT model are considered as incorrect.

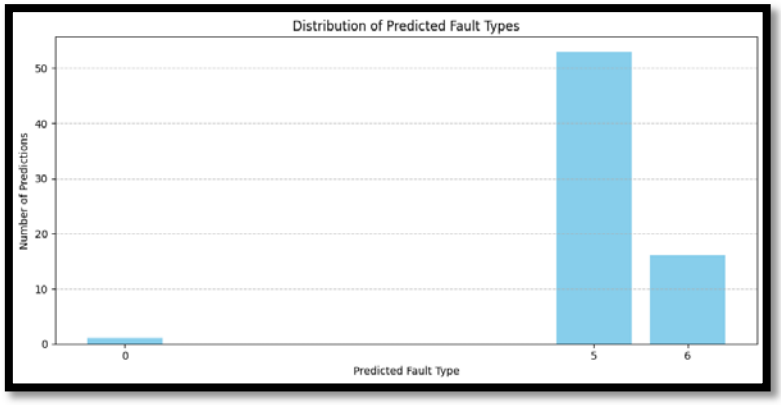

Figure 30. GBDT Predictions for Real Data

5 RESULTS ANALYSIS AND DISCUSSION

5.1 MODEL PERFORMANCE

After developing and training all the machine learning models, several key findings were established.

All the fault classification models (GBDT, SVM and GNN classification) had a high performance across both simulated LCT scenarios, being able to correctly classify the samples over 99% of the time. This proves that ML-based approaches using smart meter data (i.e. using rms voltage only) are a viable and effective solution for fault detection and classification in distribution networks. These fault analysis methods combined with predefined operational response strategies tailored to each fault type detected, could significantly improve network reliability and reduce the impact of faults. In addition, the fact that the models that were trained with the dataset from the high LCT penetration scenario maintained their high accuracy, which only a slight performance drop compared to the baseline scenario, confirms that these models are robust to the evolving grid conditions and can still be used in the future distribution grids with the same credibility. Table 1 shows the exact performance of all the classification models in both scenarios.

Classification Model	Baseline Scenario	High LCT penetration Scenario
GBDT	99.82	99.76
SVM	99.76	99.73
GNN	99.71	99.74

Table 1. Classification Model Accuracy (%)

Regarding the computational performance there were some notable differences observed in the time that took every model to train with a similar data set each. As summarized in Table 2, the SVM model had the shortest average training time, with 5 minutes, followed by the GNN classification model, with 31 minutes, being the GBDT the classification model that required the most training time. These differences are due to the algorithms that each model

uses. While SVM are more lightweight in computation, the sequential nature of the GBDT makes it a computationally intensive to train. This training time difference is a factor that has to be taken into account by DSOs when deciding which machine learning model is the most suitable one.

Model	Average Training Time (min)
GBDT Classification	41
SVM Classification	5
GNN Classification	31
GNN Fault Location	80

Table 2. Model Training Time (min)

The GNN fault location model had a different performance compared to the classification models. For Scenario 1, the model was able to correctly detect the fault in with an accuracy of 65.83% for the top-1 prediction, this accuracy improved drastically if the prediction set was expanded. The top-5 predictions reached a 94.1% accuracy and a 98.36% accuracy was obtained for the top-10 predictions. In a similar way, the model for Scenario 2 maintained a similar performance than the baseline, only having a slightly lower accuracy, but still achieving an accuracy of over 98% for the top 10 predictions. This result is particularly good if it is compared to a random baseline, which would have a probability of 0.3% for one single bus and a 3% for ten random buses. Moreover, after representing the top-10 predictions of the model in the network all the predicted buses were clustered in a single area of the network, proving that the model really leverages the graph structure and understands how faults propagates.

After evaluating the SVM and the GBDT classification models with real data from SPEN showed a notable performance difference between models. The SVM model was able to correctly classify the majority of the dataset, identifying the samples with the biggest voltage drops as single phase faults and classifying the remaining samples with constant nominal voltage as ‘no fault’. On the other hand, the GBDT model didn’t predict correctly any samples, since it classified the majority of them as phase-to-phase faults without any voltage

readings that support these results. This result can help decide which method is more suitable to implement, SVM may be a more attractive option based on this results, however, it also has to be taken into account that the available voltage readings on every sample was very limited, which likely contributed to the low performance of the GBDT model.

Overall, these results confirm that ML-based techniques can deliver high accuracy and computational efficiency. In addition, the use of GNN-based approaches brings advantages to fault location since they leverage the network topology.

5.2 RECOMMENDATIONS FOR IMPLEMENTATION

This section provides some potential improvements that can be implemented by SPEN or any DSO that decides to use machine learning models for fault analysis in distribution networks. This was elaborated by observing the limitations and challenges encountered during the development of the project. The improvements are the following:

- Increase the number of features that are fed into the model. By doing this, the models accuracy would improve and more scenarios could be simulated, giving the model more flexibility. In this project, only voltage readings were used as inputs for the model. The use of current readings, which were not used in this project due to client data privacy but are available for the DSO, would allow the model to classify faults with more precision. In addition, using external factors like weather conditions or seasonal patterns could also be used to increase the models complexity.
- Increase the number of target fault types. This project only focuses on analyzing short circuits, however, the detection of other fault conditions like overvoltage, open circuits or voltage sags could be implemented. Since these models have the ability to understand all kinds of patterns they are suitable for multi-class classification. This would also improve the models functions by supporting more comprehensive fault monitoring strategies.
- Update the smart meter data acquisition strategy. One of the biggest challenges when analyzing the real smart meter data provided by SPEN was that the voltage was only

recorded every 30 minutes. Because of this, the voltage reading after a fault alarm could happen after the fault had been already cleared, not allowing for classification or location. Increasing the sampling frequency is not a feasible solution due to the amount of data that it would generate, therefore, a proposed strategy would be to configure smart meters to trigger additional recordings every time that an alarm is detected. In that way, the needed samples would be recorded, while minimizing the storage increase.

- Increase the smart meter coverage. The analysed network had 125 loads, which should be connected to a smart meter, however, in the real dataset only voltage readings from 36 smart meters were available per sample. If ML strategies are going to be implemented in the network, more data is needed to locate faults accurately. That is why increasing the number of smart meters that report their measurements would be very beneficial for model performance.
- Use validation techniques to improve data quality. During the preprocessing stage, it was detected that some of the smart meter readings were faulty and had to be removed, since they could distort the result. The use of validation techniques like range checks or outlier detection, can help ensure that all data points are correct or in a logical range. So before the data is feed to the model these techniques can detect anomalous readings and avoid corrupted samples.
- Use machine learning techniques to handle missing data. If a data set has an erroneous measure is detected, or the data from a smart meter is missing, a method to fill missing values is necessary. Some machine learning techniques, such as K-Nearest Neighbors (KNN) imputation or Multivariate Imputation by Chained Equations (MICE), can be used to supply missing values [39].

6 CONCLUSION

The objective of the project was to evaluate the feasibility and performance of three supervised machine learning approaches for fault detection, classification and location in meshed distribution networks. In order to see if these models were able to maintain their predictive performance under challenging future scenarios, two cases were analysed, the current network configuration and a scenario with high low carbon technologies penetration.

All the classification models achieved an accuracy above 99% across both scenarios, which confirms that ML-based methods are viable and reliable complements to traditional protection schemes and will be able to be used under more complex operational conditions introduced by DER penetration.

After analysing the performance of the classification models, the one that stands out is the SVM since even if the three models achieved similar accuracy levels when tested with simulated data, the computational cost of the SVM was significantly lower, with training times up to seven times shorter than the other models. In addition, despite the non-ideal conditions for testing due to the low amount of real data, when testing GBDT and SVM with SPENs real data, only the SVM model correctly predicted the majority of the cases, reinforcing its suitability for its deployment.

The results for the GNN fault location models showed great performance, since, for the top-one prediction, the model has an accuracy above 66%, which increases to over 90% with the top-5 predictions and almost 99% when taking the top-10 predictions. Implementing this model can help DSOs to increase reliability by clearing faults much faster thanks to knowing the exact bus were the fault occurred and it highlights the benefits of leveraging the network's graph structure.

Overall, the integration of these ML approaches into distribution network fault management systems could bring substantial benefits: faster fault detection, better classification for targeted response and increased location accuracy, especially in meshed LV networks where

traditional methods face inherent challenges. Moreover, the demonstrated robustness under high LCT penetration suggests that these techniques can support the ongoing energy transition, ensuring protection systems remain effective as networks evolve. However, despite the promising results of many research studies about the use of AI in fault management, there are no known commercial AI-based protection devices available today that would replace conventional relays such as IDMT, differential or distance protection. This is due to the high standards for safety, reliability and predictability required for grid protection systems, which highlights the further need for research in this domain to develop solutions that can take advantage of the AI/ML fault classification and location models.

Future work should focus on improving the models with more input features, like current or weather conditions; extending the target fault types to cover more fault scenarios; implementing data pre-processing techniques to detect wrong measurements and handle missing data; and improving the smart meter's data acquisition strategies to ensure higher temporal resolution, enabling a more accurate fault analysis.

7 REFERENCES

[1] j. D. Clover, T. J. Overbye and M. S. Sarma, Power System Analysis and Design, CENGAGE Learning, 2017.

[2] J. D. L. Cruz, E. Gómez-Luna, M. Ali, J. C. Vasquez and J. M. Guerrero, "Fault Location for Distribution Smart Grids: Literature Overview, Challenges, Solutions, and Future Trends," *Energies,* vol. 16, no. 2280, 2023.

[3] D. Gomes and C. Ozansoy, "High-impedance faults in power distribution systems: A narrative of the field's developments," *ISA Transactions,* vol. 118, pp. 15-34, 2021.

[4] V. Sharma, S. M. Aziz, M. H. Haque and T. Kauschke, "Effects of high solar photovoltaic penetration on distribution feeders and the economic impact," *Renewable and Sustainable Energy Reviews,* vol. 131, no. 110021, 2020.

[5] McKinsey & Company, "Electric vehicles—what's ahead," McKinsey Center for Future Mobility, 2021. [Online]. Available: https://www.mckinsey.com/features/mckinsey-center-for-future-mobility/our-insights/electric-vehicles-whats-ahead. [Accessed 30 07 2025].

[6] International Energy Agency, "Global EV Outlook 2025. Trends in electric car markets," 2025.

[7] UK Government, "The Electric Vehicles (Smart Charge Points) Regulations 2021," 2021. [Online]. Available: https://www.legislation.gov.uk/uksi/2021/1467/contents/made. [Accessed 07 2025].

[8] A. Ahmad, M. Khalid, Z. Ullah, N. Ahmad, M. Aljaidi, F. A. Malik and U. Manzoor, "Electric Vehicle Charging Modes, Technologies and Applications of Smart Charging," *Energies,* vol. 15, no. 9471, 2022.

[9] M. S. Mastoi, S. Zhuang, H. M. Munir, M. Haris, M. Hassan, M. Alqarni and B. Alamri, "A Study of Charging-Dispatch Strategies and Vehicle-to-Grid Technologies for Electric Vehicles in Distribution Networks," *Energy Reports,* vol. 9, pp. 1777-1806, 2023.

[10] Renogy United States, "How Many Watts Does a Heat Pump Use," 18 11 2024. [Online]. Available: https://www.renogy.com/blog/how-many-watts-does-a-heat-pump-use?srsltid=AfmBOopOQSAPylTAuqc5pTyUflN99oDqnuescP_riVNNL6TXz7EYNaeD. [Accessed 28 07 2025].

[11] National Infrastructure Commission (NIC), "National Infrastructure Assessment," 2018.

[12] Digital Ocean, "Decision Trees Made Simple: Machine Learning Explained," 26 6 2025. [Online]. Available: https://www.digitalocean.com/community/tutorials/decision-trees-machine-learning-explained. [Accessed 7 2025].

[13] MathWorks, "Introduction to Support Vector Machine (SVM)," [Online]. Available: https://uk.mathworks.com/discovery/support-vector-machine.html#how-svm-works.

[14] Medium, "Activation Functions in Neural Networks," 15 10 2024. [Online]. Available: https://medium.com/@prasanNH/activation-functions-in-neural-networks-b79a2608a106. [Accessed 7 2025].

[15] A. A. Majeed, A. S. Altaie, M. Abderrahim and A. Alkhazraji, "A Review of Protection Schemes for Electrical Distribution Networks with Green Distributed Generation," *energies,* vol. 16, no. 7587, p. 31, 2023.

[16] S. Javadian, M.-R. Haghifam, S. Bathaee and M. F. Firoozabad, "Adaptive Centralized Protection Scheme for Distribution Systems with DG Using Risk Analysis for Protective Devices Placement," *International Journal of Electrical Power & Energy Systems,* vol. 44, pp. 337-345, 2013.

[17] M. R. M. Cruz, D. Z. Fitiwi, S. F. Santos, S. J. P. S. Mariano and J. P. S. Catalão, "Prospects of a Meshed Electrical Distribution System Featuring Large-Scale Variable Renewable Power," *Energies,* vol. 11, no. 3399, 2018.

[18] Y. Su, Z. Dong and Y. Zhang, "Active distribution network management system for the new power system," in *2025 2nd International Conference on Smart Grid and Artificial Intelligence (SGAI)*, 2025.

[19] Gobierno de España, "Plan Nacional Integrado de Energía y Clima (PNIEC) 2023-2030," Ministerio para la Transición Ecológica y el Reto Demográfico, Madrid, Spain, 2023.

[20] Electricity North West Limited, "Distribution Future Electricity Scenarios 2024," 2025. [Online]. Available: https://www.enwl.co.uk/about-us/future-forecasting/dfes/.

[21] J. P. Chaves, "Operation and planning of future distribution systems, Drivers of the change," Universidad Pontificia Comillas (ICAI), Madrid, Spain, 2023.

[22] N. Damianakis, G. R. C. Mouli and P. Bauer, "Grid impact of photovoltaics, electric vehicles and heat pumps on distribution grids — An overview," *Applied Energy,* vol. 380, no. 125000, 2025.

[23] M. Z. u. Abideen, O. Ellabban and L. Al-Fagih, "A Review of the Tools and Methods for Distribution Networks' Hosting Capacity Calculation," *Energies,* vol. 13, no. 2758, 2020.

[24] "AI in the 1980s and 1990s: The Decades That Changed Everything.," Medium, 4 Nov 2024. [Online]. Available: https://medium.com/@titanmonk90/ai-in-the-1980s-and-1990s-the-decades-that-changed-everything-a1e8c7a4735a.

[25] H. Rezapour, S. Jamali and A. Bahmanyar, "Review on Artificial Intelligence-Based Fault Location Methods in Power Distribution Networks," *Energies,* vol. 16, no. 4636, 2023.

[26] M. M. Zaben, M. Y. Worku, M. A. Hassan and M. A. Abido, "Machine Learning Methods for Fault Diagnosis in AC Microgrids: A Systematic Review," *IEEE Access,* vol. 12, pp. 20260-20298, 2024.

[27] J. Abubakar and A. Abdulkareem, "Critical Review of Fault Detection, Fault Classification and Fault Location Techniques for Transmission Network," *Journal of Engineering Science and Technology Review,* vol. 15, pp. 156-166, 2022.

[28] K. Chen, J. Hu, Y. Zhang, Z. Yu and J. He, "Fault Location in Power Distribution Systems via Deep Graph Convolutional Networks," *IEEE Journal on Selected Areas in Communications,* vol. 38, no. 1, pp. 119-131, 2020.

[29] W. Li and D. Deka, "PPGN: Physics-Preserved Graph Networks for Real-Time Fault Location in Distribution Systems with Limited Observation and Labels," in *Conference: Hawaii International Conference on System Sciences*, Hawaii, 2023.

[30] L. Liang, H. Zhang, S. Cao, X. Zhao, H. Li and Z. Chen, "Fault location method for distribution networks based on multi-head graph attention networks," *Sec. Process and Energy Systems Engineering,* vol. 12, 2024.

[31] Pacific Gas and Electric Company (PG&E), "Electric Program Investment Charge (EPIC) 3.20 Data Analytics for Predictive Maintenance," 2023.

[32] Fundamentals, "China uses AI to swiftly and successfully address grid outages," 30 05 2023. [Online]. Available: https://fundamentals.tech/china-uses-ai-to-swiftly-and-successfully-address-grid-outages/. [Accessed 7 2025].

[33] Power Technology, "Artificial intelligence: who are the leaders in AI-assisted power fault monitoring for the power industry?," 5 01 2024. [Online]. Available: https://www.power-technology.com/data-insights/innovators-ai-assisted-power-fault-monitoring-power/?cf-view. [Accessed 07 2025].

[34] Electric Power Research Institute, "OpenDSS manual," 2023. [Online]. Available: https://sourceforge.net/projects/electricdss/.

[35] Electric Power Research Institute (EPRI), "OpenDSS Documentation," 2024. [Online]. Available: https://opendss.epri.com/OpenDSSCustomScripting.html.

[36] ecoexperts, "Which Countries Are Winning the European Heat Pump Race?," 21 04 2023. [Online]. Available: https://www.theecoexperts.co.uk/heat-pumps/top-countries. [Accessed 7 2025].

[37] H. D, "Decision Trees Splitting Criteria For Classification And Regression," [Online]. Available: https://machinelearning-basics.com/decision-trees-splitting-criteria-for-classification-and-

regression/#:~:text=Friedman%20MSE%20criteria%20computes%20also,left%20side%20of%20the%20node. [Accessed 07 2025].

[38] Wikipedia, "Radial basis function kernel," 3 06 2025. [Online]. Available: https://en.wikipedia.org/wiki/Radial_basis_function_kernel. [Accessed 07 2025].

[39] L. Joel, W. Doorsamy and B. Paul, "On the Performance of Imputation Techniques for Missing Values on Healthcare Datasets," 13 03 2024. [Online]. Available: arXiv:2403.14687. [Accessed 08 2025].

[40] A. D. Patil, A. Ghasemi and H. d. Meer, "Analysis of protection blinding in active distribution grids," *IET Renewable Power Generation,* vol. 18, no. 11, pp. 1783-1797, 2024.

[41] B. L. H. Nguyen, T. V. Vu, T. -T. Nguyen, M. Panwar and R. Hovsapian, "Spatial-Temporal Recurrent Graph Neural Networks for Fault Diagnostics in Power Distribution Systems," *IEEE Access,* vol. 11, pp. 46039-46050, 2023.

ANNEX. ALIGNMENT WITH SDG

SDGs are a group of 17 goals that are part of the 2030 Agenda for Sustainable Development by the United Nations. This project aligns with several of these goals, which are closely related to energy and climate action. The use of ML for fault detection, classification and location in meshed networks with different LCT penetration scenarios helps to develop the energy sector and aligns with SDG 7, SDG 9, SDG 11 and SDG 13.

SDG 7 is Affordable and Clean Energy. Its objective is to ensure that everyone can have access to reliable, sustainable and modern energy systems. This project helps to increase stability and reliability in energy systems that integrate renewable energy sources by improving fault detection with the use of ML. This facilitates the integration of LCTs such as solar photovoltaic panels, electric vehicles and heat pumps, which makes these technologies more accessible for everyone while ensuring that the grid remains stable and secure during its operation. In addition, the penetration of LCTs like solar panels can help reduce the price of the electricity, allowing more people to have access to it at an affordable price.

SDG 9 is Industry, Innovation and Infrastructure. This objective focuses in the need of resilient infrastructure, to promote the industries of every country and to foster innovation. The use of AI methods to detect, classify and locate faults is an innovative way of fault management in distribution networks, which if it is implemented by DSOs it can help increase the reliability and advance in the use of AI technologies. This project is also closely aligned with the need of resilient infrastructure, since it supports the automation and digitalization of distribution network infrastructure, by leveraging the readings of smart meters that are being installed in all customer loads for predicting the state of the network.

SDG 11: Sustainable Cities and Communities. This project supports this objective, whose aim is to make more safe, resilient and sustainable cities. The use of meshed topologies for

distribution networks help to improve resilience in the electricity supply and this project enables the use of these kind of topology. However, meshed networks with LCTs can have safety problems with increased fault currents and bidirectional power flows, that's why this project also helps improve safety by providing sophisticated tools to manage faults effectively. These tools help reduce the duration and frequency of outages which is even more critical in cities, where electricity is related to essential services such as healthcare, public transportation, water supply, and communication systems.

SDG 13: Climate Action. This goal calls for urgent measures to combat climate change and its impacts. One of the central challenges in achieving decarbonization targets is the safe and reliable integration of LCTs into the distribution grid. Without effective fault management strategies, increased DER penetration can destabilize the network, limiting the scale at which clean technologies can be deployed. This project addresses that challenge by providing a methodology that facilitates the secure and stable operation of grids under high levels of LCT integration. Thanks to this, DSOs can control and ensure that the maximum capacity of renewable technologies is installed on the grid safely, directly supporting the energy transition.

In summary, this project contributes to the access to affordable and clean energy for everyone, to more resilient and automated infrastructure, to a more sustainable urban development and to climate change mitigations, all by using AI to analyse faults in meshed networks with LCT penetration.