Using Unsupervised Machine Learning to Analyse PRPD Patterns for High Voltage Applications

Third Year Individual Project – Final Report

April 2025

Yaseen Ahmed

10969829

Supervisor:

Dr. Qiang Liu

Contents

Word count: 9153

Abstract

This project investigated the use of an unsupervised deep learning approach to analyse and cluster Phase Resolved Partial Discharge (PRPD) patterns from high-voltage experiments, with the purpose to support the rise in demand of consistent and reliable energy. The aim was to uncover hidden structural relationships within PRPD images, that are often overlooked by manual inspection, by using AI-based tools on unlabelled data. A feature extraction process was developed combining wavelet scattering transform, a custom convolutional neural network for deep encoding, uniform manifold approximation and projection for dimensionality reduction, and K-means for clustering. Using clustering metrics such as silhouette score (0.7401), Calinski-Harabasz index (2149.32), and Davies-Bouldin index (0.3572), the model successfully grouped 388 PRPD images into five distinct clusters. These clusters revealed meaningful groupings aligned with experimental conditions such as insulation type, applied voltage, and discharge severity. Key features that influenced clustering included discharge intensity, phase-band width, and discharge symmetry. The findings highlighted the capability of unsupervised AI to extract latent features from complex electrical data, offering scalable, interpretable, and label-free solutions for condition monitoring in high-voltage systems. This approach can enhance predictive maintenance strategies and inform future intelligent diagnostics in power networks.

Keywords: Phase Resolved Partial Discharge, Convolutional Neural Network, Wavelet Scattering Transform, Unsupervised, Machine Learning.

Declaration of Originality

I hereby confirm that this dissertation is my own original work unless referenced clearly to the contrary, and that no portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual Property Statement

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Qiang Liu, for their invaluable guidance, encouragement, and support throughout the course of this project. Their expertise and insightful feedback were instrumental in shaping the direction and quality of this work.

I would also like to thank the PhD student Mr Adam Nor for conducting the partial discharge experiments and generously providing the dataset used in this study. His contributions were essential to the development and validation of the model, and I am grateful for the time and effort he dedicated to supporting this project.

1 Introduction

1.1 Background and Motivation

Electrical power systems are experiencing a transformative shift as the global demand for cleaner, more reliable, and efficient energy continues to rise [1]. As power grids are becoming increasingly complex, so does the need for intelligent monitoring systems that can predict and prevent equipment failure before it occurs [2]. High Voltage (HV) is important as it allows electrical power to be transmitted over long distances at a lower current, which reduces the energy loss. However, the stability of HV equipment heavily depends on the condition of its insulation. While HV enables more efficient system design, it also imposes risks, particularly to health and safety, such as electrical arcing, which can occur at approximately 1 kV per centimetre [3]. To address these risks, electrical systems need good insulation to withstand and contain high voltages. Insulators are non-electrically conductive materials that inhibit the flow of free electrons and create barriers between parts of a circuit [4]. When a section of high voltage insulation begins to degrade, it may no longer withstand the high electrical stress, resulting in a breakdown. This can lead to a phenomenon known as Partial Discharge, where localised electrical discharges occur within the insulation, without fully bridging the electrodes [5].

Due to this, the application of AI becomes crucial in identifying early signs of insulation degradation that may be overlooked by human observers, enabling predictive maintenance and risk mitigation in HV systems, enhancing system reliability and operational safety.

1.2 Partial Discharge (PD)

According to IEC Standard 60034-27-1 [5], partial discharge is a small electrical discharge (spark) that occurs within the insulation of HV equipment when the electric field becomes too strong. It happens when part of the insulation breaks down, but the discharge does not completely bridge the electrodes. Over time, repeated PD activity can damage the insulation, eventually leading to equipment failure.

PD typically occurs in defects in the insulation such as air bubbles, cracks, or surface irregularities. These areas become ionised under high voltage, which form tiny plasma channels called streamers [62]. This ionisation process produces charge pulses that can be detected and analysed [6,7].

Detecting PD early is important because it allows engineers to identify and fix problems before they cause expensive or dangerous breakdowns in systems like transformers, cables, and switchgear [8].

1.3 Phase Resolved Partial Discharge Patterns

Phase Resolved Partial Discharge (PRPD) patterns are a visual tool used to diagnose insulation faults in HV equipment. They are graphs that map the magnitude of partial discharge on the y-axis against the phase angle of the applied AC voltage on the x-axis. This creates a two-dimensional (2D) pattern that can reveal the type, severity, and timing of insulation defects [9].

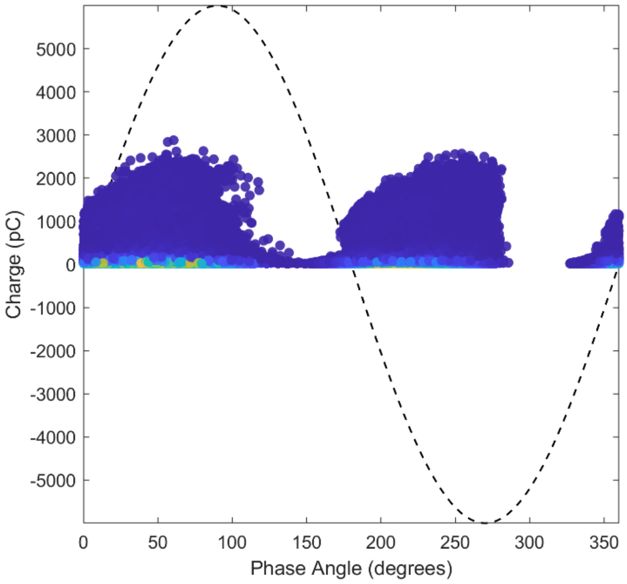


Figure 1.3.1: *Example of a PRPD pattern showing discharge activity plotted against the AC phase angle. Each point represents a partial discharge event, with charge magnitude (pC) on the y-axis and phase angle (°) on the x-axis. The sinusoidal waveform (dashed line) indicates the applied voltage cycle. Distinct clusters of activity suggest insulation defects occurring at specific phases.*

Each defect type, such as voids, cracks, or surface discharges, produces an original and unique pattern, like how star constellations have their own distinct shapes. For instance, Figure 1.3.1 presents a PRPD plot obtained from an experiment involving solid insulation with an applied voltage of 30 kV. These patterns allow engineers to detect problems early, before they lead to equipment failure [9].

PRPD graphs are often analysed by experts, but as the amount of data grows, manual analysis becomes difficult. This has led to an interest in using Artificial Intelligence (AI) tools that can automatically recognise these patterns, to make classifying the source of faults faster, more accurate, and scalable [10, 37].

1.4 Convolutional Neural Networks

Advancements in machine learning and deep learning have enabled the automation of PRPD graph analysis and classification. A prominent architecture in this field is Convolutional Neural Networks (CNNs), which is a class of deep learning models especially designed for processing image-like data [11]. CNNs have demonstrated strong performance in a range of pattern recognition tasks and are now commonly used in fault detection within electrical engineering due to their ability to extract and learn from complex visual features [12, 15, 24, 36, 38].

PRPD graphs contain unique spatial patterns associated with discharge magnitudes across the phase of an AC cycle. CNNs are well suited for analysing these types of patterns because they automatically learn hierarchical spatial features, starting with simple edges to more complex discharge textures [12]. This capability allows CNNs to model non-linear connections in the data that traditional statistical methods may overlook. Furthermore, CNNs mimic human visual perception – just as how an expert identifies subtle differences in discharge patterns, CNNs can distinguish between different defect types based on PRPD spatial signatures [13].

Compared to conventional techniques, CNNs offer three major advantages: (i) they do not require manual feature engineering, which removes human bias and intervention [14]; (ii) they are robust to noise and adaptable to unseen patterns [14]; (iii) they are scalable, allowing them to be implemented in real time monitoring systems [15]. Studies report classification accuracies exceeding 95% when CNNS are used on PRPD data, outperforming traditional approaches which often only achieve 80% [16].

CNNs can be implemented in both supervised and unsupervised learning environments. Supervised CNNs are trained on labelled datasets, to learn mappings between the input PRPD images with their predefined classification. In contrast, unsupervised CNNs learn to group similar images without labels, which is useful when patterns are unknown or ambiguous [17]. One of the limitations of supervised learning with CNNs is the need for labelled data, which is often scarce in partial discharge research [18]. Labelled data requires expert analysis, which is often time consuming and costly. To mitigate this, unsupervised approaches are increasingly explored offering a scalable way to organise and pre-label large datasets [17].

1.5 Aims and Objectives

1.5.1 Project Aim

The aim of this project was to develop and evaluate an AI-based approach for analysing and clustering PRPD graphs, with the goal of uncovering patterns and relationships that may not be easily recognised by human experts. This approach also aims to make PRPD graph classification and analysis accessible to individuals without specialised expertise in HV engineering or PD diagnostics.

1.5.2 Project Objectives

- To collect and preprocess PD data to create PRPD graphs under varying initial conditions such as applied voltage, temperature, and insulation materials.
- To design and implement an unsupervised CNN model capable of clustering PRPD images based on spatial and discharge features.
- To evaluate the clustering performance using metrics such as silhouette scores, cluster purity, and visual similarity.
- To investigate whether the identified clusters correspond to specific experimental conditions or discharge characteristics.
- To demonstrate how AI-based tools can make the analysis of complex PRPD data more intuitive and usable for non-experts.

2 Literature Review

2.1 Introduction

This literature review provides an overview of the current state of research in PD pattern recognition, with a focus on the application of AI methods to PRPD graph analysis. It begins by summarising the foundational work in traditional PD classification, and the transition towards AI based models. It then narrows the focus to studies that employ unsupervised learning techniques, such as clustering and image transforms, to discover underlying structures within PRPD image datasets. Each methodology is examined in terms of its implementation, effectiveness, and limitations. Finally, this section highlights the existing gaps in literature and introduces methodology appropriate for this project.

2.2 Search Strategy for Literature

The search for literature was conducted through a systematic and targeted search strategy aimed at identifying relevant and high-quality research studies on PD analysis, with a focus on PRPD pattern recognition using AI techniques. To ensure comprehensive coverage, the following academic databases were used: ‘IEEE Xplore’, ‘IET Digital Library’, and ‘Google Scholar’.

The search terms included combinations of key words such as “Partial Discharge”, “PRPD Pattern”, “Machine Learning”, “Convolutional Neural Networks”, “Unsupervised Clustering”, and “Feature Extraction”. Boolean operators such as AND and OR were used to refine the searches for specificity and relevance.

Studies were then shortlisted on the relevance to PRPD image analysis, the use of AI or ML models (supervised or unsupervised), and their application to high volage equipment diagnostics. Furthermore, preference was given to recent publication (2010 onward) to ensure inclusion of the latest advancements in deep learning and pattern recognition. However, foundational works on traditional PD analysis were included to provide context and contrast with AI-based approaches.

All collected literature was thematically categorised into sections based on methodology (e.g. supervised or unsupervised), experimental focus (e.g. PD analysis), and evaluation strategies (e.g. accuracy, silhouette scores, or confusion matrix).

2.3 Traditional Approaches to PD Classification

2.3.1 Traditional Statistical Methods

Early techniques for PD classification involved a visual interpretation of PD pulses displayed on an oscilloscope trace [19]. Experts would then analyse the phase position and horizontal spread of the pulses guided by knowledge bases such as CIGRE (1969) [19, 20]. These methods were highly dependent on professional experience and lacked automation or scalability [21].

The development of digital PD detectors enabled PRPD patterns to emerge [22], which could be analysed using statistical features. Metrics such as skewness, kurtosis, and standard deviation of discharge pulses and charge distributions were extracted [23]. These statistical metrics were used to train traditional classifiers such as, Support Vector Machines (SVM), Radial Basis Function Networks, Naïve Bayes, and AdaBoost [24]. SVMs in particular became a conventional choice for PRPD pattern recognition [25].

While these models improved accuracy, they still required experts to extract features and were limited in handling noise or overlapping discharge data [21].

2.4 CNNs for Supervised Classification

CNNs are a powerful development in machine learning for pattern recognition and image analysis, leading to a widespread adoption across various sectors including health and security [26, 27], with research indicating it to be more effective than traditional methods [21].

Most literature explores the use of supervised CNNs for PD classification across high voltage equipment, with a focus on enhancing the performance and interpretability of PRPD pattern analysis. For instance:

2.4.1 Overcoming SVM Limitations: A CNN Approach to Complex PRPD Scenarios

Butdee et al.'s study [15] proposed a simple and lightweight CNN for classifying PRPD patterns, outperforming SVM in scenarios with complex or overlapping discharge sources. It challenged SVMs, the convention for PRPD analysis, which struggles to generalise across complex discharge scenarios in noisy and complex PD scenarios. This issue becomes significant when dealing with overlapping defect types or changes in discharge severity.

The CNN used here is trained on a supervised learning model consisting of 1 convolutional layer with ReLU activation, 1 max pooling layer, 1 fully connected layer for feature mapping, and 1 fully connected layer with SoftMax. This terminology is explained later on. The model was trained on synthetic and real PRPD patterns, and achieved over 95% accuracy in classifying PRPD types, outperforming SVM which struggled with overlapping or ambiguous patterns, achieving only 80% accuracy. The performance was evaluated by calculating accuracy and using a confusion matrix.

Although supervised, the core principles of this paper leverage CNNs for automatic spatial feature extraction from PRPD images and directly informed the design of this project's unsupervised CNN.

2.4.2 Tackling Overlapping Defects in PRPD with Class-Specific CNNs

Mantach et al. [28] proposed a novel CNN architecture that identified multiple defect types from PRPD data. It addressed a major limitation in PRPD analysis: traditional classifiers fail on PRPD patterns from overlapping sources.

By using a shared feature extraction backbone with seven class-specific CNNs, the model learns to extract generalisable features, significantly reducing false negatives when defects overlap.

Their findings support the idea that PRPD images hold sufficient structure to learn without labels while still being able to separate underlying defects, reinforcing this project's use of unsupervised methods on raw PRPD images.

2.5 Unsupervised Clustering

While the majority of literature focuses on supervised CNNs for PD classification, there is an emerging interest in unsupervised deep learning techniques involving CNNs. There are multiple papers which support this including:

2.5.1 Clustering Partial Discharge Sources Using Unsupervised Deep Feature Learning

This study by Mantach et al. [29] explored unsupervised convolutional autoencoders to identify PD sources without prior labels. 1D time-domain PD waveforms were passed through a CNN encoder to extract features and two methods were explored for clustering: (i) Fixed k-clustering using K-means, to choose the optimum number of clusters; (ii) Adaptive clustering using cosine similarity, where every new sample is compared to the existing cluster, and a new cluster is formed if the similarity is below a certain threshold.

Results suggested that CNNs can uncover meaningful latent structure, closely aligning with the aim of this project.

2.5.2 Fixed Template vs. Adaptive Clustering: Lessons from CCNet

Abubakar et al. [21] introduced a template-based method called Cosine Cluster Net (CCNet) that compares PRPD images to a small set of pre-labelled patterns, using cosine similarity. While efficient, it fails with overlapping sources.

Unlike Abubakar's approach, this project will be using clustering instead of template matching, but both relied on K-means to partition and interpret patterns in PRPD images. This project improved on Abubakar's work by following a clustering approach which may be better suited for PRPD patterns with multiple classes, as it does not assume a fixed template. Lastly, while Abubakar validated based on similarity score to the predefined labels, this project evaluated based on metrics such as silhouette score and a visual cluster inspection, which is a more generalisable and label-independent method.

2.6 Wavelet Scattering Transform in CNNs

The Wavelet Scattering Transform (WST) presented a compelling method for enhancing CNNs in the analysis and classification of PRPD patterns. While a lot of research primarily focuses on Continuous Wavelet Transform (CWT) scalograms as a direct input to CNNs for PD classification [30, 31], the inherent properties of WST and its relationship with deep learning architectures suggest a powerful synergistic approach [32].

WST can serve as an initial, non-trainable feature extraction layer for a CNN. Instead of directly feeding raw PRPD patterns into the convolutional layers of a CNN, the patterns can first be passed through a WST to generate a set of translation invariant and deformation-stable scattering coefficients [32, 33, 34]. These coefficients capture multi-scale information and inherent structural features of PRPD patterns, which can be fed as input channels to the convolutional layers of the CNN [32]. This approach leverages the strength of WST in providing a robust and pre-analysed feature extractor, enabling the learning capacity of the CNN to identify higher-level patterns from these features.

It is important to note that there are no sources that directly present research on the specific combination of WST and CNNs for PRPD analysis and classification. However, the documented effectiveness of CWT based scalograms as input for CNNs in PD classification [30, 31] coupled with the described relationship between WST and CNNs [32] and the advantages of WST in feature extraction [33, 34], strongly suggest this as a promising avenue for future research.

2.7 Summary and Design Justification

This literature review highlights the progression of PD classification techniques, from manual visual inspection and statistical feature extraction to the adoption of machine learning methods. Traditional approaches such as SVMs and Naïve Bayes relied on handcrafted statistical features derived from PRPD patterns, however, they were constrained by their dependence on expert knowledge and limited in noisy or complex discharge scenarios.

The introduction of CNNs offer a significant advancement, enabling automatic spatial feature extraction from PRPD images. Supervised CNN models, as demonstrated by Butdee et al. and Mantach et al. showed strong performance in classifying PRPD patterns and dealing with overlapping defect types. However, they relied on large, labelled datasets, which limits scalability in real-world conditions where labelled data is scarce.

Recent literature reveals an ongoing interest in unsupervised approaches for PRPD pattern recognition. In particular, the work of Mantach et al. using convolutional autoencoders, and Abubakar et al.'s CCNet illustrates the feasibility and effectiveness of learning from unlabelled PRPD data using clustering techniques like K-means. These studies support the adoption of unsupervised leaning to uncover latent features in PRPD datasets.

To improve feature representation prior to clustering, this project integrates WST into the feature extraction pipeline, as it provides a stable and translation-invariant representation of PRPD images that retains the structural characteristics of PRPD patterns. Although direct research on WST in PRPD classification is currently limited, its proven effectiveness in general image-based pattern recognition, as outlined in [34], justifies its use as a pre-processing stage in this work.

Design choices in this project were informed by gaps identified in the literature and contributes to the field by demonstrating how unsupervised learning can reveal interpretable relationships within unlabelled PRPD datasets and provides a scalable alternative to template-based or supervised methods for HV insulation diagnostics.

3 Methodology

3.1 Overview

This section details the step-by-step approach used to develop and evaluate an unsupervised deep learning pipeline for clustering PRPD graphs. The objective was to determine whether PRPD images, obtained from experiments with varied initial conditions, could be clustered in a way that reflects those underlying differences, without access to labels.

The pipeline integrates:

- Wavelet Scattering Transform for multi-scale invariant feature extraction.
- Custom Convolutional Neural Network for deep feature encoding.
- Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction.
- K-Means for unsupervised clustering.

Cluster quality was evaluated using silhouette score, Calinski-Harabasz index, and Davies-Bouldin Index. Flowcharts for the methodology is shown in Figures 3.1.1 and 3.3.1.

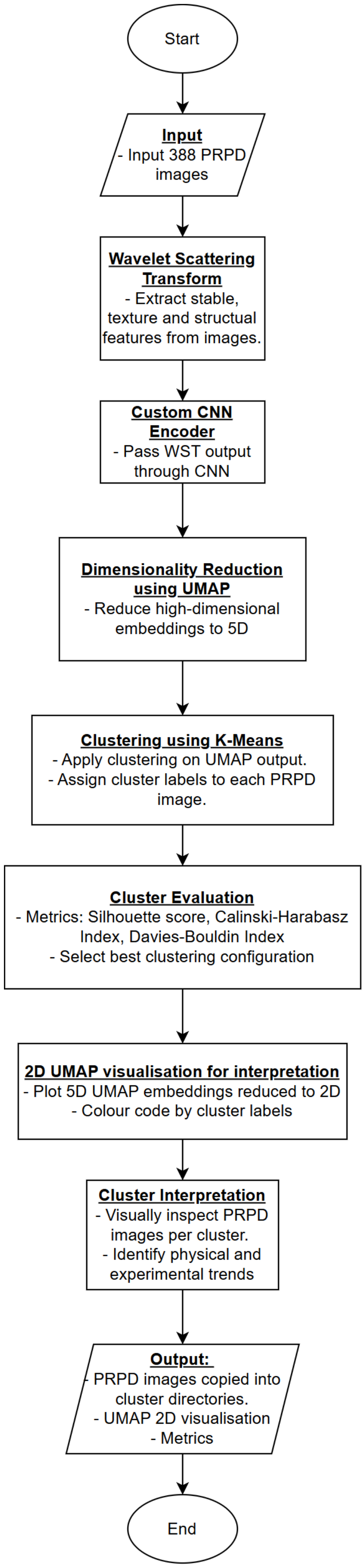

Start
Input
- Input 388 PRPD images
Wavelet Scattering Transform
- Extract stable, texture and structual features from images.
Custom CNN Encoder
- Pass WST output through CNN
Dimensionality Reduction using UMAP
- Reduce high-dimensional embeddings to 5D
Clustering using K-Means
- Apply clustering on UMAP output.
- Assign cluster labels to each PRPD image.
Cluster Evaluation
- Metrics: Silhouette score, Calinski-Harabasz Index, Davies-Bouldin Index
- Select best clustering configuration
2D UMAP visualisation for interpretation
- Plot 5D UMAP embeddings reduced to 2D
- Colour code by cluster labels
Cluster Interpretation
- Visually inspect PRPD images per cluster.
- Identify physical and experimental trends
Output:
- PRPD images copied into cluster directories.
- UMAP 2D visualisation
- Metrics
End

Figure 3.1.1: *Represents a flowchart for the clustering of the PRPD images using Python, TensorFlow. For the full code, please see Appendix D.*

3.2 Data Collection

PD data was generated from controlled experiments with varying initial conditions including electrode configuration, gap distance, temperature, viscosity, and applied AC voltage. Each experiment lasted two minutes, and output Comma Separated Values (CSV) files containing PD magnitude and corresponding phase angles. For more details on how these experiments are conducted, please see [35].

The initial conditions are summarised in Table 1.

Table 1: Initial conditions and categories.

Initial Conditions	Categories
Electrode Configuration **(EC)**	Needle-Plane (NP) Plane-Needle-Plane (PNP)
Insulation type **(IT)**	High Viscosity silicone oil Low Viscosity silicone oil Solid insulation
Temperature **(T)** in (°C)	20 80
Gap Spacing **(GS)** NP and PNP (mm)	25 45 50
Needle Protrusion **(Nee-pro)** PNP only (mm)	5 25
AC Voltage **(V)** Applied (kV)	11 – 50

This experiment setup ensures that the dataset captures real-world variations in PD behaviour due to environmental and physical changes.

3.3 PRPD Image Generation and Density Heatmap Construction

The transformation of raw PRPD data into image format suitable for unsupervised learning was implemented using MATLAB [61]. The PRPD images are generated from .csv files that contain two key columns: the phase angle at which each PD event occurred, and the corresponding charge magnitude. A .csv file was created for all 388 experiments.

3.3.1 Flowchart for Image Generation

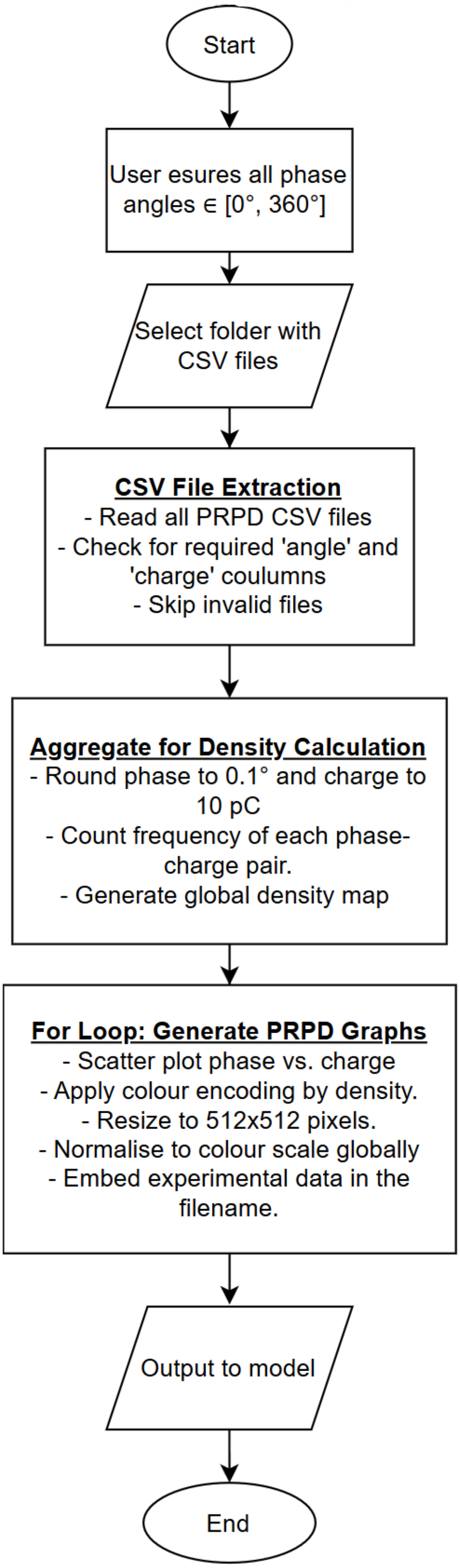

Start
User esures all phase angles ∈ [0°, 360°]
Select folder with CSV files
CSV File Extraction
- Read all PRPD CSV files
- Check for required 'angle' and 'charge' coulumns
- Skip invalid files
Aggregate for Density Calculation
- Round phase to 0.1° and charge to 10 pC
- Count frequency of each phase-charge pair.
- Generate global density map
For Loop: Generate PRPD Graphs
- Scatter plot phase vs. charge
- Apply colour encoding by density.
- Resize to 512x512 pixels.
- Normalise to colour scale globally
- Embed experimental data in the filename.
Output to model
End

Figure 3.3.1: *Represents a flowchart for the image generation code using MATLAB. For the full code, please see Appendix C.*

3.3.2 Data Aggregation and Density Mapping

To ensure consistency in the visual representation of PD density across all samples, a global density map was calculated prior to image generation. This map quantifies how often a particular phase, charge pair appears across the entire dataset and is used to assign consistent colour intensity levels to the plots. The following steps were performed for each CSV file:

1. **Phase and Charge Binning:**

- The phase angle was rounded to the nearest 0.1°.
- The charge magnitude was rounded to the nearest 10 pC.

This rounding was essential to turn the continuous PRPD data into a grid format, making it easier to count how often certain values appear, while keeping the important details.

2. **Density Map Construction:**

- An array (hash map) was created, using strings formatted as “angle_charge”. For example, “123.4_240”.
- For each angle-charge pair for every experiment, its corresponding key was incremented. This step builds a frequency count of all discharge event occurring at specific phase-charge coordinates.
- Once all CSV files were processed, the maximum frequency across the entire dataset was extracted to serve as a normalisation reference for density values.

3. **Density Assignment to Events:**

- When plotting an individual image, the rounded angle, charge coordinates for that sample were used to retrieve their corresponding density value from the global map.
- These values were then normalised by dividing the corresponding density value by the maximum observed density, ensuring all plots share a uniform scale for density visualisation at a value between 0 and 1, inclusive.

3.3.3 Graphical Rendering of PRPD Images

Each PRPD image was generated as a 512 x 512-pixel scatter plot, coloured by event density. A reference 50 Hz sinusoidal waveform was overlaid to indicate the AC phase cycle, clearly distinguishing PDs occurring in the positive and negative half-cycles. The plotting process involved:

- X-axis: Phase angle in degrees from 0° to 360°.
- Y-axis: Charge magnitude in picocoulombs (pC).

- Colour scale: using MATLAB’s parula colourmap to display relative density.
- Transparency: Each PD event was rendered with a partial transparency to enhance the visualisation of overlapping events.
- Colour Normalisation: The ‘clim’ function was used to fix the colour scale for all plots using the dataset’s max density value, preserving density uniformity.

To optimise the dataset for CNN processing, all PRPD images were generated without any axis labels, tick marks, legends, or text annotations. This decision was made to ensure the model focuses solely on learning the pattern from the distribution pattern of PD events, without being influenced by non-informative elements such as text or formatting.

Despite the removal of textual elements, all plots are rendered using a consistent and normalised axis scale. The phase angle is always displayed from 0° to 360°, and the charge magnitude on the y-axis is fixed to a -6000 pC to 6000 pC range. This guarantees that every image maintains visual consistency, allowing the CNN to interpret and compare features across the entire image dataset. An example is shown in Figure 3.3.3.

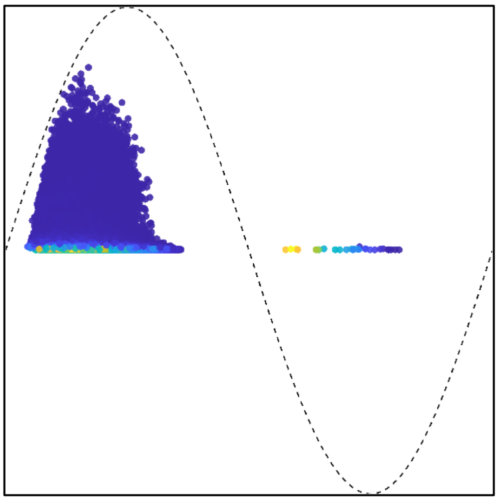


Figure 3.3.3: *Example PRPD PNG generated using the MATLAB image generation script. Colour intensity corresponds to PD event density, normalised using a global density map. Image is generated without axis labels or text.*

The output filename of each image encodes the source experimental conditions and is saved in high-resolution PNG format. This method allows each image to encode both the spatial location and the relative frequency of PDs, for capturing the patterns in PRPD data. By converting PD data into visually structured images, this process prepares the data for CNN analysis in subsequent stages.

3.4 Image Processing and Clustering

3.4.1 Wavelet Scattering Transform

Wavelet scattering transform [47] was applied to each PRPD image to extract stable and informative features. Unlike standard CNNs that learn data from applying filters, WST applies a cascade of predefined wavelet filters that are invariant to translations and small deformations [48], making it ideal for noisy and structured patterns such as PD distributions.

WST is implemented using the Kymatio [39] Python [63] library. For each 2D PRPD image:

1. The image is converted to greyscale and each pixel is normalised between 0 to 1, inclusive. This retains density information but saves computational resources by removing colour information.
2. A 2D scattering network is applied with the following parameters:
3. $\boldsymbol{J = 2}$**:** This controls the maximum scale of the wavelets and the low-pass filter, i.e. how much the image is smoothed. A larger J captures broader features. The scale is $2^J = 4$ which means that the wavelet will analyse structures up to 4 pixels across.
4. $\boldsymbol{L = 8}$**:** This controls the number of orientations of the wavelet filters. $L = 8$ ensures maximum directional coverage over the image every $(\frac{360}{8})$ ° = 45°.
5. The transform of each PRPD image involves the following steps:

a. **First Order Scattering:** The image (x) is convolved $(*)$ with a set of wavelets (ψ_{λ_1}), each with a specific scale (J), and orientation (L). Then it passes through a modulus operation, $(|\ |)$ that takes the absolute values. Finally, the result is smoothed with a low-pass filter (ϕ_J), which averages the output to make the output invariant to small shifts or noise. This is represented mathematically as [47]:

$$S_1 x = \left|x * \psi_{\lambda_1}\right| * \phi_J \tag{1}$$

b. **Second Order Scattering:** Takes the result of the first-order scattering, then convolves that with another set of wavelets. It then applies the modulus again, and another low-pass filter. This is represented mathematically as [47]:

$$S_2 x = \left|\left|x * \psi_{\lambda_1}\right| * \psi_{\lambda_2}\right| * \phi_J \tag{2}$$

This captures interactions between patterns like how edges are arranges, or how textures vary across a region.

The result is a 3D feature array that encodes descriptions of the image's content. Mathematically [47]:

$$Sx = \left\{x * \phi_J, \left|x * \psi_{\lambda_1}\right| * \phi_J, \left|\left|x * \psi_{\lambda_1}\right| * \psi_{\lambda_2}\right| * \phi_J, \dots\right\} \tag{3}$$

The first two dimensions represent spatial information from the original image and the third dimension contains all the wavelet information.

These arrays are then passed to the CNN, as inputs that already contain rich multiscale structure, allowing the CNN to learn higher-level abstractions instead of basic edges or textures.

3.4.2 CNN Feature Encoder

Understanding How CNNs Work

CNNS are composed of multiple layers that process input images in a hierarchical way [40]. The input layer receives the image data, i.e. the PRPD graph, and turns it into a matrix of pixel intensities [41]. The main building block of a CNN is the convolution layers, where it applies filters that slide over the input image to extract spatial features such as edges, gradients or textures. This produces a feature map, that highlights areas of the image where certain patterns are detected [42]. After each convolutional layer is applied to each image, activation functions such as Rectified Linear Unit (ReLU) introduce non-linearity forming more complex relationships in the network rather than linear relationships. ReLU is defined as [43]:

$$f(x) = \max(0, x) \tag{4}$$

This ensures that the function returns 0 for any negative input, and returns any positive value, x [43]. Another function applied after each convolution layer is batch normalisation which normalises the inputs of each layer to have a mean of 0 and a standard deviation of 1. This allows for stable and quicker feature learning [45].

To reduce computational complexity and enhance robustness to small shifts in the image, pooling layers are introduced. Pooling down samples the feature map by selecting the most prominent values in a small area [40].

In traditional supervised CNNs, the processed features pass through a fully connected layer which maps the extracted features to a specific class using labelled data [13, 40]. In unsupervised CNNs, however, the final classification layer is replaced, and the process features are clustered using algorithms like K-means and can be visualised using techniques like UMAP. Dropout is used to prevent overfitting, by randomly clearing half of the neurons that learned the data, so the network does not rely too heavily on a single neuron [46]. These features contain high-level information about the images, allowing the model to group visually similar PRPD graphs based on latent patterns, even without labels [44].

Design and Implementation of Custom CNN

A custom CNN was implemented in TensorFlow [62] to extract high-level features from the WST outputs. The network accepts an array of shape (32,32,81) as input which corresponds to the wavelet scattering-transformed images. The model consists of three stacked convolutional layers and pooling blocks, followed by dense (fully connected) layers to produce the final feature vectors.

The architecture consists of:

- **Input Layer:** Accepts 2D wavelet scattering coefficients with 81 channels.
- **Conv2D Layer 1:** 64 filters of size 3x3, with ReLU activation.
- **Batch Normalisation:** Applied after each convolutional layer to normalise the activations across the batch, stabilising learning and improving generalisation.
- **Conv2D Layer 2:** 128 filters, ReLU activation, followed by batch normalisation.
- **MaxPooling2D:** A 2x2 pooling operation to reduce spatial resolution while retaining key features.
- **Conv2D Layer 3:** 256 filters with ReLU activation and batch normalisation.
- **Conv2D Layer 4:** 512 filters with ReLU activation and batch normalisation.
- **MaxPooling2D:** Further reduces spatial resolution.
- **GlobalAveragePooling2D:** Aggregates spatial features into a compact vector by averaging across feature maps, reducing dimensionality.
- **Dense Layer:** 256-unit fully connected layer with ReLU activation.
- **Dropout (0.5):** Applied to prevent overfitting by randomly disabling 50% of neurons during training.
- **Dense Layer (Output):** 128-unit final layer used as the extracted feature vector for each input image.

This process was applied to the entire dataset to convert each WST input image into a 128D feature vector which was then passed to UMAP for dimensionality reduction. By separating feature learning from clustering, the CNN serves as a general-purpose encoder that transforms raw PRPD images into a format where discharge pattern similarity is captured for unsupervised grouping.

3.4.3 Dimensionality Reduction with UMAP

Following feature extraction from the CNN, the resulting 128D feature vector was compressed using UMAP [49] to enable more computationally efficient clustering. UMAP is a non-linear dimensionality reduction algorithm that preserves local and global structure of data by learning a low-dimensional representation of the high-dimensional feature vector [50].

The UMAP algorithm begins to construct a graph in the high-dimensional space, by identifying the nearest neighbours for each data point using cosine distance, which measures the angle between two vectors. Using cosine similarity better captures the structural relationship between PRPD image features [51].

UMAP then takes the Laplacian of this which describes the structure of the graph by how data points are connected. UMAP then performs eigen decomposition on this Laplacian to initialise a reconstruction of the graph in a lower-dimensional space [51], by encouraging close points in the high-dimensional space to remain close in the lower-dimensional space, while pushing apart points not connected in the original graph.

By the end of the process, the 128D CNN feature vector is mapped to a corresponding 5D point. This compact representation maintains the structure of the data, while filtering out irrelevant noise. Furthermore, this reduced representation is more computationally efficient. The parameters [51] used for UMAP included:

- **n_components = 5:** Tells the function to transform from 128D to 5D which retains sufficient feature information for accurate clustering.
- **metric = ‘cosine’:** For distance computation.
- **init = ‘spectral’:** For stable initialisation of the low-dimensional layout

This UMAP embedding was then passed to a K-means clustering algorithm, allowing similar PRPD images to be grouped based on the patterns extracted from the CNN and refined through UMAP.

3.4.4 Clustering and Evaluation

After dimensionality reduction with UMAP, the 5D feature embeddings were clustered using the K-means clustering algorithm. K-means is an unsupervised algorithm that partitions the dataset into K distinct clusters by minimising the variance of images within a cluster.

To determine the optimal number of clusters K, the algorithm runs iteratively through values from $K = 2$ to $K = \ 10$. For each K, clustering performance was evaluated using three key metrics.

- **Silhouette Score:**

  The silhouette score $s(i)$ for a sample i, quantifies how well the sample is assigned to its cluster $a(i)$ by comparing its similarity to other clusters $b(i)$.

  Mathematically, it is defined as [53]:

$$s(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)} \tag{5}$$

  Where $a(i)$ is the average distance between i to all the other points in the same cluster, and $b(i)$ is the average distance between i and points in the nearest cluster.

  Silhouette scores range from -1 to 1, with a good value higher than 0.5, and a strong value higher than 0.7. An average silhouette score close to 1 indicates well-separated clusters, while a score near 0 suggests overlapping clusters [54].

- **Calinski-Harabasz Index:**

  The Calinski-Harabasz index doesn't have a fixed range, but higher values indicate better clustering. This index is a ratio of within-cluster dispersion, to dispersion between clusters. It is calculated using this equation [55]:

$$CH = \frac{T_r(B_k)}{T_r(W_k)} \times \frac{n - k}{k - 1} \tag{6}$$

  Where $T_r(B_k)$ is the sum of the between-cluster dispersion, $T_r(W_k)$ is the sum of within-cluster dispersion matrix, n is the total number of data points, and k is the number of clusters. A high CH score means clusters are tight, and far apart [55].

- **Davies-Bouldin Index:**

  This index evaluates intra-cluster similarity and differences. Lower values indicate better clustering, with well separated and tight clusters.

  The formula is [56]:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max\left(\frac{\sigma_i + \sigma_j}{M_{ij}}\right), j \neq 1 \tag{7}$$

  Where S_i is the average distance of all points in cluster i to the centre of cluster i, M_{ij} is the distance between the centres of cluster i and cluster j, and k is the number of clusters.

For each cluster i, it finds the most similar neighbouring cluster j, and averages those over all clusters. A low index suggests low intra-cluster distances and high inter-cluster distances, which is preferred [56].

3.4.5 Cluster Visualisation, Validation, and Results

A 2D visualisation of the clustered data was generated using a projection of the first two UMAP dimensions [52]. Although clustering was performed to a 5D UMAP embedding space, reducing the dimensionality further to two dimensions allowed for a visual interpretation of the cluster results.

Each PRPD image – after being processed through wavelet scattering and the CNN feature extractor – was represented as a 5D vector. The first two components of the UMAP embedding were selected for plotting, as they captured the most dominant relationships between feature vectors post reduction.

A scatter plot was created where:

- Each point corresponds to a single PRPD image.
- The (x, y) coordinates are taken from the first two UMAP dimensions.
- The point's colour indicates the cluster it was assigned to by the K-means algorithm.

To examine the relationship between cluster assignments and experimental conditions, the metadata encoded in the image filenames was leveraged. Since filenames included information such as gap spacing, temperature, and electrode configuration, images within each cluster was traced back to their filename, and to their experimental conditions. If images from similar experimental conditions were grouped together consistently, it would suggest that the unsupervised model was able to detect latent correlations between discharge characteristics and test setup parameters. Cluster accuracy was calculated by identifying the dominant true class within each cluster, then determining the proportion of correctly grouped samples by subtracting the number of anomalies and dividing by the total number of samples.

All images were copied into cluster-specific directories, and a text file was generated listing all filenames and cluster assignments, allowing for traceability and manual review.

4 Results and Discussion

4.1 Introduction

The aim of this section is to present the results of the unsupervised learning pipeline and evaluate its effectiveness in identifying structure in PRPD images. Various clustering metrics are used to assess performance, followed by a visualisation of the five identified clusters. Each result is accompanied by discussion to contextualise the findings.

4.2 Cluster Evaluation Metrics

The optimal number of clusters was selected based on multiple unsupervised clustering metrics. After testing values of k between 2 and 10 and evaluating silhouette scores for each, the optimal value was determined to be $k = 5$, which provided the best compromise between intra-cluster similarity and inter-cluster separation. The results are shown in Table 2.

Table 2: Clustering Evaluation Metrics for $k = 5$

Metric	Value
Silhouette Score	0.7401
Calinski-Harabasz Index	2149.32
Davies-Bouldin Index	0.3572

- A high silhouette score of 0.7401 suggest strong cohesion within clusters and good separation between them.
- A high Calinski-Harabasz index of 2149.32 further supports the quality of separation.
- A low Davies Bouldin index of 0.3572 indicates minimal overlap between clusters.

4.3 Cluster Visualisation UMAP Embedding

To visualise the clustered data, the first two dimensions from the UMAP projection was plotted, allowing for a graphical representation of clustering results, shown in Figure 4.3.

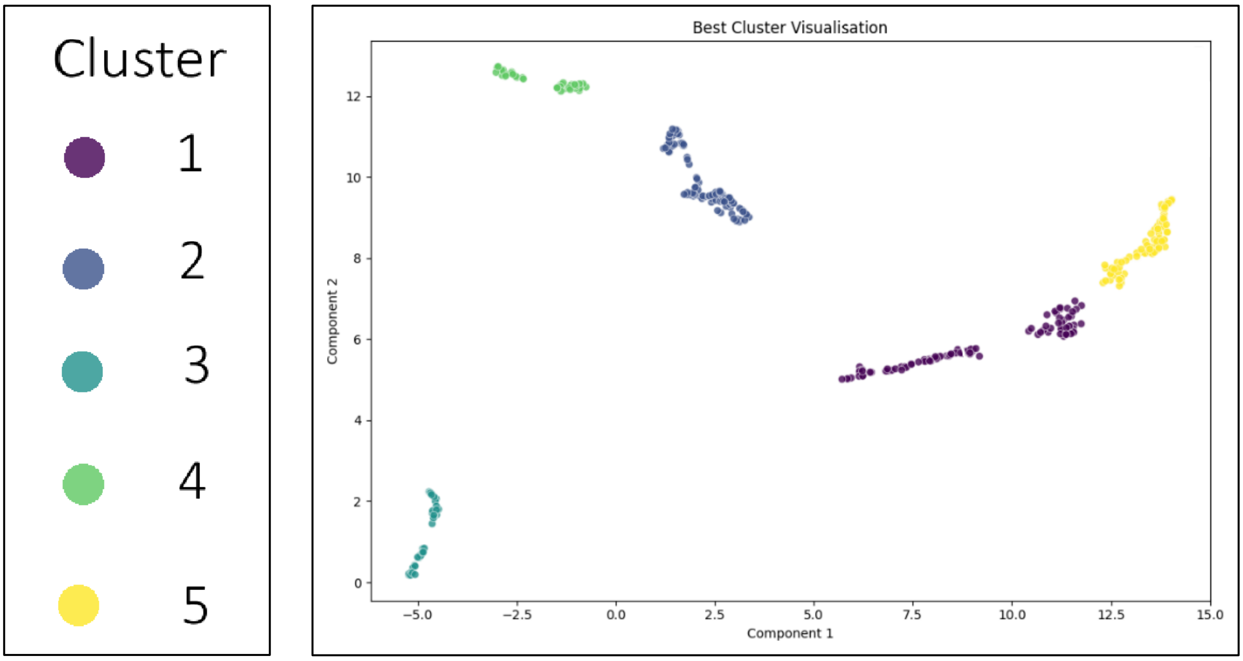


Figure 4.3: *UMAP of clusters ($k = 5$).*

Each point in Figure 4.3 represents a single PRPD image from the 5D result of the UMAP projection spliced into 2D space. Colours correspond to cluster labels assigned by K-means. No prior assumptions were made about the cluster formations, the structure observed is a result of purely unsupervised learning.

Observations:

- Clusters are well separated with minimum overlap.
- Some clusters, such as Cluster 3 (turquoise) and Cluster 5 (yellow), exhibit compact and well-defined shapes, indicating high intra-cluster similarities.
- Others, such as Cluster 1 (purple), show a more elongated spread, which may suggest greater variation in the initial conditions of the PRPD images within the cluster.

There are clusters, in particular Cluster 1, 3, and 4, that look as though they may contain internal substructures that could be further separated, as Figure 4.3 shows that there are gaps in cluster formations. While these apparent gaps might imply further subdivision is possible, it is important to recognise that this figure only reflects the first two dimensions of a five-dimensional space. The complete clustering was performed in 5D, where the algorithm had access to more information between the images. This means that points that appear far apart in 2D space, to the point where it could be its own cluster, may be grouped together intentionally when considering the complete 5D feature embedding. In practice, this suggests that ‘mergeable’ or ‘splitable’ groups in 2D do not warrant further clustering or separation.

4.4 Interpretation of Clusters – Summary Table

Table 3: Visual Summary of Cluster Observations.

Cluster	Inferred Condition	Key Features
1	Low voltage, low viscosity insulation.	• Weak discharge intensity. • Activity on positive half. • Narrow phase band.
2	High voltage, high viscosity insulation with solid insulation overlapping.	• Moderate discharge intensity. • Activity on positive and negative half. • Two broad distinct phase bands.
3	High voltage, low viscosity insulation.	• Strong discharge intensity. • Activity on positive half. • Narrow phase band.
4	Solid insulation	• Strong discharge intensity. • Activity on positive and negative half. • Three broad distinct phase bands.
5	Low magnitude and sparse PD activity with majority low voltage, low viscosity.	• Weak discharge intensity. • Activity on positive half. • Very narrow phase band.

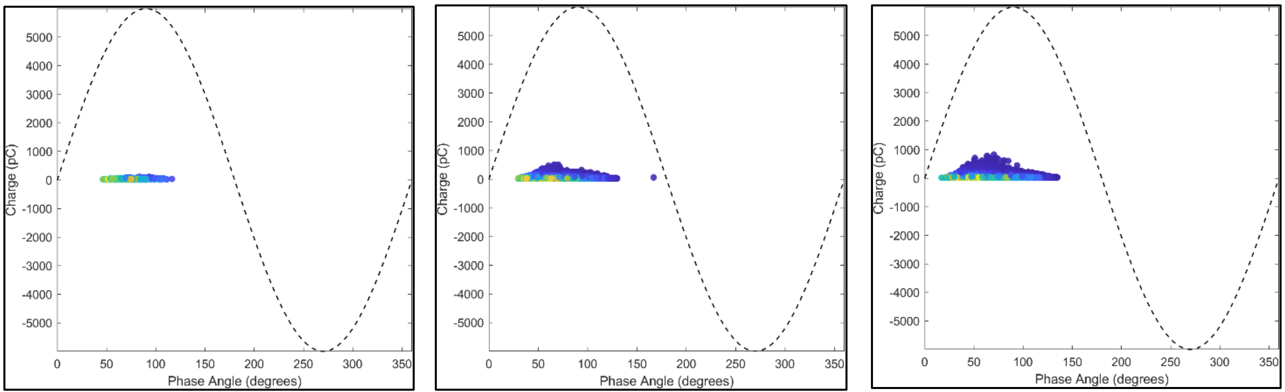
4.5 Interpretation of Clusters

This section explores the visual features of each cluster to understand what distinguished them. By tracing all 388 PRPD image to its experimental metadata (electrode configuration, viscosity, temperature, etc.) several relationships between discharge characteristics and test conditions are inferred. Please refer to Table 2 to see abbreviations.

4.5.1 Cluster 1 – Low Voltage, Low Viscosity

These PRPD images share two key features that contributed to their separation from other clusters: Low intensity and unipolar activity.

- **Size:** 108 Images
- **Cluster Accuracy:** 89%
- **Voltage Range:** 12-25 kV
- **Insulation Type:** Low Viscosity silicone oil insulation.
- **Key Visual Features:**
    - Weak discharge intensity – low charge magnitudes.
    - Activity only during the positive half of applied AC voltage.
    - Narrow band of discharge phase angles – 30° to 140°.



EC: NP	**GS:** 25 mm	**EC:** NP	**GS:** 25 mm	**EC:** NP	**GS:** 45 mm
T: 20 °C	**V:** 12 kV	**T:** 20 °C	**V:** 18 kV	**T:** 20 °C	**V:** 25 kV
IT: Low Viscosity oil		**IT:** Low Viscosity oil		**IT:** Low Viscosity oil	

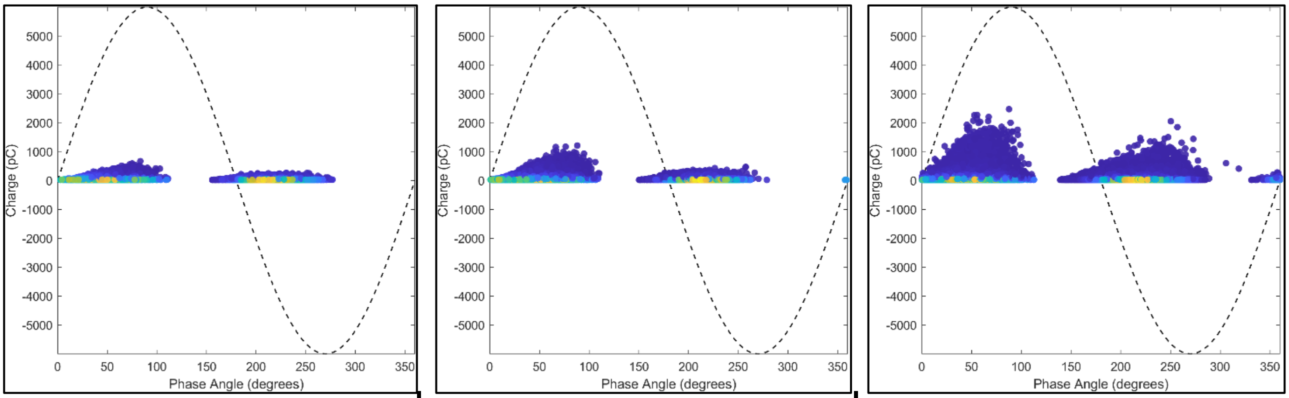
Figures 4.5.1 from left to right i, ii, iii: *Shows a sample of three images from Cluster 1.*

The consistent presence of low-intensity discharges confined to the positive half-cycle, as shown in Figures 4.5.1, suggested a common experimental setup involving low voltage and low-viscosity insulation. Despite having no access to initial condition labels, the model successfully grouped these 108 images based on their shared visual features, demonstrating its ability to uncover meaningful patterns in the data.

4.5.2 Cluster 2 – High Voltage High Viscosity, with Solid Overlapping.

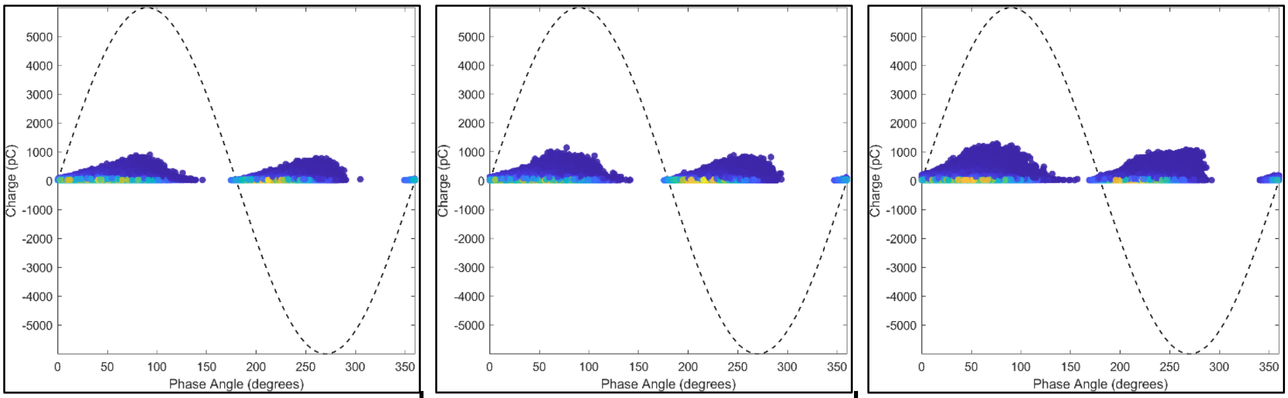
Cluster 2 presents the model's first limitation case where the unsupervised model struggled to distinguish between PRPD images generated under high viscosity, high voltage conditions, and those produced in solid insulation experiments.

- **Size:** 103 Images
- **Cluster Accuracy:** 24%
- **Voltage Range:** 25-40 kV
- **Insulation Type:** High viscosity oil and solid insulation.
- **Key Visual Features:**
  - Moderate discharge intensity.
  - Activity during the positive and negative half of applied AC voltage.
  - Two distinct bands of discharge phase angles – 0° to 90° and 150° to 280°.



EC: NP **GS:** 50 mm	**EC:** NP **GS:** 50 mm	**EC:** NP **GS:** 50 mm
T: 20 °C **V:** 25 kV	**T:** 80 °C **V:** 30 kV	**T:** 20 °C **V:** 40 kV
IT: High Viscosity oil	**IT:** High Viscosity oil	**IT:** High Viscosity oil

Figure 4.5.2a from left to right i, ii, iii: *Shows a sample of three High Viscosity oil insulation images from Cluster 2 that look similar to Solid Insulation.*



IT: Solid insulation **V:** 30 kV	**IT:** Solid insulation **V:** 30 kV	**IT:** Solid insulation **V:** 30 kV

Figure 4.5.2b from left to right i, ii, iii: *Shows a sample of three Solid insulation images from Cluster 2 that look similar to high viscosity oil insulation.*

Despite differences in experimental conditions, the images in Cluster 2 share similar structural features: the discharges are spread symmetrically across the positive and negative half of the AC voltage waveform. Furthermore, the discharge intensity is very similar in magnitude and all with a broad, low-density tail, with a concentrated core of activity near the zero crossing points. These similarities in spatial and phase distribution have confused the model's feature extraction process, leading to an overlap of conditions in the same cluster.

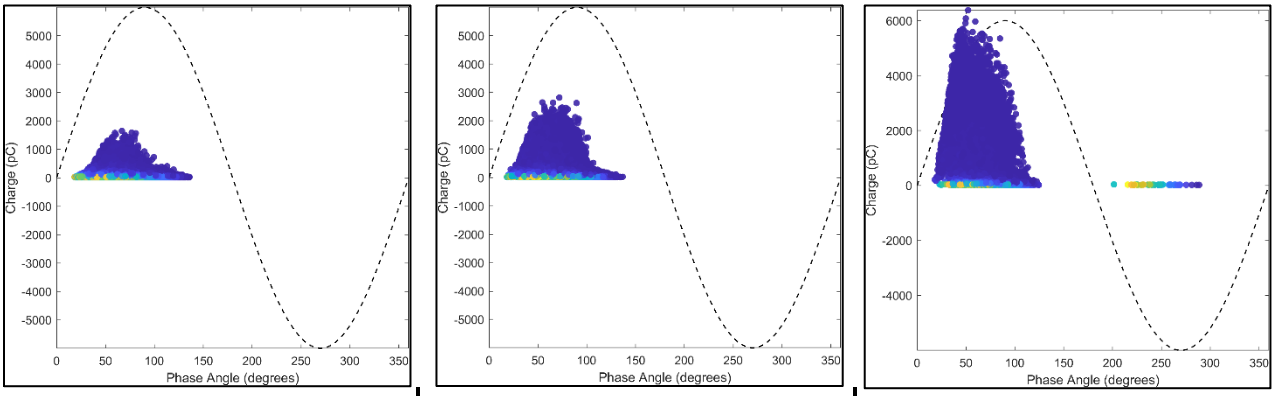
Since the clustering algorithm relied on learned visual features, it grouped these images based on similar texture, symmetry and overall shape, as shown in Figures 4.5.2. This suggests that at high applied voltage, both high viscosity and solid insulation may produce visually similar PRPD patterns, making it harder to distinguish between them using phase resolved images alone.

4.5.3 Cluster 3 – High Voltage, Low Viscosity

Cluster 3 primarily consists of PRPD images from experiments under high voltage and low viscosity insulation conditions.

- **Size:** 48 Images
- **Cluster Accuracy:** 88%
- **Voltage Range:** 25-45 kV
- **Insulation Type:** Low Viscosity oil
- **Key Visual Features:**
    - Strong discharge intensity – high charge magnitudes.

- Activity only during the positive half of applied AC voltage.
- Narrow band of discharge phase angles – 30° to 140°.



EC: NP	**GS:** 25 mm	**EC:** NP	**GS:** 25 mm	**EC:** NP	**GS:** 50 mm
T: 20 °C	**V:** 25 kV	**T:** 20 °C	**V:** 35 kV	**T:** 80 °C	**V:** 45 kV
IT: Low Viscosity oil		**IT:** Low Viscosity oil		**IT:** Low Viscosity oil	

Figure 4.5.3 from left to right i, ii, iii: *Shows a sample of three images from Cluster 3.*

All the images within this cluster share a distinct discharge pattern: a concentrated group of partial discharges located exclusively in the positive half-cycle of the AC waveform. These discharges are high in intensity, with a triangular shaped plume, as shown in Figures 4.5.3, which become a visual cue for the clustering model.

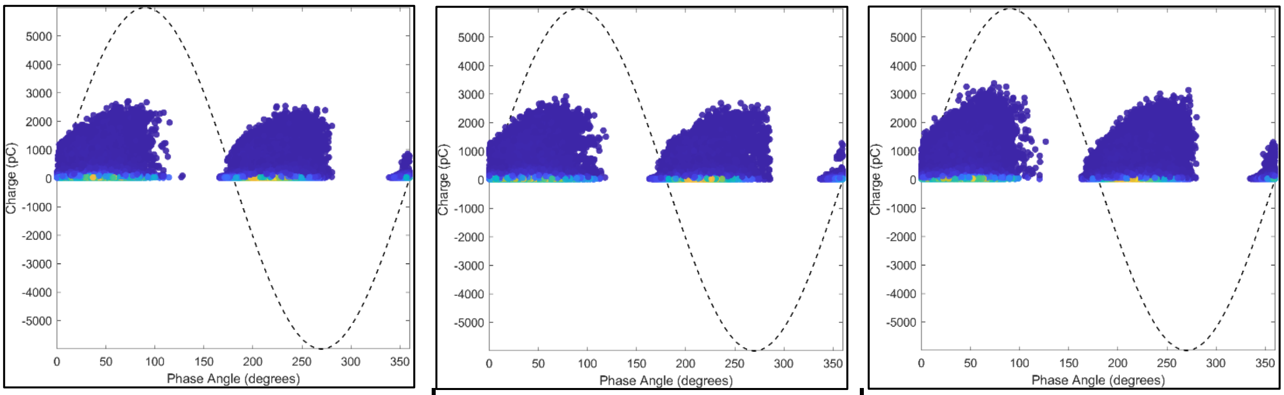
However, not all images within this cluster display partial discharges exclusively in positive half cycle. A subset, such as Figure 4.5.3iii, exhibit low-density discharge activity on the negative half cycle, typically associated with high viscosity oil insulation. Interestingly, the model still correctly assigned these to Cluster 2 indicating that, despite the presence of some negative-cycle discharges, these discharges were not structurally distinct enough to warrant placement into a separate cluster. Instead, the dominant characteristics still aligned most closely with the rest of the high voltage, low viscosity cluster.

This outcome highlights the model's ability to generalise across small variations within a class and avoid overfitting to minor deviations. It suggests that the clustering decision was driven by the overall structure of the PD pattern, specifically the strong consistent features in the positive half-cycle, rather than the occasional anomalous discharge events on the negative half-cycle. This shows that the model effectively balances sensitivity to pattern differences with robustness against noise or anomalies, which is essential for real-world diagnostic applications.

4.5.4 Cluster 4 – Solid Insulation

Cluster 4 is composed of PRPD images under solid insulation conditions, with examples shown in Figure 4.5.4a. However, three anomalous samples from high viscosity oil experiments were also grouped in this cluster, shown in Figure 4.5.4b, suggesting a partial visual overlap in discharge behaviour between high viscosity and solid. This reinforces the ambiguity observed in Cluster 2 and highlights the challenge of distinguishing certain insulation types based solely on visual discharge patterns.

- **Size:** 46 Images
- **Cluster Accuracy:** 93%
- **Voltage Range:** 30-45 kV
- **Insulation Type:** Low Viscosity oil.
- **Key Visual Features:**
    - Strong discharge intensity – high charge magnitudes.
    - Activity during the positive and negative half of applied AC voltage.
    - Three distinct bands of discharge phase angles – 0° to 110°, 170° to 270°, 340° to 360°.



IT: Solid insulation **V:** 30 kV

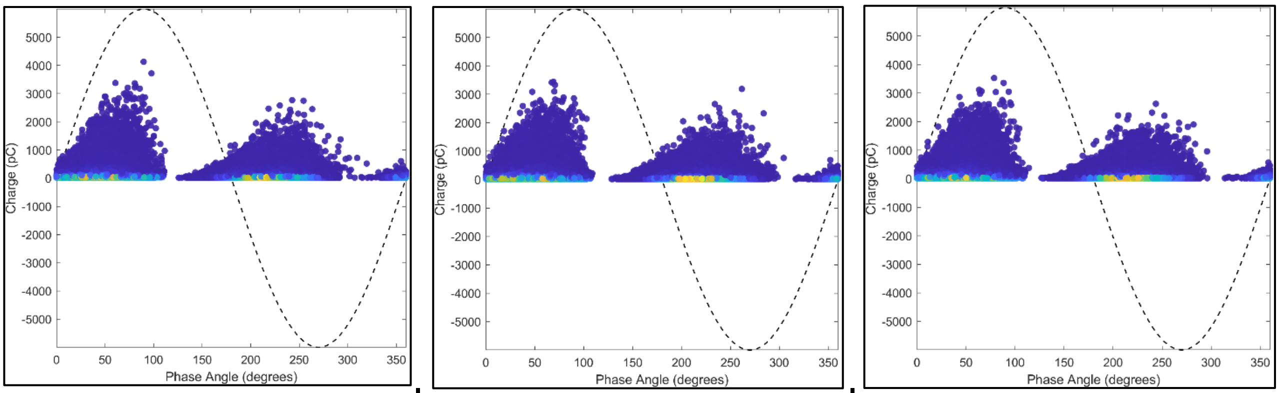
IT: Solid insulation **V:** 30 kV

IT: Solid insulation **V:** 30 kV

Figure 4.5.4a from left to right i, ii, iii: *Shows a sample of three solid insulation images from Cluster 4.*

The PRPD images in this cluster exhibit a distinct pattern characterised by dense activity in both the positive and negative half-cycles of the AC voltage waveform. Each half-cycle has a broad, high-magnitude plume that span across a wide, but consistent phase range – approximately 350° to 100° and 170° to 280°, with a distinct separation between the two. These discharge structures are symmetrical about the zero-crossing points and display significantly higher peak amplitudes than those observed in lower-voltage clusters.

This dual-lobed and phase-separated structure likely served as the dominant visual feature for clustering, with the model identifying the shape, symmetry, and distribution of the discharge density as key distinguishing features. While the majority of images in this cluster were generated under solid insulation, a small number of high-viscosity samples were also included. Their inclusion suggests a high degree of visual similarity between the discharge patterns of solid and high-viscosity insulation under certain conditions, further reinforcing the overlap observed in Cluster 2.



EC: NP	**GS:** 50 mm	**EC:** NP	**GS:** 50 mm	**EC:** NP	**GS:** 50 mm
T: 20 °C	**V:** 45 kV	**T:** 20 °C	**V:** 45 kV	**T:** 20 °C	**V:** 45 kV
IT: High Viscosity		**IT:** High Viscosity		**IT:** High Viscosity	

Figure 4.5.4b from left to right i, ii, iii: *Shows a sample of three anomalous high viscosity oil insulation images from Cluster 4.*

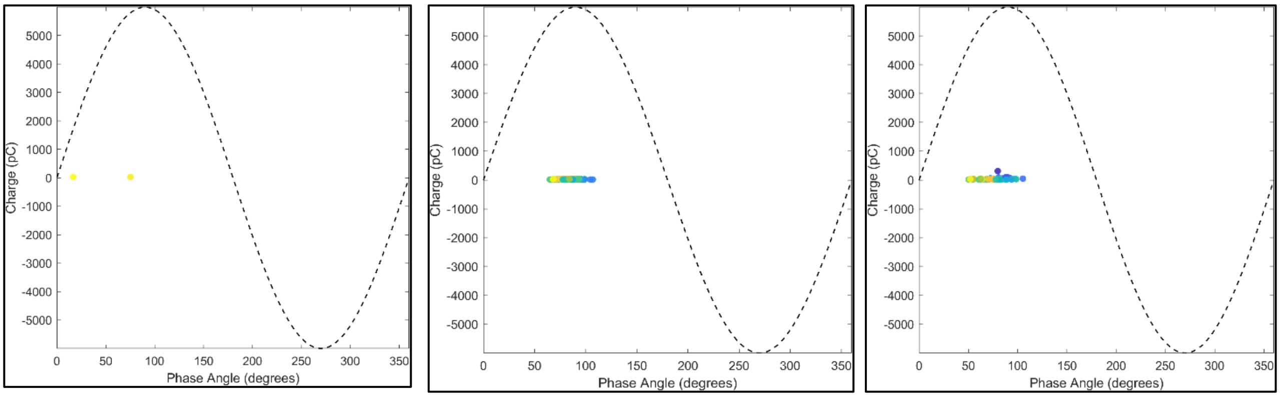
This suggests that under certain conditions, high viscosity fluid insulation can exhibit PD behaviour that visually resembles solid insulation, or vice versa. However, unlike Cluster 2, where grouping was influenced by subtle features and led to ambiguity, the clustering in Cluster 4 appears more confident and well-defined, as indicated by a higher accuracy. This is likely due to the distinct characteristics of the solid insulation images in this group, which exhibit higher discharge magnitudes and broader phase dispersion, compared to Cluster 2. These stronger, more pronounced features provided the model with clearer cues for separation, resulting in a more interpretable cluster.

4.5.5 Cluster 5 – Low Magnitude, Sparse PD Activity at Low Voltage and Low Viscosity

Cluster 5 contains PRPD images, with examples shown in Figures 4.5.5, spanning a broad range of initial conditions, including both low and high viscosity insulation, and varying voltage levels. However, a significant portion of the images in this cluster originate from experiments involving low viscosity insulation under low voltage stress.

Despite this experimental diversity, the images grouped into this cluster share one key characteristic: minimal partial discharge activity.

- **Size:** 83 Images
- **Cluster Accuracy:** 53%
- **Voltage Range:** 11-50 kV
- **Insulation Type:** Low and High Viscosity oil (No solid).
- **Key Visual Features:**
  - Very weak discharge intensity – low charge magnitudes.
  - Activity only during the positive half of applied AC voltage.
  - Very narrow band of discharge phase angles – 40° to 100°.



EC: NP	**GS:** 50 mm	**EC:** PNP	**GS:** 25 mm	**EC:** PNP	**GS:** 45 mm
T: 80 °C	**V:** 11 kV	**T:** 20 °C	**V:** 29 kV	**T:** 20 °C	**V:** 50 kV
IT: Low Viscosity oil		**Nee-pro:** 5 mm	**IT:** Low Viscosity oil	**Nee-pro:** 5 mm	**IT:** Low Viscosity oil

Figure 4.5.5 from left to right i, ii, iii: *Shows a sample of three images from Cluster 5.*

Across the cluster, discharges appear as small, scattered points with very low magnitude, often confined to a narrow phase range or dispersed randomly across different phases. In all cases, only a few discharge events are present, resulting in a dotted appearance with no clear structure, density, or intensity. This is a large contrast to the well-defined lobes or high-density plumes observed in other clusters.

The model likely prioritised this sparsity and lack of structure during clustering, using the overall weakness of discharge patterns as the defining feature. These images may represent borderline or pre-breakdown conditions, where the applied voltage stress was insufficient to generate distinct or recurring PD patterns.

4.6 Dominant Visual Features Driving Clustering Decisions

The clustering model used in this project relies solely on the extracted visual features of the PRPD images, without any access to their underlying experimental conditions. As a result, the patterns that emerge during clustering are based only on what the model identifies as structurally significant within the images. Through analysis of the five resulting clusters, it became evident that three key features play a dominant role in shaping the clustering outcome: the width of the discharge phase band, the vertical height of the discharges, and the symmetry.

4.6.1 Width of the Phase-Band – Horizontal Spread

The phase band width refers to how widely the discharge events are distributed across the 0° to 360° range of the AC waveform. PRPD patterns with a narrow phase band, where discharges occur in tightly confined regions of the waveform, tend to form distinct clusters. For example, Cluster 1 includes low-voltage images with discharges in the positive half-cycle, and Cluster 5 contains sparse patterns with random discharges. In contrast, wider phase distributions, such as Cluster 2 and 4, indicate more symmetrical PD behaviour.

This horizontal spread becomes a key differentiator when images like those in Cluster 4 with a wide and dual-lobed discharge distribution are separated from those with narrow or asymmetric activity like those in Cluster 1 or 3.

4.6.2 Height of Discharges – Vertical Spread (Charge Magnitudes)

The vertical height of the discharges in the PRPD images correspond to the magnitude of the PDs measured in picocoulombs. The clustering model clearly distinguishes between images with high-magnitude discharges that form dense, patterns that stretch vertically, then those with low-magnitude discharges, which appear as faint, low-lying dots. This contrast is clearly seen between Clusters 5 (low intensity) and Cluster 3 (high intensity).

By grouping images with similar vertical discharge heights, the model captures differences in the severity of PD activity. This allows it to identify that high voltage generate strong PD events, while low-voltage samples produce weaker activity.

4.6.3 Symmetry of the Discharge Regions

In addition to width and height, the overall shape and symmetry of the discharge regions support the formation of clusters. For example, images with symmetrical lobes around the zero-crossing point like Cluster 4 are grouped together, whereas asymmetrical patterns, like Cluster 3, are separated.

4.7 Physical Interpretation of Clustering by Viscosity and Voltage

The unsupervised clustering model revealed a strong separation between PRPD images associated with varying insulation types, insulation viscosities and applied AC voltage levels. This separation can be explained by the underlying physical mechanisms for PD activity in liquid silicone insulation.

4.7.1 Influence of High Viscosity on PRPD Structure

High-viscosity silicone oil restricts the mobility of charge carriers and gas bubbles generated during discharge events [58]. This resistance to flow in the insulation reduces the formation of streamers, resulting in more restricted PD activity at a lower magnitude, as shown in Cluster 2.

During the positive half-cycle, when the needle electrode is positively charged, electrons, which are lighter and more mobile, are attracted towards the needle [59]. This set-up supports stable streamer formation, as electrons can ionise the insulation along their path.

In the negative half-cycle, the needle becomes negatively charged, repelling electrons and attracting heavier positive ions [57, 60]. The reduced mobility of these ions makes streamer formation more difficult, so PD events are less frequent and less intense.

4.7.2 Influence of Low Viscosity on PRPD Structure

In contrast, low-viscosity silicone oil facilitates the movement of charge carriers and gas bubbles since the reduced resistance enables faster bubble movement [58], which promotes gas bubble development.

Under positive polarity, electrons are drawn towards the needle tip and interact with fast moving bubbles, creating favourable conditions for PD events [59]. This results in PRPD patterns typically dominated by discharges in the positive half-cycle like characteristics observed in Cluster 1 and 3.

However, under negative polarity, the heavier positive ions combined with fast moving gas bubbles in the low-viscous environment led to ineffective PD events [60]. This means that PRPD patterns from low-viscosity setups tend to be asymmetric and confined to the positive half of the AC cycle.

4.7.3 Voltage Effects

The magnitude of the applied voltage significantly shapes PRPD structure as a higher voltage increases the local field strength, which lowers the threshold for ionisation [60]. This results in more intense PDs of a greater magnitude, like those observed in Cluster 3.

However, lower voltages only generate pre-breakdown or weak discharges, especially in high-viscosity fluids, resulting in low-magnitude PRPD plots like those in Cluster 5 [60].

5 Conclusions and future work

5.1 Conclusions

This project investigated a novel unsupervised AI pipeline to uncover hidden patterns in PRPD graphs obtained from high-voltage experiments. The key outcomes include the successful development and implementation of a feature clustering process combining wavelet scattering transform, a custom convolutional neural network, UMAP for dimensionality reduction, and K-means for clustering. The wavelet transforms, and CNN layer depth was designed to reveal subtle structural characteristics in the PRPD images that contain distinguishing information that may be overlooked by human engineers.

The results demonstrated that the model was able to group PRPD images into clusters corresponding to different initial experimental conditions without prior labels. This capability has significant implications for automated condition monitoring, where expert interpretation is costly, time-consuming, or subjective.

Beyond the scope of electrical engineering, this research speaks to the broader trend of integrating AI into complex pattern recognition tasks across sectors. The methodology developed here can be extended to fault conditions in other systems such as biomedical scans like X-Rays and MRI scans, or even satellite imagery in geoscience.

Furthermore, the findings suggest a wider societal implication: by enabling earlier and more reliable detection of insulation degradation, utilities can prevent failures in power networks. This directly contributes to improving infrastructure resilience, reducing maintenance costs and enhancing energy security, a critical goal in global efforts to modernise ageing grid systems.

5.2 Future work

To build upon the foundation of this project, several directions are proposed for future work:

1. **Semi-Supervised Extension:** A hybrid approach that uses a small dataset of prelabelled images could guide the unsupervised model towards even more meaningful clusters, increasing its usability for diagnostic applications. This could involve using a small subset of images annotated by experts as anchor points for refining clusters.
2. **Comparison With Manual Expert Labelling:** To validate the clusters produced by the unsupervised model against classifications made by experienced high-voltage engineers. This would provide a benchmark for assessing the interpretability of the clusters and determine whether the AI model is discovering known discharge patterns or novel ones that are overlooked.
3. **Multimodal Input Fusion:** Future developments could incorporate additional data sources such as discharge magnitude, acoustic emissions, or temperature readings, alongside the PRPD images, from sensors like UV cameras or acoustic emission sensors. Fusing these inputs may enhance the model's capacity to detect complex correlations that single-input analysis might miss.
4. **Deployment in Systems:** To optimise the AI for deployment in real-time stations or facilities. This would test the pipeline's feasibility for low-latency, on-site monitoring in resource-constrained environments.

In conclusion, this work contributes not only to advancing pattern recognition in electrical diagnostics but also to the broader goal of integrating intelligent monitoring into critical infrastructure, paving the way for safer, smarter, and more sustainable engineering systems.

6 Bibliography

References

[1] IEA, “Growth in Global Energy Demand Surged in 2024 to Almost Twice Its Recent Average - News - IEA,” *IEA*, Mar. 24, 2025. https://www.iea.org/news/growth-in-global-energy-demand-surged-in-2024-to-almost-twice-its-recent-average (accessed Apr. 06, 2025).

[2] V. Rozite, J. Miller, and S. Oh, “Why AI and Energy Are the New Power Couple,” *IEA*, Nov. 02, 2023. https://www.iea.org/commentaries/why-ai-and-energy-are-the-new-power-couple (accessed Apr. 06, 2025).

[3] Elmelin Marketing, “Insulation materials for high voltage applications - Elmelin Ltd,” *Elmelin Ltd*, Nov. 04, 2021. https://elmelin.com/insulation-materials-for-high-voltage-applications/ (accessed Apr. 06, 2025).

[4] E. Staff, “How Do High Voltage Insulators Work? - Electrolock Incorporated,” *Electrolock Incorporated*, Aug. 29, 2024. https://www.electrolock.com/thought-leadership/how-do-high-voltage-insulators-work/ (accessed Apr. 05, 2025).

[5] “IEC 60034-27-2:2023,” *Webstore.iec.ch*, Dec. 07, 2023. https://webstore.iec.ch/en/publication/64620 (accessed Apr. 06, 2025).

[6] Zulbirri Faizol *et al.*, “Detection Method of Partial Discharge on Transformer and Gas-Insulated Switchgear: A Review,” *Applied sciences*, vol. 13, no. 17, pp. 9605–9605, Aug. 2023, doi: https://doi.org/10.3390/app13179605.

[7] N. Davies, “What is Partial Discharge? A Guide | EA Technology Australia,” *Eatechnology.com*, Jul. 28, 2022. https://eatechnology.com/australia/resources/blogs/high-voltage/what-is-partial-discharge-a-guide-to-understanding-and-managing-pd/ (accessed Apr. 05, 2025).

[8] Editor, “Heat Pump Summit,” *EM Magazine*, Nov. 10, 2023. https://www.energymanagermagazine.co.uk/solving-the-challenges-of-partial-discharge-testing/ (accessed Apr. 06, 2025).

[9] B. Monaghan, “What is a Phase Resolved Partial Discharge (PRPD) Plot telling you? | EA Technology Americas,” *Eatechnology.com*, Oct. 28, 2021.

https://eatechnology.com/americas/resources/faq/what-is-a-phase-resolved-partial-discharge-prpd-plot-telling-you/ (accessed Apr. 03, 2025).

[10] H. Kumar, M. Shafiq, K. Kauhaniemi, and M. Elmusrati, "A Review on the Classification of Partial Discharges in Medium-Voltage Cables: Detection, Feature Extraction, Artificial Intelligence-Based Classification, and Optimization Techniques: Energies (19961073)," *Energies (19961073)*, vol. 17, no. 5, p. 1142, Mar. 2024, doi: https://doi.org/10.3390/en17051142.

[11] I. D. Mienye and T. G. Swart, "A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications," *Information*, vol. 15, no. 12, p. 755, Nov. 2024, doi: https://doi.org/10.3390/info15120755

[12] M. Mandal, "CNN for Deep Learning | Convolutional Neural Networks (CNN)," *Analytics Vidhya*, May 01, 2021. https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ (accessed Apr. 06, 2025).

[13] Z. Keita, "An Introduction to Convolutional Neural Networks (CNNs)," *Datacamp.com*, Nov. 14, 2023. https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns (accessed Apr. 06, 2025).

[14] Tamanna, "Exploring Convolutional Neural Networks: Architecture, Steps, Use Cases, and Pros and Cons," *Medium*, Apr. 24, 2023. https://medium.com/@tam.tamanna18/exploring-convolutional-neural-networks-architecture-steps-use-cases-and-pros-and-cons-b0d3b7d46c71 (accessed Apr. 06, 2025).

[15] J. Butdee, W. Kongprawechnon, H. Nakahara, N. Chayopitak, C. Kingkan, and R. Pupadubsin, "Pattern Recognition of Partial Discharge Faults Using Convolutional Neural Network (CNN)," *2023 8th International Conference on Control and Robotics Engineering (ICCRE)*, pp. 61–66, Apr. 2023, doi: https://doi.org/10.1109/iccre57112.2023.10155616.

[16] API4AI, "How CNNs Transformed Industries Over the Past 10 Years | by API4AI | Medium," *Medium*, Jan. 10, 2025. https://medium.com/@API4AI/how-convolutional-neural-networks-transformed-industries-over-the-past-10-years-11bc651b963c (accessed Apr. 06, 2025).

[17] J. Delua, “Supervised vs. Unsupervised learning: What’s the difference? | IBM,” *www.ibm.com*, Mar. 12, 2021. https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning (accessed Apr. 06, 2025).

[18] H. T. Tai, Y.-W. Youn, H.-S. Choi, and Y.-H. Kim, “Partial Discharge Diagnosis using Semi-supervised learning and Complementary labels in Gas-insulated Switchgear,” *IEEE Access*, vol. PP, no. 99, p. 1, Jan. 2025, doi: https://doi.org/10.1109/ACCESS.2025.3556353.

[19] L. Satish and B. I. Gururaj, “Partial discharge pattern classification using multilayer neural networks,” *IEE Proceedings A Science, Measurement and Technology*, vol. 140, no. 4, p. 323, Jul. 1993, doi: https://doi.org/10.1049/ip-a-3.1993.0049.

[20] “Recognition of discharges,” *E-cigre.org*, 1969. https://www.ecigre.org/publications/detail/elt-011-2-recognition-of-discharges.html (accessed Apr. 07, 2025).

[21] A. Abubakar and C. Zachariades, “Phase-Resolved Partial Discharge (PRPD) Pattern Recognition Using Image Processing Template Matching,” *Sensors*, vol. 24, no. 11, p. 3565, May 2024, doi: https://doi.org/10.3390/s24113565.

[22] G. Stone and A. Cavallini, “The Evolution of Partial Discharge Testing in Electrical Equipment - NETAWORLD JOURNAL,” *NETAWORLD JOURNAL*, Feb. 23, 2024. https://netaworldjournal.org/the-evolution-of-partial-discharge-testing-in-electrical-equipment/ (accessed Apr. 07, 2025).

[23] K. Kothoke, “Analysis of Phase Resolved Partial Discharge Patterns using Statistical Techniques,” *International Journal of Science and Research*, vol. 5, pp. 2319–7064, 2016, Accessed: Apr. 07, 2025. [Online]. Available: https://www.ijsr.net/archive/v5i8/ART2016945.pdf

[24] S. Mantach, “Supervised and unsupervised deep learning models for partial discharge source detection and classification in electrical insulation,” *Umanitoba.ca*, Aug. 11, 2023. https://mspace.lib.umanitoba.ca/items/2d6effeb-4daf-4c34-a929-8c1bad915a68 (accessed Apr. 07, 2025).

[25] R.M. Sharkawy, R.S. Mangoubi, T.K. Abdel-Galil, M. M. A. Salama, and R. Bartnikas, “SVM classification of contaminating particles in liquid dielectrics using higher order statistics of electrical and acoustic PD measurements,” *IEEE transactions on dielectrics and electrical*

insulation, vol. 14, no. 3, pp. 669–678, Jun. 2007, doi: https://doi.org/10.1109/tdei.2007.369530.

[26] H. Ugail, A. Abubakar, A. Elmahmudi, C. Wilson, and B. Thomson, “The use of pre-trained deep learning models for the photographic assessment of donor livers for transplantation,” *Artificial Intelligence Surgery*, vol. 2, no. 2, pp. 101–119, Jan. 2022, doi: https://doi.org/10.20517/ais.2022.06.

[27] A. Elmahmudi and H. Ugail, “Deep face recognition using imperfect facial data,” *Future Generation Computer Systems*, vol. 99, pp. 213–225, Oct. 2019, doi: https://doi.org/10.1016/j.future.2019.04.025

[28] S. Mantach, A. Ashraf, H. Janani, and B. Kordi, “A Convolutional Neural Network-Based Model for Multi-Source and Single-Source Partial Discharge Pattern Classification Using Only Single-Source Training Set,” *Energies*, vol. 14, no. 5, p. 1355, Mar. 2021, doi: https://doi.org/10.3390/en14051355.

[29] S. Mantach, M. Partyka, V. Pevtsov, A. Ashraf, and B. Kordi, “Unsupervised Deep Learning for Detecting Number of Partial Discharge Sources in Stator Bars,” *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 30, no. 6, pp. 2887–2895, Dec. 2023, doi: https://doi.org/10.1109/tdei.2023.3306324.

[30] R. Sahoo and S. Karmakar, “Effectiveness of Wavelet Scalogram on Partial Discharge Pattern Classification of XLPE Cable Insulation,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–10, 2024, doi: https://doi.org/10.1109/tim.2024.3363790.

[31] R. Sahoo and S. Karmakar, “Comparative analysis of machine learning and deep learning techniques on classification of artificially created partial discharge signal,” *Measurement*, vol. 235, p. 114947, Aug. 2024, doi: https://doi.org/10.1016/j.measurement.2024.114947.

[32] MATLAB, “Machine Learning and Deep Learning with Wavelet Scattering | Understanding Wavelets, Part 5,” *www.youtube.com*, Jan. 28, 2020. https://www.youtube.com/watch?v=XyeZFo1d5aY(accessed May 04, 2024).

[33] A. W. Lone and N. Aydin, “Wavelet Scattering Transform based Doppler signal classification,” *Computers in Biology and Medicine*, vol. 167, pp. 107611–107611, Dec. 2023, doi: https://doi.org/10.1016/j.compbiomed.2023.107611.

[34] Z. Liu, G. Yao, Q. Zhang, J. Zhang, and X. Zeng, “Wavelet Scattering Transform for ECG Beat Classification,” *Computational and Mathematical Methods in Medicine*, vol. 2020, pp. 1–11, Oct. 2020, doi: https://doi.org/10.1155/2020/3215681.

[35] A. A. B. M. Nor and Q. Liu, “Effects of electrode configurations on partial discharge characteristics of silicone oil,” *IET conference proceedings.*, vol. 2023, no. 46, pp. 532–537, Apr. 2024, doi: https://doi.org/10.1049/icp.2024.0574.

[36] S. Lengsfeld, F. Rehwald, H. Ast, and O. Schroder, “Classification of Partial Discharge Patterns in Rotating Electrical Machines Using Machine Learning,” *2022 International Conference on Electrical Machines (ICEM)*, pp. 1576–1581, Sep. 2022, doi: https://doi.org/10.1109/icem51905.2022.9910894.

[37] N. Duro, “Sensor Data Fusion Analysis for Broad Applications,” *Sensors*, vol. 24, no. 12, pp. 3725–3725, Jun. 2024, doi: https://doi.org/10.3390/s24123725.

[38] B. Ding *et al.*, “Pattern recognition of partial discharge based on deep learning,” *IET Conference Proceedings*, vol. 2020, no. 1, pp. 1187–1191, Apr. 2021, doi: https://doi.org/10.1049/icp.2020.0223.

[39] kymatio, “GitHub - kymatio/kymatio: Wavelet scattering transforms in Python with GPU acceleration,” *GitHub*, 2018. https://github.com/kymatio/kymatio (accessed Apr. 07, 2025).

[40] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional Neural networks: an Overview and Application in Radiology,” *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Jun. 2018, doi: https://doi.org/10.1007/s13244-018-0639-9.

[41] M. A. Nielsen, “Neural Networks and Deep Learning,” *Neuralnetworksanddeeplearning.com*, 2019. http://neuralnetworksanddeeplearning.com/chap6.html

[42] Mayur Ingole, “Simple Convolutional Neural Network (CNN) for Dummies in PyTorch: A Step-by-Step Guide,” *Medium*, Jun. 16, 2024. https://medium.com/@myringoleMLGOD/simple-convolutional-neural-network-cnn-for-dummies-in-pytorch-a-step-by-step-guide-6f4109f6df80 (accessed Apr. 08, 2025).

[43] B. Krishnamurthy, “ReLU Activation Function Explained | Built In,” *builtin.com*, Oct. 28, 2022. https://builtin.com/machine-learning/relu-activation-function (accessed Apr. 08, 2025).

[44] Doaa Almhaithawi, A. Bellini, G. C. Chasparis, and T. Cerquitelli, “Investigating the Potential of Latent Space for the Classification of Paint Defects,” *Journal of Imaging*, vol. 11, no. 2, pp. 33–33, Jan. 2025, doi: https://doi.org/10.3390/jimaging11020033.

[45] A. Yadav, “Batch Normalization vs Layer Normalization - Biased-Algorithms - Medium,” *Medium*, Sep. 19, 2024. https://medium.com/biased-algorithms/batch-normalization-vs-layer-normalization-c44472883bf2(accessed Apr. 08, 2025).

[46] A. Yadav, “The Role of Dropout in Neural Networks - Biased-Algorithms - Medium,” *Medium*, Oct. 15, 2024. https://medium.com/biased-algorithms/the-role-of-dropout-in-neural-networks-fffbaa77eee7(accessed Apr. 08, 2025).

[47] “Wavelet Scattering,” *Mathworks.com*, 2024. https://uk.mathworks.com/help/wavelet/ug/wavelet-scattering.html (accessed Apr. 10, 2025).

[48] Zahra Baharlouei, H. Rabbani, and G. Plonka, “Wavelet scattering transform application in classification of retinal abnormalities using OCT images,” *Scientific reports*, vol. 13, no. 1, Nov. 2023, doi: https://doi.org/10.1038/s41598-023-46200-1.

[49] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,” *arXiv.org*, 2018. https://arxiv.org/abs/1802.03426 (accessed Apr. 10, 2025).

[50] A. Coenen and A. Pearce, “Understanding UMAP,” *pair-code.github.io*. https://pair-code.github.io/understanding-umap/(accessed Apr. 10, 2025).

[51] “How UMAP Works — umap 0.5 documentation,” *umap-learn.readthedocs.io*. https://umap-learn.readthedocs.io/en/latest/how_umap_works.html (accessed Apr. 10, 2025).

[52] “Using UMAP for Clustering — umap 0.5 documentation,” *umap-learn.readthedocs.io*. https://umap-learn.readthedocs.io/en/latest/clustering.html (accessed Apr. 10, 2025).

[53] Scikit-learn, “sklearn.metrics.silhouette_score — scikit-learn 0.21.3 documentation,” *Scikit-learn.org*, 2019. https://scikitlearn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html (accessed Apr. 10, 2025).

[54] LinkedIn Community, “How can you calculate the silhouette score for a clustering algorithm?,” *www.linkedin.com*. https://www.linkedin.com/advice/0/how-can-you-calculate-silhouette-score-clustering-algorithm-w9bcc (accessed Apr. 10, 2025).

[55] M. Sv, “Calinski-Harabasz Index for K-Means Clustering Evaluation using Python | Towards Data Science,” *Towards Data Science*, Mar. 15, 2022. https://towardsdatascience.com/calinski-harabasz-index-for-k-means-clustering-evaluation-using-python-4fefeeb2988e/ (accessed Apr. 10, 2025).

[56] V. Artus, “Clustering metrics: evaluate the complex, make it simple,” *Medium*, Feb. 29, 2024. https://medium.com/@vladimir-artus/%D1%81lustering-metrics-evaluate-the-complex-make-it-simple-6ae70c0f164b (accessed Apr. 10, 2025).

[57] S. Suwarno, “Partial Discharge in High Voltage Insulating Materials,” *International Journal on Electrical Engineering and Informatics*, vol. 8, no. 1, pp. 147–163, Mar. 2016, doi: https://doi.org/10.15676/ijeei.2016.8.1.11.

[58] Q. Xue, I. Timoshkin, M. P. Wilson, M. Given, and S. J. MacGregor, “Mobility of charge carriers in mineral oil and ester fluids,” *High Voltage*, vol. 6, no. 6, pp. 1040–1050, May 2021, doi: https://doi.org/10.1049/hve2.12098.

[59] F. M. Francis, “A model for the initiation and propagation of electrical streamers in transformer oil and transformer oil based nanofluids,” *Mit.edu*, 2007, doi: http://dspace.mit.edu/handle/1721.1/40504.

[60] M. Pompili, “Partial discharge measurements in dielectric liquids,” *2008 IEEE International Conference on Dielectric Liquids*, pp. 1–7, Jun. 2008, doi: https://doi.org/10.1109/icdl.2008.4622542.

[61] MATLAB, version 24.2.0.2712019 (R2024b), The MathWorks Inc., Natwick Massachusetts, USA, 2024. [Online]. Available: https://uk.mathworks.com/products/matlab.html

[62] M. Abadi *et al.*, “TensorFlow: A System for Large-Scale Machine Learning This paper is included in the Proceedings of the 12th USENIX Symposium on Operating Systems Design

and Implementation (OSDI ’16). TensorFlow: A system for large-scale machine learning,” Nov. 2016. Accessed: Apr. 11, 2025. [Online]. Available: https://www.tensorflow.org

[63] Python Software Foundation, Python Language Reference, version 3.9.10*. [Online]. Available: https://www.python.org (accessed Apr. 7, 2025).

Appendices

A Project outline

Project Proposal (Old) – High Voltage: Advanced Analysis of Partial Discharge (PD) Measurement Results

Background and Motivation:

Partial Discharges (PDs) are small electrical discharges that occur in high voltage equipment due to imperfections in the insulation, such as voids, cracks, or improper cable joints. These discharges can lead to long-term damage and can eventually result in failure of a system. Detecting and analysing PD is crucial for maintaining the health of high voltage equipment such as cables, transformers and switches.

Traditional PD detection methods involve analysing electrical signals, acoustic emissions, and electromagnetic wave detection. However, advanced techniques, such as electric field analysis, machine learning, and image processing, can be used to interpret PD data, making diagnostics more precise and predictable.

The motivation for this project comes from the increasing demand for efficient and reliable high voltage systems and PD detection is a critical tool to prevent equipment failure. Such tools can be enhanced through advanced mathematical techniques, AI, and image analysis, for example, MATLAB can be used for data processing and image analysis, and ANSYS can be used for simulations.

Project Aim and Objectives:

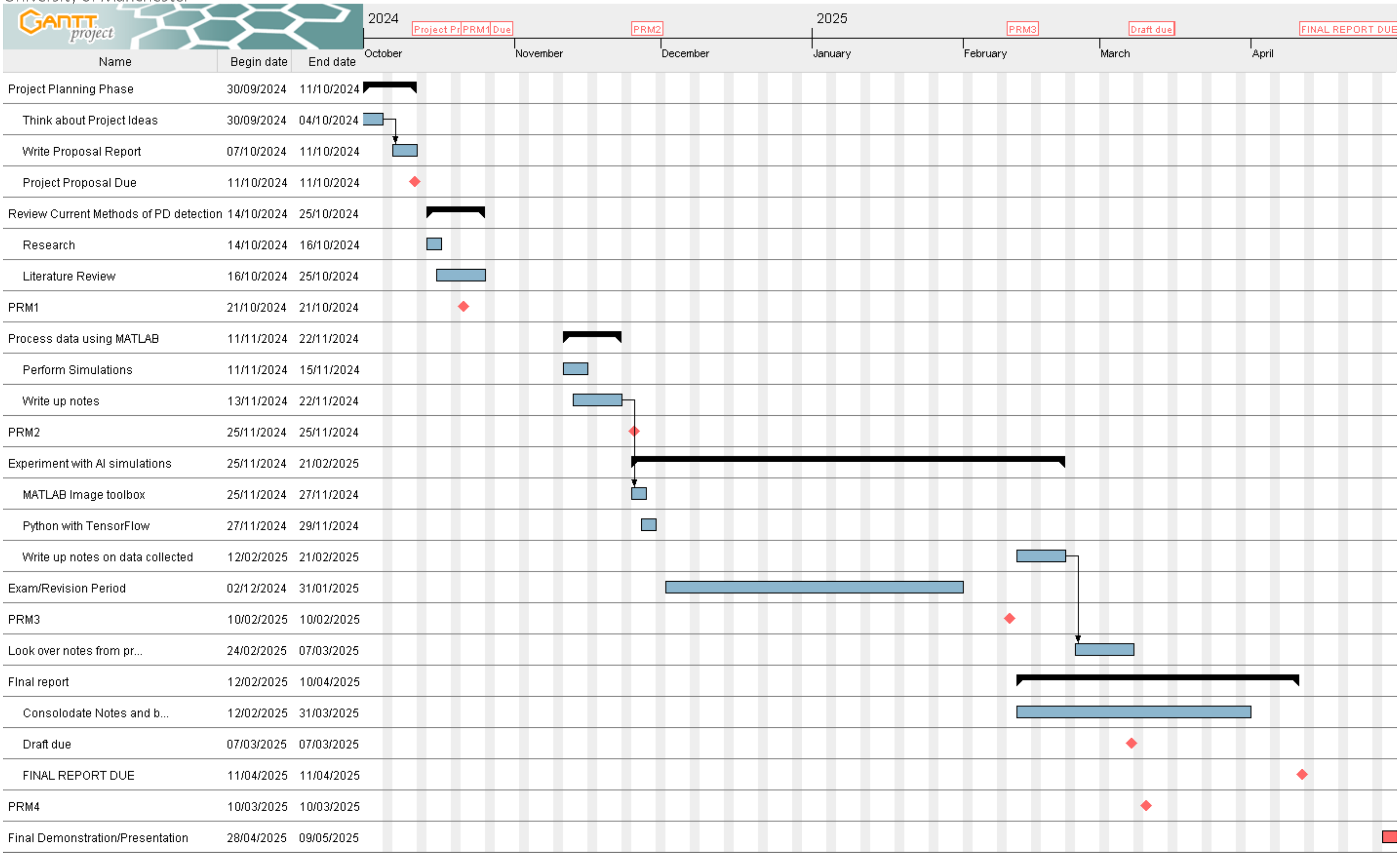
1. Review current methods to detect and analyse PD –
    a. a. This can be a literature review investigating limitations and gaps in current techniques.
2. Simulate Electric Fields using ANSYS –

a. a. Simulate electric fields in high voltage equipment and analyse regions prone to PD occurrence.

b. b. This can also show the relationship between electric field strength, insulation strength and application, and PD occurrence.

3. Process PD data using MATLAB –
    a. Utilise advanced mathematical techniques to process PD data, including Coulombs, phase angles and PD patterns.
    b. MATLAB’s image processing toolbox can be used to enhance or supress certain features in an image, such as edge detection. It can also analyse and classify PD patterns based on their shape, intensity, or texture.
4. Compare simulated data with experimental data if available.
5. Analyse the impact of improper cable insulation such as voids, edges, or bubbles in the insulation on PD occurrence and propagation.

MANCHESTER
1824
The University of Manchester



Name	Begin date	End date
Project Planning Phase	30/09/2024	11/10/2024
Think about Project Ideas	30/09/2024	04/10/2024
Write Proposal Report	07/10/2024	11/10/2024
Project Proposal Due	11/10/2024	11/10/2024
Review Current Methods of PD detection	14/10/2024	25/10/2024
Research	14/10/2024	16/10/2024
Literature Review	16/10/2024	25/10/2024
PRM1	21/10/2024	21/10/2024
Process data using MATLAB	11/11/2024	22/11/2024
Perform Simulations	11/11/2024	15/11/2024
Write up notes	13/11/2024	22/11/2024
PRM2	25/11/2024	25/11/2024
Experiment with AI simulations	25/11/2024	21/02/2025
MATLAB Image toolbox	25/11/2024	27/11/2024
Python with TensorFlow	27/11/2024	29/11/2024
Write up notes on data collected	12/02/2025	21/02/2025
Exam/Revision Period	02/12/2024	31/01/2025
PRM3	10/02/2025	10/02/2025
Look over notes from pr...	24/02/2025	07/03/2025
FInal report	12/02/2025	10/04/2025
Consolodate Notes and b...	12/02/2025	31/03/2025
Draft due	07/03/2025	07/03/2025
FINAL REPORT DUE	11/04/2025	11/04/2025
PRM4	10/03/2025	10/03/2025
Final Demonstration/Presentation	28/04/2025	09/05/2025

B Risk assessment

General Risk Assessment Form

Date: 11/10/24	Assessed by: Yaseen Ahmed	Checked / Validated* by:	Location: Working from home/university labs	Assessment ref no (5)	Review date: (6)
Task / premises: (7)					

Activity (8)	Hazard (9)	Who might be harmed and how (10)	Existing measures to control risk (11)	Risk rating (12)	Result (13)
Working from home	Lone working Prolonged looking at screen Affected Wellbeing (stress) Incidents	Student	1. Please refer to the University Lone Working policy and guidance for more information 2. Please refer to the new University Working at Home guidance 3. Please refer to the new University Wellbeing Support website Staff to remain in regular direct contact with line manager and colleagues via phone, Skype, Zoom, Slack or email 1. Floors and walkways kept clear of items, e.g. boxes, packaging, equipment etc. 2. Furniture is arranged such that movement of people and equipment are not restricted 3. Make sure all areas have good level of lighting 4. Reasonable standards of housekeeping maintained 5. Trailing cables positioned neatly away from walkways 6. Cabinet drawers and doors kept closed when not in use 7. Report the event to personal line manager and the School Safety Advisor to complete an incident form	Low	A

Activity (8)	Hazard (9)	Who might be harmed and how (10)	Existing measures to control risk (11)	Risk rating (12)	Result (13)
At home or University workspaces	Electrical Hazards from electronics Poor posture, repetitive movements.	Student, other users in vicinity.	1. All office equipment used in accordance with the manufacturer's instructions 2. Visual checks before use to make sure equipment, cables and free from defects 3. Avoid daisy chaining and do not overload extension leads 4. University IT equipment brought home should already be PAT tested 5. The domestic electrical supply and equipment owned by the employee is the responsibility of the employee to maintain 6. Liquid spills cleaned up immediately Defective plugs, cables and equipment should be taken out of use 1. Please refer to the DSE policy, guidance and poster for more information on how to set up your workstation properly 2. Complete DSE self-assessment for guidance on how to set up workstation properly 3. Set up workstation to a comfortable position with good lighting and natural light where possible 4. Take regular breaks away from the screen 5. Regularly stretch your arms, back, neck, wrists and hands to avoid repetitive strain injuries. Refer to workstation exercises here Set up a desktop working space where possible and try to avoid working on a laptop without a docking station	Med	A
Moving from University to home (vice versa)	Tripping, incident, injury	Student / others in vicinity	Ensure clean and clear working spaces.		

Project Title:	Using Unsupervised Machine Learning and Wavelet Scattering Transformation to Analyse PRPD Patterns for High Voltage Applications		Submission Date:	11/4/25
Student Name:	Yaseen	Ahmed		

Project Risk	**Severity**			**Potential**			**Score** **(Severity x Potential)**	**Mitigation Measures**
	L	M	H	L	M	H	L=1, M=2, H=3	
Loss of documents			X	X			3	Backup everything
Code not working	X					X	3	Debug code to remove errors
Laptop not working			x	x			3	Fix laptop or use University computers
Emergencies			X	X			3	Include extra time in schedule for falling behind
Missing deadline			X	X			3	Set timers and alarms

Continuing Professional Development Log

Name: *Yaseen Ahmed*

Current and recent CPD activity:

CPD Activity Title	Description	Dates	CPD Hours
Python Coding	Throughout my project I had to develop my ability to code in Python	October 24 to April 25	150
Machine Leaning and AI knowledge development	I had to develop my knowledge in AI and ML to be able to carry out my project. This learning process was supported by the internet and reading papers. I learned about CNNs, unsupervised and supervised, and how to code them in MATLAB and Python	-	-
MATLAB coding	Throughout my project I had to develop my ability to code in MATLAB	-	-
Report writing skills	Throughout this project I had to develop my ability to writing code, and ability to express thoughts, actions, and theories in text to people who are unfamiliar with topics.	-	-
Project planning and risk assessment	I had to develop my abilities in planning a project by creating a Gantt chart to create a schedule and think about potential risks.	-	-

Planned and Future CPD activity:

CPD Activity Title	Description	Skills addressed	Dates
Presentation Skills	I have the opportunity to develop my presentation skills when presenting my projection presentation.	Presentation, communication	May 25
Applying for internships and grad schemes	I am currently and applying for internships to do during the summer and after graduation	Job skills	All the time
Tutoring	I will be tutoring GCSE and A-Level science and maths across multiple platforms	Communication, Teaching	Foreseeable future
MEng	I will be doing MEng next year	Academic	September 25

C PRPD Image Generation code using MATLAB

```
%% Combined PRPD Dataset Expansion with Improved Augmentation & Density Normalization
% - Reads CSV files containing 'angle' and 'charge' columns.
% - Aggregates a global density map for consistent visual scaling across plots.
% - Applies augmentation with random phase shifts, charge scaling, and noise.
% - Normalizes density maps independently for original and augmented data to ensure visual consistency.

clc; clear; close all;
rng(42); % Set random seed for reproducibility

%% Select Folder Containing CSV Files
folderPath = uigetdir(pwd, 'Select Folder Containing CSV Files');
if folderPath == 0
    error('No folder selected. Exiting.');
end

% Retrieve all CSV files from the selected folder
csvFiles = dir(fullfile(folderPath, '*.csv'));
fprintf('Processing %d CSV files...\n', length(csvFiles));

%% Step 1: Aggregate Global Density Map
% This step creates a shared density lookup for consistent visual scaling
allPhases = [];
allCharges = [];
densityMap = containers.Map('KeyType', 'char', 'ValueType', 'double');

for fileIdx = 1:length(csvFiles)
    fileName = csvFiles(fileIdx).name;
    filePath = fullfile(folderPath, fileName);

    try
        data = readtable(filePath, 'PreserveVariableNames', true);
    catch
        fprintf('  Could not read file: %s. Skipping.\n', fileName);
        continue;
    end

    % Ensure necessary columns are present
    if all(ismember({'angle', 'charge'}, data.Properties.VariableNames))
        phase = round(data.angle, 1);
        charge = round(data.charge, -1);

        allPhases = [allPhases; phase];
        allCharges = [allCharges; charge];

        % Create key for each data point and accumulate count
        for i = 1:length(phase)
            key = sprintf('%0.1f_%d', phase(i), charge(i));
            if isKey(densityMap, key)
                densityMap(key) = densityMap(key) + 1;
            else
                densityMap(key) = 1;
            end
        end
    else
        fprintf('  Required columns missing in file: %s. Skipping.\n', fileName);
    end
end

if isempty(densityMap)
    error('No valid data found. Exiting.');
end

globalMaxDensity = max(cell2mat(values(densityMap)));
fprintf('Global maximum density found: %d\n', globalMaxDensity);

%% Step 2: Augmentation Configuration
```

```
numAugmentedSamples = 10;             % Number of augmented versions per file
maxPhaseShift = 10;                   % Maximum phase shift (degrees)
chargeScalingRange = [0.97, 1.03];    % Range for charge scaling
noiseStdDev = 25;                     % Standard deviation of Gaussian noise

fprintf('Generating PRPD plots with improved augmentation...\n');

%% Step 3: Process Each File and Generate Plots
for fileIdx = 1:length(csvFiles)
    fileName = csvFiles(fileIdx).name;
    filePath = fullfile(folderPath, fileName);
    fprintf('Processing file: %s\n', fileName);

    try
        data = readtable(filePath, 'PreserveVariableNames', true);
    catch
        fprintf('  Could not read file: %s. Skipping.\n', fileName);
        continue;
    end

    if ~all(ismember({'angle', 'charge'}, data.Properties.VariableNames))
        fprintf('  Required columns missing in file: %s. Skipping.\n', fileName);
        continue;
    end

    %% Step 3A: Plot Original PRPD Data
    phaseOrig = round(data.angle, 1);
    chargeOrig = round(data.charge, -1);

    densityOrig = zeros(size(phaseOrig));
    for i = 1:length(phaseOrig)
        key = sprintf('%0.1f_%d', phaseOrig(i), chargeOrig(i));
        if isKey(densityMap, key)
            densityOrig(i) = densityMap(key);
        end
    end

    origMaxDensity = max(densityOrig);
    if origMaxDensity > 0
        densityOrig = densityOrig / origMaxDensity;
    end

    createPRPDPlot(phaseOrig, chargeOrig, densityOrig, fileName, folderPath, 'Original');

    %% Step 3B: Generate and Plot Augmented Versions
    for augIdx = 1:numAugmentedSamples
        augPhase = mod(data.angle + randi([-maxPhaseShift, maxPhaseShift], size(data.angle)), 360);

        scalingFactors = chargeScalingRange(1) + ...
                         (chargeScalingRange(2) - chargeScalingRange(1)) * rand(size(data.charge));

        augCharge = data.charge .* scalingFactors + noiseStdDev * randn(size(data.charge));
        augCharge = max(augCharge, 0); % Ensure non-negative charge

        augPhaseRounded = round(augPhase, 1);
        augChargeRounded = round(augCharge, -1);

        augDensity = zeros(size(augPhaseRounded));
        for i = 1:length(augPhaseRounded)
            key = sprintf('%0.1f_%d', augPhaseRounded(i), augChargeRounded(i));
            if isKey(densityMap, key)
                augDensity(i) = densityMap(key);
            end
        end

        augMaxDensity = max(augDensity);
        if augMaxDensity > 0
            augDensity = augDensity / augMaxDensity;
        end

        label = sprintf('Augmented_%d', augIdx);
        createPRPDPlot(augPhase, augCharge, augDensity, fileName, folderPath, label);
    end
```

```
end

fprintf('Processing complete. PRPD graphs saved in the same folder as the CSV files.\n');

%% Function to Create and Save PRPD Plot
function createPRPDPlot(phase, charge, density, fileName, folderPath, label)
    figure('Units', 'Pixels', 'Position', [100, 100, 512, 512]); % Standard fixed size
    hold on;

    % Plot sine wave reference for phase context
    phaseAngles = linspace(0, 360, 1000);
    sineWave = 6000 * sin(deg2rad(phaseAngles));
    plot(phaseAngles, sineWave, 'k--', 'LineWidth', 1);

    % Scatter plot with color based on density
    scatter(phase, charge, 30, density, 'filled', 'MarkerFaceAlpha', 0.9);
    colormap("parula");

    % Set color axis limits based on density
    if max(density) > 0
        clim([0 max(density)]);
    else
        clim([0 1]);
    end

    % Set axis labels
    xlabel('Phase Angle (degrees)', 'FontSize', 12);
    ylabel('Charge (pC)', 'FontSize', 12);
    axis tight;
    box on;
    set(gca, 'FontSize', 10);
    set(gca, 'Position', [0.13 0.11 0.775 0.75]);

    % Export the plot as PNG
    outputFileName = sprintf('PRPD_%s_%s.png', fileName, label);
    outputFileName = strrep(outputFileName, ' ', '_');
    exportgraphics(gcf, fullfile(folderPath, outputFileName), ...
        'Resolution', 300, 'BackgroundColor', 'white');

    close(gcf);
    hold off;
end
```

Appendix D Unsupervised methodology including WST, CNN, UMAP, K-Means.

```
import os
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns
import shutil
from kymatio import Scattering2D
from tensorflow.keras import layers, models
from tensorflow.keras.preprocessing import image
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score, calinski_harabasz_score, davies_bouldin_score
from sklearn.preprocessing import StandardScaler
import umap

# Configure TensorFlow to manage GPU memory growth
gpus = tf.config.experimental.list_physical_devices('GPU')
if gpus:
    try:
        for gpu in gpus:
            tf.config.experimental.set_memory_growth(gpu, True)
        print("GPU memory growth enabled.")
    except RuntimeError as e:
        print(f"Could not enable memory growth: {e}")

# Function to load grayscale images from a directory
```

```python
def load_images_from_directory(directory, target_size=(128, 128)):
    images, filenames, filepaths = [], [], []
    valid_extensions = (".png", ".jpg", ".jpeg", ".bmp", ".tif", ".tiff")
    for root, _, files in os.walk(directory):
        for filename in files:
            if filename.lower().endswith(valid_extensions):
                img_path = os.path.join(root, filename)
                try:
                    img = image.load_img(img_path, target_size=target_size, color_mode='grayscale')
                    img_array = image.img_to_array(img) / 255.0  # Normalize pixel values
                    images.append(img_array)
                    filenames.append(filename)
                    filepaths.append(img_path)
                except Exception as e:
                    print(f"Failed to load {img_path}: {e}")
    return np.array(images), filenames, filepaths

# Load image dataset from a specified directory
image_dir = r"directory"
images, filenames, filepaths = load_images_from_directory(image_dir)
images = np.expand_dims(images, axis=-1)  # Add channel dimension

# Apply wavelet scattering transformation to extract stable features
scattering = Scattering2D(J=2, shape=(128, 128))

def apply_wavelet_scattering(images):
    scattering_results = np.array([scattering(img.squeeze()) for img in images])
    print(f"Wavelet scattering result shape: {scattering_results.shape}")
    return np.transpose(scattering_results, (0, 2, 3, 1))  # Reorder dimensions for CNN input

images_wst = apply_wavelet_scattering(images)

# Define CNN model architecture for feature extraction
def build_feature_extractor(input_shape=(32, 32, 81)):
    model = models.Sequential([
        layers.InputLayer(input_shape=input_shape),
        layers.Conv2D(64, (3, 3), activation='relu', padding='same'),
        layers.BatchNormalization(),
        layers.Conv2D(128, (3, 3), activation='relu', padding='same'),
        layers.BatchNormalization(),
        layers.MaxPooling2D((2, 2)),

        layers.Conv2D(256, (3, 3), activation='relu', padding='same'),
        layers.BatchNormalization(),
        layers.Conv2D(512, (3, 3), activation='relu', padding='same'),
        layers.BatchNormalization(),
        layers.MaxPooling2D((2, 2)),

        layers.GlobalAveragePooling2D(),
        layers.Dense(256, activation='relu'),
        layers.Dropout(0.5),
        layers.Dense(128, activation='relu')  # Output feature vector
    ])
    return model

# Extract features from the scattering-transformed images
model = build_feature_extractor()
features = model.predict(images_wst, batch_size=4)

# Standardize the feature vectors
features = StandardScaler().fit_transform(features)

# Apply UMAP for non-linear dimensionality reduction
umap_reducer = umap.UMAP(n_components=5, init='spectral', metric='cosine', random_state=42)
reduced_features = umap_reducer.fit_transform(features)

# Determine the optimal number of clusters using silhouette score
def find_optimal_clusters(data, max_clusters=10):
    silhouette_scores = []
    cluster_range = range(2, min(max_clusters, len(data) - 1))
    for k in cluster_range:
        kmeans = KMeans(n_clusters=k, n_init=10, random_state=42)
        cluster_labels = kmeans.fit_predict(data)
        if len(set(cluster_labels)) > 1:
            silhouette_scores.append(silhouette_score(data, cluster_labels))
    return cluster_range[np.argmax(silhouette_scores)]

optimal_clusters = find_optimal_clusters(reduced_features)

# Apply KMeans clustering using the optimal number of clusters
kmeans = KMeans(n_clusters=optimal_clusters, n_init=20, random_state=42)
clusters = kmeans.fit_predict(reduced_features)

# Evaluate clustering performance
sil_score = silhouette_score(reduced_features, clusters)
ch_score = calinski_harabasz_score(reduced_features, clusters)
db_score = davies_bouldin_score(reduced_features, clusters)

# Save results and organize clustered images
output_dir = "Clustered_Images"
if os.path.exists(output_dir):
    shutil.rmtree(output_dir)
os.makedirs(output_dir, exist_ok=True)

# Save clustering evaluation and image assignments to text file
with open(os.path.join(output_dir, "cluster_assignments.txt"), "w", encoding="utf-8") as f:
    f.write("Clustering Evaluation Metrics:\n")
    f.write(f"Silhouette Score: {sil_score:.4f}\n")
    f.write(f"Calinski-Harabasz Index: {ch_score:.2f}\n")
    f.write(f"Davies-Bouldin Index: {db_score:.4f}\n\n")

    for cluster_id in range(optimal_clusters):
        f.write(f"Cluster {cluster_id}:\n")
        cluster_folder = os.path.join(output_dir, f"Cluster_{cluster_id}")
        os.makedirs(cluster_folder, exist_ok=True)
        for path, label in zip(filepaths, clusters):
            if label == cluster_id:
                shutil.copy(path, cluster_folder)
                f.write(f"{os.path.basename(path)}\n")
        f.write("\n")

# Visualize cluster assignments in 2D
plt.figure(figsize=(10, 7))
sns.scatterplot(x=reduced_features[:, 0], y=reduced_features[:, 1], hue=clusters, palette="viridis",
alpha=0.8)
plt.title("Cluster Visualisation (UMAP Projection)")
plt.xlabel("UMAP Component 1")
plt.ylabel("UMAP Component 2")
plt.legend(title="Cluster", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.savefig(os.path.join(output_dir, "best_cluster_plot.png"))

print("Clustering completed. Results and visualisations have been saved.")
```

D GitHub Repository

All code, results, and data used and created in this project is available at this link:

https://github.com/Y45E3N-A/UsingUnsupervisedML