****

**A Hybrid Approach for Partial Discharge Classification: Combining Traditional machine learning and Deep Neural Network**

**Submitted by: Ding Shaobo**

**Supervisor: Jiang Xudong**

### School of Electrical & Electronic Engineering

A final year project report presented to the Nanyang Technological University

in partial fulfillment of the requirements of the degree of

Bachelor of Engineering

**2023**

# Table of Contents

[Abstract i](#_Toc197854799)

[List of Figures ii](#_Toc197854805)

[List of Tables i](#_Toc197854806)ii

[Chapter 1 Introduction 1](#_Toc197854807)

[1.1 Motivations 1](#_Toc197854808)

[1.2 Objectives and Scope 1](#_Toc197854809)

[1.3 Organisations 1](#_Toc197854810)

[Chapter 2 Literature Review](#_Toc197854807)  ……………………………………............................................. ..2

[2.1 Background and Concept 2](#_Toc197854808)

[2.2 PD denoising 2](#_Toc197854809)

[2.3 PD Detection and Classification 2](#_Toc197854809)

[Chapter 3 Experiment](#_Toc197854807) Process …………………………............................................. 3

[3.1 Brief introduction of experiment process 3](#_Toc197854808)

[3.2 Data Description 3](#_Toc197854809)

[3.3 Feature extraction method 3](#_Toc197854809)

[3.4 Compare different clustering methods and Features 3](#_Toc197854809)

[3.5 Simple CNN approach 3](#_Toc197854809)

[3.6 Triplet Loss Approach 3](#_Toc197854809)

[Chapter 4 Conclusions and Future Work…………………………………………….... 4](#_Toc197854807)

[4.1 Conclusions 4](#_Toc197854808)

[4.2 Recommendation for Future Work 4](#_Toc197854809)

Reflection on Learning Outcome Attainment …………………………………………………………. 5

References………………………………………………………………………………………………. 6

# Abstract

Partial discharge (PD) is a critical issue in high-voltage equipment, and the accurate detection and classification of PDs are essential for preventing equipment failure. In recent years, various approaches have been proposed for PD classification, including traditional machine learning methods and deep learning techniques.

Traditional machine learning algorithms, such as decision trees, support vector machines (SVM), and k-nearest neighbors (KNN), have been widely used for PD classification. However, these methods rely on manual feature extraction, which can be time-consuming and may not capture the complete range of PD characteristics. In contrast, deep learning techniques, including CNN and RNN, have shown promising results in PD classification by enabling the automatic extraction of relevant features from PD data. However, it requires a large amount of training data.

This study proposes a novel approach for PD classification, combining traditional machine learning algorithms with deep neural networks to perform transfer learning. Firstly, manual feature extraction is conducted to extract PD features. Traditional machine learning clustering algorithms, such as K-means and affinity propagation clustering will be applied to these features to separate noises from PDs. Subsequently, the Partial Discharge Pattern Recognition and Diagnosis (PRPD) is plotted and fed into a CNN to classify each cluster. In order to apply it in real-life applications, minimizing the missing detection rate is considered the priority of the tunning process. The proposed method can effectively detect and classify PD which can aid in the development of effective PD diagnosis systems and contribute to the safe and reliable operation of high-voltage equipment.

# List of Figures

[Figure 2.3.1.1: (a) Structure of a BPNN. 12](#_Toc197854900)

[Figure 2.3.1.2: (a) SVM example. 14](#_Toc197854901)

[Figure 2.3.2.1: (a) RNN example. 16](#_Toc197854902)

[Figure 2.3.2.2: (a) Simple convolution operation 18](#_Toc197854903)

[Figure 2.3.2.2: (b) Simple non-linear layer with relu (c) 2\*2 kernel max-pooling operation. 19](#_Toc197854904)

[Figure 3.1:(a) Experiment framework. 23](#_Toc197854904)

[Figure 3.2.1: (a) A single waveform data captured. 24](#_Toc197854904)

[Figure 3.2.1: (b) Captured waveform data list of one artificial electronic signal 25](#_Toc197854904)

[Figure 3.2.2: (a) Captured PRPD data list of one artificial electronic signal 25](#_Toc197854904)

[Figure 3.2.2: (b) Example of one PRPD pattern 26](#_Toc197854904)

[Figure 3.5.1: (a) Less noisy PRPD pattern (internal PD) 29](#_Toc197854904)

[Figure 3.5.1: (b) Example of desired one cluster of less noisy PRPD pattern (c) Noisier PRPD pattern (internal PD) 30](#_Toc197854904)

[Figure 3.5.1: (d) Example of desired one cluster of noisier PRPD pattern 31](#_Toc197854904)

[Figure 3.5.2: (a) mean-shift + FFT (b) mean-shift + PSD 32](#_Toc197854904)

[Figure 3.5.2: (c) mean-shift + CWT (d) K-Means Elbow + FFT 33](#_Toc197854904)

[Figure 3.5.2: (e) K-Means Elbow + PSD (f) K-Means Elbow + CWT 34](#_Toc197854904)

[Figure 3.5.2: (g) DBSCAN + FFT (h) DBSCAN + PSD 35](#_Toc197854904)

[Figure 3.5.2: (i) DBSCAN + CWT 36](#_Toc197854904)

[Figure 3.6.1: (a) CNN structure of first approach 39](#_Toc197854904)

[Figure 3.6.2.1: (a) Balance data code snippet (b) Multiply corona and internal class weights by 2 40](#_Toc197854904)

[Figure 3.6.2.3: (a) Downsampling. 42](#_Toc197854904)

[Figure 3.6.2.4: (a) Data split (b) Compile setup (c) Fit setup. 43](#_Toc197854904)

[Figure 3.6.2.4: (d) Confusion plot of CNN performance result. 44](#_Toc197854904)

[Figure 3.6.2.4: (e) Accuracy curve and confusion plot of CNN 5 (f) Accuracy curve and confusion plot of CNN 9. 47](#_Toc197854904)

[Figure 3.6:(a) Structure of CNN For capturing embeddings. 49](#_Toc197854904)

[Figure 3.6:(b) The embeddings generated and the loss curve of embedding CNN (c) confusion plot of CNN triplet with best overall acc. 50](#_Toc197854904)

# List of Tables

[Table 2-3-2-2: Result of Song et al. 21](#_Toc197854967)

[Table 3-4-3-a: processing time of each combination of features and clustering method for less noisy PRPD test pattern. 36](#_Toc197854968)

[Table 3-4-3-b: processing time of each combination of features and clustering method for noisier PRPD test pattern.](#_Toc197854969) 37

[Table 3-6-2-4-a: Result of CNN model 1, 2, 3, 4 45](#_Toc197854970)

[Table 3-6-2-4-b: Result of CNN model 5, 6, 7, 8, 9 46](#_Toc197854970)

# Introduction

## Motivations

This thesis deals with the problem of PD detection and classification for high-voltage insulation systems due to the fact that PD could result in serious defects in the insulation system.

## Objectives and Scope

* The model should be able to detect PD in electronic analog signals. Noises and PDs should be successfully separated after the clustering method.
* The model should be able to classify different types of PDs.
  + The model should be able to classify between different classes.
  + The model should be able to classify between two different but similar classes.
  + The missing detection situation (PD predicted as non-PD) should be minimized.
  + Intra-class distances should be minimized and smaller than inter-class distances.

# Literature Review

## Background and Concept

### PD introduction

International Electrotechnical Commission (IEC) 60270 Standard defines PD as “a localized electrical discharge that only partially bridges the insulation between conductors and which may or may not occur adjacent to a conductor” [1]. The occurrence of this phenomenon is usually limited to specific locations within the insulation system where the insulation material has a higher breakdown strength compared to the defect point [2]. Failure to promptly address PD could worsen the deterioration of electrical insulation in high voltage (HV) equipment and potentially lead to major power failure [3]. Therefore, the importance of partial discharge activity detection and classification in monitoring the condition of insulation systems can be understood. [4].

PD generally results in a transient current pulse that rises extremely quickly and whose pulse width varies depending on the type of discharge or defect that is present [5]. The PD can exhibit a range of source patterns, and the extent of the defects is influenced not only by the strength of the PD signal but also by the location of the defective area within the insulation material. [3].

### PD pattern representations

The two main categories of PD pattern representations currently employed in PD-related research are phase-resolved and time-resolved PD [6].

According to [5], the phase-resolved PD (PRPD) pattern is commonly employed for PD characteristic representation and displays the association between the electric charge and the calculated phase in the electronic signal. A PD detector will record each PD signal and quantify each pulse based on these features and generate the corresponding PRPD [7]. For computational purposes, when storing the pertinent amplitude, phase angle, and pulse numbers of PRPD patterns, the matrix format is commonly chosen [8].

As for time-resolved PD, because individual pulse shapes may be seen and there is a correlation between the PD signal's shape and the type of insulation defect, time-resolved data patterns have some intriguing advantages that reveal the age of the insulation system [9]. Furthermore, compared to phase-resolved PD, the time-resolved PD measuring process often requires a less expensive measurement apparatus [6].

Since each PD pulse has a strong relationship with the PRPD patterns, phase-resolved data are more frequently used in PD classification research because they can depict the physical process at the PD spot [10].

### Different Kinds of PD

Basically, PD can be classified into three different types, namely: void (internal) discharge, corona discharge, and surface discharge [3].

#### Void (internal) discharge

Internal PD occurs within a void when the electric field in the void surpasses the inception field and an initial free electron triggers an avalanche process. The appearance of void discharge is uncertain because the emergence of a free electron is a stochastic event. The electric field in the void conforms to the sinusoidal pattern of the applied voltage waveform. Consequently, the Partial Discharge Pattern Recognition and Diagnosis (PRPD) presents a curvilinear configuration that aligns with the sinusoidal voltage waveform [11].

#### Surface discharge

Surface discharge occurs when the electric field at the surface of an electrode exceeds the gas's breakdown strength, causing ionization of the air near the electrode surface and initiating partial discharge (PD). This phenomenon is caused by the high tangential field on the insulation surface, which triggers the formation of PDs along the surface of the material. The surface discharge continues in a sequence until the transient activity fades away [11].

#### Corona discharge

When an electric field is present and the voltage at the sharp point of an electrode exceeds the breakdown strength of the gas, ionization of air molecules near the electrode will occur. The formation of positive and negative streamers is determined by the polarity of the applied voltage. These streamers are known as corona discharges and continue until the transient activity subsides. If the discharge becomes self-sustaining, a steady glow will appear near the anode. [12].

## PD denoising

In an ideal scenario, the classification of PD patterns can be accomplished by examining the distinctive combination of PD phase distribution, pulse magnitude, and temporal variations [13]. During the measurement of PD, a major challenge is the presence of external noise interference, which results in reduced sensitivity and accuracy of PD detection [6]. Three main types of interferences happen during PD measuring: discrete spectral interferences, stochastic pulse-shaped interferences, and periodic pulse-shaped interferences [14]. According to W.J.K. Raymond et al. [15], traditional machine learning models are commonly trained using clean PD data obtained from laboratory measurements but are expected to function in on-site scenarios where a certain degree of noise or interference is inevitable. W.J.K. Raymond et al. [15] also mentioned that when evaluated using clean PD data, most machine learning models can easily achieve classification accuracy above 90%. Nevertheless, classification accuracy substantially decreases when these models are assessed using PD data that is corrupted by noise. The issue of noise in PD detection has been so significant that researchers have resorted to incorporating artificially generated noise into their PD data to assess the effectiveness of their PD classification models in a more practical scenario [16].

To eliminate noises from PD signals, various denoising methods were implemented and compared. Such as thresholding methods [17], wavelet analysis [14], fast Fourier transform, and different kinds of filtering methods [18]. According to the research [18], among 28 different types of denoising methods, wavelet analysis was proven to be the best methodology for PD denoising with the Mean Square Error method used.

## PD Detection and Classification

Various sensing modalities have been investigated to detect PD activities, including electrical current impulses, chemical by-products, acoustic emission, and electromagnetic radiation [3]. Two types of detection methods exist for partial discharge detection: offline and online, based on the measurement approach. Online monitoring is gaining in popularity because it can reduce power outages and disturbances to the system's operation, while also achieving similar levels of performance [3]. This study will introduce and summarize a few detection methods.

### Traditional machine learning method

#### Artificial Neural Network (ANN)

For many years, artificial neural networks (ANNs) have been utilized in diagnosing partial discharge (PD) [3]. The multilayer perceptron (MLP) is a type of feedforward neural network that is part of artificial neural networks (ANNs) that are inspired by biological neural networks. MLPs are composed of a minimum of three fully connected layers, including one or more hidden layers that activate nonlinearly, along with one input and one output layer. The backpropagation algorithm is commonly used for supervised learning to train feedforward neural networks, which are also referred to as backpropagation neural networks (BPNNs) [3]. The typical structure of BPNN is illustrated in Figure 2.3.4.1a.

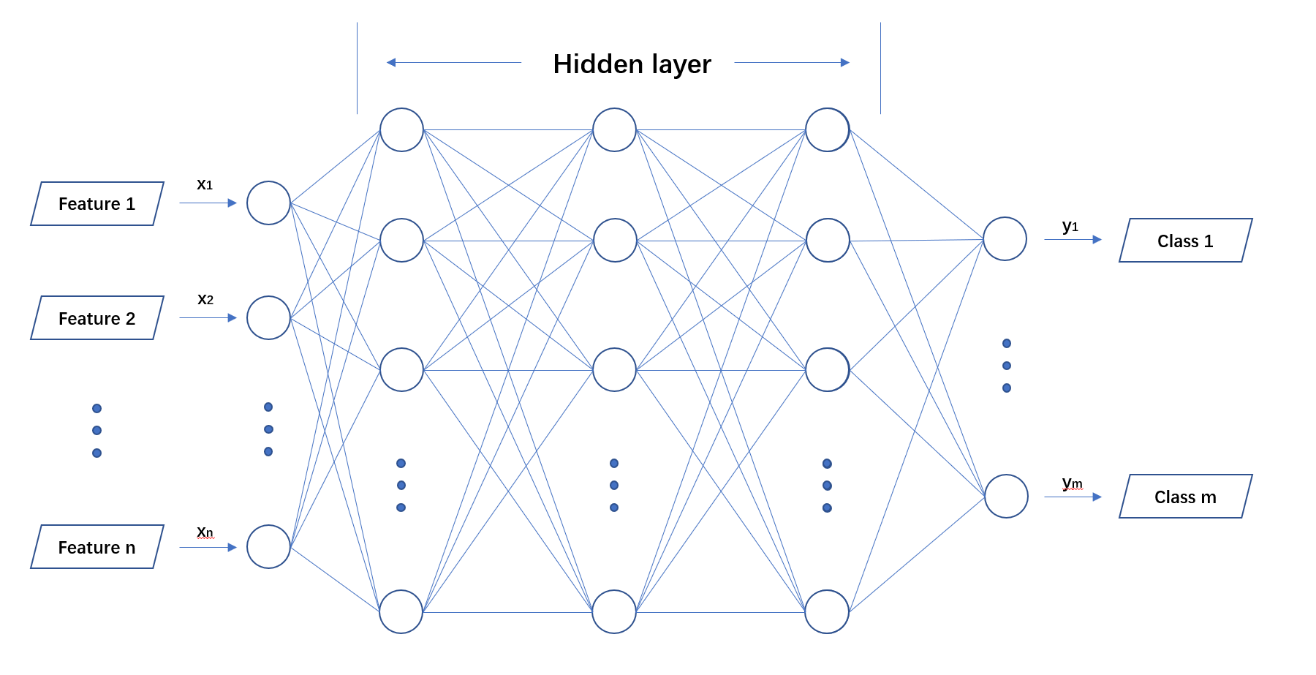


Figure 2.3.1.1a: Structure of a BPNN

There have been numerous studies conducted on utilizing artificial neural networks (ANNs) for the detection and classification of partial discharge (PD). In [19], Kainaga et al proposed a classification method based on a traditional ML algorithm for detecting PD in high-voltage direct current (HVDC) systems. Results showed that with statistical features as input, the model resulted in a classification rate greater than 95%. Polisetty et al. proposed a PD classification method for outdoor insulators using BPNN and harmonic components of acoustic emission signals [20]. The method involved extracting FFT features from UHF signals and using a BPNN-based classifier for PD classification.

ANNs provide various benefits, including the ability to handle large amounts of data and input variables and effectively solve multi-classification problems. Nevertheless, ANNs' black-box nature results in low interpretability. Furthermore, larger ANNs may struggle to identify the global optimal solution, making them vulnerable to overfitting [3].

#### Support Vector Machine (SVM)

SVMs are a type of supervised machine learning algorithm that can be used for classification and regression analysis. The aim of the Support Vector Machine (SVM) algorithm is to find an optimal hyperplane that separates a given dataset into two classes with a maximal margin [21]. Using the linear kernel SVM for a binary classification problem as an example. The formula can be illustrated in equation (1) [3].

From equation (1), the W and b are parameters to determine the hyperplane.

Subsequently, the SVM determines the positive and negative boundaries by identifying the nearest point from the hyperplane for each group. A point that falls above the positive boundary is assigned the label 1 and categorized in one class, while a point below the negative boundary is assigned the label -1 and categorized in the other class following the equation (2). By scaling the distance of the nearest point from the hyperplane in each group to 1, the SVM ensures that the selected hyperplane satisfies the condition for separating the dataset [3].

Lastly, the objective becomes to adjust the optimal hyperplane to achieve the maximum margin between two boundaries. The illustration can be shown in Figure 2.3.4.2a and the problem can be formulated in equation (3) [21]. In addition, to handle datasets that can not be separated linearly, SVM simply adds regularization terms in the cost function in equation (3).

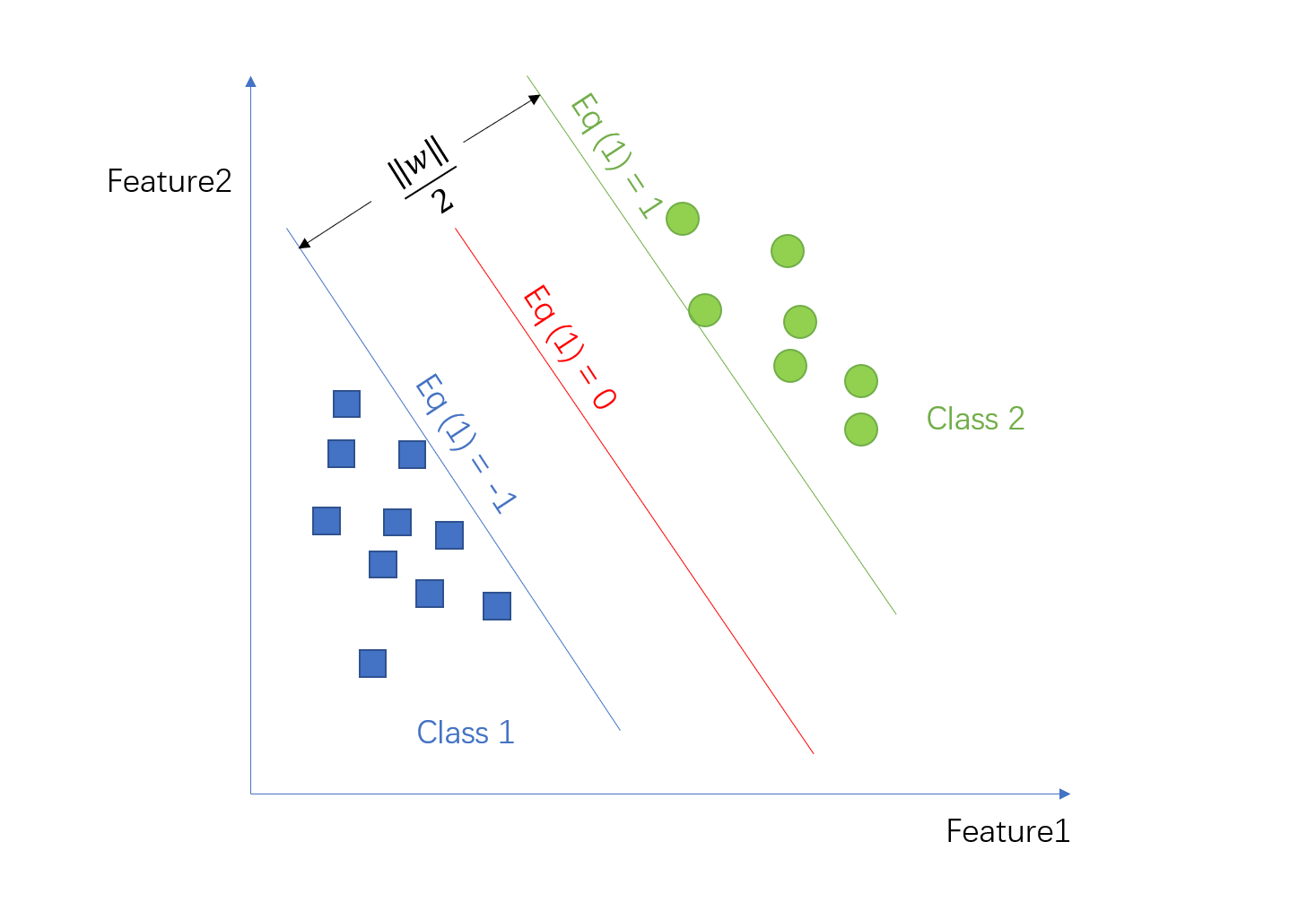


Figure 2.3.1.2a: SVM example

Generally, the basic SVM is only designed for solving binary classification problems [3]. It could simply label non-PD as one class and PD as another for PD detection. However, PD classification usually involves multiclass. As a result, some techniques like one-against-all (OAA) and one-against-one (OAO) were developed in order to reduce the multiclass classification problem to a number of binary classification problems. In addition, the accuracy of an SVM classifier is significantly influenced by the choice of kernel used. There are various types of kernels available, including linear, polynomial, and radial basis function (RBF) kernels. Each kernel type has its own characteristics and can affect the accuracy of the SVM classifier differently [3].

There are many researches about using SVM in PD detection and classification. With statistical features of PRPD as input, in [39], Herath et al evaluated and compared the performance of different supervised machine learning classifiers for the classification of PD. The study found that most of the tested machine learning techniques achieved a high classification accuracy of over 95%. Among the classifiers tested, SVM with a linear kernel and MLP had the highest accuracy. Yao et al. establish the polar coordinate pattern to categorize PD kinds in GIS in [22] using data from the UHF PRPS dataset. Transfer learning method by first applying K-means clustering to the polar coordinate pattern's parameters. Afterward, SVM with Gaussian kernel is then implemented with the generated parameter vector as an input to perform PD classification. The proposed method had an average improvement of 4.4% compared to the conventional method.

### Deep Learning Method

#### Recurrent neural network (RNN)

A Recurrent Neural Network (RNN) is a type of neural network that has feedback connections, allowing it to analyze and make predictions on sequential data. It is specifically designed to handle data with temporal dependencies. RNNs have the unique property of being able to retain information about previous inputs in their internal memory, which allows them to capture long-term dependencies in the data. A simple RNN structure is shown in Figure 2.3.5.1a. It includes a single hidden layer on the left and the temporally expanding layers on the right. The figure clearly shows the unique property of RNN which is being able to retain information about previous inputs in their internal memory by taking inputs x(t) and y(t-1) at every time step t [5]. The neuron in RNN has two sets of weights for two inputs: for and for a, and the formulation can be formulated in equation (4).

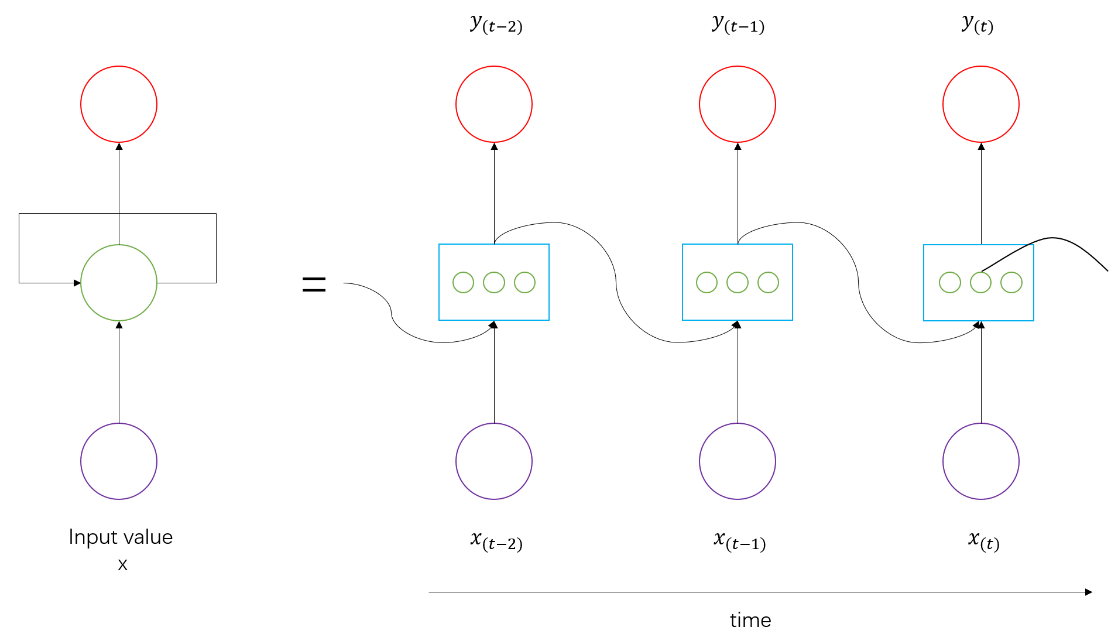


Figure 2.3.2.1a: RNN example

In [23], Nguyen et al. conducted research on classifying various partial discharge sources and an artificial noise source using a recurrent neural network with LSTM layers, trained through an Adam optimizer and early stop technique. The study found that the best results were obtained with two LSTM layers and resulting in an accuracy of 96.62%. However, the experimental data set was imbalanced, which may have biased the classification algorithm toward the majority class and insufficiently learned the characteristics of the minority class, as discussed in [24] [25]. In their conclusion, The proposed model outperformed other techniques they have tried such as Linear SVM, and conventional ANN [23].

#### Convolutional neural network (CNN)

Convolutional Neural Networks (CNNs) have been successful in various tasks, including signal processing and recognition, and are not limited to visual perception [5]. In image classification tasks, a CNN recognizes features such as edges, curves, and ridges, and their combinations [26].

A typical CNN architecture includes several layers, including convolutional layers, nonlinear layers, pooling layers, and fully connected layers.

* Convolutional layers: In these layers, filters or kernels are applied to the input image at different positions, generating feature maps through convolution operations. Each filter learns to recognize a specific feature in the image, such as edges or corners.

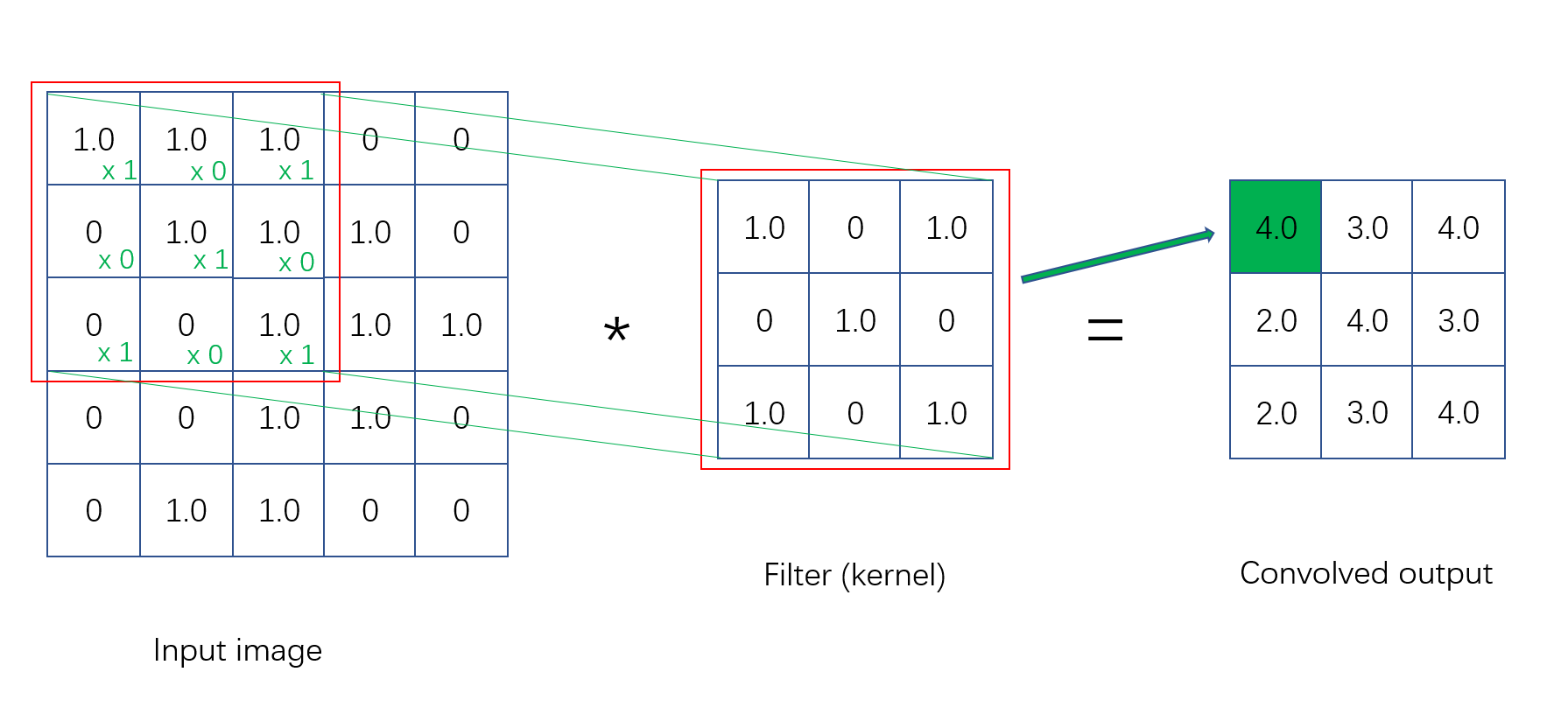


Figure 2.3.2.2a: Simple Convolution operation

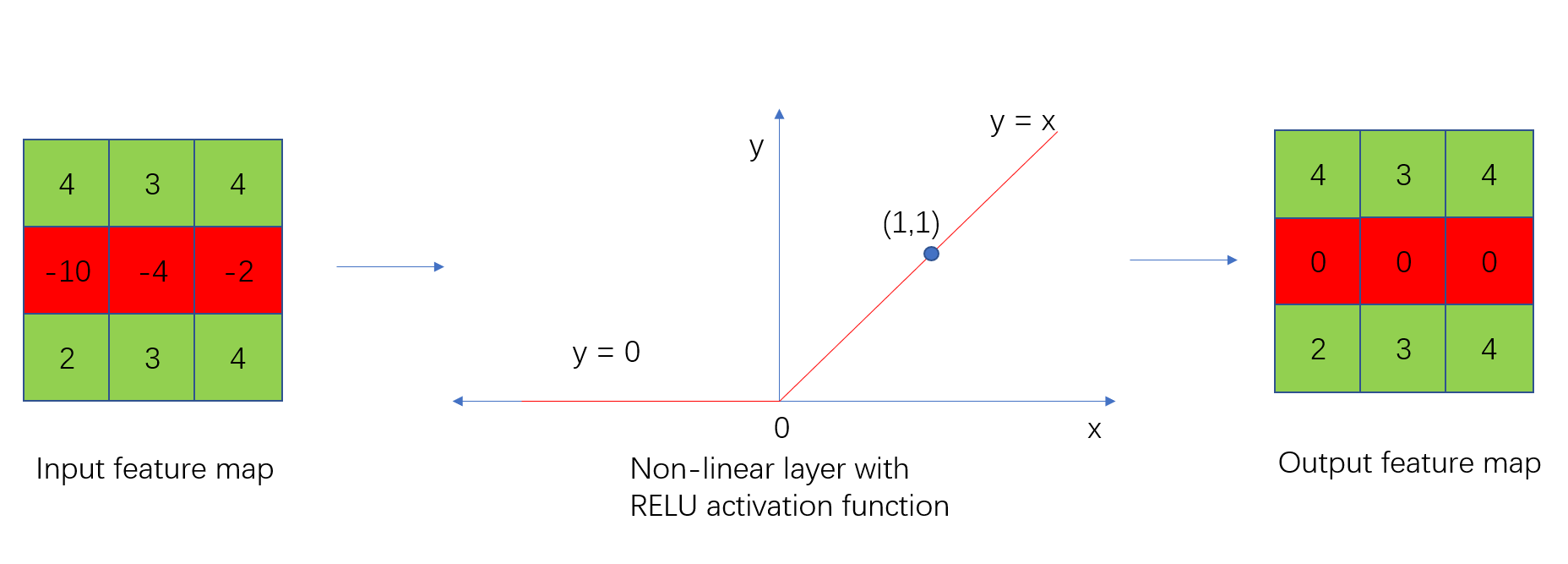
* Nonlinear layers: These layers introduce nonlinearity in the network to capture more complex relationships between features. They use activation functions such as the Rectified Linear Unit (ReLU), which outputs the input value if it's positive, and 0 if it's negative.

Figure 2.3.2.2b: simple non-liner layer with relu

* Pooling layers: These layers reduce the computational load by reducing the size of the feature maps, making the network more efficient. There are three types of pooling operations: max-pooling, min-pooling, and average-pooling. In addition, they also introduce positional invariance by downsampling the feature maps, which makes the network more robust to small changes in the input image.

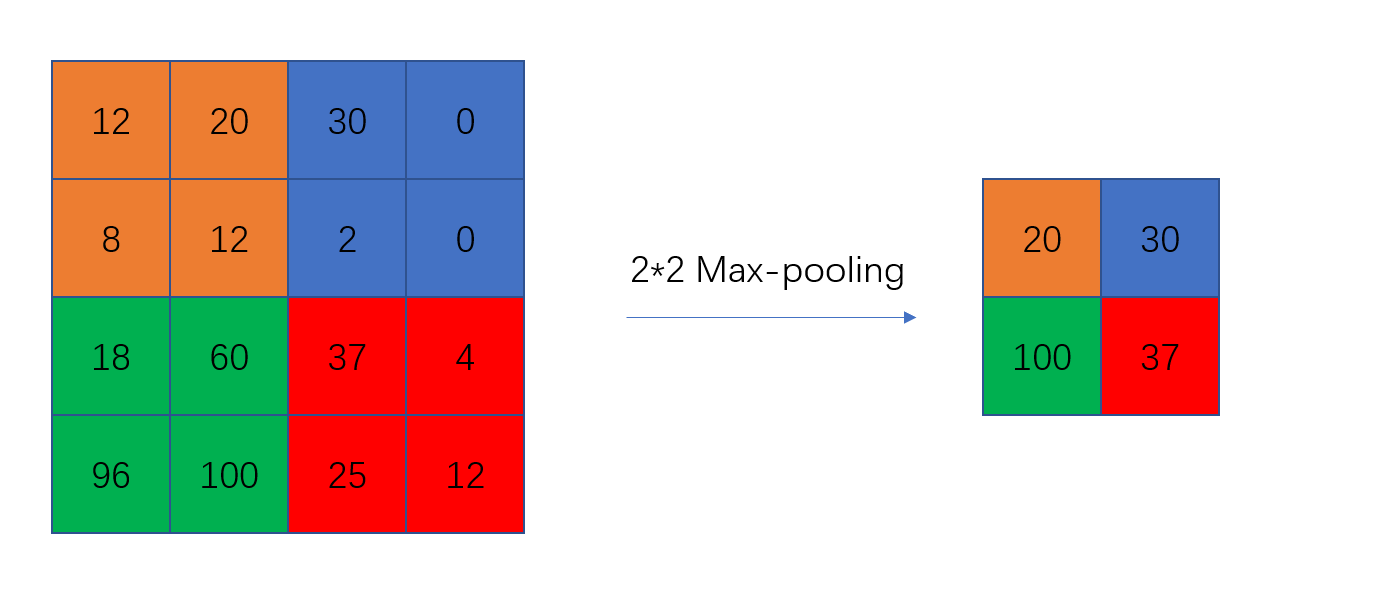


Figure 2.3.2.2c: 2\*2 kernel max-pooling operation

* Fully connected layers: These layers use an Artificial Neural Network (ANN) to make a prediction based on the convolutional features extracted from the previous layers. The output is generated using a softmax function, which provides a probability distribution over the possible classes.

The network's performance is evaluated by backpropagating and calculating the output MSE (mean square error) and minimizing it. The backpropagation process is established by various loss functions. Finally, an optimizer is used to improve the network's accuracy by adjusting the weights and biases of the network during training [5].

In [27], Lu et al. proposed a CNN with five layers, ReLu activation functions, and a softmax output layer to detect PD signals. The input used was time-frequency spectra images and three other methods: Pulse Current, Ultrasonic, and Existing Transient Earth voltage (TEV) were used for comparison. After training, the proposed CNN was found to have the best accuracy in classifying different PD types, disguising PDs from noises, and taking less time to perform PD detection than the other method.

Dey et al. [28] conducted a CNN model with time-series data as input and tested it on a real transformer with an unknown design by simulation. They discovered that when the settings of the transformer changed, the model's performance dropped heavily. However, the CNN model still outperformed competing approaches like self-organizing maps, fuzzy logic, and SVM.

PRPD data was used by Song et al. in paper [29] together with Autoencoder to generate preliminary features for CNN’s input. On top of this transferring method, in the paper, 800 random data PRPD samples were trained and got an accuracy of 89.7%. In addition, the model’s performances were also tested with a mix of half laboratory data and real-life data. The result was shown in Table 2.3.5.2.

|  |  |  |
| --- | --- | --- |
| Model | Only laboratory Data | Mixed data |
| CNN | 95.6 | 86.7 |
| SVM | 91.5 | 74.2 |
| ANN | 90.6 | 70.2 |

Table 2-3-2-2: Result of Song et al

# Experiment Process

## Brief Introduction to the experiment process

With inspiration from previous papers, this study conducted a method that includes traditional machine learning with manual feature extraction before the clustering process [30]. Afterward, different cluster data will be fed into CNN for PD classification to perform the transfer learning method [29].

As shown in Figure 3.1a, firstly features will be extracted from time-series waveform data before the clustering process. Subsequently, extracted features will be used as input for the clustering algorithm to plot PRPD for different clusters in order to separate noises and different kinds of PDs. Lastly, these PRPD data will be fed into CNN for PD classification purposes. In order to apply this model in real life, the miss detection rate’s minimization is crucial since if a PD is classified as non-PD, potential defects will still exist in the insulation system. The equation for calculating the miss detection rate (false negative rate) is shown in equation (5). As a result, another approach with a new layer of transfer learning was implemented by using Siamese Network CNN [31] to create embeddings with triplet loss function [32] from different clusters’ PRPD data. Finally, PD classification tasks will be performed by different traditional ML classification methods for comparison purposes.

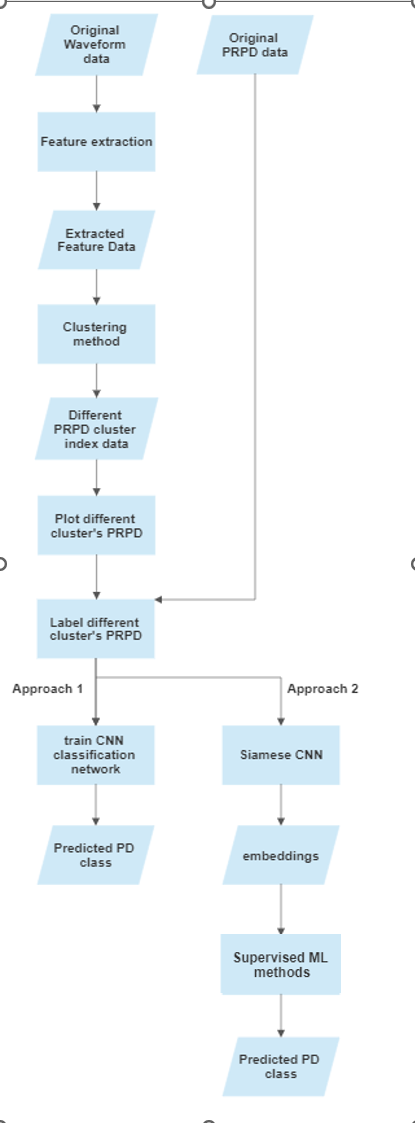


Figure 3.1a: Experiment Framework

## Data description

Two sets of data were captured from the laboratory for each electronic analog signal. One of them is waveform data and another one is PRPD data.

### Waveform data

Waveform data was captured from an artificially generated electronic signal. The equipment was set with a 1000Hz sampling rate and a predetermined time frame of 0.256s. Therefore, for each waveform, 256 data points will be recorded. An example of waveform data is shown in Figure 3.2.1a. For each electronic signal, multiple waveforms will be selected and sampled. The waveform data was saved in .csv or .txt format for computational purposes.

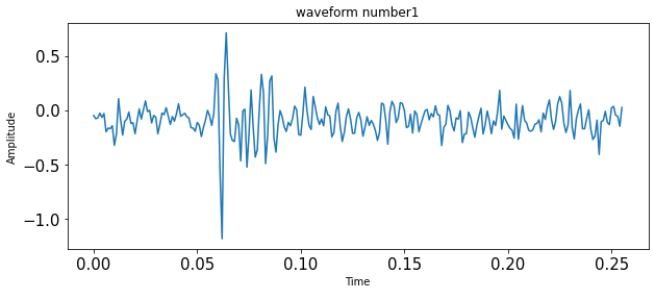


Figure 3.2.1a: a single waveform data captured

Figure 3.2.1b is the captured waveform data list of one generated signal’s shape. The shape of the waveform data list is (5110, 256) which means 5110 waveforms were recorded and each waveform consisted of 256 data points.

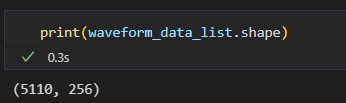


Figure 3.2.1b: captured waveform data list of one artificial electronic signal

### PRPD data

Every waveform captured will be corresponding to one data point in PRPD data. The data point in waveform data with the maximum absolute value would be selected and the corresponding phase angle was calculated. Subsequently, the amplitude and the phase angle of the selected data point were recorded as the data point for the PRPD data. As a result, the shape of the PRPD data of one electronic signal would be [ \_, 2]. An example of PRPD shape is shown in Figure 3.2.2a and the example of PRPD pattern plotted is shown in Figure 3.2.2b.

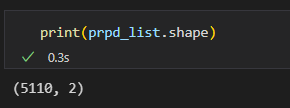


Figure 3.2.2a: captured PRPD data list of one artificial electronic signal

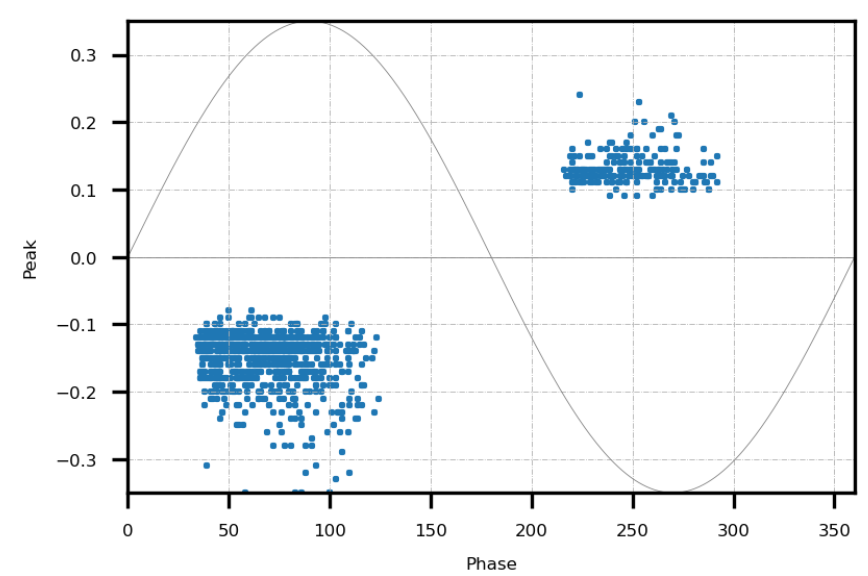


Figure 3.2.2b: Example of one PRPD pattern

## Feature extraction methods

### Fast Fourier Transform (FFT)

The Fast Fourier Transform (FFT) is an algorithm used to efficiently compute the discrete Fourier transform (DFT) of a sequence. The DFT is a mathematical transformation that converts a sequence of time-domain samples into its equivalent representation in the frequency domain. As a result, Fast Fourier Transform (FFT) can be used to analyze the frequency content of the PD signals. The FFT algorithm can convert PD signals from the time domain to the frequency domain which is useful for PD detection and classification because different types of PD and noise normally have different core frequencies.

The equation for the DFT of a sequence x of length N is shown in equation (8).

where k is the frequency index and X[k] is the corresponding frequency component of the DFT.

By iteratively dividing the input sequence into smaller subproblems of length N/2 and then combining the outputs in a certain fashion, the FFT algorithm calculates the DFT. This procedure is repeated until the subproblems are trivial, which happens when the length of the subproblems equals 1. Due to its utilization of the symmetry and periodicity of the sinusoidal functions employed in the DFT, the FFT algorithm is significantly faster than the conventional DFT approach. As a result, the method can reuse computations and reduce unnecessary computations.

### Power Spectrum Density (PSD)

The Power Spectral Density (PSD) provides an easier way to visually represent the distribution of signal frequency components, compared to the complex Discrete Fourier Transform (DFT). The PSD shows how much of the total signal power is contributed by each frequency component of a voltage signal, where power is computed as V^2/R. To obtain the PSD, the mean squared amplitude of each frequency component is determined by using the DFT and averaging the results over the n samples in the digitized record. However, since only half of the frequency components are unique, the two halves of the DFT are combined, which doubles the power of each component. The resulting values are then plotted as the lower components, k = 1 … n/2+1, and the equation is shown in equation (9).

### Continous Wavelet Transform (CWT)

The Continuous Wavelet Transform (CWT) is a mathematical tool used in signal processing and analysis which enables the breakdown of a signal into its time-frequency representation. It is a sort of time-frequency analysis that examines signals simultaneously in the time and frequency domains using a collection of wavelet functions.

A set of wavelets that have been scaled and translated in time are convolved with the signal in the CWT process. This method enables the simultaneous detection of various frequency components of a signal. The CWT creates a two-dimensional representation of the signal in time and frequency, with the frequency axis being the inverse of the wavelet scale and the time axis being the original time domain of the signal.

The CWT can be applied to a variety of signal-processing tasks, including feature extraction, pattern identification, and signal denoising. The CWT's ability to provide a higher time-frequency resolution than other time-frequency analysis methods like the Short-Time Fourier Transform is one of its benefits (STFT).

## Compare different clustering methods and features

### Test data

Two sets of signal data with different levels of noise were given as a benchmark for evaluating the performances of combinations of different features and clustering methods

The PRPD patterns with noises are shown in Figure 3.5.1a and Figure 3.5.1c.

The desired PRPD patterns separated using the clustering method are shown in Figure 3.5.1b and Figure 3.5.1d.

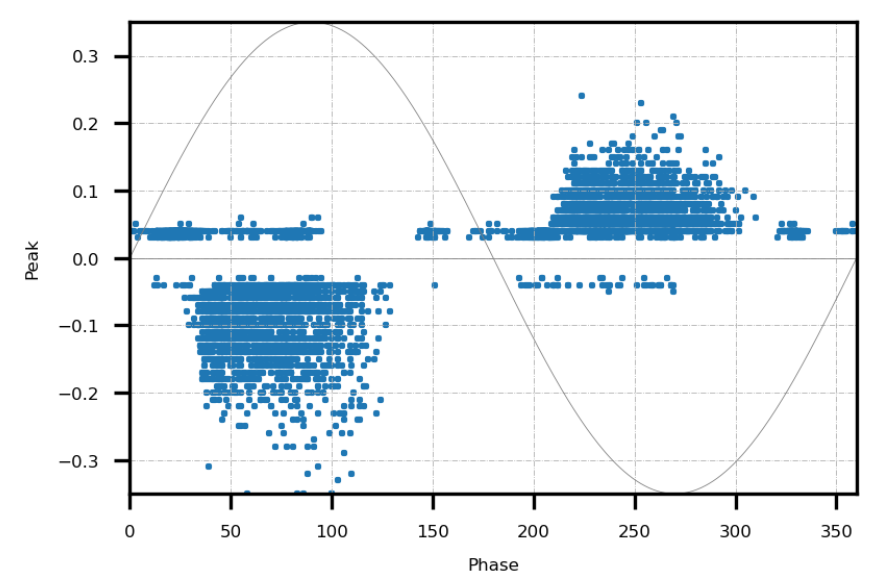


Figure 3.5.1a: Less noisy PRPD pattern (internal PD)

Chart

Description automatically generated

Figure 3.5.1b: Example of desired one cluster of less noisy PRPD pattern

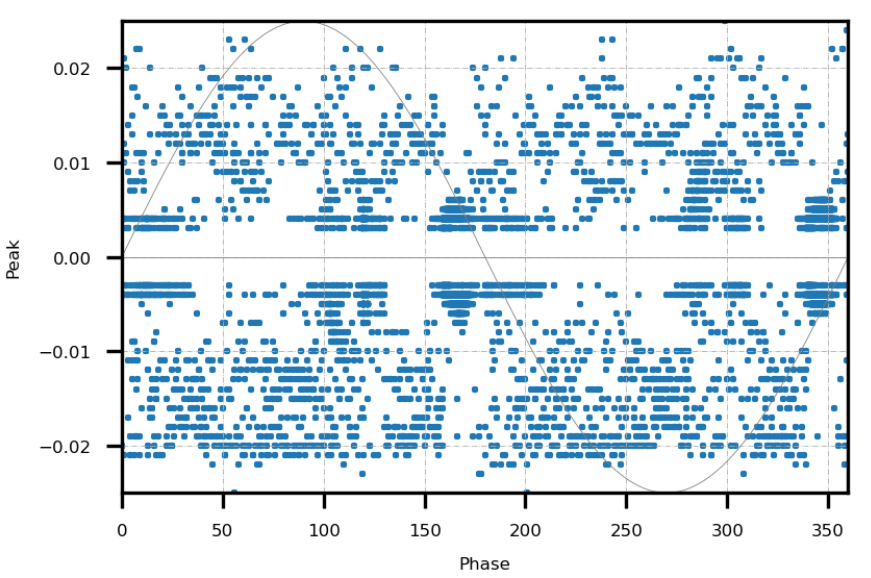


Figure 3.5.1c: Noisier PRPD pattern (internal PD)

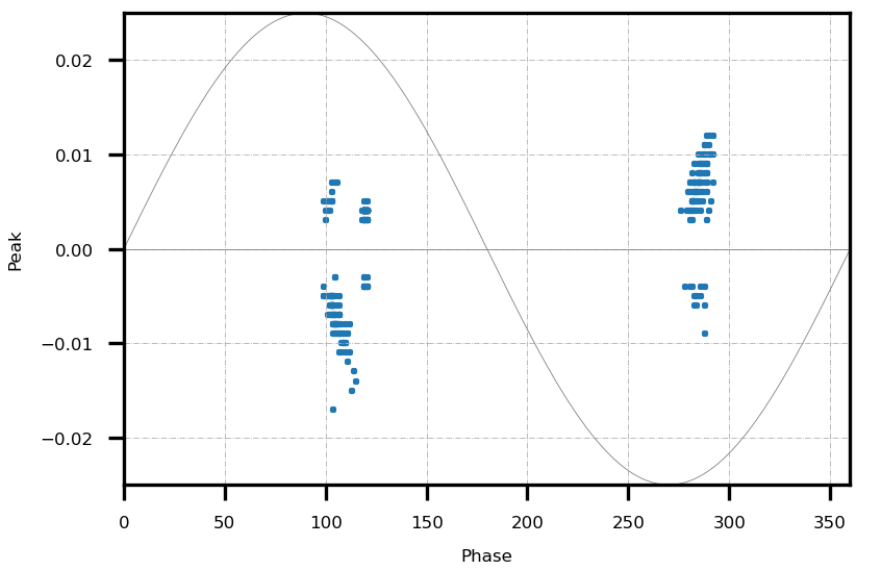


Figure 3.5.1d: Example of desired one cluster of noisier PRPD pattern

### Clustering result of chosen algorithm and feature (denoising process)

An unsupervised clustering method will be chosen to separate data points of the two sets of testing PRPD data. In addition, one extracted feature from the corresponding waveform data will be selected as input to the clustering algorithm. The results of each combination of the clustering algorithm and extracted feature are listed below.

The goal of the clustering method is that the selected clustering method with the extracted feature should be able to separate the desired cluster’s PRPD from the original PRPD pattern and the less processing time taken the better. Different combinations of results are shown below.

Diagram, engineering drawing

Description automatically generated

Figure 3.5.2a: mean-shift +FFT

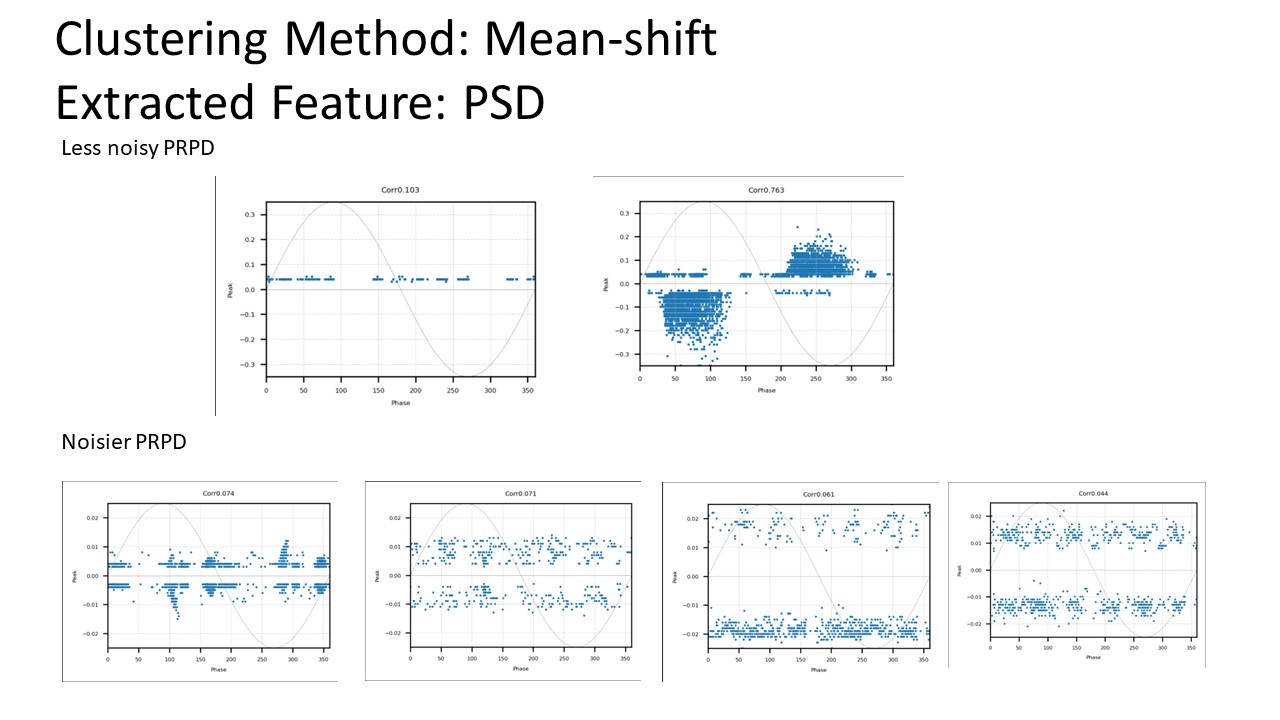


Figure 3.5.2b: mean-shift + PSD

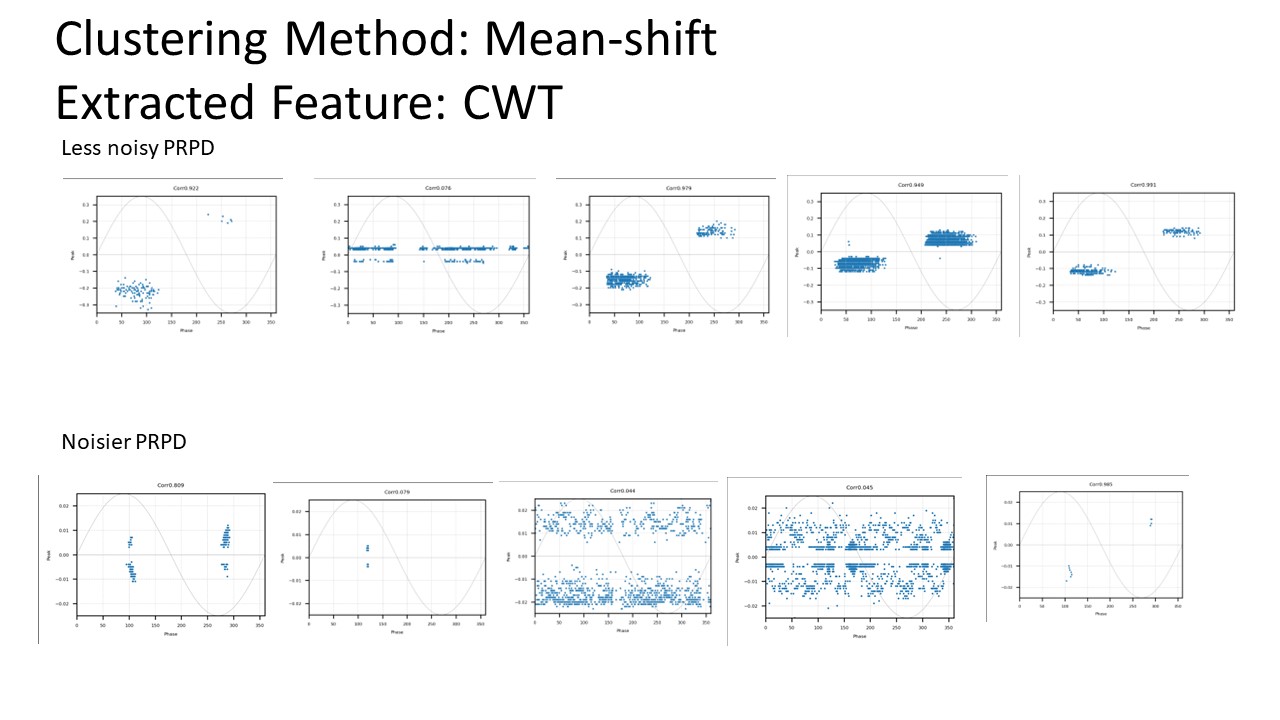


Figure 3.5.2c: mean-shift + CWT

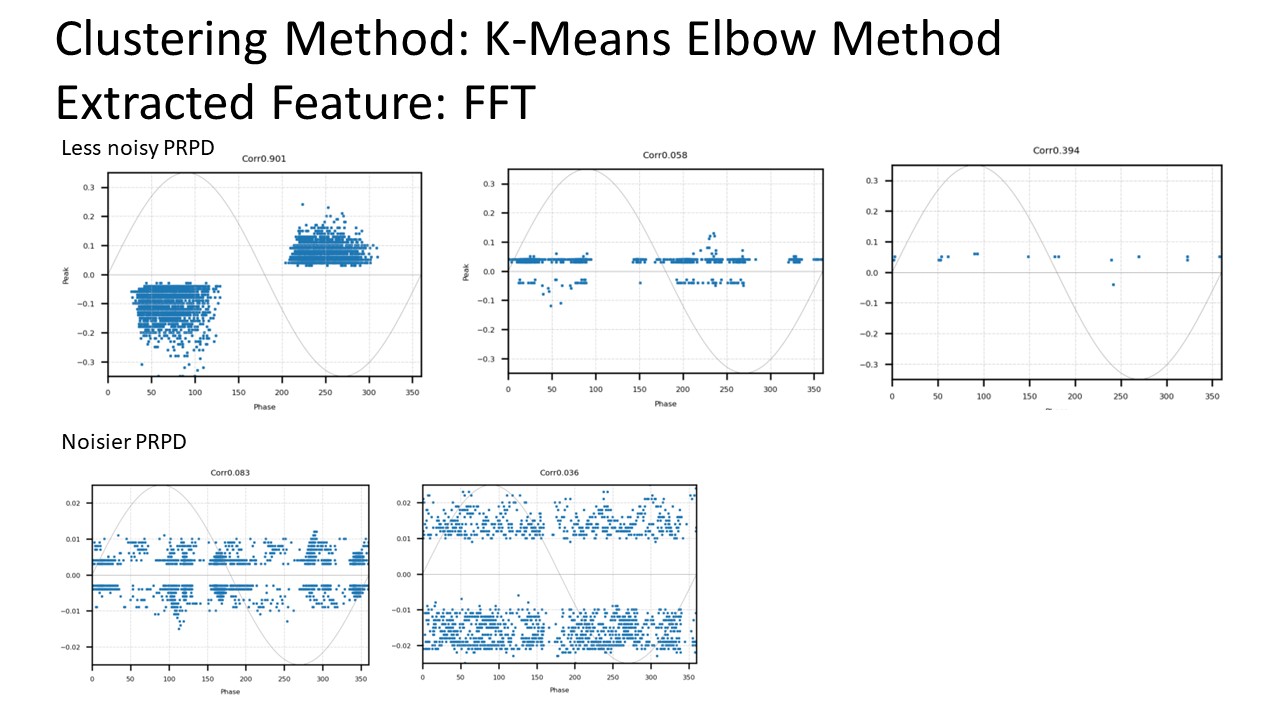


Figure 3.5.2d: K-Means Elbow + FFT

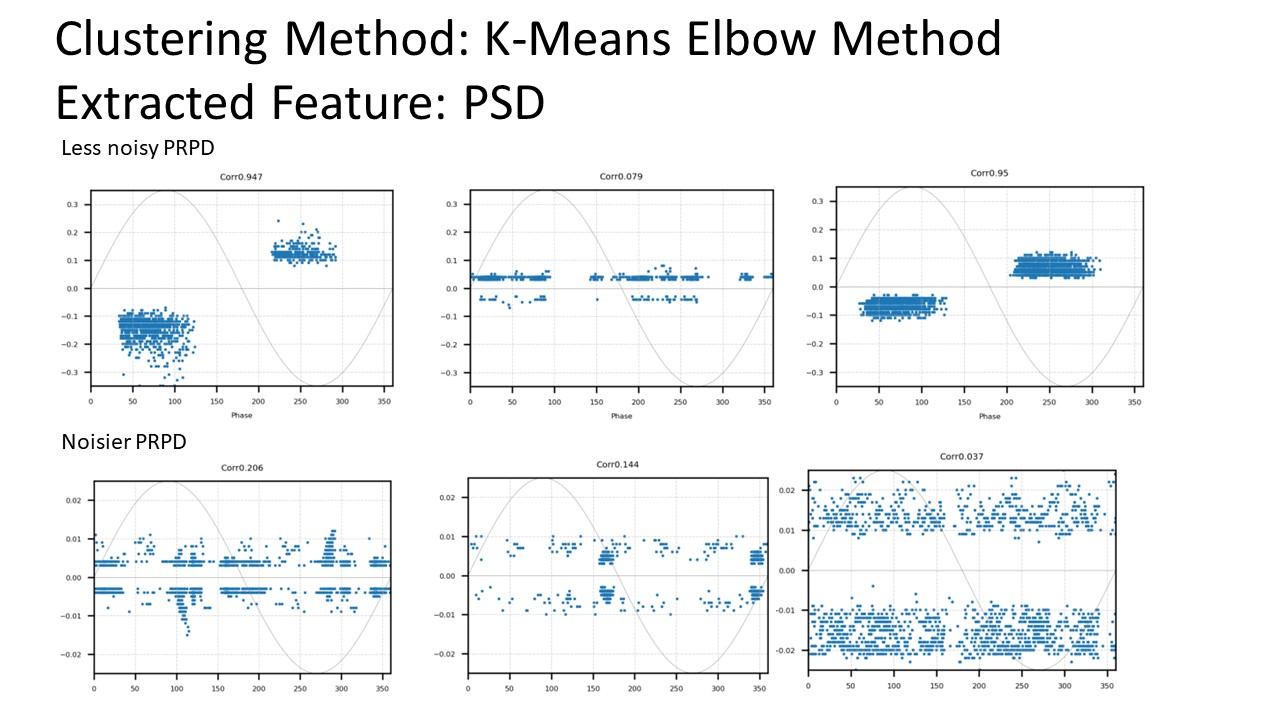


Figure 3.5.2e: K-Means Elbow + PSD

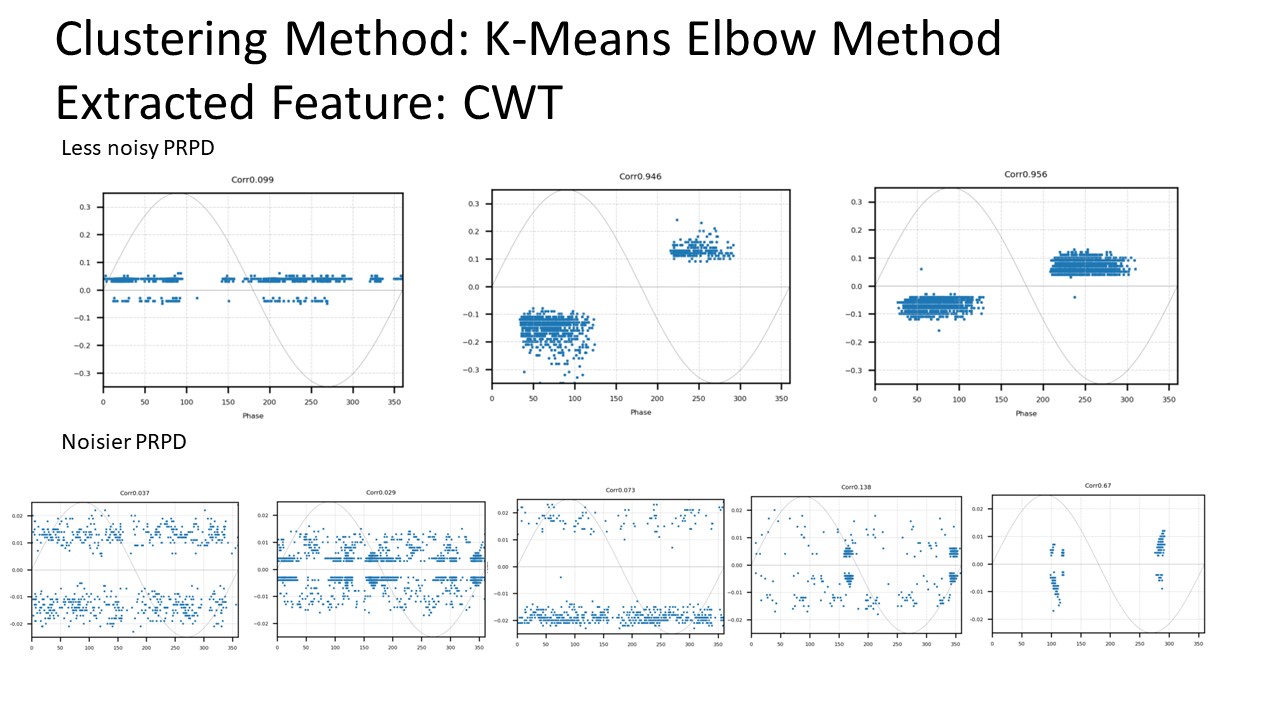


Figure 3.5.2f: K-Means Elbow + CWT

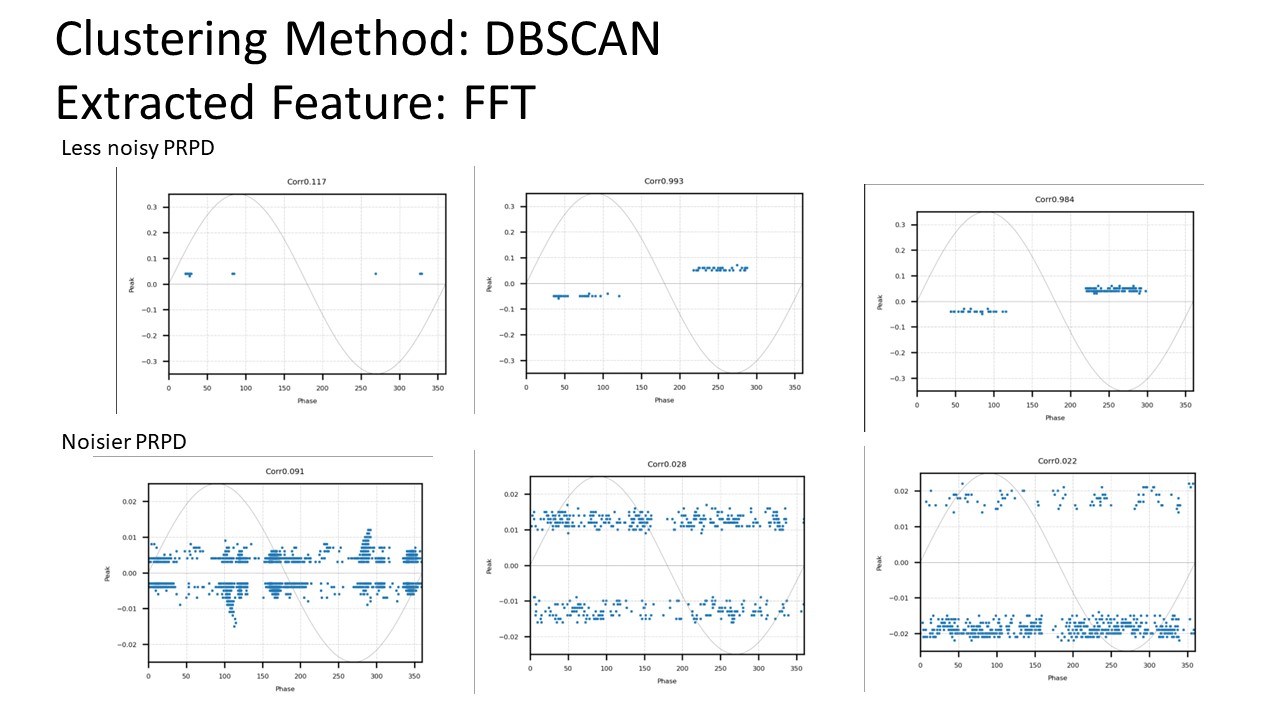


Figure 3.5.2g: DBSCAN + FFT

A picture containing chart

Description automatically generated

Figure 3.5.2h: DBSCAN + PSD

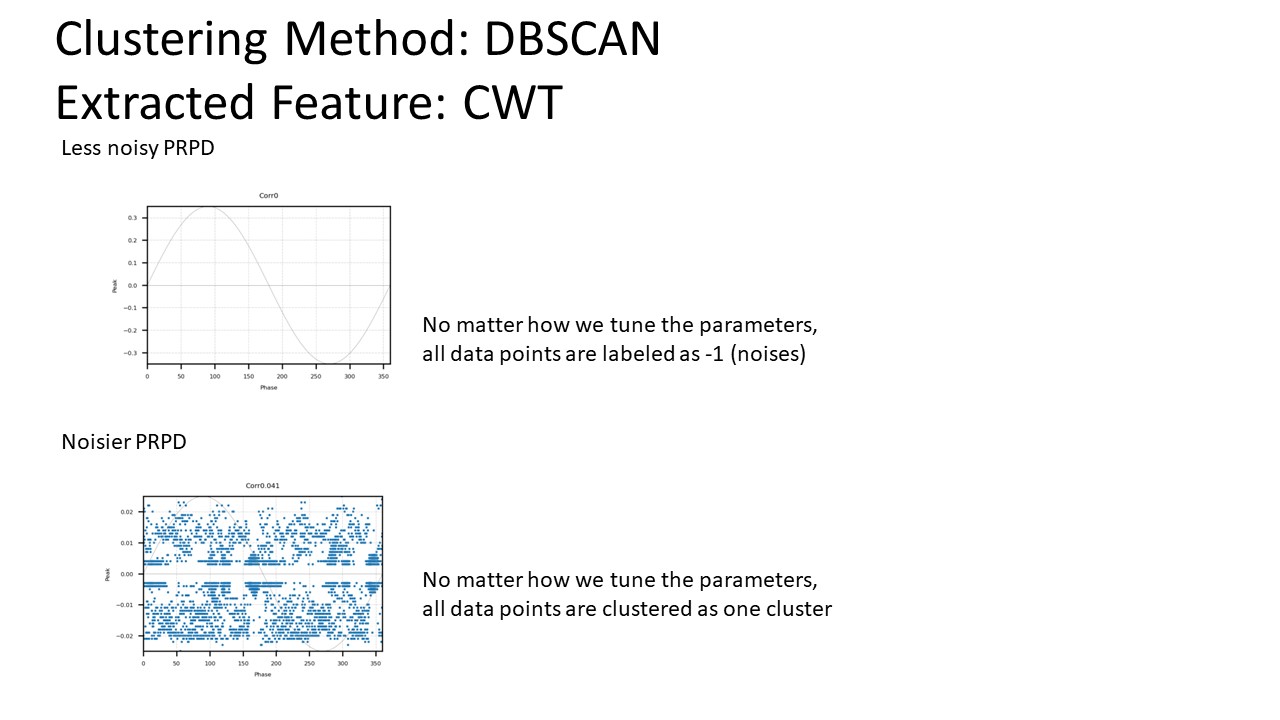


Figure 3.5.2i: DBSCAN + CWT

**Processing time table**

|  |  |  |  |
| --- | --- | --- | --- |
| Less noisy PRPD | Mean-shift | DBSCAN | K-means elbow |
| CWT | 515s | 63.7s | 71.23s |
| FFT | 60.79s | 1.82s | 4.29s |
| PSD | 249s | 3.8s | 7.21s |

Table 3-4-3-a: processing time of each combination of features and clustering method for less noisy PRPD test pattern

|  |  |  |  |
| --- | --- | --- | --- |
| Noisier PRPD | Mean-shift | DBSCAN | K-means elbow |
| CWT | 815s | 189s | 104s |
| FFT | 132s | 4.89s | 6.74s |
| PSD | 118s | 5.8s | 11.2s |

Table 3-4-3-b: processing time of each combination of features and clustering method for noisier PRPD test pattern

### Result discussion

Here we selected three clustering methods which are the K-means elbow method, mean-shift, and method Density-Based Spatial Clustering of Applications with Noise (DBSCAN) method to compare and test the performance of each extracted feature. Because we don’t know how many types of PDs and noises will be in one signal, the clustering method we selected are all able to determine the best cluster numbers.

* **DBSCAN can’t get the result**

From all the combinations we could discover that DBSCAN sometimes can’t get any result from the feature selected, such as PSD and CWT. We think it might be because our features are all high dimensional data which is not suitable for DBSCAN’s input.

* **Clustering performance**

From the performance graph of clustering, we could find that only the combination of the K-means elbow method + CWT features and mean-shift + CWT features could get the desired result for both less noisy and noisier PRPD test patterns.

* **Processing time**

Comparing the processing table vertically, we could find CWT usually takes the longest time to process. Compare to the processing table horizontally, the mean-shift method generally requires more time to get the result. In addition, usually, noisier PRPD requires more time to process.

* **Why CWT can retrieve the desired result**

From the clustering result, we can see that CWT is the best feature for clustering. PSD and FFT usually can get good results for less noisy PRPD patterns but it is difficult to separate the desired cluster from noisier PRPD. From Chapter 2.11 we know that PD typically results in a transient current pulse that climbs incredibly quickly and whose pulse width varies depending on different kinds of patterns. So not only the frequency changes very quickly during this process, the pulse width is time-related. As a result, we will need both time and frequency features to perform clustering. Therefore, CWT which captures both the time and frequency features could result in better performance than PSD and FFT which only represent the frequency component of the signal.

* **Conclusion**

To conclude, CWT + Mean-shift or K-means elbow could both retrieve the desired result. However, the processing time of the K-means elbow method taken is much less than the Mean-shift method. So from the experiment result, we think the features should be using CWT and apply the K-means elbow clustering method.

## Simple CNN approach

### CNN structure

The first approach to performing PD classification will be simple CNN approaches. The network structure presented in this study has 3 convolution layers, each followed by a batch normalization layer and a RELU activation. In addition, 3 max-pooling layers, 1 fully connected layer, 1 dense layer, and another dense layer with softmax activation function were also included in the network. The structure of the model is shown in Figure 3.6.1a.

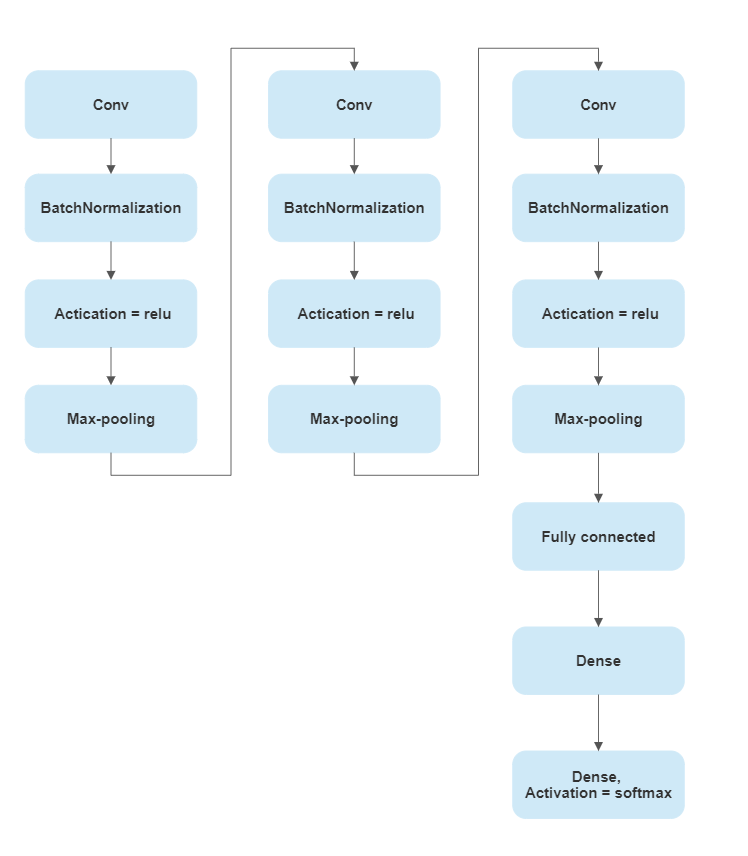


Figure 3.6.1a: CNN structure of the first approach

### Train CNN

#### Balance data

The training data has 3 classes: 0 for noise with 73 images, 1 for corona PD with 116 images, and 2 for internal PD with 128 images. The training data set is unbalanced which could cause potential problems for the training process and result [24] [25]. Therefore data set balanced process needs to be taken before training the CNN. The code snippet is shown in Figure 3.6.2.1a.

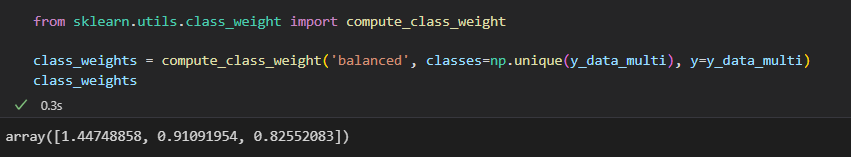


Figure 3.6.2.1a: Balance data code snippet

However, one of the model’s goals is to make FNR as low as possible in order to prevent miss detection problems. Therefore, after creating a balanced class weight, we could make the weights of the corona and internal higher. The reason is that classes with higher weights will be penalized more during the backpropagation process which could result in the model capturing more characteristics about the corresponding class. As a result, after a few times of experiments, we chose to multiply 2 for the original balanced class weight of corona and internal class which is shown in Figure 3.6.2.1b.

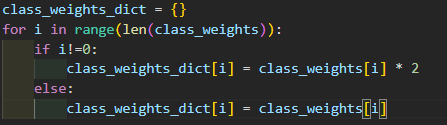


Figure 3.6.2.1b: Multiply corona and internal class weights by 2

#### Prevent overfitting

According to [33], overfitting is a common problem in deep neural networks and it is caused by the continuous updating of gradients and the sensitivity of the cross-entropy loss to scale. This problem happens because generally deep neural networks are overparameterized during the training process which may undermine the effectiveness of their applications in solving different network problems [34]. In order to prevent the overfitting problem, early stop and reduced learning rate techniques were employed in this experiment.

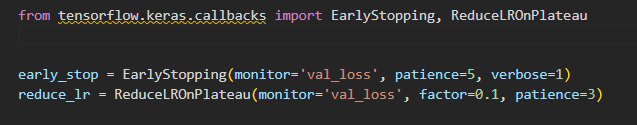


Figure 3.6.2.2a: Early stop and reduce learning rate setup

#### Data preprocessing

Data preprocessing is needed for data to be fed into CNN. The PRPD pattern data was stored in JSON format without any normalization which could not directly feed into CNN because CNN only receives image data with shapes like (50, 32, 32, 3). 50 is the number of images, 32 \* 32 is the dimension of the input image and 3 is the channel number. However, the shape of stored PRPD data is (num\_patterns, data\_points\_per\_pattern, 2). 2 represents the data of phase angle and amplitude. The problem is that data\_points\_per\_pattern is not consistent for each PRPD pattern.

To solve this problem, downsampling and normalization process were performed for each PRPD pattern data. The target shape is (num\_images, 36, 36, 1). The dimension of the PRPD pattern was predetermined to be (36, 36) which means the original x-axis of 360 degrees was split into 36 parts which are 10 degrees per division. In addition, the y-axis of amplitude was split into 36 parts but each PRPD pattern will have different ranges of amplitude. To normalize the PRPD image data, the maximum amplitude of each PRPD pattern was taken out and divided by 36 to calculate the y-axis’s division value. Finally, each division will contain multiple data points and the number of data points per division will be used for the last dimension 1. The detailed operation and code snippet are shown in Figure 3.6.2.3a.

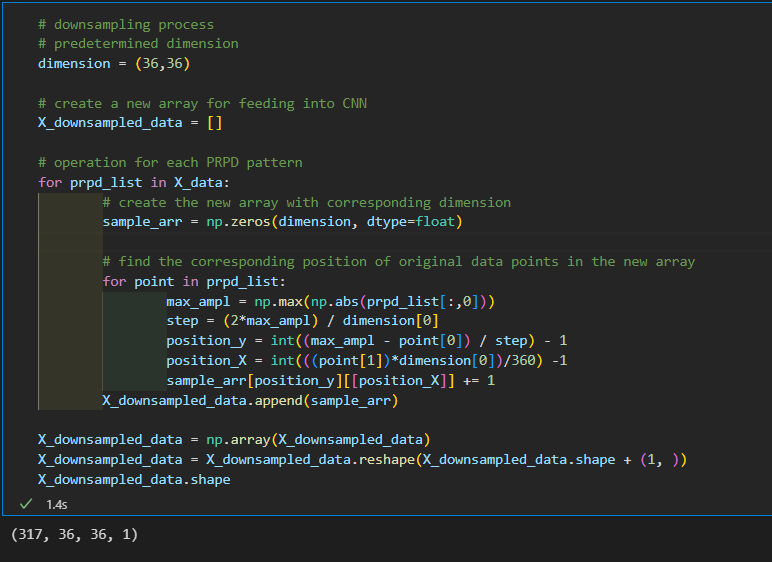


Figure 3.6.2.3a: Downsampling

#### Tunning parameters and comparing the results

In order to train the CNN, we will first need to split our data into train and test sets. To test the model’s power, we use only 20% to train the network but 80% to test as shown in Figure 3.6.2.4a.

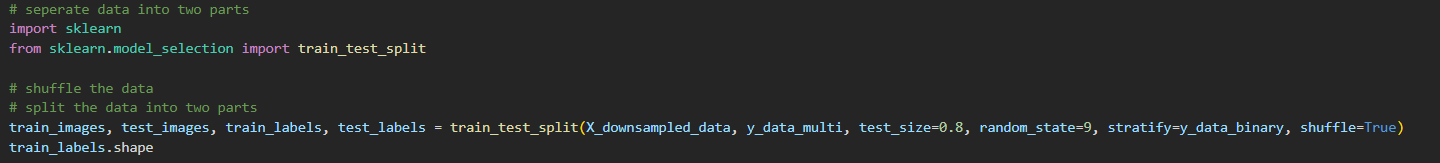


Figure 3.6.2.4a: Data split

Subsequently, the model will be compiled with the selected optimizer, learning rate, and proper loss function in Figure 3.6.2.4b.

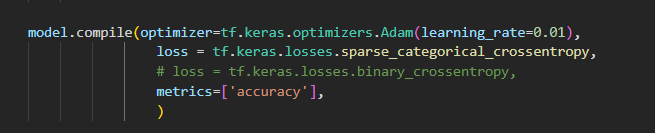


Figure 3.6.2.4b: Compile setup

Finally, the early stop, reduce learning rate, and class weights were all put in during the fitting process as shown in Figure 3.6.2.4c. The batch size was set to 16 due to few training images. In addition, epochs were set to 100 because we have early stop and reduce learning rate techniques, we don’t need to worry too much about overfitting.

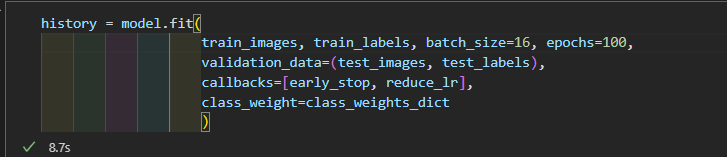


Figure 3.6.2.4c: Fit setup

The result is shown in a confusion matrix plot (Figure 3.6.2.4d). The FNR is calculated by equation (6) and overall accuracy is calculated by equation (7). The class weight with “special design” means the corona and internal class weight were multiplied by 2.

Chart

Description automatically generated

Figure 3.6.2.4d: Confusion plot of CNN performance result

Here the coronanoise means the number of corona but predicted as noise. Corona means simply the total number of real corona values. For example in Figure 3.6.2.4d, the FNR is (5 + 7) / (5+87+1+7+3+93) = 0.0612.

As a result, the overall accuracy in Figure 3.6.2.4d is (48 + 87 + 93) / (48+10+0+5+87+1+7+3+93) = 0.8976.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model Name** | CNN 1 | CNN 2 | CNN 3 | CNN 4 |
| **conv layer kernel-size** | 3 \* 3 | 5 \* 5 | 3 \* 3 | 5 \* 5 |
| **neurons per con layer** | 64,128,128 | 64,128,128 | 32,64,32 | 32,64,32 |
| **fc neurons** | 64 | 64 | 64 | 64 |
| **learning rate** | 0.001 | 0.001 | 0.001 | 0.001 |
| **class weight** | special design | special design | special design | special design |
| **FNR** | 0.1581 | 0.0816 | 0.1173 | 0.0612 |
| **overall accuracy** | 0.811 | 0.8701 | 0.8386 | 0.8976 |

Table 3-6-2-4-a: Result of CNN model 1, 2, 3, 4

When comparing both CNN1 with CNN2 and CNN3 with CNN4, we could discover that the 5\*5 kernel has a better performance compared with the 3\*3 kernel. Then, when comparing CNN2 and CNN4, we can see that the neurons per convolution layer are not the more the better. In contrast, CNN4 with half neurons per convolution layer of CNN2 has a better result. Therefore, we can fix our kernel size to be 5 and neurons per convolution layer to be 32,64,32 to continue our experiment

Next, the learning rate will be changed and also the effect of special design will be shown in Table 3.6.2.4d.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model Name** | CNN 5 | CNN 6 | CNN 7 | CNN 8 | CNN 9 |
| **conv layer kernel-size** | 5\*5 | 5 \* 5 | 5 \* 5 | 5\*5 | 5 \* 5 |
| **neurons per con layer** | 32,64,32 | 32,64,32 | 32,64,32 | 32,64,32 | 32,64,32 |
| **fc neurons** | 64 | 64 | 64 | 64 | 64 |
| **learning rate** | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| **class weight** | special design | special design | special design | simple balanced | simple  balanced |
| **FNR** | 0.0051 | 0.0306 | 0.0306 | 0.0357 | 0.0255 |
| **overall accuracy** | 0.9213 | 0.9370 | 0.8858 | 0.9331 | 0.9449 |

Table 3-6-2-4-b: Result of CNN model 5,6,7,8,9

From Table 3-6-2-4-b, for three models CNN5, 6, and 7 with special design class weight, we consider CNN 5 to be the most desired model because it has the lowest FNR rate and a rather high overall accuracy. For models with simple design class weight, two models CNN8, 9 with the best performance were selected and put into Table 3-6-2-4-b. When we compare CNN 9 with CNN5, we can see that CNN9 has higher overall accuracy than CNN5 but higher FNR than CNN5 which shows that the special design of class weight forced the model to capture more features of corona and internal PD.

The accuracy curve and confusion plot of CNN 5 and 9 will be listed below.

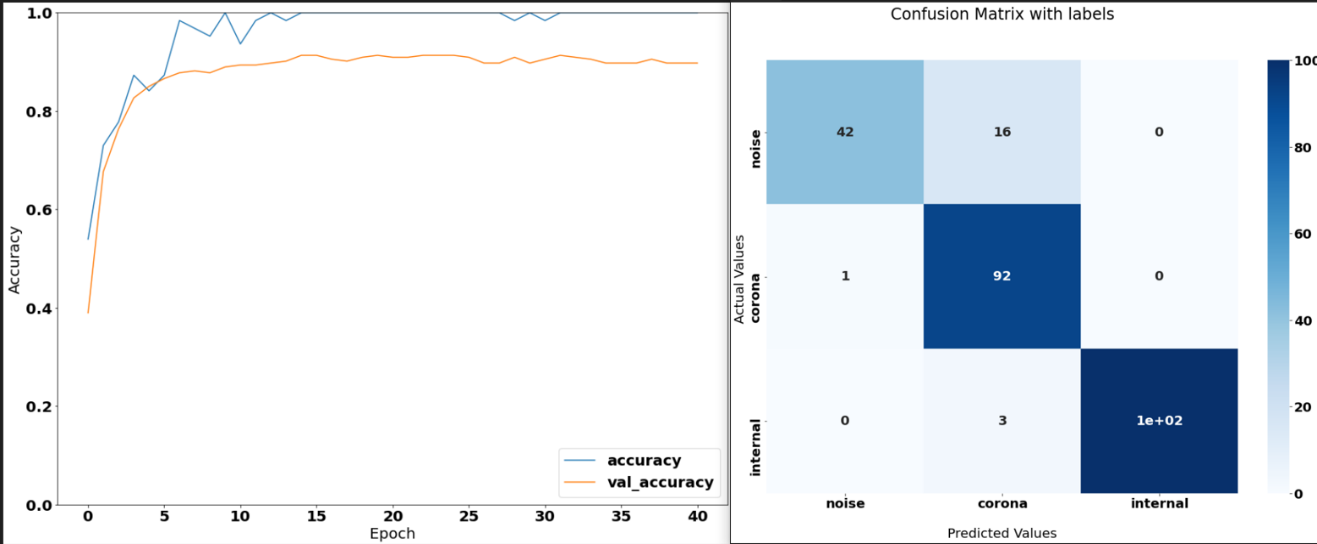


Figure 3.6.2.4e: Accuracy curve and confusion plot of CNN 5

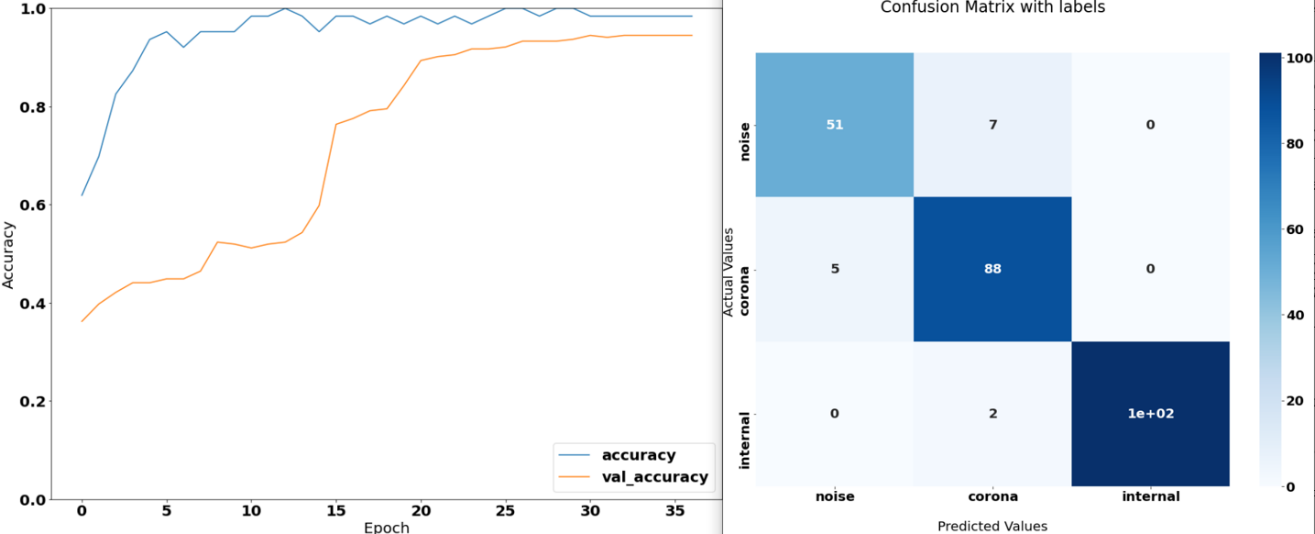


Figure 3.6.2.4f: Accuracy curve and confusion plot of CNN 9

## Triplet Loss Approach

Triplet loss metrics were first proposed in [32] for metric learning tasks, particularly in image and video recognition. Metric learning aims to learn a distance metric between pairs of data points, such that similar data points are mapped close to each other and dissimilar data points are mapped far apart from each other. In our experiment, due to the equipment’s limitation, the captured corona PD looks similar to some of the noise data samples. To prevent the phenomenon of intra-class distance being bigger than inter-class distance for our trained CNN, we proposed to use triplet loss approaches in our experiment. In addition, we will need to minimize the miss detection rate which also could use the triplet loss function to achieve.

The equation of triplet loss calculation is shown in (10).

The input is a triplet which includes anchor (a), negative sample (n), and positive sample (p). The positive sample is supposed to come from the same class as the anchor, and the negative sample should be selected from a different class as the anchor. The similarity calculation between samples is achieved by optimizing the distance between anchor examples and positive examples to be smaller than the distance between anchor examples and negative examples.

Moreover, the triplet can be classified into three types.

Easy triplet: L = 0, which means . This situation doesn’t need optimization.

Hard triplet: L > margin, which means . This situation has a serious problem that needs optimization.

Semi-hard triplet: L < margin, which means . Although the loss isn’t the biggest but still need optimization.

For our experiment, we tried to use semi-hard and hard triplets to train our CNN embeddings and plot the loss curve during the training process. The same early stop and reduced learning rate techniques were implemented here to prevent overfitting. The embedding CNN network architecture is shown in 3.6a.

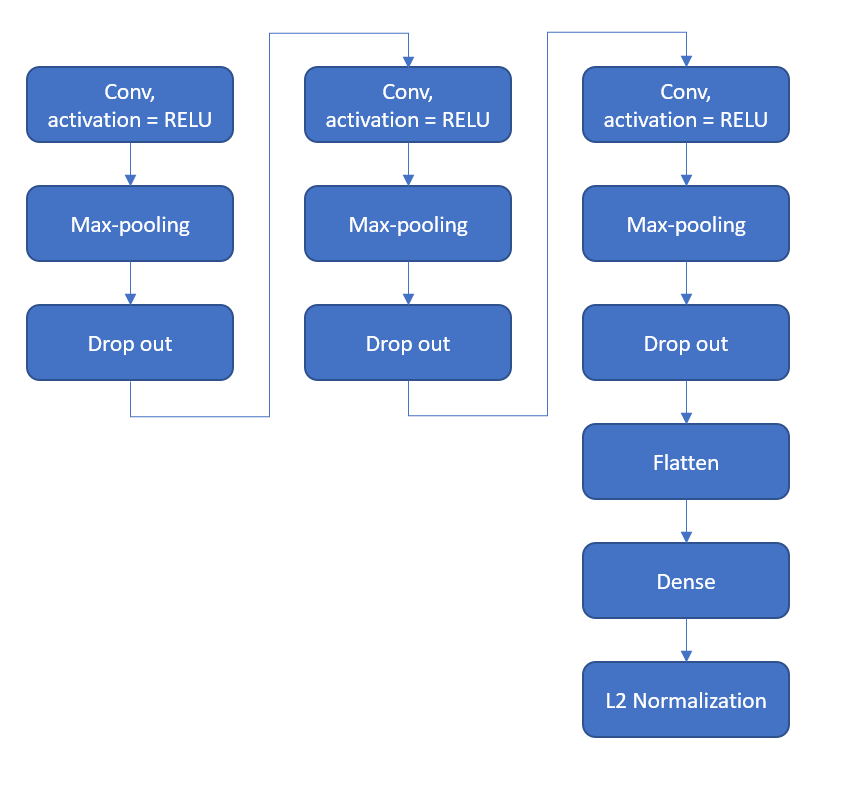


Figure 3.6a: Structure of CNN For capturing embeddings

The best result we have is training using hard triplet with margin L = 1.0 and Adam optimizer with 0.0005 learning rate. The embedding of CNN’s result is shown below.

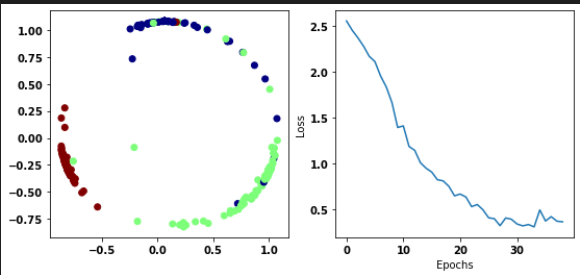


Figure3.6b: The embeddings generated and the loss curve of embedding CNN

The embedding with 3 classes blue (noise), green (corona), and red (internal). We can see that the red cluster is away from the other two clusters. But there are some blue and green points mixed together which basically means the noise and corona samples have some similar samples in our dataset. The loss curve looks good because it is continuously deceased.

Then we use the embedding as an input for SVM to classify different PDs. We get an accuracy of 94.12%, with FNR = 3.8% and the confusion matrix is shown in Figure 3.6c. In addition, we used the same parameters and trained another model with 93.6% accuracy and 2.5% FNR.

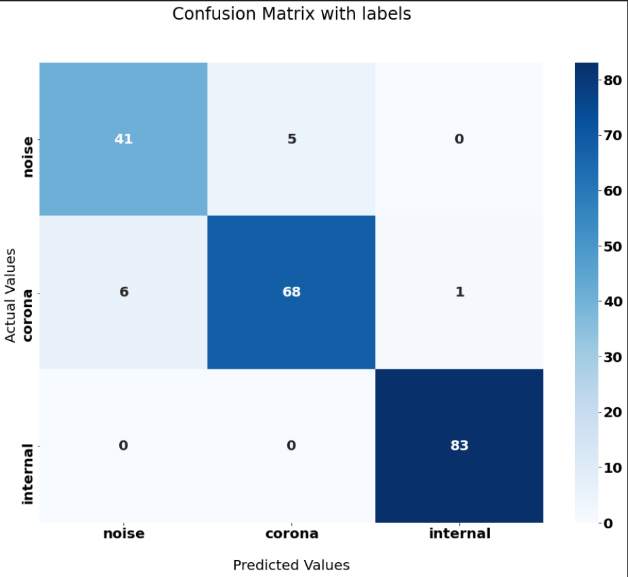


Figure 3.6c: confusion plot of CNN triplet with best overall acc

# Chapter 4

**Conclusions and Future Work**

# 4.1 Conclusions

In our experiment, the proposed transfer learning method based on manual feature extraction, traditional machine learning clustering method, and CNN can be used to perform PD classification. The best-extracted feature and clustering method combination after comparison was CWT and K-mean elbow method. If using simple CNN approaches, the miss detection rate minimization problem can be addressed using designed class weight during training. The best accuracy that can achieve is 94.49% with 2.55% FNR. The lowest FNR that can achieve is 0.51% with an overall accuracy of 92.13%. In addition, a second approach with similarity analysis using triplet loss can address the intra-class greater than inter-class distance is also discussed and experimented with in this study. The best accuracy that can achieve is 94.12% with 3.8% FNR. The lowest FNR can achieve is 2.53% with overall accuracy of 93.13%.

Comparing the first and second approaches, during the experiment, the fluctuations of training accuracy of the first approach are larger than the second approach. Moreover, the data separation of train and test has a great impact on the training result of simple CNN. However, for the second approach, every time the accuracy are pretty much the same even with such small data but the class weights don’t have so much effect on the training process.

To conclude, the second approach can achieve a relatively low FNR rate and high overall accuracy for a small training dataset. The first approach’s result is highly dependent on the data provided, sometimes you can get high overall accuracy and low FNR but may not perform well in real-life applications. But it is easier to modify the network in the first approach to get the lowest FNR.

# 4.2 Recommendation for Future Work

* Data

In our experiment, we only have 317 PRPD patterns for training. In addition, our train and test split is 0.2 to 0.8. Because we want to test the power of the model so the split percentage should be maintained. As a result, in the future, we can provide other types of data and more data for each kind of data to achieve better performance. In addition, implementing this on a simulated real-life application can be a more accurate and powerful measurement of the model’s performance.

* Methods

In our second approach, we actually feed the embeddings into an SVM to classify different kinds of PDs. Maybe using a deep neural network can achieve better results and the specially designed class weight method could be implemented to achieve lower FNR.

# Reflection on Learning Outcome Attainment

*Reflect on your experience during your FYP and the achievements you have relating to at least three of the points below:*

* *Engineering knowledge*
* *Problem Analysis*
* *Investigation*
* *Design/development of Solutions*
* *Modern Tool Usage*
* *The Engineer and Society*
* *Environment and Sustainability*
* *Ethics*
* *Individual and Team Work*
* *Communication*
* *Project Management and Finance*
* *Lifelong Learning*

At the start of the FYP, my supervisor and my tutor had given me a clear idea of how we are going to carry on the project which allowed me to structure and organized my FYP well. During the FYP project, I have learned what is PD and why should we do PD detection and classification. In addition, I had a lot of discussions with my tutor during the FYP period which could always correct and develop my thoughts.

The feature extraction process and clustering master section helped me learn various clustering methods and their implementations which improved my engineering knowledge in machine learning and problem analysis skills. Moreover, all the comparisons trained me on how to investigate the bests approaches among all the options. Last but not least, the triplet loss makes me get in touch with more advanced techniques which inspired me to have more ideas about designing the solution to our current problem.

# References

|  |  |
| --- | --- |
| [1] | *High-voltage test techniques – Partial discharge measurements,* 60270:2000 ed., IEC, 2000. |
| [2] | H. Illias, G. Chen and P. L. Lewin, "Partial Discharge Behavior within a Spherical Cavity in a Solid Dielectric Material as a Function of Frequency and Amplitude of the Applied Voltage," *IEEE Transactions on Dielectrics and Electrical Insulation,* vol. 18, pp. 432-443, 2011. |
| [3] | S. Lu, H. Chai, A. Sahoo and B. T. Phung, "Condition Monitoring Based on Partial Discharge Diagnostics Using Machine Learning Methods: A Comprehensive State-of-the-Art Review," *IEEE Transactions on Dielectrics and Electrical Insulation,* vol. 27, no. 6, December 2020. |
| [4] | I. Kemp, "Partial Dischargc Plant-monitoring Technology: Present and Future Developments," *IEE Proc. Sci. Meas. Tech,* vol. 142, no. 1, pp. 4-10, 1995. |
| [5] | S. Barrios, D. Buldain, M. P. Comech, I. Gilbert and I. Orue, "Partial Discharge Classiﬁcation Using Deep Learning Methods—Survey of Recent Progress," *energies,* 2019. |
| [6] | W. J. K. Raymond, H. A. Illias, A. H. A. Bakar and Hazlie, "Partial discharge classifications: review of," *Measurement,* vol. 68, pp. 164-181, 2015. |
| [7] | N. C. Sahoo, M. M. A. Salama and R. Bartnikas, "Trends in partial discharge pattern classification: a survey," *IEEE Transactions on Dielectrics and,* vol. 12, pp. 248-264, 2005. |
| [8] | A. Contin, G. C. Montanari and C. Ferraro, "PD source recognition by Weibull processing of pulse height distributions," *IEEE Transactions on Dielectrics,* vol. 7, pp. 48-58, 2000. |
| [9] | F. H. Kreuger, E. Gulski and A. Krivda, "Classification of partial discharges," *IEEE Transactions on Electrical Insulation,* vol. 28, pp. 917-931, 1993. |
| [10] | B. Karthikeyan, S. Gopal and S. Venkatesh, "Partial discharge pattern classification using composite versions of probabilistic neural network inference," *Expert Systems with Applications,* vol. 34, pp. 1938-1947, 2008. |
| [11] | H. Illias, T. S. Yuan, A. H. A. Bakar, H. Mokhlis, G. Chen and P. L. Lewin, "Partial Discharge Patterns in High Voltage Insulation," in *IEEE International Conference on Power and Energy (PECon)*, Kota Kinabalu Sabah, 2012. |
| [12] | E. Kuffel, W. S. Zaengl and J. Kuffel, High Voltage Engineering: Fundamentals, 2nd ed., Butterworth-Heinemann, 2000. |
| [13] | X. Ma, C. Zhou and I. J. Kemp, "Automated wavelet selection and thresholding for PD detection," *Electrical Insulation Magazine,* vol. 18, pp. 37-45, 2002. |
| [14] | L. Satish and B. Nazneen, "Wavelet-based denoising of partial discharge signals buried in excessive noise and interference," *IEEE Transactions on Dielectrics and Electrical Insulation,* vol. 10, pp. 354-367, 2003. |
| [15] | W. J. K. Raymond, C. W. Xin, L. W. Kin and H. A. Illias, "Noise invariant partial discharge classification based on convolutional neural network," *measurement,* 2021. |
| [16] | L. Satish and W. S. Zaengl, "Artificial neural networks for recognition of 3-d partial discharge patterns," *IEEE Transactions on Dielectrics and Electrical,* vol. 1, pp. 265-275, 1994. |
| [17] | E. Gulski, "Computer-aided measurement of partial discharges in HV equipment," *IEEE Transactions on Electrical Insulation,* vol. 28, pp. 969-983, 1993. |
| [18] | S. Sriram, S. Nitin, K. M. M. Prabhu and M. J. Bastiaans, "Signal Denoising Techniques for Partial Discharge Measurements," *IEEE Transactions on Dielectrics and Electrical Insulation,* pp. 1182-1191, 2005. |
| [19] | S. Kainaga, A. Pirker and U. Schichler, "Identification of partial discharges at DC voltage using machine learning methods," in *Int. Symp. High Volt. Eng. (ISH)*, 2017. |
| [20] | S. Polisetty, A. El-Hag and S. Jayram, "Classification of common discharges in outdoor insulation using acoustic signals and artificial neural network," *High Volt,* vol. 4, no. 4, pp. 333-338, 2019. |
| [21] | V. Kecman, "Support Vector Machines: Theory and Applications," *Springer Science & Business Media,* 2005. |
| [22] | R. Y. e. al, "A New Discharge Pattern for the Characterization and Identification of Insulation Defects in GIS," *Energies,* vol. 11, no. 4, p. 971, 2018. |
| [23] | M.-T. Nguyen, V.-H. Nguyen, S.-J. Yun and Y.-H. Kim, "Recurrent Neural Network for Partial Discharge Diagnosis in Gas-Insulated Switchgear.," *Energies,* vol. 11, no. 5, 2018. |
| [24] | S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng and P. Kennedy, "Training deep neural networks on imbalanced data," in *International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC,Canada, 2016. |
| [25] | Y. Yan, M. Chen, M. Shyu and S. Chen, "Deep Learning for Imbalanced Multimedia Data Classiﬁcation," in *IEEE International Symposium on Multimedia (ISM)*, Miami, FL, USA, 2015. |
| [26] | Y. LeCun, K. Kavukcuoglu and C. Farabet, "Convolutional networks and applications in vision," in *IEEE International Symposium on Circuits and Systems*, Paris, France, 2010. |
| [27] | Y. Lu, R. Wei, J. Chen and J. Yuan, "Convolutional Neural Network Based Transient Earth Voltage Detection," in *International Symposium on Parallel and Distributed Computing (ISPDC)*, Fuzhou, China, 2016. |
| [28] | D. Dey, B. Chatterjee, S. Dalai, S. Munshi and S. Chakravorti, "A deep learning framework using convolution neural network for classification of impulse fault patterns in transformers with increased accuracy," *IEEE Transactions on Dielectrics and Electrical Insulation,* vol. 24, no. 6, p. 3894–3897, 2017. |
| [29] | H. Song, J. Dai, G. Sheng and X. Jiang, "GIS partial discharge pattern recognition via deep convolutional neural network under complex data source," *IEEE Trans. Dielectr. Electr. Insul.,* vol. 25, no. 2, pp. 678-685, 2018. |
| [30] | Y. Khan, "Partial discharge pattern analysis using PCA and back propagation artificial neural network for the estimation of size and position of metallic particle adhering to spacer in GIS," *Electr. Eng.,* vol. 98, no. 1, pp. 29-42, 2016. |
| [31] | Y. Taigman, M. Yang, M. Ranzato and L. Wolf., "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014. |
| [32] | F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015. |
| [33] | S. Salman and X. Liu, "Overfitting Mechanism and Avoidance in Deep Neural Networks," January 2019. [Online]. Available: arXiv:1901.06566. |
| [34] | M. Xiao, Y. Wu, G. Zuo, S. Fan, H. Yu, Z. A. Shaikh and Z. Wen, "Addressing Overfitting Problem in Deep Learning-Based Solutions for Next Generation Data-Driven Networks," *Wireless Communications and Mobile Computing,* 2021. |
| [35] | C. Gao, L. Yu, Y. Xu, W. Wang, S. Wang and P. Wang, "Partial Discharge Localization inside Transformer Windings via Fiber-Optic Acoustic Sensor Array," *Gao, Chaofei; Yu, Lei; Xu, Yue; Wang, Wei; Wang, Shijie; Wang, Peng (2018). Partial Discharge LocaliIEEE Transactions on Power Delivery,* 2018. |
| [36] | H. Chai, B. T. Phung and S. Mitchell, "Application of UHF sensors in power system equipment for partial discharge detection: A review," *Sensors,* vol. 19, no. 5, p. 1029, 2019. |
| [37] | H. Chai, S. Lu, B. T. Phung and S. Mitchell, "Comparative Study of Partial Discharge Localization based on UHF Detection Methods," in *Int. Conf. Exhib. Electr. Distrib. (CIRED)*, 2019. |
| [38] | S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory.," *Neural Comput.,* vol. 9, no. 8, pp. 1735-1780, 1997. |