Master's Thesis

Evaluation of Classification Algorithms for Partial Discharge Diagnosis in Gas-Insulated HVDC Systems

Joris Clement
Email (no longer exists): j.clement@campus.tu-berlin.de
Course of Study: Computer Science
Faculty IV - Electrical Engineering and Computer Science
Technische Universität Berlin

1st Supervisor: Prof. Dr.-Ing. Ronald Plath
1st Daily Supervisor: Moritz Geske, M.Sc.

2nd Supervisor: Prof. Dr. Dr. h.c. Sahin Albayrak
2nd Daily Supervisor: Marcus Voß, M.Sc.

Submitted on 2021/08/17
Rebuilt with minor changes on February 17, 2023

Declaration of Originality

I hereby declare that the thesis submitted is my own, unaided work, completed without any unpermitted external help. Only the sources and resources listed were used.

The independent and unaided completion of the thesis is affirmed by affidavit:

Berlin, (February 17, 2023)

Joris Clement

Abstract

To reduce the impact of global warming the electricity grid is undergoing drastic changes. Due to the superior transmission efficiency of direct current (DC) over alternating current (AC) voltage, the usage of high voltage direct current (HVDC) is increasing in the grid. The main driver of this change is the integration of regenerative energy sources. Since gas-insulated systems are space-saving, they are e.g. being used as gas-insulated substation (GIS) on offshore converter platforms to distribute wind energy or have potential to be used as underground HVDC transmission lines (GIL) in urban areas. With the increase in HVDC transmission, the use of gas-insulated DC systems is also becoming more important. Failures in such a system can lead to its fallout and might impact the stability of the power grid. One method to prevent failures are partial discharge (PD) measurements, which detect potential defects in the system before they cause problems. PD measurements are a sufficient diagnostic method for AC equipment. However, the diagnostic is much more difficult for DC due to the lack of AC phase correlation. To ensure the reliability of gas-insulated systems other methods than what is used for AC must be found for the analysis of direct current PD measurements.

The purpose of this master's thesis is to evaluate different machine learning techniques for the diagnosis of partial discharge defects in gas-insulated DC systems. Here, the aspect of correctly classifying different PD defect types is investigated. Apart from comparing multiple feature-based classifiers other time series classification algorithms such as Dynamic Time Warping are evaluated. To analyze the generalizibility of the approaches the algorithms are verified on different test sets. Different groups of machine learning approaches are compared: A group of feature-based models and two groups of non-feature-based time series classifiers.

It was shown that the feature-based approaches coupled with modern classification algorithms can achieve good results for the analyzed PD measurements. When comparing the classification algorithms that were utilized in the feature-based models with each other, it was observed that the more complex ones can better utilize the extracted features. Among all models, two feature-based models utilized absolute and the others normalized PD amplitudes. It was found that the models with normalized values generalized better to variations which were not present in the training data. Additionally, one Dynamic Time Warping model achieved results, which were only slightly worse than the results of the feature-based models. This means that other, more-sophisticated time series classification algorithms might be worth exploring in future work. These models have the advantage that they do not rely on feature engineering as the feature-based models do. Furthermore, Deep Learning models might also be worth to explore in future studies, if enough data is available.

Zusammenfassung

Um die Auswirkungen des Klimawandels zu reduzieren, erfährt das Stromnetz drastische Veränderungen. Aufgrund der überlegenen Übertragungseffizienz von Gleichspannung gegenüber Wechselspannung steigt der Einsatz von Hochspannungs-Gleichstrom-Übertragung (HGÜ) im Netz. Der Haupttreiber dieser Veränderung ist die Integration erneuerbarer Energiequellen. Da gasisolierte Systeme platzsparend sind, werden sie z.B. als gasisolierte Schaltanlagen auf Offshore-Konverterplatformen für Windenergie eingesetzt. Mit der Zunahme der HGÜ-Übertragung wird auch der Einsatz von gasisolierten Gleichspannungssystemen wichtiger. Fehler in solchen Systemen können zum Ausfall der betreffenden Komponente führen und die Stabilität des Stromnetzes beeinträchtigen. Eine Methode zur Vermeidung von Ausfällen sind Teilentladungsmessungen, die potenzielle Defekte im System erkennen, bevor sie Probleme verursachen. Teilentladungsmessungen sind eine ausreichende Diagnosemethode für Wechselstromgeräte. Allerdings ist die Diagnose bei Gleichspannung aufgrund des fehlenden Phasenbezugs wesentlich schwieriger. Um die Zuverlässigkeit von gasisolierten Anlagen zu gewährleisten, müssen andere Methoden zur Analyse von Teilentladungsmessungen bei Gleichspannung gefunden werden, als sie für Wechselspannung verwendet werden.

Das Ziel dieser Masterarbeit ist es verschiedene Verfahren des maschinellen Lernens für die Diagnose von Teilentladungsfehlern in gasisolierten Gleichstromsystemen zu evaluieren. Hier wird der Aspekt der korrekten Klassifizierung verschiedener Teilentladungsdefekttypen untersucht. Neben dem Vergleich mehrerer merkmalsbasierter Klassifikatoren werden auch andere Zeitreihenklassifikationalgorithmen, wie z.B. Dynamic Time Warping, evaluiert. Um die Generalisierbarkeit der Ansätze zu analysieren, werden die Algorithmen auf verschiedenen Testdatensätzen verifiziert. Es wurden verschiedene Gruppen verglichen: Eine Gruppe von merkmalsbasierten Modellen und zwei Gruppen von Zeitreihenklassifikationsalgorithmen.

Die merkmalsbasierten Ansätze konnten in Verbindung mit modernen Klassifikationsalgorithmen gute Ergebnisse erzielen. Innerhalb der merkmalsbasierten Modelle wurde beobachtet, dass die komplexeren Klassifikationsalgorithmen überlegen sind. Unter allen Modellen nutzten zwei merkmalsbasierte Modelle absolute und die Anderen normalisierte Teilentladungsamplituden. Es wurde festgestellt, dass die Modelle mit normalisierten Werten besser generalisierten. Desweiteren war ein Dynamic Time Warping Modell nur wenig schlechter als die merkmalsbasierten Modelle. Es kann daher lohnenswert sein andere Zeitreihenklassifikatoren in zukünftigen Arbeiten zu erforschen. Diese Modelle haben nämlich den Vorteil, dass sie nicht auf die Entwicklung von Merkmalen angewiesen sind. Darüber hinaus könnten Deep-Learning-Modelle in zukünftigen Studien eine Untersuchung wert sein.

Contents

List of Abbreviations **iv**

List of Figures **vi**

List of Tables **viii**

1 Introduction **1**

1.1 Motivation 1

1.2 Related Work 2

1.3 Contribution 3

2 Fundamentals **5**

2.1 Gas-Insulated DC Systems 5

2.1.1 Partial Discharges 7

2.1.2 Defect Types in Gas-Insulated DC Systems 7

2.1.3 Partial Discharge Measurement Techniques 11

2.1.4 Noise 12

2.1.5 PD Analysis in Gas-Insulated DC Systems 13

2.2 Machine Learning 15

2.2.1 Classifiers 15

2.2.2 Time Series Classification 20

2.2.3 Cross-Validation 23

3 Methods **25**

3.1 Datasets 26

3.2 Data Preprocessing 27

3.2.1 Data Cleaning 27

3.2.2 Further Preprocessing Steps 27

3.3 Data Splitting 28

3.3.1 Training and Test 28

3.3.2 Cross-Validation on Training Set 28

3.4 Machine Learning Models and Training 29

3.4.1 Feature-Based Models 29

3.4.2 1D-Resample Models . . . 33
3.4.3 Raw Data Models . . . 35
3.4.4 Measurement File Prediction . . . 36
3.4.5 Noise Data Classification . . . 36
3.5 Evaluation . . . 37
3.5.1 Model Selection & Performance . . . 37
3.5.2 Prediction Time Comparison . . . 37
3.5.3 Feature Attribution Analysis . . . 38

4 Results 39
4.1 Data Properties . . . 39
4.2 Hyperparameters and Validation Scores . . . 40
4.2.1 Feature-Based Models . . . 40
4.2.2 1D-Resample Models . . . 42
4.2.3 Raw Data Models . . . 42
4.3 Test Scores . . . 45
4.4 Noise Data Classification . . . 45
4.5 Prediction Time Comparison . . . 48
4.6 Feature Attribution Analysis . . . 48

5 Discussion 51
5.1 Generalization to Other Settings . . . 51
5.2 Feature-Based versus Other Classifiers . . . 52
5.3 Comparison of Feature Sets . . . 53
5.4 Contribution of Certain Features . . . 54
5.5 Sliding Window . . . 54
5.6 Other Measuring Techniques . . . 55
5.7 Limitations . . . 55

6 Conclusion 57

Appendices 59

A Methods 60
A.1 Graphical User Interface . . . 60
A.2 Weibull Distribution . . . 60
A.3 Features of Reimplemented Sets . . . 61
A.4 Feature Tuning . . . 62

B Results 63
B.1 Tuned Hyperparameters . . . 63
B.2 Validation Scores . . . 64
B.3 Test Scores . . . 65

B.4 Noise Data Classification . . . 68
B.5 Feature Attribution Analysis . . . 69

Bibliography **70**

List of Abbreviations

AC	alternating current
ANN	Artificial Neural Network
BOSS	Bag-of-SFA-Symbols
CNN	Convolutional Neural Network
CSV	Comma Seperated Values
COTE	Collective of Transformation Ensembles
DC	direct current
DT	Decision Tree
DTW	Dynamic Time Warping
EMC	electromagnetic compatibility
FDR	False Discovery Rate
FN	False Negatives
FP	False Positives
GAN	Generative Adversarial Network
GIL	gas-insulated transmission line
GIS	gas-insulated substation
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HIVE-COTE	Hierarchical Vote Collective of Transformation-based Ensembles
HVAC	high voltage alternating current
HVDC	high voltage direct current
HV	high voltage
IEC	International Electrotechnical Commission
k-NN	k-Nearest Neighbor
LGBM	Light Gradient Boosting Machine
LR	Logistic Regression
LSTM	Long-Short Term Memory
ML	machine learning

MLP	Multi-Layer Perceptron
−DC	negative DC voltage
NoDi*	Normalized Differenced
OHL	overhead line
OvO	one-vs-one
OvR	one-vs-rest
PD	partial discharge
PDIV	partial discharge inception voltage
+DC	positive DC voltage
PRPD	phase-resolved partial discharge
PSA	pulse sequence analysis
RAM	random-access memory
RBF	Radial Basis Function
RF	Random Forest
RNN	Recurrent Neural Network
Rocket	RandOm Convolutional KErnel Transform
RUS	Random Undersampling
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine
TP	True Positives
TSC	time series classification
UCR/UEA	University of California, Riverside/University of East Anglia
UHF	ultra high frequency
WEASEL	Word ExtrAction for time SEries cLassification
BAC	balanced accuracy
ACC	accuracy

List of Figures

2.1 Applications of DC-GIS connecting different transmission types 5
2.2 Structure of a GIL 6
2.3 Classes of insulation defects in GIS/GIL 8
2.4 Typical movement patterns of free moving particles in DC gas-insulated systems 9
2.5 Illustration of the parameters of PD measurements under DC voltage . . . 11
2.6 UHF PD disk sensor 12
2.7 PSA pattern of a protrusion 14
2.8 k-NN with two classes and different k 16
2.9 SVM with two classes and a maximum-margin hyperplane 17
2.10 Illustration of an Artificial Neural Network 18
2.11 Illustration of a RF consisting of multiple DTs 19
2.12 Illustration of Dynamic Time Warping (DTW) 22
2.13 Illustration of stratified k-fold cross-validation 23

3.1 Workflow for constructing ML models in this thesis 25
3.2 Overview of implemented ML models 29
3.3 Illustration of the sliding window 30
3.4 Set diagram of feature sets 31

4.1 Overview of all measurement files in relation to their PD count and duration 39
4.2 PD activity of an example measurement for each of the six defect classes . 40
4.3 Occurrence of the polarity in relation to the defect types 41
4.4 Boxplot on number of PDs per defect class when using sliding window . . 41
4.5 Validation Scores of the tuned models for each feature set 43
4.6 Recall and precision of the LGBM-SFip model on the four test sets 46
4.7 Recall and precision of the MLP-RelFip model on the four test sets 47
4.8 Recall and precision of the Raw-k-NN-DTW model on the four test sets . 47
4.9 Recall and precision on the test sets of the model trained also with the noise data 48
4.10 SHAP feature attribution of top 10 features of the LGBM-RelFip model . 49

A.1 Screenshot of GUI 60
A.2 Visualization of 3 features in a pair plot 62
A.3 Correlation matrix to exclude redundant features 62

B.1 Recall and precision of the LGBM-AbsFip model on the four test sets . . 65
B.2 Recall and precision for the LGBM-OFip model on the four test sets . . . 66
B.3 Recall and precision for the LGBM-Tsfresh model on the four test sets . . 66
B.4 Recall and precision of the 1D-k-NN-DTW model on the four test sets . . 67
B.5 Confusion matrix on the test sets of the model trained also with the noise data 68
B.6 Feature attribution of all features of the LGBM-RelFip model 69

List of Tables

3.1 Description of datasets 26
3.2 Overview of dataset splitting into training and test 28
3.3 Features used for the feature set RelFip 32
3.4 Additional features not in RelFip, but considered for the feature set AbsFip 32
3.5 Hyperparameters of all feature-based classifiers tuned via grid search . . . 34
3.6 Hyperparameters options of the 1D-Resample k-NN models tuned via grid search 35
3.7 Hyperparameters of the Raw data classifiers tuned via grid search 36

4.1 Tuned hyperparameters of 1D-Resample models 44
4.2 Validation scores of the tuned 1D-Resample models 44
4.3 Tuned hyperparameters of Raw Data models 44
4.4 Validation scores of the tuned Raw Data Models 45
4.5 BAC and ACC of best models on the test sets 46
4.6 Average prediction times in seconds of best models on the Mixture test set 49

A.1 Description of the features used by the feature sets OFip and SFip 61

B.1 Optimal hyperparameters of feature-based classifiers 63
B.2 Validation Scores of the tuned models for each feature set 64
B.3 Support of models for the validation score 64
B.4 Support of best models on the test sets 65
B.5 BAC and ACC on the test sets of the model trained also with the noise data 68

1 Introduction

1.1 Motivation

Greenhouse gas emissions caused by the generation of electricity need to decrease drastically in order to reduce the effects of global warming [1, p. 51]. To accomplish this, regenerative energy sources like wind and solar are added to the electricity distribution grid. However, the decentralized generation of power from these sources makes the grid more complex [2]. This leads to the necessity of long distance energy transport. In the case of Germany, additional infrastructure is necessary to on-demand transmit wind energy from the north of the country to the south, or vice versa in case of solar energy [3]. To minimize the loss of energy, the distribution over such long distances is done with high voltages of 380 kV or more [4]. Although most of the grid is operated with alternating current (AC) voltage, direct current (DC) has advantages in certain situations. High voltage direct current (HVDC) exhibits, in contrast to high voltage alternating current (HVAC), a lower loss of energy for long distances [5], can stabilize the grid against cascading disturbances [2], and allows connecting asynchronous AC systems [6].

High voltage transmission lines are mainly realized with overhead lines or underground cables insulated with materials such as oil or polyethylene [7; 6, p. 7]. An alternative are gas-insulated systems, consisting of gas-insulated transmission line (GIL) [8, p. 41] and gas-insulated substation (GIS) [9].

To ensure that gas-insulated systems have a high level of quality, are reliable, and do not suffer from unexpected failures such as their complete fallout, partial discharge (PD) measurements are necessary to detect insulation defects. One of the best techniques for such measurements is the ultra high frequency (UHF) method [10]. Information gained from these investigations are used to detect underlying defects such as protrusion, floating electrode, or void. Depending on the defect, early actions can be taken to prevent failures. The detection of those defects is well established for AC voltage with methods such as phase-resolved partial discharge (PRPD) patterns [10; 11]. Multiple studies also utilize machine learning (ML) algorithms for the PD analysis, such as [12] by using an Recurrent Neural Network (RNN) and [13] by evaluating multiple Artificial Neural Networks (ANNs). However, these methods can not be simply adopted to DC voltage because the AC phase is missing there [10; 11]. Since there is not enough experience with the analysis of PD measurements conducted with DC voltage yet, recent recommendations are to use AC measuring also for gas-insulated DC systems [14, Sec. 6.2.2; 15, Sec. 5.1]. But relying solely on AC measurements might not be sufficient for DC systems. Hence, utilizing the PD analysis also with DC measuring might improve the identification of possible PD defects. Current methods of DC PD analysis include the evaluation of pulse sequence analysis (PSA) patterns to visually discriminate the defect types from each other [10; 16]. Techniques from the field of machine learning and Pattern Recognition might be helpful here to automate the analysis of the measurements and improve the accuracy. Currently, only a few studies such as [11] use ML algorithms and show promising results.

1.2 Related Work

This work is based on ongoing research by Moritz Geske. The measurements which were used in the present work were performed by him. One of the aims of his research was to analyze the application of the PSA method to a real-sized DC-GIL test arrangement. It was shown that defect types such as the moving particle could be classified by visual discrimination with an enhanced PSA method [10]. The present thesis extended the analysis of the data by applying ML methods to detect the different defect types.

There are several related studies and theses that have been conducted in this domain. One of them is the master thesis by Ott [17] that was performed like the present thesis in the Department of High Voltage Engineering at the Technische Universität Berlin. Ott used a subset of the data, which was also available for this work. Multiple statistical parameters which are important for the analysis of PD activity were examined. Out of 36 statistical parameters (features) 12 were identified as relevant. This feature set, also called fingerprint, was utilized to differentiate the same defect classes from each other as in this work without the void defect [17, Sec. 6.5]. The fingerprint was used to evaluate three approaches [17, Chap. 7]. The first approach consisted of a visual discrimination by using a graphical representation of the feature set. In the second approach the means of the normalized parameters were calculated for every defect type. An unknown measurement was assigned to the defect type with the mean vector having the smallest distance to the measurement. For the last approach a modified version of the recognition rate method by [18] was applied, which did not yield results as good as the other two approaches. Of Ott's three approaches, the second is the most similar to the classification algorithms implemented in the present work. It was therefore reimplemented as a baseline and compared to other, more complex ML based algorithms.

Schober, Schichler, and Pirker et al., affiliated with the Institute of High Voltage Engineering and System Performance at TU Graz, contributed so called Normalized Differenced (NoDi*) pattern and ML techniques to the problem of defect classification in HVDC GIS/GIL [11; 19–21]. For the automatic classification with machine learning four different algorithms, a Decision Tree (DT), Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), and an ANN, were used. The best results were generated with an SVM [20]. The input data for all ML algorithms was a feature set derived from statistical parameters of PD measurements. The set consisted of 14 parameters such as the minimal, maximal, and mean time between adjacent partial discharges [20]. For comparison and replication this feature set was reimplemented here. They used these statistical features because they claimed that the raw data can not be used for classification as it would lead to insufficient results.

Their work differed from this work in four aspects. Firstly, the data in their work relied on the conventional measuring method as the apparent charge q was used [11; 20; 21]. The UHF technique was utilized as well, but these were according to [11] not used for the ML algorithms. Contrary to that this work utilized UHF measurements for the ML models. Secondly, the classification differentiated between other defect types (protrusion on HV, floating electrode, bouncing particle, particle in firefly, and noise data). In contrast to that, the present work discriminated additionally the defect types protrusion on enclosure, particle on insulator, and void in insulator, but merged bouncing particle and particle in firefly into one class. Thirdly, in the present work raw data was not rejected, if no significant patterns, e.g. NoDi* or PSA, could be detected. Fourthly, the data was split in this thesis into training and test and cross-validation was performed on the training set. Thus, cross-validation was used for tuning (model selection) and the test set to assess an unbiased score (model performance). Contrary to the present study, Schober and Schichler utilized cross-validation for both tuning and estimating the model performance [19–21]. Thus, an independent dataset to verify the validation score was missing.

Most of further research on the classification of defects in gas-insulated systems which makes use of ML methods focused on AC rather than DC voltage [22]. An example for this is [12] utilizing an Long-Short Term Memory (LSTM)-RNN. Further related research on the broader topic of PD diagnosis, also for other systems than gas-insulated systems, can be found in a survey by Lu et al. [22].

1.3 Contribution

Due to the increasing importance of HVDC transmissions, and the need to ensure its reliable operation, the purpose of this master's thesis is to evaluate different machine learning techniques for the analysis of partial discharge defects in gas-insulated DC systems. One aspect in this analysis is the correct classification of different PD defect types. Only a few studies have been conducted so far on this topic [22]. Apart from comparing multiple feature-based algorithms other time series classification algorithms such as Dynamic Time Warping are evaluated. The algorithms are verified on different test sets. As mentioned by [22, Sec. 5] most papers just divide their whole dataset into one training and one test set. In this study part of the overall test set consists of measurements which have different characteristics than the measurements in the training set.

The rest of this document is structured as follows. The next chapter describes the fundamentals on gas-insulated DC systems and ML methods which can be applied to the PD analysis in these systems. Chapter 3 presents the datasets and the methods used for training, tuning and testing the ML models. The results are shown in Chapter 4 and discussed in Chapter 5. This thesis ends with a summary and conclusion (Chapter 6).

2 Fundamentals

2.1 Gas-Insulated DC Systems

Both, gas-insulated transmission line (GIL) and gas-insulated substation (GIS) are gas-insulated systems. Just like other transmission lines, the function of a GIL is to transmit electricity from one location to another [8]. One function of a GIS is to connect transmission lines, possibly of different type (see Figure 2.1). In contrast to other transmission systems, gas-insulated systems use (special) gases in an enclosure to achieve electrical insulation [8; 9]. To increase the insulation capabilities, the utilized gas is usually put under high pressure. This allows the usage of high voltage (HV) and compact constructions. Due to the compact constructions DC-GIS need up to 95% less space than air-insulated substations [23]. Additional advantages of GIS are a long operating lifetime, high availability, and high reliability [8; 9]. The other application type of gas-insulated systems are GIL. Figure 2.2 shows the cross section of such a transmission line. A GIL is a coaxial pipe system consisting of an inner conductor pipe and an enclosure pipe [8, p. 93]. The HV conductor is secured by post insulators made of cast resin [8, Sec. 3.3.5]. The insulation gas fills the space between the conductor and the surrounding enclosure. Besides the advantages mentioned for GIS, GIL have low transmission losses compared to other transmission systems such as overhead lines [8, Sec. 4.3]. They furthermore can also be constructed underground [8] and cause less environmental problems than overhead lines [7; 8]. A disadvantage are the high installation costs, which however

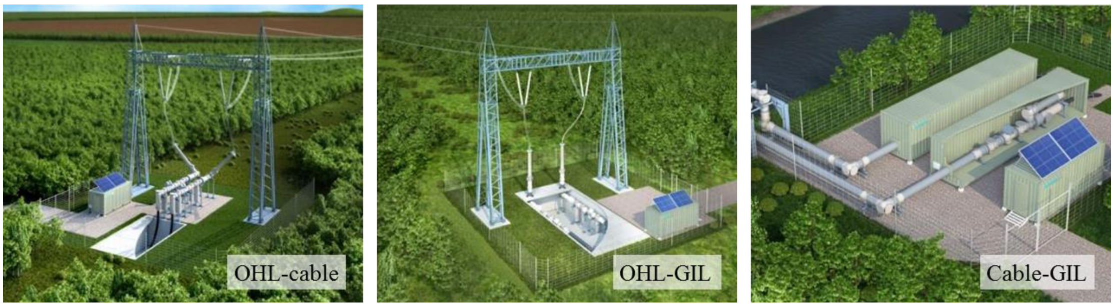


Figure 2.1 **Applications of DC-GIS connecting different transmission types** [23, Fig. 6]. Left: A GIS connects an overhead line (OHL) and an underground cable. Center: A GIS connects an OHL and an underground GIL. Right: A GIS connects an underground cable and an above-ground GIL.

might be worth the investment because of potential lower life cycle costs [8]. Due to the high installation costs GIL currently only realized under specific circumstances [8, p. 20].

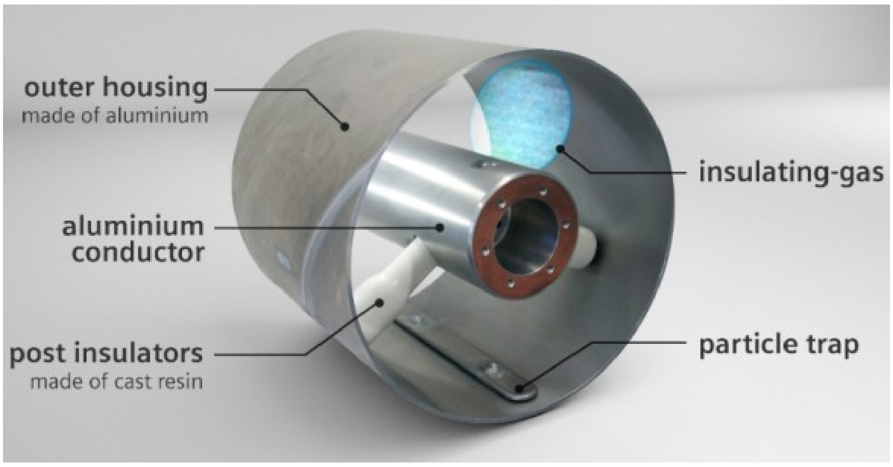


Figure 2.2 **Structure of a GIL** [24].

Different gases can be used in gas-insulated systems. Especially in GIS the most commonly used gas is SF_6 (Sulfurhexafluoride), which main advantage is its high insulation capabilities. The biggest disadvantages are a comparably high price and its strong global warming potential [8, Sec. 3.1.4; 9, Sec. 2.8]. Therefore, research is done to replace or reduce SF_6. A common alternative is to add N_2 (Nitrogen) to SF_6, which still offers similar insulation capabilities [8, p. 49]. Another alternative that does not contain SF_6 is CleanAir [25, Sec. 5.1.3], also known as technical air. It is a mixture of 80% N_2 and 20% O_2. The disadvantage of CleanAir is its reduced dielectric strength, which require larger constructions dimensions and higher gas pressures to achieve similar insulation capabilities. Therefore CleanAir is not being used widely yet [8, p. 47]. Other alternatives to SF_6 are gas mixtures with synthetic gases such as Fluoronitrile Novec™ 4710 [26] or C4 (C_4F_7N) [27].

Currently most of the grid is operated with AC voltage. Therefore, gas-insulated systems are mainly in use with high voltage AC (HVAC). Since HVDC is more efficient than HVAC over long distances, it is gaining importance [6, Sec. 1.1]. This is specially due to the rise of renewable energy sources, which increase the need for longer electricity transmission. Therefore also GIL and GIS operated with HVDC are getting more significant. To connect offshore wind power to the grid, DC-GIS are currently in use on offshore converter platforms because they are compact and space is expensive on these platforms [3; 10; 23; 28]. Furthermore, DC-GIS can be used to connect different transmission lines as shown in Figure 2.1. DC-GIL are currently only in operation in prototype installations [3; 29–31]. It has not seen industrial application yet as certain requirements are not met due to e.g. the risk posed by a moving particle defect (see

Section 2.1.2) [31].

To ensure the safe and reliable operation of these systems, diagnostic methods are needed to assure that no defects such as moving particles cause a breakdown of the system. One appropriate method for HVAC are PD measurements. In contrast to AC voltage, there is not yet a reliable diagnostic procedure for DC voltage [10; 11].

2.1.1 Partial Discharges

For gas-insulated systems, an electrical discharge is the release of electric charge in an electric field through the insulation gas. Electrical discharges can be divided into flashovers and partial discharges (PDs). A flashover is the electric breakdown of a system, which happens when the applied voltage exceeds the insulation capabilities of the system. In contrast to that, a PD is a spatially bounded discharge, which only partially bridges the insulation between the conductor and the enclosure [25, Sec. 3.6]. They are brought up by locally intensified electric fields or regions with lower insulation capabilities. These are in turn caused by defects such as moving particles or protrusions. PDs do usually not affect the short-term insulation properties [25, Sec. 3.6]. However, the underlying defect and the resulting partial discharges weaken the insulation of the system and may cause its breakdown. A breakdown can then happen in the case of a voltage spike due to e.g. a ligtning strike [25, Sec. 2.2]. Therefore it is important to analyze their occurrence. Important characteristics of PDs are among others their magnitude, repetition rate, and the partial discharge inception voltage (PDIV). The PDIV is the voltage upon which partial discharges start to occur continuously [25]. For DC voltage, one common definition of this continuity is a continued PD rate of at least 3 PDs per 30 seconds [10; 11; 32].

2.1.2 Defect Types in Gas-Insulated DC Systems

As described in the previous section defects and their PDs reduce the insulation capabilities of the gas-insulated system, which may cause a breakdown. In other words, they lower the system's breakdown voltage. Defects themselves can be caused by friction or vibration during the operation, or by insufficient quality during the manufacturing or assembling of the components [10].

There are different types of defects in gas-insulated systems [10; 11; 9, p. 210]. As shown in Figure 2.3 these are: Moving particle (Particle (moving) on enclosure), particle on insulator, protrusion on enclosure/ground and HV, floating, delamination, and void (cavity in insulating material). Except for the moving particles all defect classes are stationary. Defects can happen regardless whether AC or DC voltage is applied on the gas-insulated system. The important difference is though that the PD activity differs between AC and DC voltage due to a different field distribution (AC: capacitive, DC: resistive). It consists of fewer PD events and their occurrence is more irregular for DC voltage [25].

Another influential factor on partial discharges is the voltage polarity on the two electrodes, the HV conductor and the enclosure. The polarity indicates which electrode has the higher and which the lower electric potential. Two values are used for that, namely positive (+) for the higher and negative (-) for the lower potential. In contrast to AC voltage where the polarity alternates, it is constant for DC. Due to that, the voltage polarity can have a major influence on the PD activity of a defect for DC voltage [25].

In the following, all relevant defect classes examined in this work are described. Therefore, the defect's key properties such as their behavior or the effect of the polarity are mentioned.

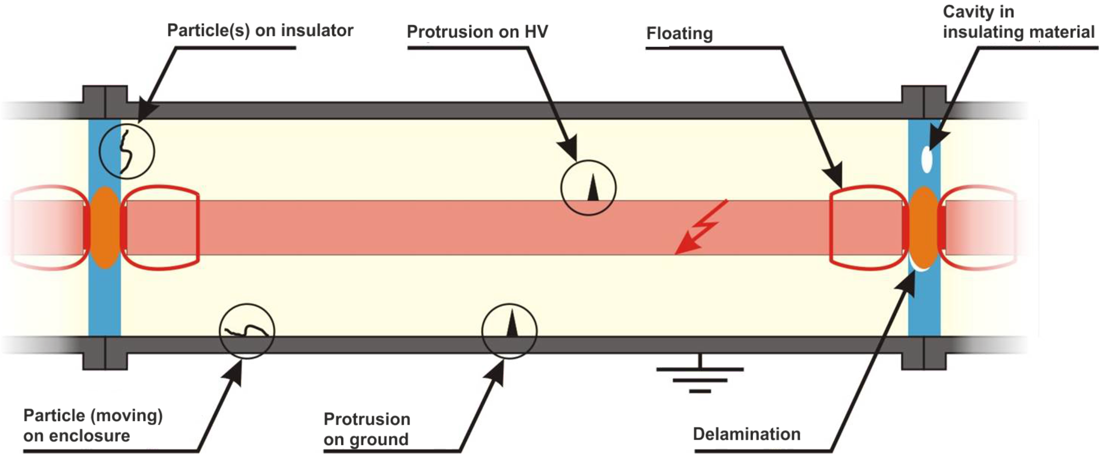
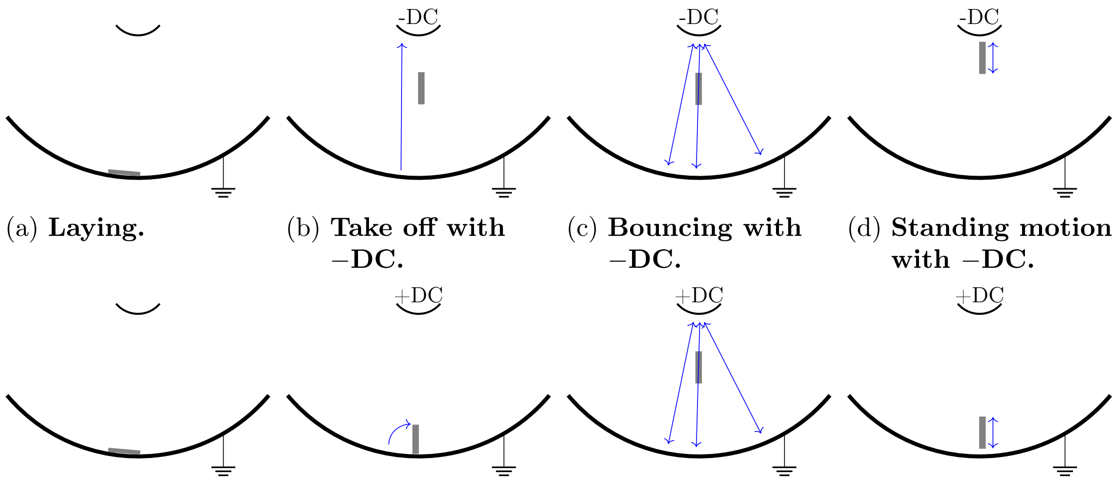


Figure 2.3 **Classes of insulation defects in GIS/GIL**[33, Slightly adapted].

Moving Particle A (free) moving particle is a small metallic particle, which can move inside the gas compartment. It usually lies on the bottom part of the enclosure. When such a particle is then affected by an electric field, it charges up. If the electrical charge of the particle is larger than the gravitational force, the particle starts to raise from the enclosure [10]. Depending on the form and the size of the moving particle different effects can happen.

Particle movements differ between AC and DC voltage [36]. For DC, two typical movement patterns are bouncing and standing motion [10; 36], see Figure 2.4. A bouncing particle bounces between the HV conductor and the enclosure. The bouncing usually starts on the enclosure and happens because the particle absorbs charge of the same polarity as the enclosure and is thereby repelled. With increasing voltage the bouncing height increases until the particle reaches the HV conductor. As it absorbs there charge of the other polarity it is repelled back to the enclosure restarting the process [10; 17]. A particle in standing motion behaves differently (see Figures 2.4c and 2.4g compared to 2.4d and 2.4h). The movement happens at the negative electrode, i.e. it happens at the HV conductor if negative DC voltage is applied and at the enclosure if positive voltage is applied. Similar to the bouncing particle the particle absorbs charge from the



(a) **Laying.**

(b) **Take off with −DC.**

(c) **Bouncing with −DC.**

(d) **Standing motion with −DC.**

(e) **Laying.**

(f) **Stand up with +DC.**

(g) **Bouncing with +DC.**

(h) **Standing motion with +DC.**

Figure 2.4 **Typical movement patterns of free moving particles in DC gas-insulated systems**. Figures 2.4a to 2.4d show the movement with −DC and Figures 2.4e to 2.4h with +DC. Based on [34, p. 30f; 35, Fig. 6.7].

negative electrode. In contrast to the bouncing motion, the standing motion happens when partial discharges occur at a higher rate. The particle gets ongoing charges from the negative electrode and emits these to the positive electrode [10]. That is why it does not move towards the counter electrode. Hence, the free moving particle performs a standing motion [10; 36]. It may happen that the particle in standing motion keeps on quickly recharging causing the region to illuminate, the so called firefly effect [11; 16; 36].

The moving particle defect is mainly caused by metallic particles coming into existence during the construction of the system. Such metallic particles may also arise due to erosion in the system over time during operation. The moving particle defect is seen as the most critical compared to other defect classes [17]. Therefore the foremost goal is to avoid that particles get into the gas-insulated system. As this can not be ensured completely, one established technique is coating. For this, a dielectric coating material is applied onto the electrodes (the HV conductor and the inside of the enclosure) [37]. This will usually prohibit the occurrence of a bouncing particle. Another established method is to apply particle traps on the enclosure, see Figure 2.2 for an example. As the electric field in such a particle trap is very low, a particle in that trap is no longer dangerous [8, p. 79]. Whereas for AC voltage particle traps are sufficient on the enclosure, this does not hold true for DC voltage as particles may also be located on the HV conductor. To handle that phenomena, particle traps should also be attached there [36].

Particle on Insulator A particle on insulator is a particle, which is attached to a post insulator and hence does not move. A moving particle may result in a particle on insulator, when it comes in contact with the surface of the insulator [38]. It causes irregularities in the electric field, which might easier cause a breakdown, because the insulation capabilities are lower around the post insulators than in other sections of the gas-insulated system. Investigations with SF_6 as the insulation gas showed that especially at high pressures partial discharges are almost undetectable and initial PDs may lead to a flashover [28]. Additionally, a longtime presence of the particle may cause erosion of the post insulator and therefore lower the insulation capabilities over time [28].

Protrusions A protrusion in a gas-insulated system is a sharp edge or scratch in the metallic surface [38]. It may also be a fixed particle. Furthermore, it is either located on the enclosure or the HV conductor [38]. The protrusion causes local increases in the electric field, which lead to PDs. The effects of a protrusion are influenced by its shape, size, the pressure of the insulation gas, and the polarity. The shapes are categorized into more rounded and sharp protrusions [38]. In general, a bigger protrusion is seen as more dangerous. Apart from the protrusion itself, the voltage polarity has influence on its behavior: A protrusion with negative polarity has a lower PDIV and a higher breakdown voltage than the protrusion with positive polarity. This is the so called polarity effect [10; 25, p. 190]. Furthermore, Juhre et al. found that when CleanAir is used instead of SF_6 as the insulation gas the PD inception voltage is reduced by around half [32].

Floating Electrode The floating electrode defect class is caused by improperly connected or untightened parts in the gas-insulated system. During the operation due to the electrical stress these parts might start to vibrate and can partially detach from the HV conductor or the enclosure and become a floating electrode [38].

This defect class is characterized by a constant repetition of many partial discharges with low amplitudes and a few discharges with high amplitude [11]. The time intervals shorten with increasing voltage.

Void This defect class is caused by a void/cavity inside a post insulator. The post insulators are made out of cast resin. Mainly due to entrapped air, cavities may form inside the insulator within the manufacturing process during the cool down of the cast resing. In new gas-insulated systems voids can be rarely found due to improved manufacturing and quality control [38].

As the cavity is filled with air instead of the insulation gas the insulation is lower in that area. This lead may lead to the occurrence of PD activity. The PD activity of this defect class is unstable, it may take a while before PDs occur and the occurrence may also fade away [38]. For DC voltage it is characteristic that the duration between two PDs is very long compared to other defect classes.

2.1.3 Partial Discharge Measurement Techniques

The measurements of partial discharges (PDs) enables the detection of defects in gas-insulated systems. PD recordings consists of a sequence of PD events consisting of their timestamp and amplitude. Figure 2.5 illustrates the properties of a measurement under DC voltage [10]. The timestamps are denoted by t_n, t_{n+1}, ... and the difference between timestamps by Δt_n respectively. The measured amplitude of a PD is denoted by A_n.

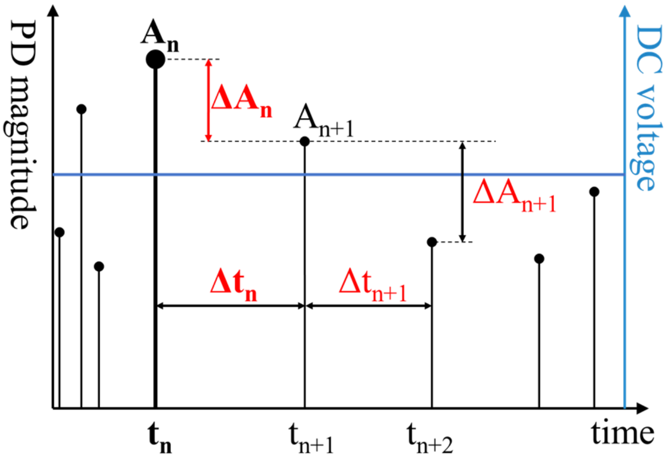


Figure 2.5 **Illustration of the parameters of PD measurements under DC voltage** [10].

Multiple techniques are available to perform PD measurements: There are the conventional measurement technique, the UHF method, acoustic measuring, optical measuring and chemical measuring [25, Sec. 6.4.2; 9, Sec. 4.2.2.2]. In contrast to the first two methods the latter three do not use the electrical signal of a partial discharge. Because of that they are not applicable in all situations [25, Sec. 6.4.2.9]. These methods are not described here further.

Conventional Method

The most commonly used method is the so called conventional method. It has been standardized in IEC 60270 (International Electrotechnical Commission). It works by connecting the test equipment to the section of the gas-insulated systems, where PD activity is to be measured. By use of a coupling capacitor the so called apparent charge q of a partial discharge is measured through a tiny voltage pulse [25, Sec. 6.4.2.2]. Through the apparent charge q the actual charge magnitude of a PD can be measured by proper calibration of the test equipment. The main disadvantage of this method is that the GIS/GIL section must often be disconnected from the grid by using a blocking impedance in order to gain reliable measurements [25, Sec. 6.4.2.1].

It is a common practice for AC voltage according to IEC 60270 to weight the PD amplitudes by their occurence [39]. However, this technique is not applicable to DC voltage [11; 39] due to an expected lower PD repetition rate [10]. Moreover, IEC 60270 contains initial but overall little guidance for PD measuring with DC voltage [11].

UHF Method

An alternative especially for gas-insulated systems is the ultra high frequency (UHF) method [25, Sec. 6.4.2.8]. Since partial discharges generate electromagnetic waves the UHF method works by detecting these electromagnetic waves in the range of around 100 MHz to 2 GHz. Special sensors are needed to detect these waves. These sensors can be integrated into a gas-insulated system in the form of so called disk sensors, see Figure 2.6.

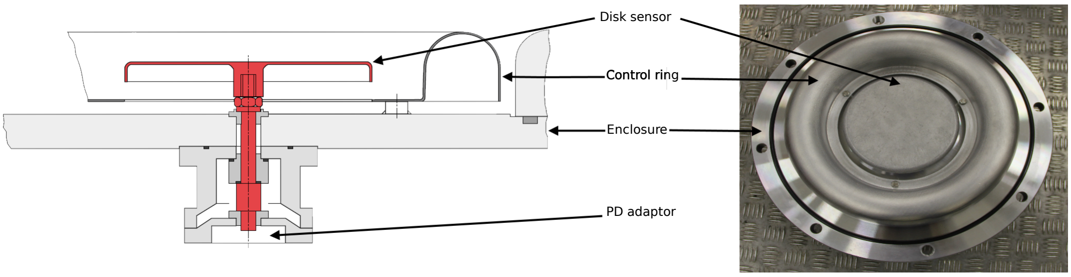


Figure 2.6 **UHF PD disk sensor**. The drawing on the left depicts the structure and the right part shows a picture of such a sensor. Taken and slightly adapted from [40, Fig. 3.2]. Source of left part is [41] and of right is [40, Fig. 3.2].

In contrast to the conventional method, the magnitude of a partial discharge can not be measured because there is no direct, clear relation between the amplitude of the UHF signal and the PDs electric charge magnitude. It depends on multiple factors such as the location of the defect in the gas-insulated system and the reflection of the UHF signal. Nevertheless, the UHF magnitude is and can be used by relying on the so called sensitivity verification check, which has to be done before the actual measurement [42]. This ensures that the UHF method will be as sensitive as the conventional method. Hence, it also fulfills the sensitivity requirements according to IEC 60270 [42].

An advantage of the UHF over the conventional method is that it is easier to use [42]. As the sensors can be integrated into gas-insulated systems another advantage is that the systems do not need to be disconnected from the grid and thus monitoring is possible [25, Sec. 6.4.2.8].

2.1.4 Noise

As the test equipment with both the conventional method and the UHF method is used to measure tiny voltage changes, they can be disturbed by other sources emitting electromagnetic waves. Thus, electrical measurement equipment needs to ensure so called electromagnetic compatibility (EMC), i.e. the ability of a (measurement) system to function properly in its interfered environment [25, Sec. 6.3.8]. For the conventional method one important source of noise is grid-based interference. As indicated in Section 2.1.3, to prevent this interference the relevant section of the gas-insulated system should be disconnected from the grid [25, Sec. 6.4.2.5]. Multiple noise sources also exist for the UHF method. Some noise sources are mobile radio communication and television [25, Sec. 6.4.2.8]. Examples causing interfering signals in the case of mobile radio communication are the LTE and GSM technology.

To reduce the disturbance of the UHF method by such signals multiple actions can be taken. Since partial discharges occur less often and less periodically for DC than for AC voltage these actions are even more important there [25, p. 440]. One of the actions is to shield off the equipment. In a HV laboratory setup this is useful and mostly sufficient. However, for on-site measurement this can often not be accomplished. Hence, noise-free measurements are more difficult on-site [25, Sec. 6.4.2.5]. Some noise signals, which disturb on-site measurements, such as radio broadcasting are caused by electromagnetic waves sent only on certain frequencies. A useful technique to ignore these can be to perform narrow-banded measurements [42; 25, Sec. 6.4.2.5]. For these the equipment is set to a frequency band with less or no interfering signals and a small bandwidth of frequencies is measured, i.e. the noise frequencies are not in this bandwidth. For DC voltage, due to the (much) lower PD repetition rate too few partial discharges might be detected with this technique though. Therefore, broadband instead of narrow-banded measurements are recommended with DC as e.g. done by [10]. Another action is to measure the partial discharges at multiple locations. This allows the separation of multiple noise sources and potentially multiple defects causing PDs [25, Sec. 6.4.2.7]. Furthermore, there are digital filters that can separate the partial discharges from the interfering signals after recording [25, Sec. 6.4.2.5].

2.1.5 PD Analysis in Gas-Insulated DC Systems

PD measurements and the analysis of these measurements are carried out in multiple situations, e.g. on-site measuring within commissioning tests [10; 29]. The identification of defects using the DC PD analysis is not reliable so far since the PD behavior differs between AC and DC voltage and there is not enough experience with DC yet. Hence, the recommendations are to perform the PD analysis with AC measurements also for gas-insulated DC systems [28, p. 103; 3; 15, Sec. 5.1; 14; 23]. As shown by [29] there can be situations where a free moving particle is not identified with AC but with DC voltage at the same voltage stress level since the PD amplitudes were too low with AC. Moritz Geske also experienced similar situations during his research [43, Abb. 2]. From these, one could infer that there may be situations where solely relying on AC measurements might not be sufficient for DC systems. Hence, it might improve the dielectric performance to additionally utilize the PD analysis with DC measuring. Correspondingly, the CIGRÉ working group WG D1.63 states that "*PD measurement at DC voltage is desirable*" because of "*various physical effects related to DC voltage stress, such as accumulations of space charges or the temperature dependence of the field dis-*

tribution"[15, Sec. 5.2]. Aside from tests such as on-site measurements, PD analysis is done within monitoring [25, Sec. 6.4.8], which is continuous measuring during operation. This requires that the analysis is, of course, based on DC measuring and is performed automatically [11].

The analysis of PD activity aims for an accurate risk-assessment of defects in gas-insulated systems. Essential for this assessment is the correct identification and classification of one or even multiple defects as not only one defect may be present in the gas-insulated system [10; 35, p. 57]. Additionally, it is important to find the location of the defects in case certain actions are required. To know the location and class of a defect, information and data gained through one or multiple of the above described measurement methods can be used. Conventionally, the classification of a defect is done by letting a human expert interpret graphical visualizations of the data. The expert compares therefore the visualizations to reference measurements. For AC voltage the most used visualization type are PRPD pattern. The partial discharges are thereby put in relation to the AC phase. Due to the lack of phase correlation PRPD patterns can not be used for DC voltage. Useful visualizations are there PSA pattern [10], see Figure 2.7, and also NoDi* pattern [44].

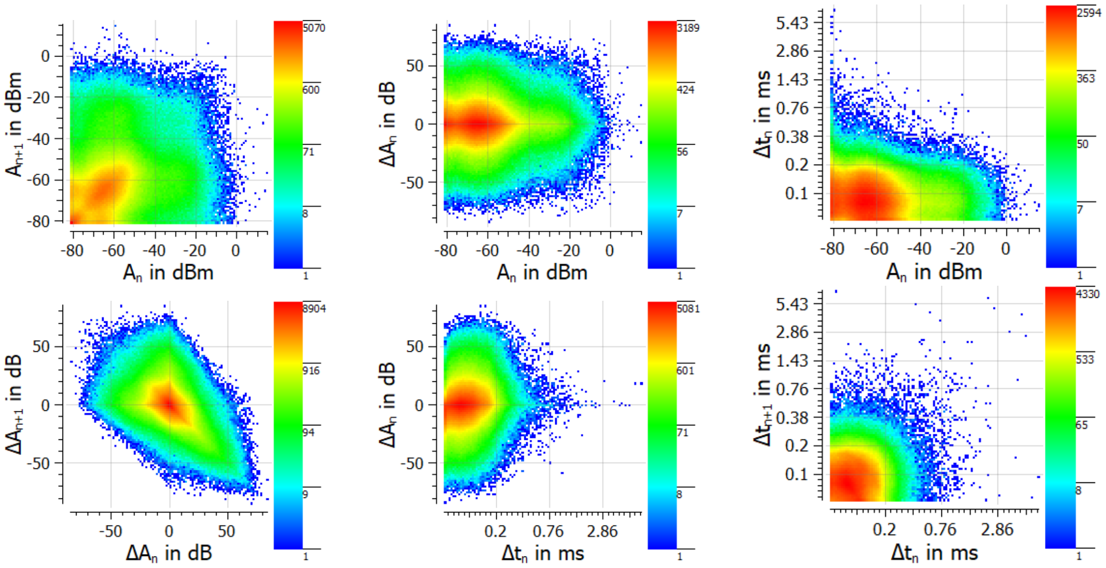


Figure 2.7 **PSA pattern of a protrusion** [10]. The applied voltage is −550 kV.

There are several disadvantages of relying on human expertise alone for defect identification[11]. A human expert needs to be trained and might make subjective or wrong decisions. Thus, machine learning based systems are gaining popularity in this domain [25, Sec. 6.4.2.6; 22; 20]. Here, ML algorithms have the potential of automating the defect identification and improving the detection accuracy. This might facilitate defect detection and even reduce costs.

2.2 Machine Learning

Machine learning algorithms are methods that learn a specific task based on given data. After training a ML algorithm, a model is obtained that is typically used to make predictions on a similar dataset. There are two different types of ML algorithms: supervised and unsupervised ML methods [45, Chap. 1]. Supervised ML algorithms are applied to data with a corresponding target or label. This target can be for example a certain class that the data sample belongs to and that should be predicted by the model. If the aim is to assign each data sample to one of a specific number of categories, then this is called a classification problem. In the case of two categories it is named binary. In case that at least three different categories can be assigned, then it is termed a multiclass classification problem. If the output is a continuous variable instead of a category, the problem is called regression. In contrast to that, unsupervised ML algorithms learn from data without any corresponding label. Here, a potential aim could be identifying different groups within the data or finding similar data samples. This work will focus on a supervised multiclass classification problem.

Different ML classifiers can be used to address such classification problems. The ML classifiers used in this work are described in Section 2.2.1. As the data used in this work are time series, time series classification (TSC) is introduced next. The ML fundamentals ends with a description of the cross-validation method to validate the models' performance.

2.2.1 Classifiers

k-Nearest Neighbor

K-Nearest Neighbor (k-NN) is an ML algorithm that makes use of the similarity between data samples [46]. It belongs to the class of non-generalizing classifiers. This means that k-NN assigns a sample according to its local neighborhood. When training this classifier, all data samples are simply stored in an n-dimensional space where n is the amount of features, see Figure 2.8 for $n = 2$. When predicting a new sample, the distance to the k closest neighbors is computed. The predicted class is then the class most neighboring samples belong to [45, Sec. 2.5.2]. The classes of the neighboring samples can either be weighted uniformly or by their respective distance to the data sample. The number of neighbors k can differ and is selected based on the best classification performance. Another important hyperparameter that can be tuned is the distance measure. Common distance measures include the euclidean distance or the Manhattan distance. For time series data, an established metric is DTW [47–49]. As all the training data is stored by the classifier the prediction can be quite expensive as the unknown sample is compared to every stored sample. To overcome this certain techniques such as a KD-Tree can be used to speed up this processing by indexing the data samples to some extent during the training.

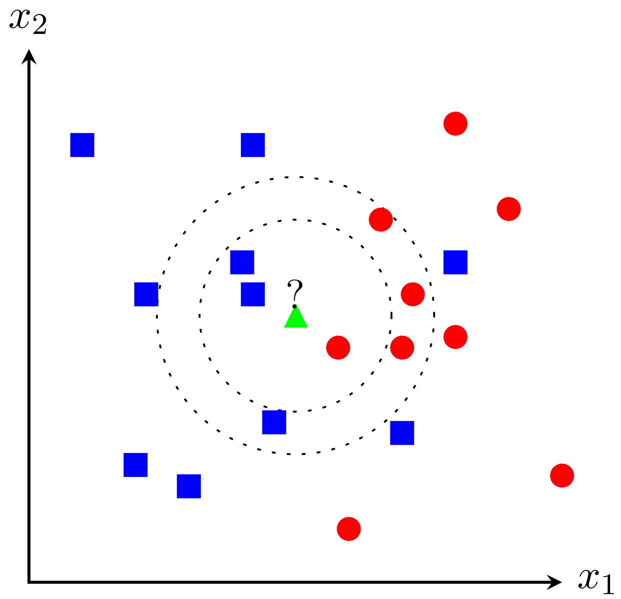


Figure 2.8 **k-NN with two classes and different k**. The values for k are 3 and 7.

Logistic Regression

Logistic Regression (LR) is in its basic form a binary ML classifier. The LR classifier is based on the logistic function, also called sigmoid function, which is defined as [50]:

$$\sigma(t) = \frac{1}{1+e^{-t}}, t \in \mathbb{R} \tag{2.1}$$

Following this definition the resulting value $\sigma(t)$ is in the open interval $(0, 1)$. This characteristic is used to form a probability, for which the decision boundary between the two classes is 0.5. Hence it is a binary classifier.

The classifier is trained by learning weights [50]. An input data vector x with n features, $x = (x_1, x_2, \ldots, x_n)$ is combined with the weights vector w as follows:

$$t = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \cdots + w_n \cdot x_n \tag{2.2}$$

The resulting vector t is the input of the logistic function. During the training process the weights w are optimized using gradient descent. The LR classifier is extended to support multiclass problems by techniques such as one-vs-one (OvO), one-vs-rest (OvR), or the multinomial approach [50]. In the one-vs-one approach all possible pairs of classes within the multiclass classification problem are classified alone. After that, majority voting determines the class prediction of a sample. An alternative would be the one-vs-rest approach. Each class is here classified against all samples from the other classes. Again, the final prediction is decided by majority voting.

Support Vector Machine

Support Vector Machines (SVMs) are algorithms that try to find the best classification boundary between two groups of samples, which is illustrated in Figure 2.9. This boundary is called a hyperplane. The distance between the hyperplane and the nearest samples is called the margin. The computation of the hyperplane and the margin is done by calculating all distances between the samples and determining the maximal margin [45, Sec. 7.1]. During the computation either a hard or a soft margin can be used. The soft margin allows for misclassification while a hard margin does not [45, Sec. 7.1.1]. Samples that lie directly on the hyperplanes are called support vectors. If samples cannot be separated by a linear hyperplane, i.e. are not linearly separable, kernel functions can assist. For that, all samples can be mapped into a higher dimensional space using a kernel function. Here, the samples might be linearly separable. Thus, all distances are calculated in the higher dimensional space and the hyperplane is computed. Since the data itself is not directly transformed into the higher dimensional space but only the relationships between the samples are calculated, SVMs are still computationally inexpensive. This is also called the kernel trick [45, Chap. 6]. Different functions can serve as a suitable kernel here. Among them are linear kernels, polynomial kernels or Radial Basis Functions (RBFs).

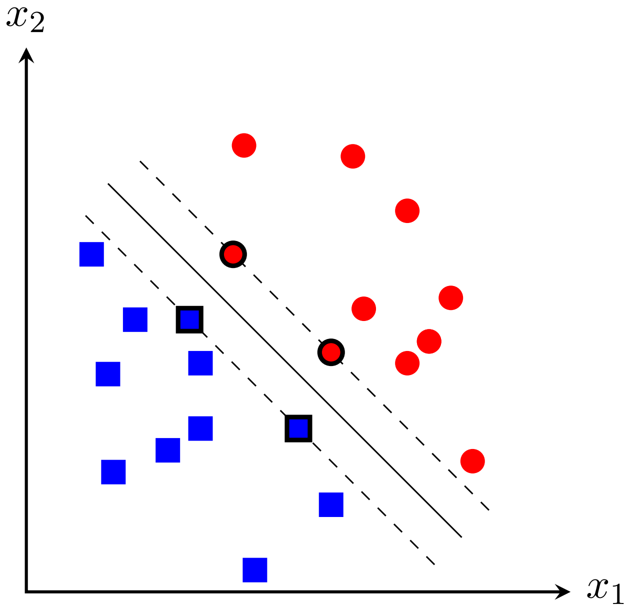
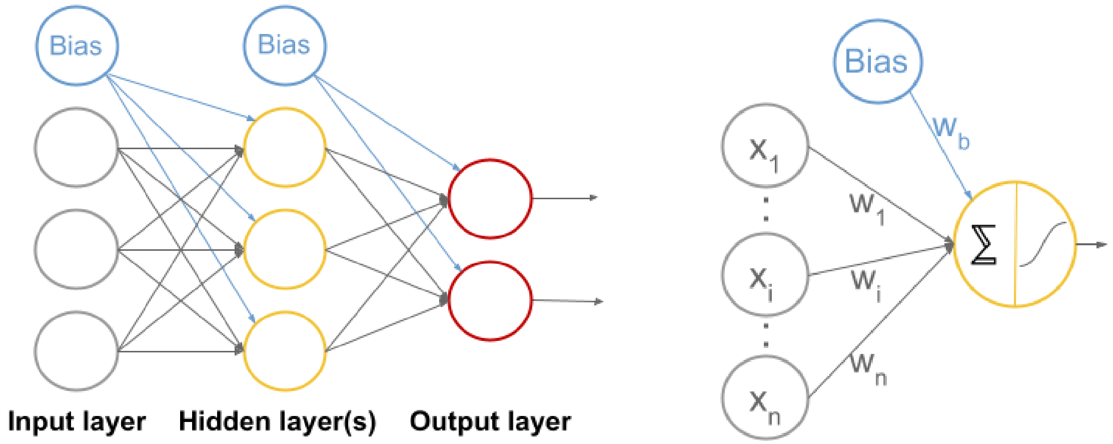


Figure 2.9 **SVM with two classes and a maximum-margin hyperplane**. The support vectors are encircled in black.

SVM is by default a binary classifier, i.e. it can only classify two classes. It can be converted into a multiclass classifier by using the one-vs-one (OvO) or one-vs-rest (OvR) approach [45, Sec. 7.1.3] (see Section 2.2.1). Furthermore, since the hyperplane is making binary, hard decisions the classifier's output does not provide probabilities by default.

Artificial Neural Network



(a) **Structure of an ANN**

(b) **Properties of a single neuron**

Figure 2.10 **Illustration of an Artificial Neural Network**.

Artificial Neural Networks (ANNs) are ML algorithms that are inspired by biological neural networks of the brain. They consists of different neurons that are connected with each other. The neurons are grouped into different layers, see Figure 2.10a. Each ANN consists of an input layer and an output layer [45, Sec. 5.1]. Depending on the complexity of the ANN there can be several hidden layers in between the input and output layer. With increasing number of hidden layers the networks gets more complex and is able to learn more complex relationships between input and output. The input layer represents a sample of the input data. This is then passed to the first hidden layer. In this way each neuron in the hidden layer gets information from several neurons of the previous layer. Additionally, there is a bias node that can add further information. The amount of information a neurons receives from each connected previous neuron is controlled by a weight w. All weights are multiplied by the input of the preceding neuron, e.g. x if it belongs to the input layer, and then summed up. This is then passed into an activation function that imposes non linearity and constructs the output, see Figure 2.10b. Different activation functions can be applied. The most common one are the sigmoid and rectified linear unit (ReLU) functions. This procedure is done until the input layer is passed through all hidden neurons to the output layer. The output layer then corresponds to the number of classes that should be predicted. In order to train a neural network, backpropagation is used [45, Sec. 5.3]. This is an algorithm that computes the error between the predicted output and actual output and passes the error back through the network, from output layer to input layer. In this fashion the weights for each neuron in the different layers can be adapted to reduce the error.

Random Forest

Random Forest (RF) is an ensemble classifier, i.e. the classifier is based on several base classifiers. A RF is based on Decision Trees (DTs) [51]. Decision trees use nodes to classify data based on certain features (see Figure 2.11). After splitting the data samples at the first node into two groups, both are directed separately to the next level that splits the data based on another feature-based criterion further. The splitting continues until a stopping criterion occurs. A possible criterion could be that all samples in the node have the same class or that a minimum number of samples per node is reached. After that, the class information of the different samples are stored in so called leaf nodes, which are at the bottom of the trees depicted in Figure 2.11.

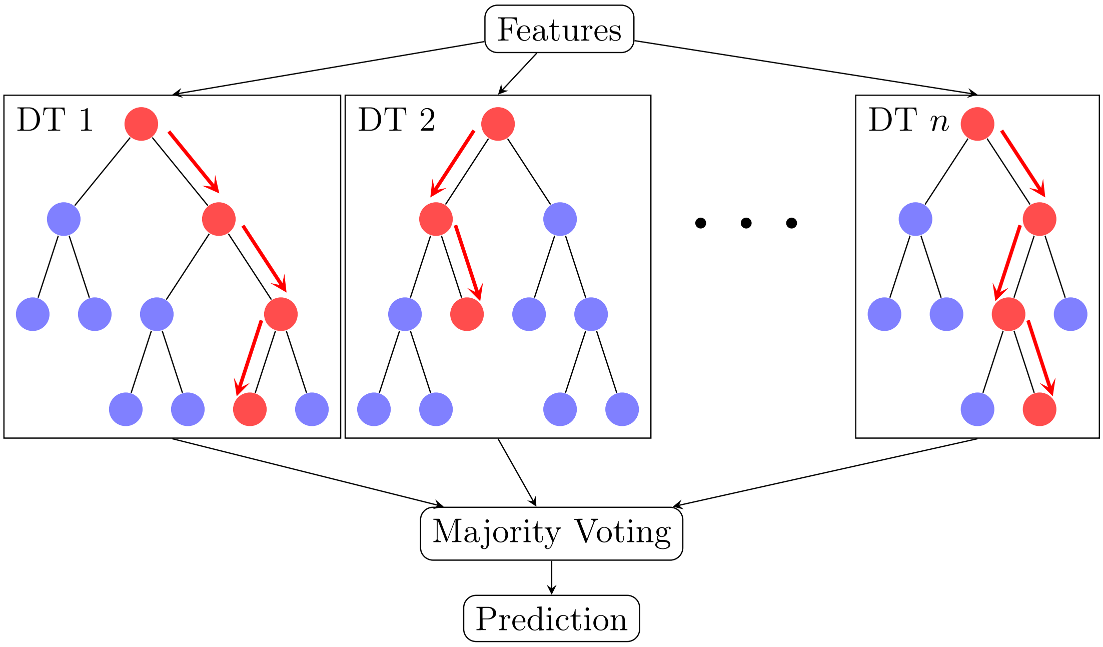


Figure 2.11 **Illustration of a RF consisting of multiple DTs**. After a Random Forest (RF) is trained the prediction works by passing an unknown sample to all Decision Trees (DTs) and combine their prediction to an overall prediction through majority voting.

Figure 2.11 illustrates the prediction process done by a RF. The first step in the RF algorithm is usually to randomly select a subset of the original training set, i.e. the bootstrapped dataset. This subset is used to build a decision tree. Then, for each split when building a decision tree typically only a randomly selected subset of the features is used. This procedure is independently done for more trees until the wanted number of trees is reached. The classification of a new sample is then determined by majority voting of the decision trees or by averaging the prediction probabilities.

Light Gradient Boosting Machine

The Light Gradient Boosting Machine (LGBM) is a ML classifier that is - as Random Forest - based on decision trees [52]. It differs from RF in how the decision trees are built. In contrast to Random Forest LGBM builds the decision trees iteratively. Decision trees are added one at a time to fit the error of the previous prediction. For the next prediction all trees are then combined and weighted by a learning rate. The algorithm is finished once the maximum number of trees is reached or the change in prediction error is below a certain threshold. Moreover, LGBM is considered to be a fast algorithm [52].

2.2.2 Time Series Classification

A time series can be seen as a sequence of values recorded over time. A time series T with real values $v_i \in \mathbb{R}$ can be defined as:

$$T = (v_1, v_2, \dots, v_i, \dots, v_n) \tag{2.3}$$

Time series are univariate as shown in the definition above, i.e. only one variable is recorded, or multivariate, i.e. multiple variables are recorded over time. As the data used in this work is univariate, the focus will lie on this kind. An alternative definition is to store pairs consisting of a timestamp t_i and measured value v_i [53]:

$$T = ((t_1, v_1), (t_2, v_2), \dots, (t_i, v_i), \dots, (t_n, v_n)) \tag{2.4}$$

This representation can be useful, if the recordings are not equidistant, formally $t_i - t_{i-1} \neq t_{i+1} - t_i$. It can also be useful, if most of the recorded values are 0 which is the case for the data used in this work.

Time series may have different lengths. Section 2.2.2 describes how to deal with such time series. After that multiple approaches to perform time series classification (TSC) are described and categorized.

Varying Lengths

As many TSC algorithms require time series to be of equal length, samples with differing length may need to be manipulated. One reason for different lengths is that time series were created with different sampling rates [47]. In the case of PD measurements this may be caused by using different sensors having different sampling rates. One way to handle this is to rescale the time series to be of equal length [47]. Another reason is that time series were created with a different start or end point, resulting in a different duration. In the case of PD measurements this may be caused by stopping a measurement earlier due to a very high rate of PD events. One approach to handle this is by prepending or appending random values or zeros to the time series [47]. Another approach is to truncate the data to the shortest series [49]. An additional alternative is to apply the sliding window technique to get segments of equal length [54]. The sliding window works by sliding over the time series and extracting subseries of a defined length. A more detailed description of this approach can be found in Section 3.4.1.

Classification Algorithms

Multiple approaches exist for TSC. Time series and sequence classification approaches can be grouped into multiple categories: Feature-based, Distance-based, Shapelets, Dictionary-based, Deep Learning, and Combinations. The categorization is based on [49; 53].

The idea of the **Feature-based** category is the problem to a conventional classification task. It works by extracting features such as the mean or the variance from the time series. These extracted features are then treated as tabular data and passed to a conventional classifier such as one of the classifiers described in Section 2.2.1 [53].

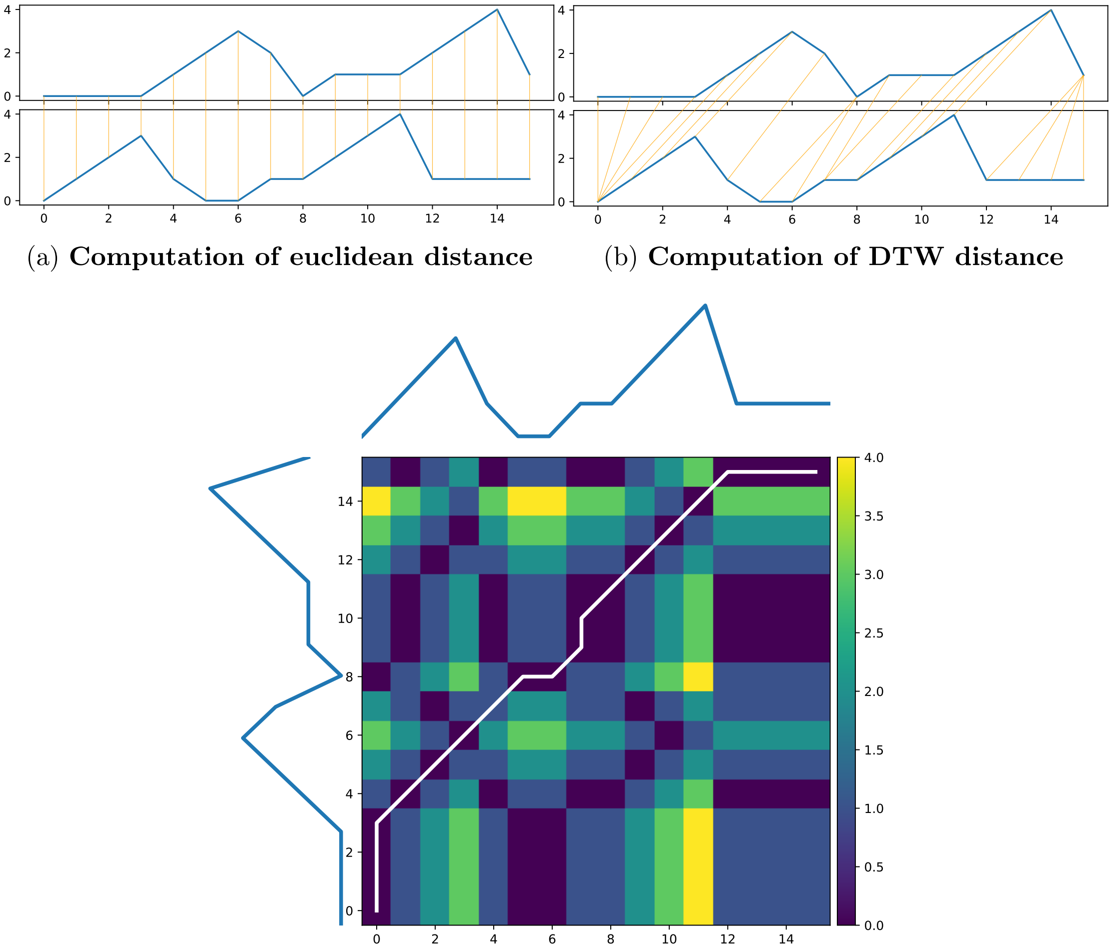
Another way to perform TSC is **Deep Learning**. It makes use of deep neural networks such as Convolutional Neural Networks (CNNs) or LSTM networks to classify time series [55]. A disadvantage is here that they usually need a lot of data for training and are computationally expensive, i.e. they are time-intensive and often require the use of a Graphical Processing Unit (GPU).

Approaches in the **Distance-based** category, called Whole Series by [49], use a distance measure to quantify the similarity between time series [53]. For that, a k-NN classifier is often utilized, which requires a distance measure. A potential candidate for the distance measure is the euclidean metric, which is illustrated in Figure 2.12a. Since the euclidean distance does not compensate for local distortions nor allows temporal shifts, other distance measures such as Dynamic Time Warping (DTW) are usually more suitable. DTW is often being used as the standard benchmark for TSC problems [48; 49]. Figure 2.12b illustrates how the distance is computed. Here, the distance between two time series is calculated by finding the optimal warping path between them. A distance matrix of the two time series $T_1 = (s_1, \dots, s_i, \dots, s_n)$ and $T_2 = (t_1, \dots, t_i, \dots, t_n)$ is constructed. The $n \times n$ matrix $M(T_1, T_2)$ is then filled with the following distance values:

$$\forall i, j \in \{1, 2, \dots, n\}\colon\ M(T_1, T_2)_{i,j} = (s_i - t_i)^2 \qquad (2.5)$$

The DTW distance is the sum of values of the optimal warping path through the matrix M (see Figure 2.12c). The optimal path minimizes this sum under certain constraints such as a limit on the warping amount [49]. Computing this optimal path can be quite time consuming, the run time is $\mathcal{O}(n^2)$ for time series of length n. To speed up the computation and potentially also improve the accuracy the amount of warping can be limited by setting a warping window, also called Sakoe-Chiba band [48; 56]. For the Sakoe-Chiba band it is $\mathcal{O}(nr)$, $r \leq n$, with r being the warping window size [48], also called Sakoe-Chiba radius. Other options limiting warping are to apply a so called Itakura parallelogram or to penalize warping (Weighted DTW) [49; 56].

The idea behind **Shapelets** is that time series can be discriminated by short patterns.



(a) **Computation of euclidean distance**

(b) **Computation of DTW distance**

(c) **Computation of DTW illustrated by the distance matrix.** Yellow denotes high and blue low values. The optimal warping path is shown in white.

Figure 2.12 **Illustration of DTW** by comparing it to the euclidean distance and and by showing the optimal warping path in a distance matrix.

It consists of algorithms that find short pattern discriminating the classes and using the presence or absence of these for classification. The found pattern are often called shapelets. A drawback of algorithms in this category is that discovering (good) shapelets can be computationally expensive depending on the data and the used algorithm [49].

The **Dictionary-based** category is based on the idea that the distribution of found features is more useful than e.g. the presence or absence of a single feature. The algorithms in this category have in common that they rely on the sliding window method. With the help of the sliding window method so called words are formed, which are then used for the classification. An example in this category is Bag-of-SFA-Symbols (BOSS) [57]. The words are formed with the use of symbolic Fourier approximations.

The **Combinations** category consists of approaches which combine multiple classifiers into one ensemble classifier. They consequently often achieve a higher accuracy than classifiers from the other categories [49; 58]. This comes usually at the cost of higher space and time complexity regarding both training and prediction such as in the case of COTE (Collective of Transformation Ensembles) [59]. Ensemble classifiers can be divided into homogeneous and heterogeneous ensembles. Homogeneous ensembles combine multiple instances of the same core classifier. In contrast to that heterogeneous ensembles utilize different classifiers. An example of a heterogeneous ensemble is COTE, which combines 35 classifiers belonging to different categories and using different transformations applied to the time series data. Some classification algorithms used in COTE are k-NN, SVM, and RF. A homogeneous ensemble is Word ExtrAction for time SEries cLassification (WEASEL), which is despite multiple classifiers supposed to be fast also on long time series [58]. WEASEL is similar to BOSS. It relies on multiple sliding windows of different lengths and utilizes feature selection. It can therefore be seen as an dictionary-based ensemble.

2.2.3 Cross-Validation

To get a robust and fair estimate of the model's performance, cross-validation can be performed. Cross-validation is a technique that aims to assess the generalizability of the model to an independent dataset. For that, the dataset is split into three different parts: training set, validation set and test set. The training set is utilized to train the ML model. The performance of this model is then evaluated on the validation set. After that, the ML model is trained and validated with different hyperparameters. The model with the best validation performance is then evaluated on the test set. Thus, the test performance gives an estimate of the generalizable performance that is not biased by a specific hyperparameter setting.

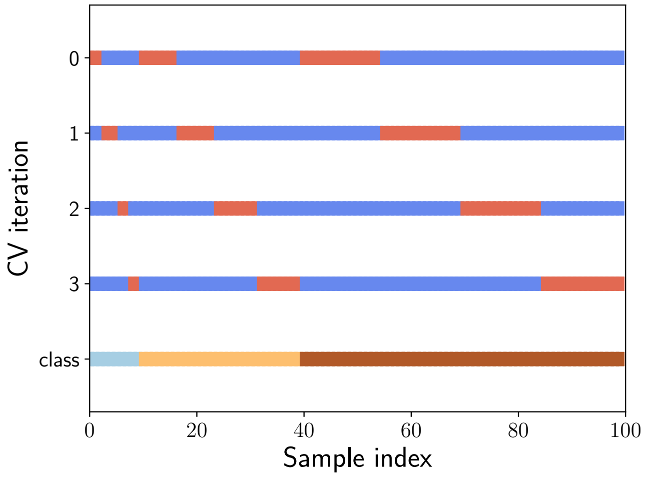


Figure 2.13 **Illustration of stratified k-fold cross-validation**. k is here 4 and the number of samples in the dataset is 100. Blue depicts the training and orange the validation in each fold. Adapted from [60].

In this work, two variations of cross-validation are used: stratified k-fold cross-validation and leave-one-out cross-validation [45, Sec. 1.3]. For k-fold cross-validation, the dataset is first split into training and test set. After that, the training set is split into k parts. The parts 2 to k are then used for training, whereas the validation is performed on the first part. Then, the second part is used for validation and the rest for training. This is repeated k times so that each of the parts was once used for validation. The validation performances are then averaged. The model parameters of the model with the highest average validation performance are then used to trained again on the whole training and validation set. This model's performance is then evaluated on the test set. For stratified k-fold cross-validation, the folds are additionally split in such way that the class distribution is preserved in each fold [61], see Figure 2.13 for an illustration. This is particularly useful if there is an imbalance of classes within the dataset.

The second cross-validation variation is leave-one-out cross-validation. It is a special version of k-fold cross-validation with k set to the number of data samples. Thus, every validation fold consists of one sample.

3 Methods

This chapter describes the methods used in this work. A workflow of how the classification models are constructed is shown in Figure 3.1. First, the data used in this work is described in Section 3.1. After that, the data is cleaned and further preprocessed (Section 3.2). The following section describes how the data is split into training and test and the usage of cross-validation (Section 3.3). In Section 3.4 the machine learning models are described. The model construction is divided into three groups: Feature-based models, 1D-Resample models, and raw data models. The three groups are visualized in Figure 3.1 by the different paths.

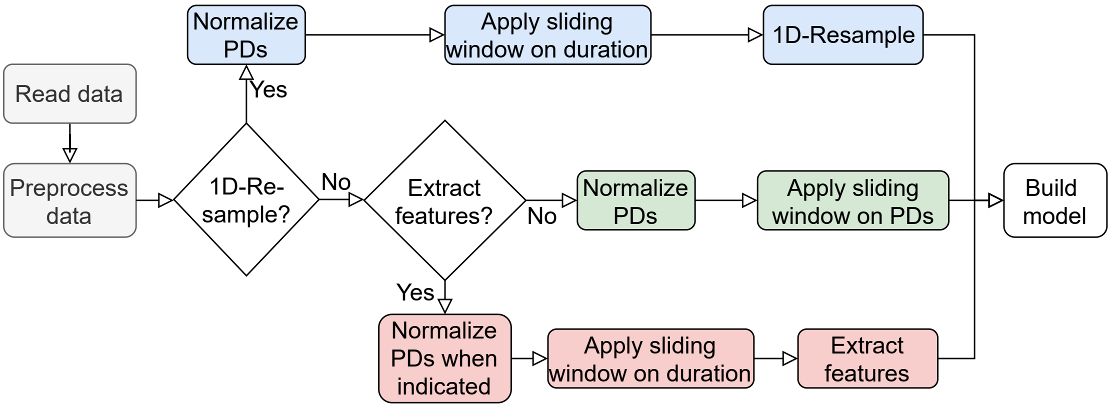


Figure 3.1 **Workflow for constructing ML models in this thesis**.

For data management and good software practices the guidelines by Wilson et al. were implemented [62]. The software libraries that were used for the specific models are mentioned in the corresponding sections. The source code to this work is stored in a repository of the Technische Universität Berlin GitLab instance [1]. A Graphical User Interface (GUI) was implemented to simplify the use of the ML models after completion of this work (see Figure A.1). For a quick and easy installation the source code and a ML model are provided as a Python package, a so called Wheel [2].

[1] Source code repository: `https://git.tu-berlin.de/flyingdutchman/thesis_implementation`

[2] Download Wheel here: `https://git.tu-berlin.de/flyingdutchman/thesis_implementation/-/packages`

3.1 Datasets

The data used in this work was provided by the Department of High Voltage Engineering at the Technische Universität Berlin. The underlying measurements were performed within research work with focus on the diagnosis of defects in gas-insulated DC systems [10]. The data consisted of partial discharge (PD) measurements on Gas-Insulated HVDC systems undertaken with the ultra high frequency (UHF) method. The measurements were conducted for the most part inside a laboratory setup with few on-site (OS) measurements done on a DC-GIL setup in Griesheim [29]. Further details on the laboratory experiments and data retrieval can be found in [10; 17].

The data consisted of six different datasets. In total 166 measurement files were available. Each dataset contained measurements executed on different setups, which are described in Table 3.1. The setups differ by which gas has been used, on which gas-insulated system the measurements have been performed, and whether there was a defect present during the experiment. Furthermore, the table states the number of included measurement files and an abbreviation for each dataset. The abbreviations will be used to refer to the datasets throughout this work.

Table 3.1 **Description of datasets**. The number of measurement files (Num) is mentioned for each dataset and an abbreviation (Abbr.) is attached for further references.

Abbr.	**Description**	**Num**
Mixture	laboratory setup filled with a mixture of 60% N_2 and 40% SF_6	115
SF6	laboratory setup filled with 100% SF_6 gas	6
CleanAir	laboratory setup filled with 100% CleanAir gas	20
OS-Normal	large on-site prototype installation in Griesheim [29]	8
OS-V2	large on-site prototype installation in Griesheim [29] with slightly different measurement tools (a different amplifier)	4
Noise	laboratory setup without any defects	13

Each dataset contained measurements that can be assigned to one of six defect classes: Moving Particle, Particle on Insulator, Protrusion on HV, Protrusion on Enclosure, Floating electrode, and Void. The datasets are imbalanced regarding the quantity of the defect classes: The datasets SF6, OS-Normal, and OS-V2 contain only defects of the type moving particle. The CleanAir dataset contains the defects moving particle and protrusion on HV and the Mixture dataset contains measurements of all defect classes. The remaining dataset called Noise contained as the name says only noise data. Noise data consisted of measurements, where no defect was present in the Gas-Insulated System, but where external disturbing factors caused the sensors to measure PDs, although none occurred (see Section 2.1.3). As the main goal of this work was to distinguish the defects from each other the noise data was analyzed separately (see Section 3.4.5).

3.2 Data Preprocessing

3.2.1 Data Cleaning

The raw data was stored in two different file formats: CSV and MAT (used by the software MATLAB). To simplify further processing steps the MAT files were converted to CSV.

Some of the measurements were included in parts or as a whole in other measurement files. The "whole" duplicates were removed. The "part" duplicates were identified and compared to each other. Here, the longer measurement file was included for further analysis while the shorter one was removed.

3.2.2 Further Preprocessing Steps

The following preprocessing steps were applied to the data:

Test Voltage Refusal Some of the measurement files contained the applied test voltage when PDs occurred. As only a few measurements contained this information, it could not be used and was therefore dropped.

Absolute value of PD amplitude Some of the measurements contained negative PD magnitudes. Negative PD values are invalid and were caused by the measurement device [10]. As the amplitude of UHF PD measurements can physically only be positive the absolute value was taken. This is common practice in the field and was thus also applied here.

Scale Time The time values were stored in the unit seconds. Due to the sampling rate of the sensors the minimal difference between adjacent PDs could be 50 microseconds. To increase the numerical difference between the timestamps all time values were converted to the unit milliseconds.

Δt Use Since most of the received measurement files were only segments of longer observations, the start was often missing and the initial time value was much greater than zero. To overcome this issue the difference of the time values Δt was calculated and used in place of the time values.

PD Normalization Depending on the used model the PD amplitudes A were normalized individually per measurement file by dividing all PD values by the maximum value. This resulted in all values being in a range of $(0, 1]$. In contrast to the other preprocessing steps this step was not done for all models. If a model did not utilize normalized data, it was explicitly stated so in Section 3.4.

3.3 Data Splitting

3.3.1 Training and Test

The datasets were split into training and test. Roughly 70% of the data was used for training and the rest for test. To test generalizability to slightly different setups, the four smaller datasets (SF6, CleanAir, OS-Normal, OS-V2) were assigned as a whole to either training or test (see Table 3.2). CleanAir and OS-V2 were only included in the test set. While SF6 was included only for training, CleanAir was included only for testing to assess if the models' performances can extend to a dataset that used different gas mixture within the laboratory setup. Furthermore, the OS-V2 dataset was included in the test set whereas OS-Normal was used for training to test if the models' predictions can generalize to a slightly different measurement system. The Mixture dataset was split randomly while maintaining a similar class distribution.

Table 3.2 **Overview of dataset splitting into training and test**.

Dataset	**Training**	**Test**
Mixture	90	25
SF6	6	-
CleanAir	-	20
OS-Normal	8	-
OS-V2	-	4

3.3.2 Cross-Validation on Training Set

Cross-validation was utilized as a technique to validate the models' performances regarding the used parameters. Hereby, the training set was split into a training and a validation set. Whereas the ML models were trained on the training set for different sets of hyperparameters, the evaluation was performed on the validation set. The best validation performance then determined the best hyperparameter configuration. Here, two different cross-validation mechanisms were used: leave-one-out cross-validation and stratified-k-fold cross-validation from the scikit-learn library [60]. The two strategies are described in Section 2.2.3. Due to computational restrictions only one parameter setting per model could be calculated with the leave-one-out cross-validation. The stratified-k-fold cross-validation was utilized within grid search. K was chosen to be equal to the number of measurements in the minority class, the minority class being the class with lowest number of measurements.

3.4 Machine Learning Models and Training

In this work, several machine learning models were utilized. Figure 3.2 gives an overview of the different methods grouped by the representation of their input data. The groups are referred to as Feature-based models, 1D Data models, and Raw Data models (Sections 3.4.1 to 3.4.3).

As the data was subsampled using the sliding window technique for all models multiple classifications exist per measurement file. Therefore, it is described how these predictions were combined to form the actual prediction (Section 3.4.4). This section finishes by integrating the noise as an additional class into on of the models (Section 3.4.5).

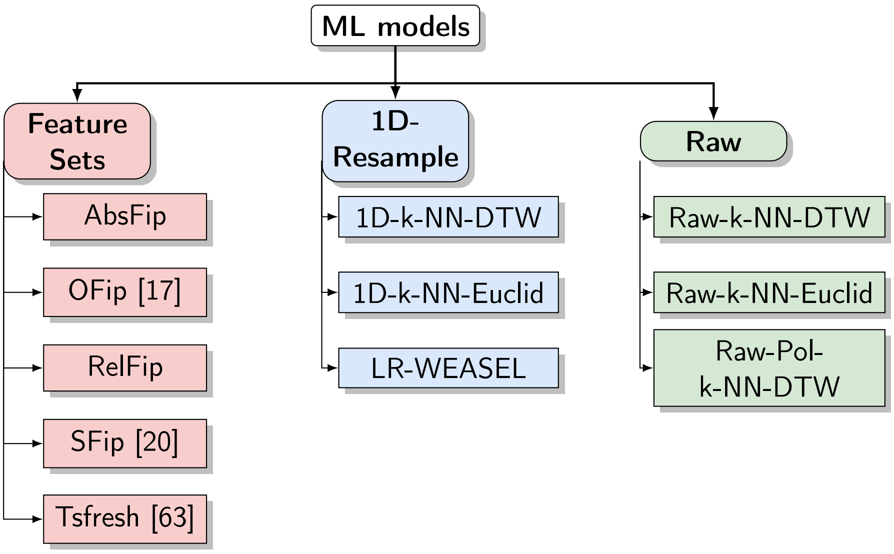


Figure 3.2 **Overview of implemented ML models**.

3.4.1 Feature-Based Models

Sliding Window

The data showed high imbalance regarding the number of samples per class and high variation among the measurements regarding their duration and number of partial discharges. To counteract this and to increase the number of training samples, the data was subsampled using the sliding window technique. Here, a window (sample) of a certain length, called the window size, is extracted starting at the beginning of the measurement file (see Figure 3.3). Then the sliding window was moved by the step size and the next sample was generated. This continued until the end of a measurement file. In this way, overlapping samples of the same size were created.

To select a suitable window size for the measurement files, two aspects needed to be considered. On the one hand the duration for each sample should have a minimum length so that every sample contains a minimal number of PDs. On the other hand it should be short enough so that most measurement files would be longer than the set duration.

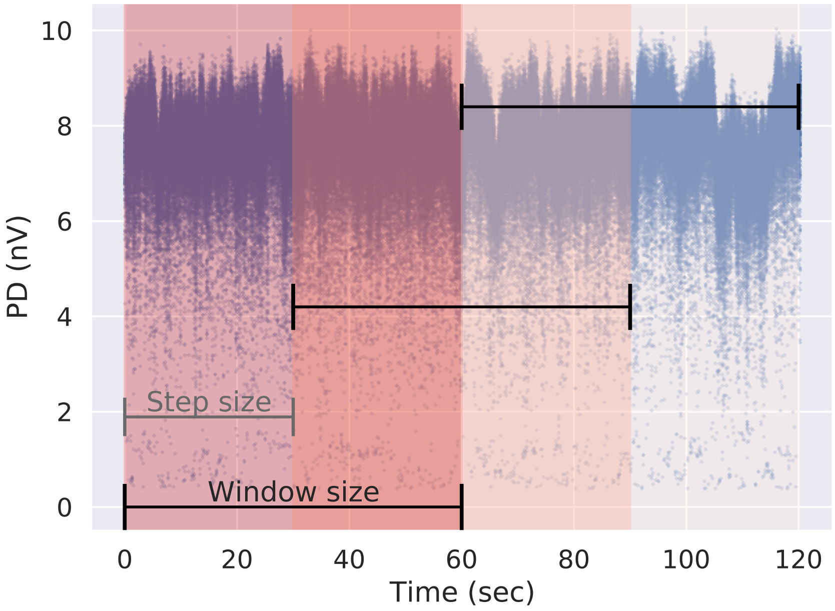


Figure 3.3 **Illustration of the sliding window**. It is shown on a measurement of a moving particle defect. Here, the window size is 60 and the step size 30 seconds.

After selecting the window size, all files which were too short were then excluded from further processing. Related to the common definition of the PDIV (Section 2.1.1) all windows that contained fewer than 3 PDs per 30 seconds were also excluded. Due to these mentioned requirements the window size and step size were regarded as hyperparameters that were tuned.

Feature sets

As described conceptually in Section 2.2.2 features were extracted from the data in this group of ML models. These were then used for classification. Figure 3.4 shows the (logical) relation between the five implemented feature sets. To build upon previous work, two existing feature sets established by Ott [17] and Schober and Schichler [20] were reimplemented and used as a baseline. The feature set from Ott is referred to as OFip (Fip standing for Fingerprint). It consists of 12 features, which are described in Table A.1a. The feature set by Schober and Schichler [20] will be referred to as SFip. It consists of 14 features, which are described in Table A.1b. The main difference between the two feature sets is that OFip did not use normalized PD amplitudes. Hence, for this feature set the preprocessing step of normalizing the PD values was skipped.

In this work two additional feature sets were established. The two feature sets are similar to OFip, and SFip and also used features from [21] and six features from Tsfresh, a library for automatic feature extraction and selection of time series data [63]. The first features set will be referred to as RelFip (Relative Fingerprint). It contains 24

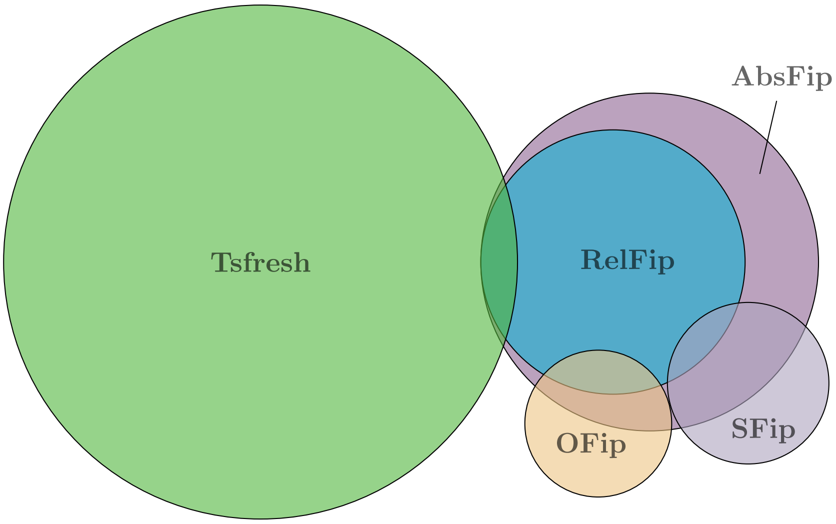


Figure 3.4 **Set diagram of feature sets**. The scale of the circles illustrate the sets' cardinality differences.

features which are described in Table 3.3. In addition to the description, Table 3.3 contains an abbreviation, by which each feature can be referenced later. The second feature set AbsFip (Absolute Fingerprint) contains all the features from RelFip plus certain features depending on the absolute PD value, in total 29 features. All features, which are in AbsFip, but not in RelFip are described in Table 3.4. As for OFip the preprocessing step of normalizing the PD values was skipped here.

For both feature sets, the features were selected and their parameters were tuned manually by visualizing them and calculating the validation score with changed features. For the visualization and feature selection so called pairplots were used (see Figure A.2) [64]. Additionally, to find and exclude redundant features correlation matrices, which show the correlation between all features, were utilized (see Figure A.3).

In addition to the manually constructed feature sets the automatic feature extraction library Tsfresh has been used for further comparison [63]. The library automatically calculates a large number of features and provides tools for feature selection. The library does not take into account whether the data has equidistant time stamps or not. For the Tsfresh feature extraction the data could not be transformed into equidistant time series due to memory restrictions. Therefore, the feature extraction was done on the PD value sequence without the time values. Tsfresh offers two suitable feature sets, a feature set consisting of all its features (ComprehensiveFCParameters) and a smaller feature set skipping the calculation of features with high computational cost (EfficientFCParameters) [63]. To limit computational time the EfficientFCParameters set was utilized. Further, Tsfresh allows to tune the feature selection. For the feature selection two parameters were tuned. The first parameter was the p-values threshold. P-values are a measure of the significance of a feature with respect to a binary classification prob-

Table 3.3 **Features used for the feature set RelFip**. The time difference values are denoted by Δt. PD amplitudes are denoted by A and amplitude differences by ΔA.

Feature ID	**Description**
+DC/-DC	Polarity of DC voltage applied on GIS/GIL
A-Num->-mean	Number of A higher than A mean [63]
A-norm-Weib-α-β	Weibull parameters α, β of fit to distribution of A. See Section A.2 for further information on Weibull.
A-Num-peaks	Number of PD value peaks of support n. A peak has support n, if it is bigger than its n neighbours to the left and to the right [63]. Chosen values for n is 5
A-Max-len-<-mean	Length of the longest part with only A values smaller than mean A
A-Var	Variance of PD
PDs/Sec	PDs per second
Time-norm-Weib-α-β	Weibull parameters α, β of fit to normalized time values
Δt-Change-quantile	Variance of absolute change of Δt values in the quantile $(0.0, 0.3]$ [63]
Δt-Kurt	Kurtosis of Δt
Δt-Median	Median of Δt
Δt-Skew	Skewness of Δt
$A/\Delta t$-Weib-α-β	Weibull parameters α, β of fit to distribution of $\frac{A}{\Delta t}$
Auto-Corr-A	Pearson autocorrelation of A with lag l. Chosen values for l are 1, 2, 3
Auto-Corr-Δt	Pearson autocorrelation of Δt with lag l. Chosen values for l are 1, 5, 10
Corr-A-to-ΔA	Pearson correlation of A and ΔA
Corr-A-to-Δt	Pearson correlation of A and Δt

Table 3.4 **Additional features not in RelFip, but considered for the feature set AbsFip**. PD amplitudes are denoted by A.

Feature ID	**Description**
A-Max	$\max A$
A-Min	$\min A$
A-Mean	mean A
A-Median	median A
A-Sum	$\sum A$

lem. Higher p-values indicate a lower significance. The p-values threshold, which is also called False Discovery Rate (FDR) level by the Tsfresh package, then defines the highest p-value for which a feature is regarded as significant [63]. During the feature selection p-values were created for all extracted features for each one-vs-rest binary classification problem. Since the present classification problem is multiclass, this were six OvR binary problems here as the number of classes was six. The other parameter was the number of classes. This parameter describes the number of classes for which a feature needs to be significant to be selected. This automatic feature set will be referred to as Tsfresh.

Classifiers

As a baseline the classification algorithm by Ott was reimplemented and used with the OFip feature set [17]. I will refer to the classification algorithm as OttAlgo. In contrast to the other models no sliding window was used here as it was not used by Ott.

Multiple classifiers were applied on the five feature sets (OFip, SFip, RelFip, AbsFip, Tsfresh): RF, LGBM, k-NN, SVM, and Multi-Layer Perceptron (MLP). More information on the classifiers can be found in Section 2.2.1. For the classifiers RF, k-NN, and SVM the scikit-learn library was used for implementation [60]. The LightGBM library by Microsoft was used to implement the LGBM classifier [65]. Keras was used to implement the MLP classifier [66].

For some of the classifiers feature scaling was performed: MinMax normalization for the k-NN classifier, z-score normalization for the MLP, and for the SVM the feature scaling method was part of the hyperparameter tuning. For all classifiers hyperparameter tuning was done performing grid search over selected parameters. These hyperparameters and selected values are described in Table 3.5 for all classifiers.

3.4.2 1D-Resample Models

This section describes models using data that was transformed and resampled to a certain sampling rate. In addition to the resampling transformation the same sliding window method as for the feature-based models was used (see Section 3.4.1).

Data Transformation: 1D-Resample

The preprocessed data is a sequence of pairs of time difference Δt and PD amplitude A, for which the Δt values are unequal. In other words, the data is 2D. To have equidistant times and 1D samples, the preprocessed data was converted (resampled) according to the sensors' sampling rate. The used sensors were able to detect up to 20000 PDs per second [10]. Hence, each converted sample consists of 1200000 equidistant values per minute. All values between the values of the PD events were set to zero because nothing was detected there.

Due to computational and/or memory limitations the 1D series were downsampled

Table 3.5 **Hyperparameters of all feature-based classifiers tuned via grid search**.

Classifier	Name	Values
RF	Minimum samples per leaf node	1, 2, 5, 10, 30
	Perform Bootstrapping	Yes, No
	Balance class weights	Yes, No
LGBM	Feature fraction per tree	1.0, 0.8
	Min. gain when splitting a node	0.0, 0.01, 0.1, 0.2
	Max. bins for feature values	63, 127, 255
	Max. leaf nodes per tree	10, 30, 50
	Balance class weights	Yes, No
k-NN	Number of neighbors	1, 5, 10, 30
	Weight neighbors	Uniform, Distance
SVM	Decision function shape	OvO, OvR
	Type of kernel	RBF, Poly, Linear
	Balance class weights	Yes, No
	Feature scaling	MinMax, Z-Score
MLP	Batch size	1, 10
	Dropout (Same for each layer)	0.0, 0.05, 0.2
	Neurons per layer	(5), (20), (5,3), (20, 10)
	Balance class weights	Yes, No
	Epochs to train	50, 100, 150

to a lower sampling rate. The chosen sampling rate depended on the used classifiers. This data transformation will be referred to as 1D-Resample in following sections.

Classifiers

Here, k-NN with the two distance measures, DTW and the euclidean metric, was used. The Tslearn library was used for the computation of the DTW distance measure [67].

The hyperparameter grids for the two classifiers are described in Table 3.6. Since DTW is computationally expensive regarding the sample size parameter tuning was limited. The both models were first used with different window sizes, namely 60 s, 30 s, 15 s, and 5 s. The window step size was set to the same value as the window size. For the 1D-k-NN-DTW model the sampling rate was adjusted to keep the computational time reasonable. Moreover, to deal with the class imbalance in the data Random Undersampling (RUS) is used as a hyperparameter. RUS ensures that all classes contain the same number of samples by eliminating samples randomly from the majority classes. The two models are referred to as 1D-k-NN-DTW and 1D-k-NN-Euclid.

Further, the dictionary-based method WEASEL was applied [58]. The WEASEL method computes a large set of words (features) by using multiple sliding windows of different sizes. For each sliding window features are generated by converting the window to a word by using Symbolic Fourier Approximation. A LR classifier was used to perform

Table 3.6 **Hyperparameters options of the 1D-Resample k-NN models tuned via grid search**.

Model	Name	Values
1D-k-NN-DTW	Number of neighbors	1, 5, 15
	Weight neighbors	Distance
	RUS	Yes, No
1D-k-NN-Euclid	Number of neighbors	1, 3, 7, 15, 30
	Weight neighbors	Distance, Uniform
	RUS	Yes, No
	Sampling rate	20 000 Hz

the classification on these generated features [58]. For the implementation of the LR classifier scikit-learn was used and for the implementation of WEASEL a Python library called pyts library was utilized [60; 68].

WEASEL offers several hyperparameters such as the word size or the window size to be tuned. Due to computational restrictions no systematic hyperparameter tuning was performed as the computationally highest feasible sampling rate was used. This model will be referred to as LR-WEASEL further.

3.4.3 Raw Data Models

Sliding Window

For the raw data model the sliding window was also applied. In contrast to the 1D-Resample and feature-based models (see Section 3.4.1) it was applied here on the number of PDs instead of the duration.

Classifiers

A k-NN classifier was implemented for this group of models using the DTW and the euclidean distance. In the following sections the two models will be referred to as Raw-k-NN-DTW and Raw-k-NN-Euclid.

Moreover, an adaption of Raw-k-NN-DTW was implemented. Usually when implementing k-NN classifiers, every sample is compared to all other samples to calculate their nearest neighbors (see Section 2.2.1). In this adaption the neighbors were only calculated among samples with the same polarity. It is referred to as Raw-Pol-k-NN-DTW.

Additionally, as for the 1D-Resample models Random Undersampling (RUS) was applied here as a hyperparameter to deal with the data class imbalance. For the Raw-k-NN-DTW model an alternative undersampling strategy was to take the first n samples of each measurement file. The tuned hyperparameters for the three classifiers are described in Table 3.7.

Table 3.7 **Hyperparameters of the Raw data classifiers tuned via grid search**. "First n" stands for extracting the first n windows from each measurement file.

Model	Name	Values
Raw-k-NN-DTW	Number of neighbors	1, 3, 7, 15
	Sakoe-Chiba radius [67]	4, 8, 16, 32, 64, No
	Weight neighbors	Uniform, Distance
	Undersample	RUS, First n
Raw-k-NN-Euclid	Number of neighbors	1, 3, 7, 15
	Weight neighbors	Uniform, Distance
	RUS	Yes, No
Raw-Pol-k-NN-DTW	Number of neighbors per Polarity	1, 3, 7, 15
	Weight neighbors	Uniform, Distance
	RUS	Yes, No

3.4.4 Measurement File Prediction

Since the sliding window approach led to multiple predictions per measurement file, these predictions had to be combined to an overall prediction. This was done by averaging the probability predictions of all samples of a measurement. The defect with the highest probability was then selected as a prediction for the whole file. As the SVM implementation did not offer probability prediction by default another method had to be used with this classifier. Here, the method was to select the defect which was most often predicted among the samples of a measurement file.

3.4.5 Noise Data Classification

To distinguish noise as a reason for observing PDs from actual defects, the noise data (see Section 3.1) is treated as an additional class. As the models relying on normalized PD amplitudes are supposed to generalize better to measurements on other gas-insulated systems only these were considered. Of these models, the one with the highest validation score was selected for which feature attribution analysis (see Section 3.5.3) could be performed. The noise data was integrated into the existing training and test datasets. The noise measurements did not have information on the polarity of the gas-insulated system because the detection of noise signals is not affected by it. As the RelFip and AbsFip feature sets include the polarity, the noise data measurement files were duplicated by having one with negative and another one with positive polarity for each file. The validation and test scores were calculated here in the same manner as for the other models.

3.5 Evaluation

3.5.1 Model Selection & Performance

For the evaluation multiple metrics were used: Accuracy, precision per defect, recall per defect, and balanced accuracy. Accuracy (ACC) is defined as:

$$\frac{\text{Number of correct predictions}}{\text{Number of predictions}} \tag{3.1}$$

Precision and recall are metrics used in binary classification problems and therefore needed to be used per class for multiclass classification problems. To define precision and recall several terms need to be introduced. True Positives (TP) are defined as the correct predictions of a class. False Positives (FP) are defined as the predictions which are predicted as a certain class without really belonging to it. Further, False Negatives (FN) are defined as the predictions, which are predicted to be not of a certain class while actually being samples of that class. By using these definitions, precision is then defined as:

$$\frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

In contrast to that, recall is defined as:

$$\frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

Balanced accuracy (BAC) is defined here as the mean of the recall of each class. Due to the imbalance of defects in the data the main metric used to compare models was balanced accuracy.

The test scores were computed for the models with a validation score higher than 60% and also for at least the best model from each of the three groups. The test set can be divided into four datasets: The Mixture test dataset, a manipulated Mixture test set, the OS-V2 dataset, and the CleanAir set (see Section 3.1). The manipulated Mixture set was created to check the robustness of the model against measurements for which less partial discharges were detected. It will be called Mix-0.4nV and was generated by performing the following steps:

1. Substract each PD amplitude A by $0.4\,\text{nV}$.
2. Drop all PDs, where $A < 0.4\,\text{nV}$.

3.5.2 Prediction Time Comparison

The accuracy of a model is not the only relevant aspect for comparisons. Another important aspect is the time a model needs to predict a sample. Here, it is especially useful because of the high time complexity of DTW. The prediction time is recorded for all models on the Mixture test set. The resulting numbers are averaged. To make the results more comparable the recordings were done when no other user application was running and with no model using more than one CPU core.

3.5.3 Feature Attribution Analysis

To evaluate the contributions of each feature in a ML model, a feature importance analysis was performed. For this, the Shapley values were computed for one feature-based model. Shapley values are a concept from game theory [69], which are named after the author. Basic components of this concept are a cooperative game, the players, and the overall gain for the players. Shapley values measure the contribution of each player to the gain. In other words, they give an answer to the importance of that player. The importance of a player is measured by comparing the gain of a player combination with that player to the combination without that player. As the dependencies between the players may be non-linear this comparison has to be performed for all player combinations. The final importance for a player is then computed by averaging the gain differences in a particular way. When adopting this to machine learning the game stands for producing the prediction of a ML model for one sample, the players are the features of the model, and the gain is the difference between the prediction of that one sample and the average prediction of all samples. As the Shapley values for each feature are calculated per sample an overall feature impact is measured by averaging the impact for all samples.

Here, the feature attribution analysis based on Shapley values was done for the RelFip feature set on the training data. For the implementation, SHapley Additive exPlanations (SHAP) was used [70]. As the exact computation of Shapley values is challenging, especially with an increasing number of features, SHAP approximates the values [70]. SHAP works with a variety of different classifiers such as LGBM and also offers a model agnostic approach. However, due to implementational constraints SHAP could not be applied to MLP[3]. Therefore it was used with a different classifier.

[3] See this Github issue: `https://github.com/slundberg/shap/issues/1741`

4 Results

4.1 Data Properties

Measurement Properties As mentioned in the Methods chapter the data varies greatly by the duration and the number of PD events. This can be seen in Figure 4.1 for all defect types and also noise measurements. In general it can be said that measurements with longer durations have fewer PDs. The minimum duration among all measurement files was 8.6 seconds.

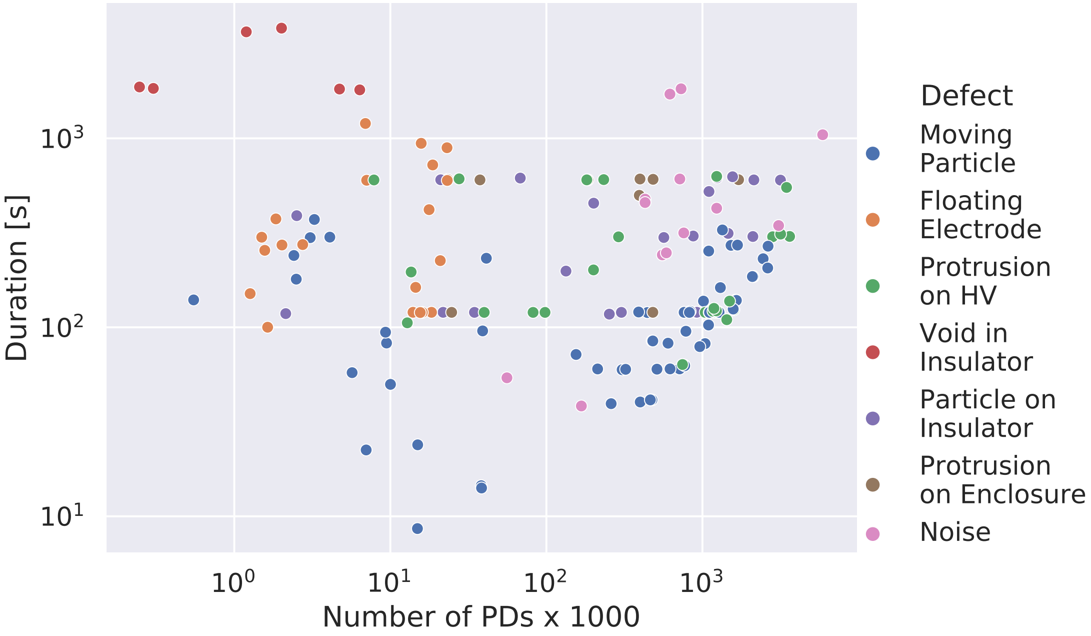
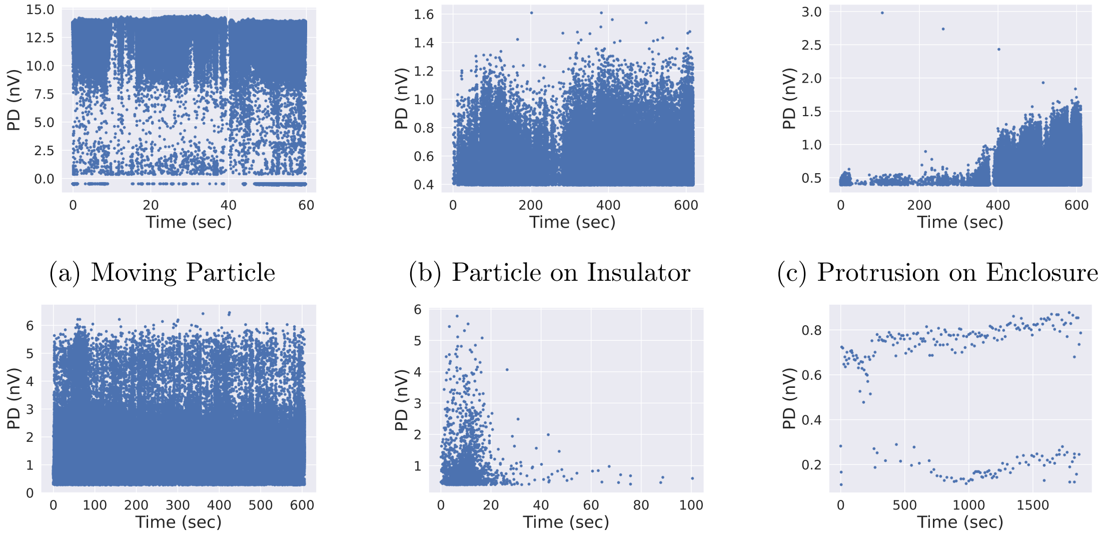


Figure 4.1 **Overview of all measurement files in relation to their PD count and duration.**

Apart from varying durations the files show different PD behavior. Samples of certain classes such as the moving particle show local, but no global differences. Other classes such as the floating electrode defect show global differences (see Figure 4.2).

Polarity and Class Imbalance The polarity occurrence in the measurements is imbalanced for three defect types. In other words, there exist more moving particle measurements with negative polarity than with positive polarity (see Figure 4.3). Since the



(a) Moving Particle

(b) Particle on Insulator

(c) Protrusion on Enclosure

(d) Protrusion on HV

(e) Floating Electrode

(f) Void in Insulator

Figure 4.2 **PD activity of an example measurement for each of the six defect classes**. Each example shows the measured UHF amplitudes over time. Certain examples show global pattern such as Figures 4.2c and 4.2e whereas the other do not.

noise data does not have polarity information it is not depicted in Figure 4.3. In addition to the polarity imbalance, Figure 4.3 shows the imbalanced distribution of classes.

Mix-0.4nV Test Set As described in Section 3.5.1 the Mix-0.4nV dataset contains manipulated measurements from the Mixture dataset. Due to this manipulation no void defect measurements were in this test set because nearly all their PD values were dropped due to the manipulation.

4.2 Hyperparameters and Validation Scores

Due to different window sizes used in the models different numbers of measurements were excluded. Table B.3 shows the corresponding support for the validation scores of the different models.

4.2.1 Feature-Based Models

The window size for all feature-based models was set to 60 seconds. The window step size was set to 30 seconds for all feature-based models as early results showed that overlapping the windows with that step size was a good trade off between accuracy and computing effort. When a sliding window with these parameters was applied, 919 (sub)samples were generated. Figure 4.4 shows the number of PDs and the sample count for each defect class in the training data when this sliding window was applied.

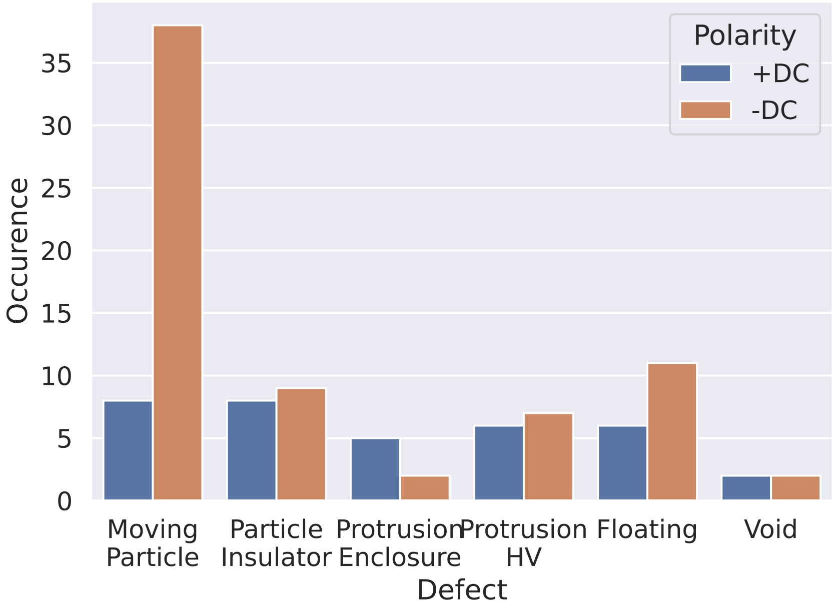


Figure 4.3 **Occurrence of the polarity in relation to the defect types**. This plot shows all defect measurements from the training set.

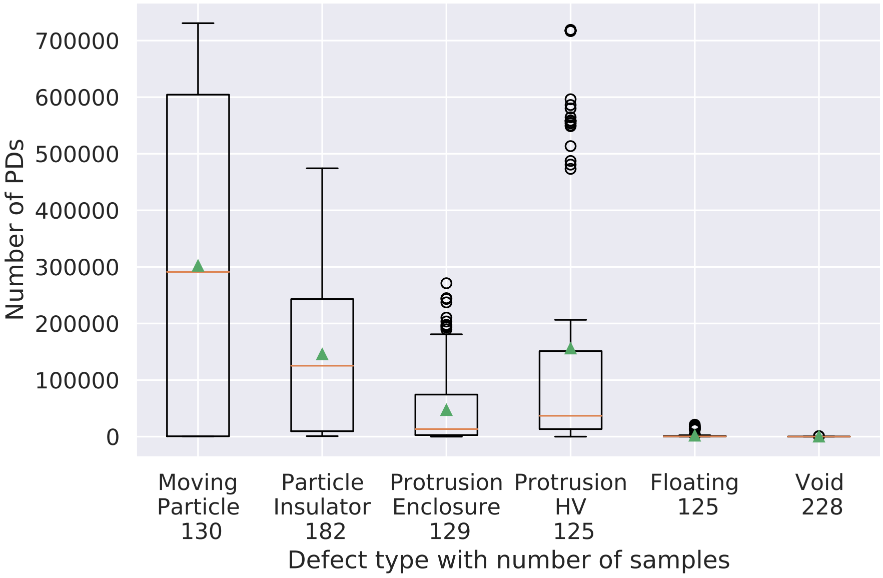


Figure 4.4 **Boxplot on number of PDs per defect class when using sliding window**. The sliding window has window size of 60 seconds and a step size of 30 seconds. The triangles in the boxplot represent the mean.

The tuned hyperparameters are described in Table B.1 in the appendix. Figure 4.5 shows the validation scores after doing hyperparameter tuning, the actual numbers are shown in Table B.2 in the appendix. The model with the overall highest validation score was an LGBM with AbsFip. The best model which made use of normalized PD values was RelFip in combination with MLP. The scores of that feature set were overall higher than those of OFip, which used absolute PD amplitudes. Tsfresh had the overall lowest validation scores. As the Tsfresh set consists in contrast to the other feature sets of automatically extracted and selected features the number of features was not manually set. 356 features were selected here. This was achieved by setting the feature selection significance level (FDR level) to 0.05 and the number of classes for which a feature needs to be significant to 3.

When comparing the classification algorithms, Figure 4.5 shows that LGBM performs overall best. In general, it can be seen that more complex classifiers such as LGBM or also MLP are better than simple classifiers such as k-NN. This effect is larger for the feature sets with more features, namely AbsFip, RelFip, and Tsfresh, than for the two smaller feature sets OFip and SFip.

4.2.2 1D-Resample Models

As the window sizes differed between the models of this group they are described together with the tuned hyperparameters in Table 4.1. Regarding the tuned hyperparameters the following should be highlighted: To keep the computational time reasonable the sampling rate was much lower for 1D-k-NN-DTW than the other two models. The highest feasible sampling rate of the LR-WEASEL model was a fourth of the sensors' actual rate. The WEASEL computation was memory intensive as a machine with 32 GB random-access memory (RAM) was necessary for the computation.

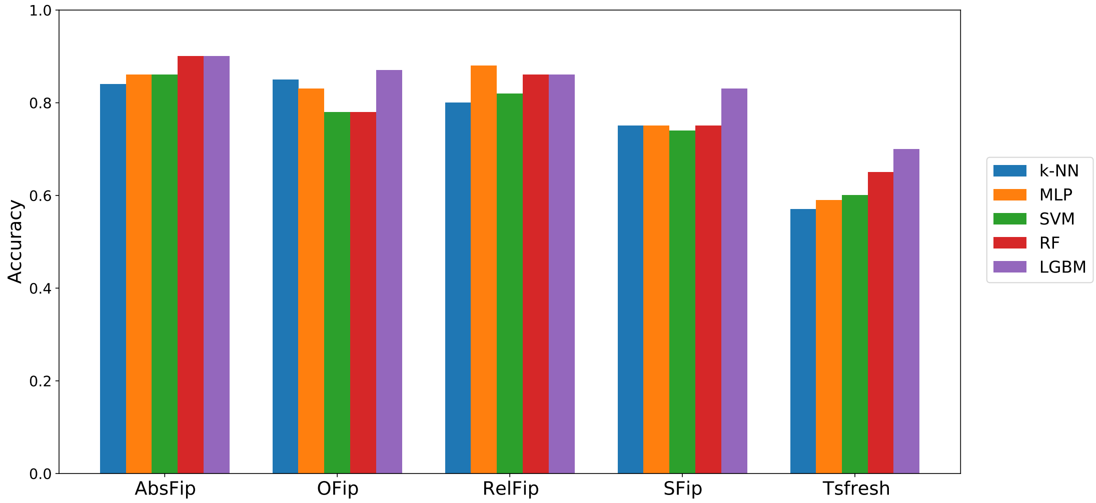
Table 4.2 shows the validation scores of the 1D-Resample models. The 1D-k-NN-DTW model achieved the highest validation score.

4.2.3 Raw Data Models

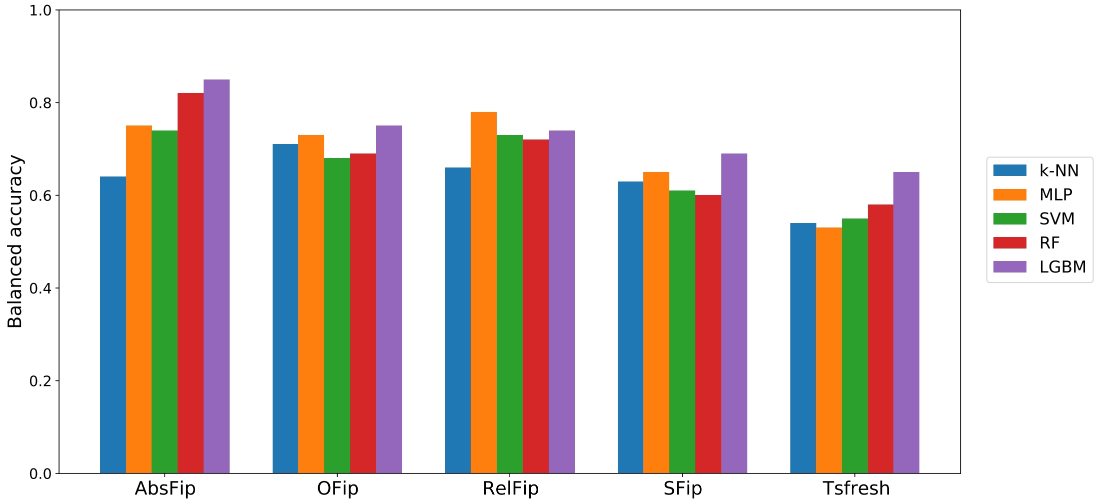
The sliding window parameters were equal for the three models of this group. The window size was set to 246 as the shortest measurement file in the training set was slightly larger. The window step size was set to the same value.

The hyperparameters can be seen in Table 4.3. The models used different undersampling strategies. Contrary to the other the models of this group the Raw-k-NN-Euclid model utilized Random Undersampling, which completely removed the class imbalance. That model also had a smaller gap between the BAC and ACC score than the other two, namely a difference compared of 0.08 compared to 0.17 (Raw-k-NN-DTW) and 0.14 (Raw-Pol-k-NN-DTW).

The validation scores are shown in Table 4.4. The best model was the one utilizing DTW without separating the samples by their polarity, namely Raw-k-NN-DTW.



(a) **Balanced accuracy**



(b) **Accuracy**

Figure 4.5 **Validation Scores of the tuned models for each feature set**. The two feature sets AbsFip and OFip rely on absolute PD amplitudes and the other on normalized values.

Table 4.1 **Tuned hyperparameters of 1D-Resample models**.

Model	**Hyperparameter**	**Tuned value**
1D-k-NN-DTW	Number of neighbors	15
	Weight neighbors	Distance
	RUS	Yes
	Window size	15 s
	Window step size	15 s
	Sampling rate	50 Hz
1D-k-NN-Euclid	Number of neighbors	1
	Weight neighbors	Uniform
	RUS	Yes
	Window size	60 s
	Window step size	30 s
	Sampling rate	20 000 Hz
LR-WEASEL	Sampling rate	5000 Hz
	Window size	60 s
	Window step size	30 s
	RUS	Yes
	WEASEL Windows	500, 2000, 10000, 40000
	LR parameter: Dual	Yes
	LR parameter: Solver	liblinear

Table 4.2 **Validation scores of the tuned 1D-Resample models**.

Model	**BAC**	**ACC**
1D-k-NN-DTW	**0.53**	**0.58**
1D-k-NN-Euclid	0.35	0.37
LR-WEASEL	0.23	0.31

Table 4.3 **Tuned hyperparameters of Raw Data models**.

Model	**Hyperparameter**	**Tuned value**
Raw-k-NN-DTW	Number of neighbors	1
	Weight neighbors	Uniform
	Sakoe-Chiba radius	32
	Undersample	First 12 windows
Raw-k-NN-Euclid	Number of neighbors	1
	Weight neighbors	Uniform
	RUS	Yes
Raw-Pol-k-NN-DTW	Number of neighbors	+DC: 1, −DC: 15
	Weight neighbors	Distance
	RUS	No

Table 4.4 **Validation scores of the tuned Raw Data Models**.

Model	**BAC**	**ACC**
Raw-k-NN-DTW	**0.62**	**0.79**
Raw-k-NN-Euclid	0.49	0.57
Raw-Pol-k-NN-DTW	0.52	0.66

4.3 Test Scores

Table 4.5 shows the test scores (BAC and ACC) on the four different test sets for the models with a high enough validation score. As the models utilized different window sizes a slightly different number of measurement files was used for computing the test scores (see Table B.4). In contrast to the feature-based models, the 1D-Resample and Raw Data model did not exclude a measurement file from the test set.

Only the Mixture dataset contained contrary to the other test sets all defect classes. The best score on this set was achieved by the LGBM-AbsFip model, which classified all measurements correctly. The second best score was achieved by the MLP-RelFip model with a balanced accuracy of 0.97. Slightly worse were the LGBM-SFip and LGBM-OFip model, which both misclassified one measurement more. The scores of the two models from the two other groups, 1D-k-NN-DTW, Raw-k-NN-DTW, are worse than these three, namely 0.75 and 0.83 balanced accuracy. For the Raw-k-NN-DTW model this means that it misclassified 2 measurements more than the MLP-RelFip model. Apart from the Tsfresh model, which has a score on this test that is lower than its validation score, the BAC score of each model on this test set is higher than the corresponding validation score, e.g. LGBM-SFip with a validation score of 0.69 and score of 0.91 on this test set. The accuracy scores are more similar, which are for LGBM-SFip 0.83 validation and 0.92 on this test set.

The two test sets, Miv-0.4nV and OS-V2, show that the absolute PD models perform overall worse than the models with normalized PD amplitudes. They show the highest differences between the scores of the Mixture and Mix-0.4nV set. The only exception is that the 1D-k-NN-DTW model has a big difference as well. For the OS-V2 dataset, which contains four measurements, all normalized models are better than the other two.

Overall the scores are lower on the CleanAir test set. Most models classify here none of the measurements of the protrusion on HV defect correctly (see Figure 4.8 and Figures B.1 to B.4). The two exceptions are the models MLP-RelFip and LGBM-SFip, which classified around half of these measurements correctly (see Figures 4.6 and 4.7).

4.4 Noise Data Classification

The validation scores for the LGBM-RelFip model including the noise data are: Balanced accuracy 0.82 and accuracy 0.89. Figure 4.9 shows the recall and precision on the

Table 4.5 **BAC and ACC of best models on the test sets**. The models are divided by their usage of normalized PD amplitudes. See Table B.4 for the number of samples the scores were calculated on. Balanced accuracy (BAC) and accuracy (ACC) are the same for OS-V2 as it only contains one class. *Metric is calculated on 4 instead of 3 samples

Model	**Mixture**		**Mix-0.4nV**		**OS-V2**		**CleanAir**	
	BAC	ACC	BAC	ACC	BAC	ACC	BAC	ACC
Absolute PD values								
LGBM-AbsFip	1	1	0.60	0.67	0.33	0.33	0.46	0.58
LGBM-OFip [17]	0.92	0.92	0.76	0.76	0	0	0.42	0.53
Normalized PD values								
LGBM-SFip [20]	0.91	0.92	0.79	0.81	**1**	**1**	**0.74**	**0.79**
MLP-RelFip	**0.97**	**0.96**	**0.96**	**0.95**	**1**	**1**	0.67	0.73
LGBM-Tsfresh [63]	0.49	0.67	0.43	0.67	**1**	**1**	0.50	0.63
1D-k-NN-DTW	0.75	0.84	0.42	0.63	**0.75***	**0.75***	0.46	0.6
Raw-k-NN-DTW	0.83	0.88	0.75	0.82	0.5	0.5	0.42	0.55

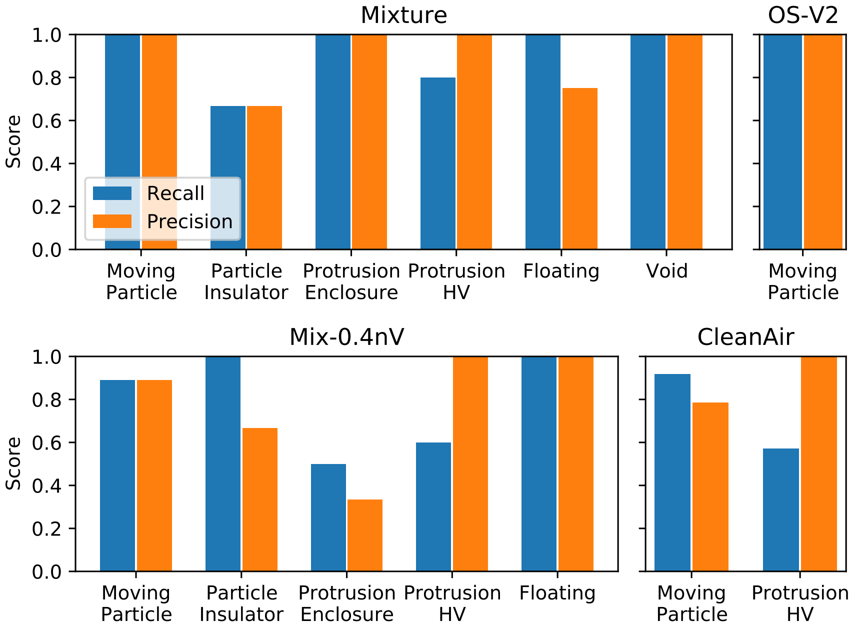


Figure 4.6 **Recall and precision of the LGBM-SFip model on the four test sets**.

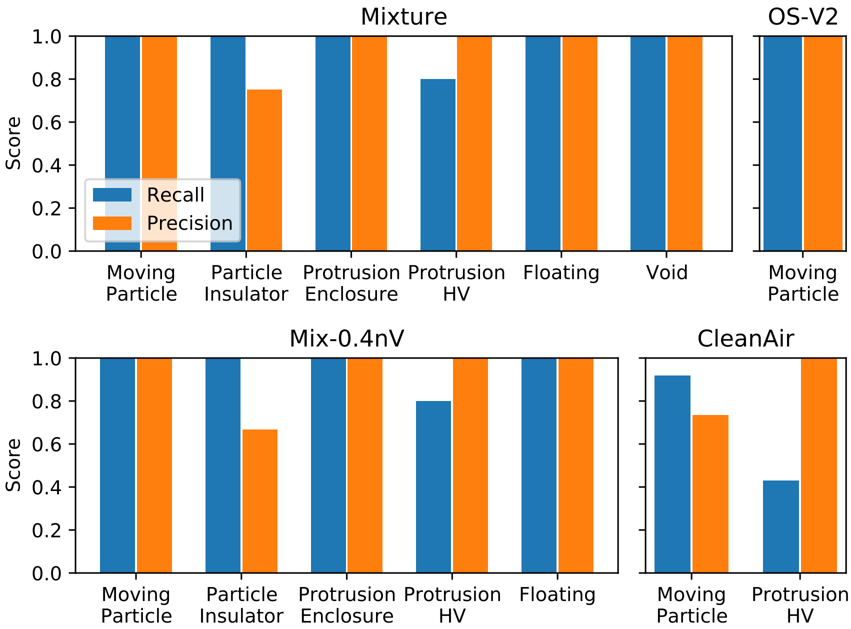


Figure 4.7 **Recall and precision of the MLP-RelFip model on the four test sets.**

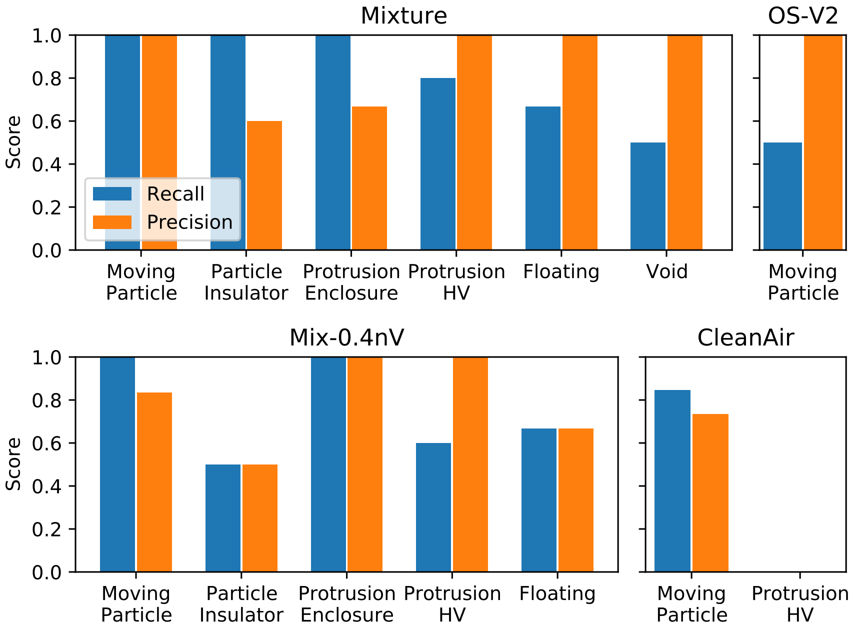


Figure 4.8 **Recall and precision of the Raw-k-NN-DTW model on the four test sets.**

test sets, Table B.5 shows the corresponding BAC and ACC scores and Figure B.5 the confusion matrix. All of the five noise measurements were correctly separated from the defects. Apart from the noise data, the test results of this model look similar to the results of the MLP-RelFip which was trained without the noise data (see Figure 4.7 for comparison). A notable difference between the two is that the model with the noise data misclassified all protrusion on HV measurements of the CleanAir test set.

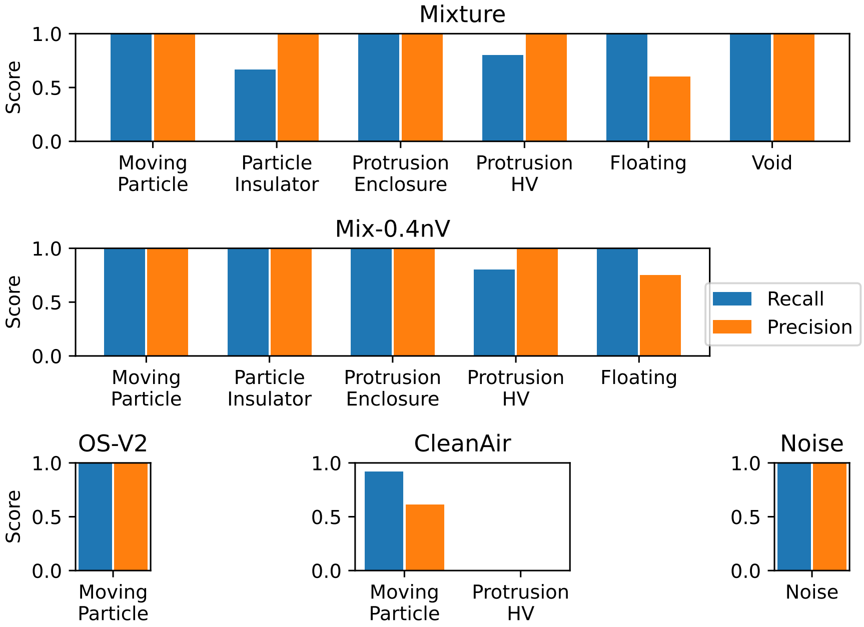


Figure 4.9 **Recall and precision on the test sets of the model trained also with the noise data**. The model is LGBM-RelFip, which was trained also on the noise data.

4.5 Prediction Time Comparison

The average prediction time varies highly between different models (see Table 4.6). LGBM-SFip model achieved the best prediction time, namely 13 seconds. The other feature-based models are in the same order, 21 and 25 seconds. One of the two measured DTW models is also in this magnitude, Raw-k-NN-DTW with 20 seconds. The outlier is the other DTW model, which is an order of 100 slower.

4.6 Feature Attribution Analysis

The feature attribution analysis was performed on the LGBM-RelFip model (see Section 3.5.3). Figure 4.10 shows the 10 features with the highest summarized SHAP values (see Figure B.6 for all features). According to these values the median time difference between the partial discharges (Δt-Median) is the most important feature for the model,

Table 4.6 **Average prediction times in seconds of best models on the Mixture test set**. As these computations were performed on a personal computer the results should mainly be seen as relative to each other.

Model	Avg. Prediction Time (Sec)
Absolute PD values	
LGBM-AbsFip	21.1
LGBM-OFip [17]	25.2
Normalized PD values	
LGBM-SFip [20]	13.2
MLP-RelFip	21.0
1D-k-NN-DTW	2011.1
Raw-k-NN-DTW	19.5

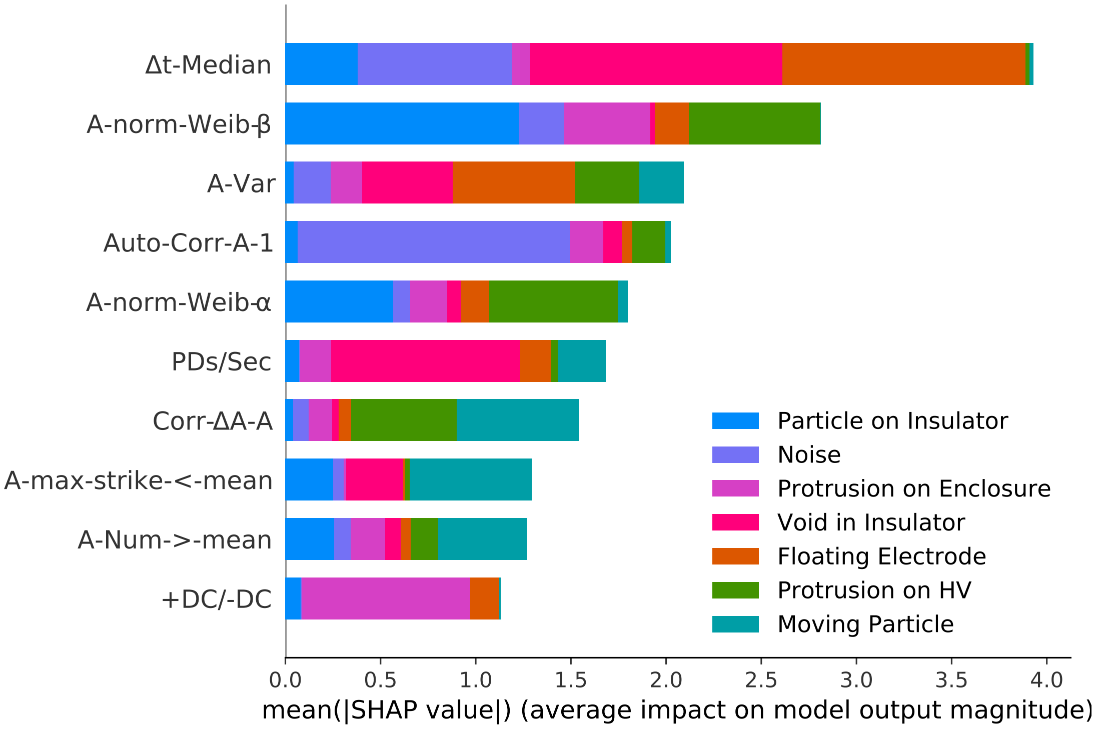


Figure 4.10 **SHAP feature attribution of top 10 features of the LGBM-RelFip model**. The model was trained on the complete training data including noise and SHAP was applied on the same data. Figure B.6 shows the feature attribution of all features.

which is much higher than the related feature PDs/Sec. Overall, most of the depicted features were significant for the majority of the classes. This was not the case for the polarity feature +DC/−DC, which only contributed to the particle on insulator, protrusion on enclosure, and floating electrode defect types.

5 Discussion

In this thesis, three different groups of machine learning approaches for the classification of UHF partial discharge defect measurements in gas-insulated high voltage DC systems were compared. These groups included feature-based models and non-feature-based time series classifiers, the 1D-Resample and Raw Data models. When comparing the validation scores of the feature-based models with each other, it was observed that more complex classifiers such as LGBM or MLP can better utilize the extracted features. Consequently, the models with the overall highest test scores were based on extracted features using normalized PD values (MLP-RelFip followed by LGBM-SFip). Among all models, two utilized absolute and the others normalized PD amplitudes. It was found that the models with absolute values were superior on the test which was from its properties equal to the training set. However, when the test set was modified or measured with different settings, models based on normalized values had a lower drop in performance. Thus, these models might be more robust to changes. Additionally, it was found that the time series classification approaches, as utilized in the Raw Data models group, might be worth exploring in future works. These models have the advantage that they do not rely on manual feature extraction. While feature-based models still outperformed these models, the difference could be explained by few samples.

5.1 Generalization to Other Settings

As noted by [22] one drawback of most related studies is that the test set is from the same setup as the training data. A problem is that other (on-site) measurements might be done on different settings such as different gas pressures, gas mixtures, measurement equipment, or GIL size. Hence, the test set might not represent the underlying problem [22]. To deal with this, test data consisting partially of measurements conducted in other settings was used in this thesis. As expected, the present study shows an indication that normalized PD amplitudes generalize better to other settings than absolute ones. Two of the models here, LGBM-AbsFip and LGBM-OFip, utilized absolute PD amplitudes. There is also at least one other GIS study that utilized absolute PD values [71] (with AC). In this thesis, the two models with absolute PD amplitudes (AbsFip and OFip) achieved overall higher test scores on the Mixture test set, which is the set that has the same properties as most of the training data. The two models performed overall worse on the manipulated test set and the two test sets containing measurements conducted under different settings. Contrary to that, the models with normalized PD amplitudes did not show such a performance decrease, especially on the datasets Mix-0.4nV and OS-V2. Other related studies also avoid the absolute PD information such as [72; 73] normalizing the amplitudes as well or [20] avoiding features which depend on these. Accordingly, Schober and Schichler state that absolute values "*can differ between different PD measurement systems and methods*"[20].

The models operating with normalized PD values generalized well to a different extent depending on the test set. Whereas the models showed overall good performance on the Mix-0.4nV and the OS-V2 dataset, the performance on CleanAir dataset was worse as most models misclassified all the protrusion on HV measurements of that dataset. This might relate to the different behavior of the protrusion on HV defect for CleanAir compared to other gases such as SF_6 [32]. This is corroborated by the overall worse performance for this specific defect and dataset in my findings (see Figures B.1 to B.4 and 4.6 to 4.9). However, it remains unexpected that the models MLP-RelFip and LGBM-SFip were still able to classify around half of these protrusion measurements correctly (see Figures 4.6 and 4.7). Future work could investigate whether there are reasons for this and, if so, what they are. In general, one could deduce from these results that models generalize only to certain variations like slightly different measuring equipment. Hence, this underlines the importance that the data should consist of measurements from various settings to generalize better [22].

5.2 Feature-Based versus Other Classifiers

A comparison between the feature-based algorithms and the other TSC algorithms, more specifically the algorithms utilizing DTW, shows that the feature-based models achieved higher validation and test scores. This suggests that feature-based algorithms are still preferred for this kind of problem. A potential explanation for that could be the characteristics of the dataset. The dataset consists for some classes of relatively few cases and in general of long time series which makes time series classification hard according to [49]. Nevertheless, the Raw-k-NN-DTW model misclassified only a few measurements more than the feature-based ones. This gives it the potential to be a baseline not only in the TSC community [49], but also in this domain. Furthermore, it could be utilized in an ensemble classifier together with a feature-based model as ensembles are generally highly competitive in TSC [49] and might improve accuracy here. Apart from possible application of DTW, its main advantage is that no feature engineering is necessary. A problem with DTW can be its time consuming computation [49; 56]. Here the results showed that the prediction time is similar to the feature-based models for Raw-k-NN-DTW as the sample length was kept reasonably short and the quantity of samples stored by the k-NN classifier was limited because undersampling was used.

Because the DTW models achieved already reasonable results, other TSC algorithms such as COTE [59] or the improvement HIVE-COTE [74] are likely to achieve better results as they perform better than DTW on the publicly UCR/UEA TSC datasets. A problem with the suggested classifiers is that they have high computational complexities [74; 75]. A suitable candidate might be Rocket, the RandOm Convolutional KErnel Transform, coupled with a ridge regression classifier. It is much faster than e.g. HIVE-COTE in terms of training and prediction on the UCR/UEA datasets [75].

Apart from the mentioned TSC algorithms Deep Learning might be an alternative for future work. As Deep Learning was very successful for image classification it is being applied to time series as well [55; 76]. Jiang compared Deep Learning models to distances measures such as DTW on the aforementioned UCR/UEA datasets [76]. They found two properties of datasets where Deep Learning performed better than the compared distance measures: First, it worked better for longer time series, which is normally the case for PD measurements due to the high sampling rate [22]. Second, Deep Learning achieved better results on datasets with more training samples. This underlines the fact that Deep Learning models need much data to work well, which is often not the case for PD measurements [22]. Transfer Learning can be a solution in such cases with insufficient training data [22; 77]. Here, a Deep Learning model is trained on a source dataset and then the learned features, also called weights, are transferred to a target dataset for fine-tuning [77; 78]. Transfer Learning can improve the accuracy, the model's generalization capabilities, and the time to train a model [77; 78]. Ismail Fawaz et al. applied Transfer Learning on all UCR/UEA datasets and showed that by choosing a good source dataset the performance improved considerably [78]. A good source dataset is often a dataset, which is similar to the target dataset. To guide this choice of finding a similar dataset Ismail Fawaz et al. employed DTW as an inter-dataset similarity measure [78]. Related to the present thesis, future work could find a similar dataset where a suitable Deep Learning model was trained on and could fine-tune that on the data here. As noted by [22, Sec. 6] there exists though no comprehensive open-access dataset yet for PD measurements.

Additionally to Transfer Learning, an option might be to utilize more training data beyond performing more experiments by generating synthetic data [22, Sec. 5]. A related study applied e.g. Generative Adversarial Networks to create artificial samples and used these to improve the PD defect classification in HVAC cables [79].

5.3 Comparison of Feature Sets

Among the feature-based models the two proposed models LGBM-AbsFip and MLP-RelFip achieved the highest validation scores and also test scores on the Mixture test set. This is not surprising as the feature engineering for these feature sets was contrary to the other feature sets done on the training data used within this thesis. OFip was designed on a much smaller subset of the data used in the present work. The feature set SFip was engineered on different data measured with different equipment and for different defect classes [20]. Nevertheless, we could replicate the findings by [20] to a certain extent. The scores were lower here than what was reported by [20], which could indicate a lack in generalization to the present dataset. The same may of course happen, if classification algorithms are trained with RelFip on different PD measurements such as the data used by [20].

When comparing the classification algorithms of the feature-based models it was found that the more complex algorithms, MLP and LGBM, performed best. This might be because the influence of certain features such as the polarity is complex and might not be handled well by simple algorithms such as k-NN. Gradient boosting models like LGBM have been used rarely in related studies. Only one of the studies surveyed by [22] used a Gradient Boosting model, namely [80]. It was furthermore also utilized by [81], where it was slightly better than the used RF model. Since the LGBM classifier performed especially well in this study gradient boosting models might be a good candidate for other related studies relying on feature engineering.

5.4 Contribution of Certain Features

To make the decision making process of the RelFip feature set more explainable the importance of its features was analyzed. Unsurprisingly, the feature attribution showed that the median of the time difference between consecutive partial discharges was the most important feature (see Section 4.6). This fits to the fact that the repetition rate differs considerably between some defect classes [10; 11] (see Section 2.1.2). This median might be superior to the related feature PDs/Sec due to its robustness against outliers. Another utilized feature was the polarity. The idea was that the classifiers might detect different patterns for the two polarities. According to the feature attribution, this likely did not work since the polarity contributed only towards the correctness of defect classes with imbalanced polarities in the training set, namely protrusion on enclosure, particle on insulator, and floating electrode (see Figure 4.3). In future work, this feature should be balanced when training a model to actually be able to make use of it.

Apart from the polarity, a feature which could not be utilized was the test voltage. Within future work it might improve classification performance to utilize the voltage level e.g. in relation to the partial discharge inception voltage because the PD activity is influenced by the applied voltage and specifically its changes [10; 11].

5.5 Sliding Window

To have a sufficient number of samples for each class and to ensure their equal duration, the sliding window technique was utilized for all models. This means, the actual measurements were subsampled like it was also done by [12] for AC-GIS measurements. However, some of the measurements examined in this thesis showed global pattern (see Figure 4.2). Due to the sliding window these pattern could not be used. Therefore it might be useful to analyze these global patterns in future work, when more data is available for every class.

Furthermore, an improvement can be to also allow measurements with a duration shorter than 60 seconds, if they contain enough partial discharges. The different duration should not be a problem because e.g. no feature in RelFip depends (directly) on the duration.

5.6 Other Measuring Techniques

Apart from improving the ML aspects, the measuring technique can be adapted as well. This thesis relied on the UHF method to detect partial discharges. As mentioned in Section 2.1.3 there are alternatives such as the conventional method or optical measuring. The optical method works by installing optical sensors into the gas-insulated system. These sensors record then the light signals caused by PDs. An advantage of this method is that it can not be disturbed by electromagnetic interfering signals [22, Sec. 2.4]. Thus, ML models could be enhanced by combining the UHF with the optical method as it was done by [82] for the defect classification in an AC-GIL.

All measurements used in the present thesis had one UHF sensor as input. Multiple sensors were used within the experiments, but they were not combined for the analysis [10]. All of the discussed ML models could in general utilize multiple sources of input. Hence, it could increase the correctness of the models to rely on multiple sensors. As mentioned in Section 2.1.4 multiple sensors also improve the detection of noise signals. Multiple sensors are further necessary for the localization of a defect. Li et al. used UHF sensors to localize defects in an AC-GIS with the help of time-frequency analysis and SVM [83].

5.7 Limitations

This study has several limitations. First, apart from two small datasets coming from a different gas-insulated system the data was only from one system. This might be a problem as measurements on other setups might look significantly different. Furthermore, the ambient conditions can differ between on-site and the laboratory measurements used in this thesis. This could mean that the noise signals might differ such that they are not classified as noise. This kind of limitation was also noted by [22, Sec. 5] about other related studies. Hence, generalization to other setups is limited.

Second, the analysis performed here was done only for single defects. Multiple defects can however exist simultaneously in a gas-insulated system [35, p. 57]. It is left for future work to tackle such cases.

Third, some classes contained only very few measurements. As surveyed on other related studies [22, Sec. 5], this caused class imbalance in the data (see Section 4.1).

Although measures such as appropriate evaluation metrics, sample weighting, and undersampling strategies were used [22], the classes with more measurements such as the free moving particle achieved overall higher scores. As a consequence, this resulted in high gaps between the accuracy and balanced accuracy score for some models. To overcome this, more sophisticated undersampling or oversampling methods than Random Undersampling could be utilized in future work.

Fourth, computational resources were a limiting factor. Certain algorithms, namely WEASEL, partially DTW, and Tsfresh, required significant memory and/or processing power. Thus, tuning and computations were limited resulting in perhaps lower scores than possible. Due to this limitation other TSC models such as the COTE ensemble and also Deep Learning were not considered.

6 Conclusion

In this thesis, machine learning algorithms were applied to the PD analysis of gas-insulated DC systems, in particular the classification of defect types using UHF measurements. It was shown that current feature-based approaches coupled with modern classification algorithms can achieve good results for measurements recorded under laboratory conditions. As generalization to different settings is important for such approaches, it was shown that normalization is superior to the usage of absolute PD amplitudes. Furthermore, other time series classification algorithms were applied. They did however not outperform the feature-based models, but as discussed in Section 5.2 could still function as a baseline since they do not require feature engineering. The best classifier utilized the self-engineered feature set RelFip, which may have only outperformed the related sets because it was engineered on the data used in this thesis. Additionally, it was shown for a RelFip model that noise can be successfully separated from the defect measurements. Finally, to make the decisions of that model more explainable the importance of its features for the prediction was analyzed (Section 5.4).

As the examples mentioned in Section 2.1.5 show only relying on AC measuring during commissioning tests might not be sufficient for gas-insulated DC systems. Therefore, in contrast with recommendations by CIGRÉ [14, Sec. 6.2.2; 15, Sec. 5.1], utilizing the feature-based approach could be a possible extension to the prevalent PD measurements with AC voltage. To make use of this approach the measurements need to be performed as in [10] and the model should be trained on an enhanced dataset. Besides utilizing the feature-based approach within commissioning tests, this thesis is as well a step towards reliable PD monitoring of DC systems. However for monitoring, a current shortcoming of the presented approach is that fewer partial discharges can occur than are needed for the analysis.

In future work, the ML models developed in this work should be applied to a bigger, more balanced, and ideally more heterogeneous dataset. In addition, the accuracy might increase by adapting the classifiers themselves and not the data, e.g. by modifying the sliding window method. Instead of improving the utilized models, also more advanced TSC algorithms or even Deep Learning could be employed (Section 5.2). Apart from enhancing the ML aspects of the PD analysis, the measurement technique can be improved as well. As discussed in Section 5.6 it may improve the PD analysis to feed classifiers with the data from multiple sensors or to combine the UHF with the optical method.

Acknowledgments

First, I would like to thank Prof. Plath for the opportunity to write my master's thesis in his department. Furthermore, I am grateful to Prof. Albayrak for co-supervising this thesis.

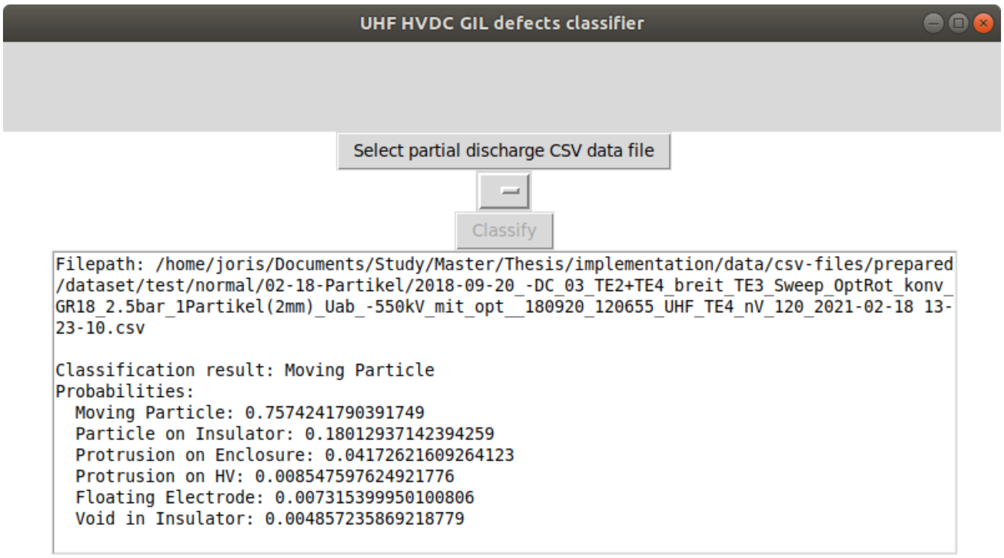
I also want to thank Moritz Geske for his regular and constant mentorship from the beginning of this thesis. I am especially grateful for his extensive explanations and feedback regarding the electrotechnics aspects of the topic, which helped me gain knowledge about those. Furthermore, special thanks belong to Marcus Voß, who mentored me on the machine learning aspects of this work. I am very thankful for his helpful suggestions to make use of certain techniques and his valuable feedback.

I thank my partner Tabea Kossen for sharing her knowledge on machine learning, her critical reviews, and final proofreading, which substantially improved the quality of this work. Furthermore, I like to thank my father Jan Clement for additional final proofreading. Last but not least I want to thank my friends and parents for their continuous support.

Appendices

A Methods

A.1 Graphical User Interface


UHF HVDC GIL defects classifier
Select partial discharge CSV data file
Classify

```
Filepath: /home/joris/Documents/Study/Master/Thesis/implementation/data/csv-files/prepared
/dataset/test/normal/02-18-Partikel/2018-09-20_-DC_03_TE2+TE4_breit_TE3_Sweep_OptRot_konv_
GR18_2.5bar_1Partikel(2mm)_Uab_-550kV_mit_opt__180920_120655_UHF_TE4_nV_120_2021-02-18 13-
23-10.csv

Classification result: Moving Particle
Probabilities:
  Moving Particle: 0.7574241790391749
  Particle on Insulator: 0.18012937142394259
  Protrusion on Enclosure: 0.04172621609264123
  Protrusion on HV: 0.008547597624921776
  Floating Electrode: 0.007315399950100806
  Void in Insulator: 0.004857235869218779
```

Figure A.1 **Screenshot of GUI**.

A.2 Weibull Distribution

The Weibull distribution is a continuous probability distribution consisting of a scale parameter α and a shape parameter β [17; 20]. It is defined as:

$$f(x) = \frac{\beta}{\alpha}(\frac{x}{\alpha})^{\beta-1}e^{-\frac{x}{\alpha}^{\beta}}, x \geq 0 \tag{A.1}$$

Within this work, the parameters α and β were estimated using the scipy library [84].

A.3 Features of Reimplemented Sets

Table A.1 **Description of the features used by the feature sets OFip [17] and SFip [20]**. A stands for the PD amplitude and Δt for the time differences between the PD events.

(a) OFip feature set

Index	Description
1	Mean of A
2	Maximum of A
3	Coefficient of variation of A
4,5	Weibull parameter α, β of Weibull fit to A with location parameter fixed to 0.
6	Mean of ΔA
7	Variance of ΔA
8	Kurtosis of ΔA
9	Weibull parameter α of Weibull fit to ΔA with location parameter fixed to 0.
10	Median Δt
11	Pearson correlation between ΔA and A
12	Pearson correlation between A_n and A_{n+1}

(b) SFip feature set

Index	Description
1	Variance of A
2	Skewness of A
3	Kurtosis of A
4,5	Weibull parameter α, β of Weibull fit to ΔA distribution with location parameter fixed to 0.
6	Maximum of Δt
7	Mean of Δt
8	Minimum of Δt
9	Variance of Δt
10	Skewness of Δt
11	Kurtosis of Δt
12,13	Weibull parameter α, β of Weibull fit to Δt distribution with location parameter fixed to 0.
14	PDs per second

A.4 Feature Tuning

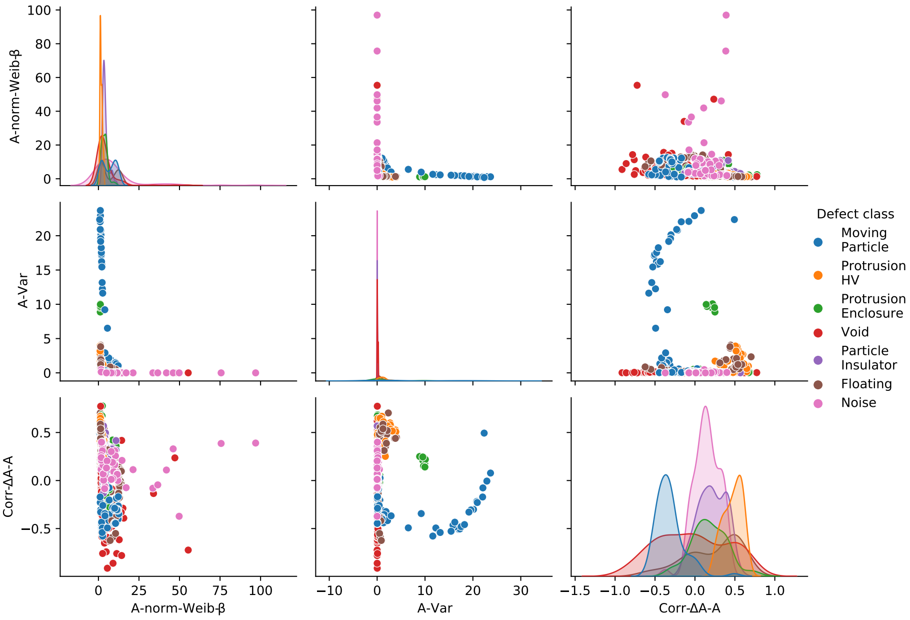


Figure A.2 **Visualization of 3 features in a pair plot**. These plots were used to visualize, select and tune the features. The plots were generated with Seaborn [64]

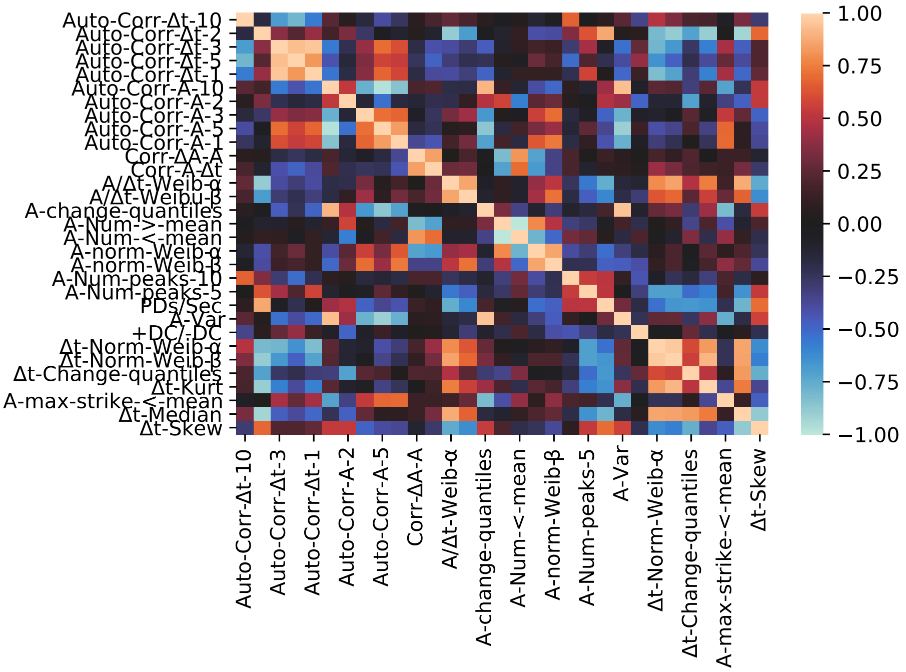


Figure A.3 **Correlation matrix to exclude redundant features**. These plots were used to visualize, select and tune the features.

B Results

B.1 Tuned Hyperparameters

Table B.1 **Optimal hyperparameters of feature-based classifiers**. The tuned hyperparameters are shown for each combination of classification algorithm (Algo) and feature set such as RelFip. The values were tuned among the options described in Table 3.5. Dist. stands for by distance.

Algo	**Hyperparameter**	**AbsFip**	**OFip**	**RelFip**	**SFip**	**Tsfresh**
RF	Minimum samples per leaf node	2	30	2	5	5
	Perform Bootstrapping	No	Yes	Yes	Yes	Yes
	Balance class weights	Yes	No	Yes	No	Yes
LGBM	Feature fraction per tree	0.8	1.0	1.0	1.0	0.8
	Min. gain when splitting a node	0.0	0.0	0.1	0.0	0.0
	Max. bins for feature values	63	63	127	127	63
	Max. leaf nodes per tree	30	30	30	30	30
	Balance class weights	Yes	Yes	Yes	Yes	Yes
k-NN	Number of neighbors	5	5	5	10	5
	Weight neighbors	Dist.	Uniform	Dist.	Dist.	Uniform
SVM	Decision function shape	OvR	OvR	OvR	OvR	OvR
	Type of kernel	RBF	Linear	RBF	RBF	Linear
	Balance class weights	Yes	Yes	Yes	No	Yes
	Feature scaling	Z-score	Z-Score	Z-Score	Z-Score	MinMax
MLP	Batch size	1	1	10	1	1
	Dropout (Same for each layer)	0.05	0.2	0.0	0.0	0.05
	Neurons per layer	(20)	(20)	(20)	(20)	(20,10)
	Balance class weights	No	No	No	No	No
	Epochs to train	50	150	100	100	150

B.2 Validation Scores

Table B.2 **Validation Scores of the tuned models for each feature set**. Balanced accuracy (BAC) is used as the validation metric. Two of the feature sets use absolute PD amplitudes and the other normalized values.

(a) **Absolute PD amplitudes**. *OttAlgo did not use sliding window.

Feature Set	**Classifier**	**BAC**	**ACC**
AbsFip	k-NN	0.64	0.84
	MLP	0.75	0.86
	SVM	0.74	0.86
	RF	0.82	0.90
	LGBM	**0.85**	**0.90**
OFip	k-NN	0.71	0.85
	MLP	0.73	0.83
	SVM	0.68	0.78
	RF	0.69	0.78
	LGBM	**0.75**	**0.87**
	OttAlgo*	0.39	0.39

(b) **Normalized PD amplitudes**

Feature Set	**Classifier**	**BAC**	**ACC**
SFip	k-NN	0.63	0.75
	MLP	0.65	0.75
	SVM	0.61	0.74
	RF	0.60	0.75
	LGBM	**0.69**	**0.83**
RelFip	k-NN	0.66	0.80
	MLP	**0.78**	**0.88**
	SVM	0.73	0.82
	RF	0.72	0.86
	LGBM	0.74	0.86
Tsfresh	k-NN	0.54	0.57
	MLP	0.53	0.59
	SVM	0.55	0.60
	RF	0.58	0.65
	LGBM	**0.65**	**0.70**

Table B.3 **Support of models for the validation score**. Support stands for the number of samples the scores shown in Table 4.5 were computed on.

Model	Support
feature-based models	90
1D-k-NN-DTW	101
1D-k-NN-Euclid	90
LR-WEASEL	90
Raw Data models	104

B.3 Test Scores

Table B.4 **Support of best models on the test sets**. Support stands for the number of samples the scores shown in Table 4.5 were computed on.

Model	Mixture	Mix-0.4nV	OS-V2	CleanAir
Absolute PD values				
AbsFip	24	21	3	19
OFip [17]	24	21	3	19
Normalized PD values				
SFip [20]	24	21	3	19
RelFip	24	21	3	19
Tsfresh	24	21	3	19
1D-Resample	25	22	4	20
Raw Data	25	22	4	20

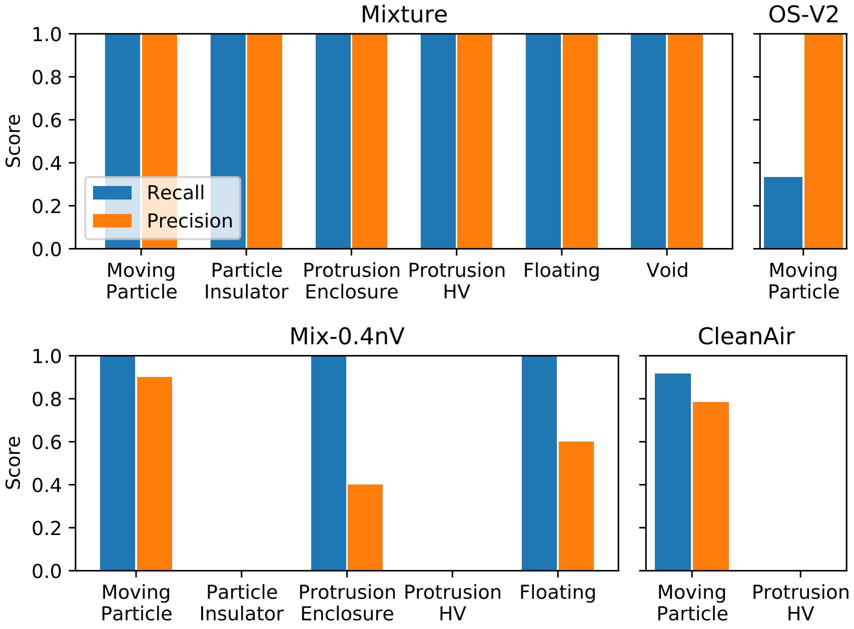


Figure B.1 **Recall and precision of the LGBM-AbsFip model on the four test sets**.

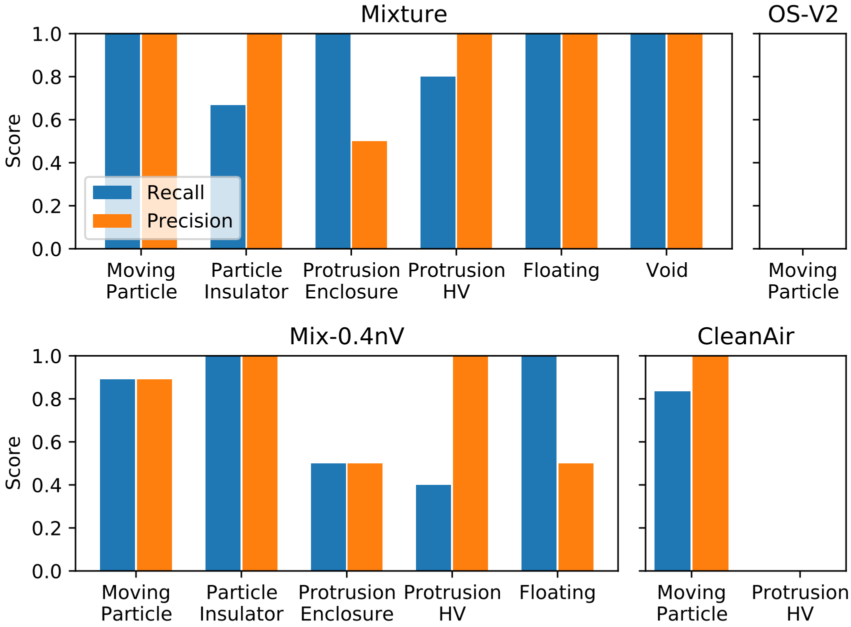


Figure B.2 **Recall and precision for the LGBM-OFip model on the four test sets.**

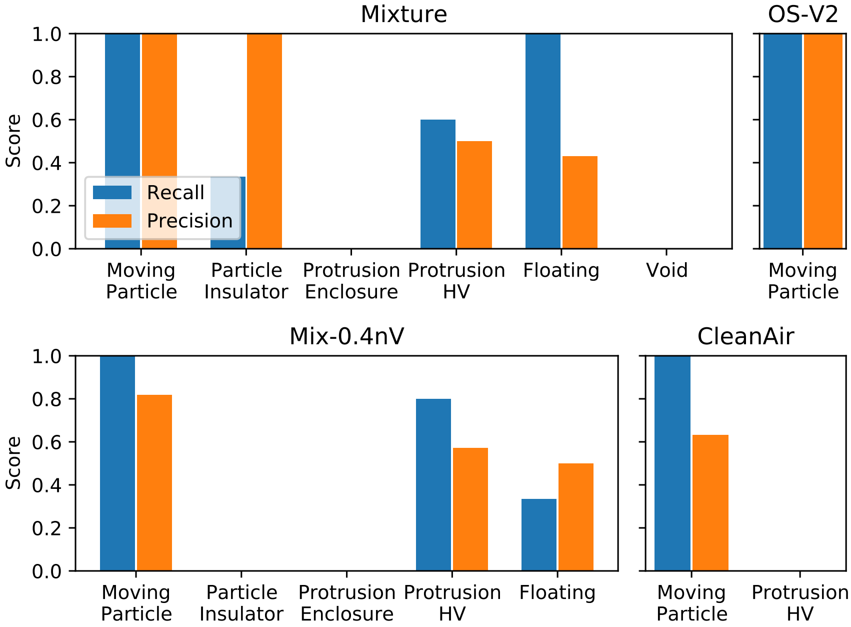


Figure B.3 **Recall and precision for the LGBM-Tsfresh model on the four test sets.**

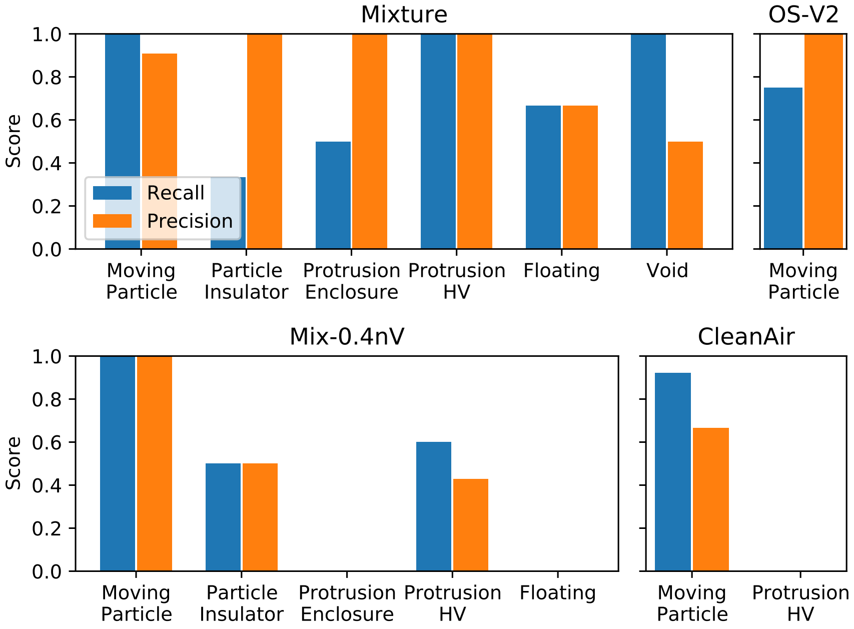


Figure B.4 **Recall and precision of the 1D-k-NN-DTW model on the four test sets.**

B.4 Noise Data Classification

Table B.5 **BAC and ACC on the test sets of the model trained also with the noise data**. The model is LGBM-RelFip, which was also trained with the noise data. BAC stands for balanced accuracy and ACC for accuracy

Mixture		Mix-0.4nV		OS-V2		CleanAir		Noise	
BAC	ACC	BAC	ACC	BAC	ACC	BAC	ACC	BAC	ACC
0.91	0.92	0.96	0.95	1	1	0.46	0.58	1	1

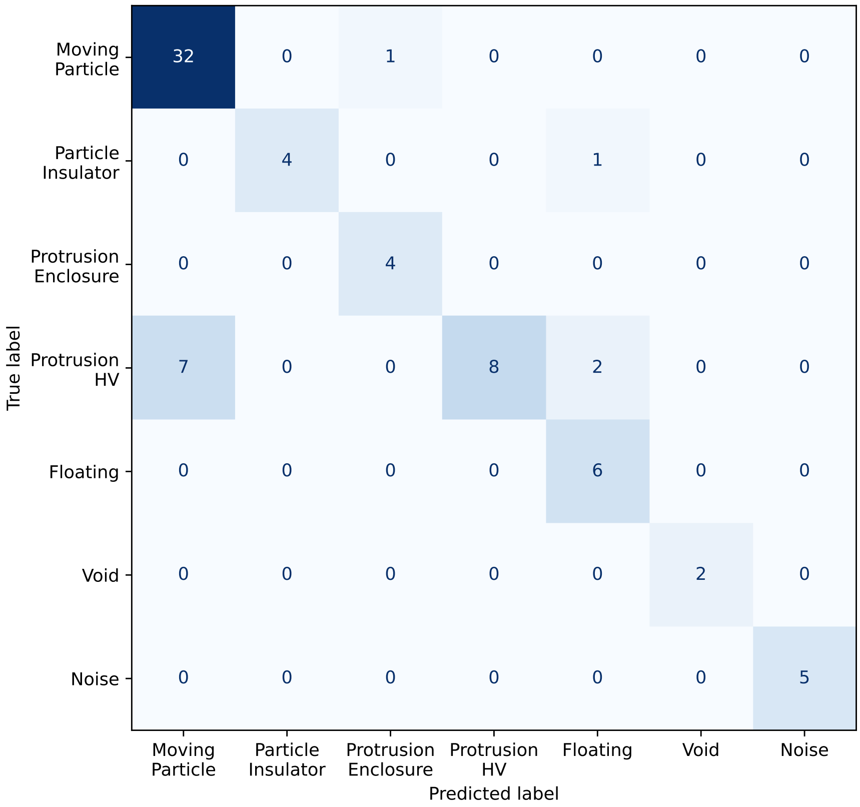


Figure B.5 **Confusion matrix on the test sets of the model trained also with the noise data**. The model is LGBM-RelFip, which was also trained with the noise data. Confusion matrix was calculated over all test sets, including the noise test set, combined.

B.5 Feature Attribution Analysis

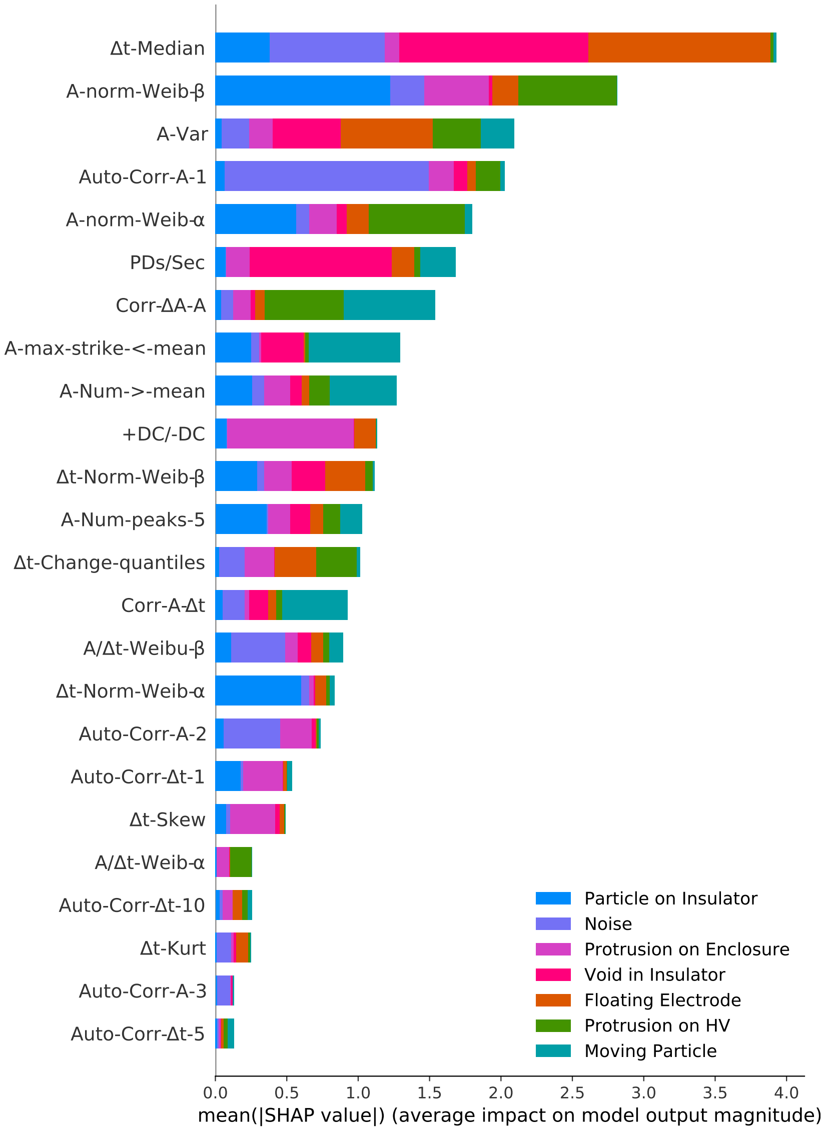


Figure B.6 **Feature attribution of all features of the LGBM-RelFip model**. Plot is based on the same model and data as Figure 4.10, which shows only the top 10 features.

Bibliography

[1] International Energy Agency (IEA). “Chapter 2: Energy Sector Investment to Meet Climate Goals.” In: International Energy Agency (IEA) and International Renewable Energy Agency (IRENA). *Perspectives for the Energy Transition: Investment Needs for a Low-Carbon Energy System.* 2017, pp. 51–120.

[2] J. Dorn et al. “Transformation of the Energy System in Germany - Enhancement of System Stability by Integration of innovative Multilevel HVDC in the AC Grid.” In: *International ETG-Congress 2013; Symposium 1: Security in Critical Infrastructures Today.* 2013, pp. 1–6.

[3] T. Magier, M. Tenzer, and H. Koch. “Direct Current Gas-Insulated Transmission Lines.” In: *IEEE Transactions on Power Delivery* 33.1 (Feb. 2018). Conference Name: IEEE Transactions on Power Delivery, pp. 440–446. ISSN: 1937-4208. DOI: 10.1109/TPWRD.2017.2716182.

[4] A. Gaviano, K. Weber, and C. Dirmeier. “Challenges and Integration of PV and Wind Energy Facilities from a Smart Grid Point of View.” In: *Energy Procedia* 25 (2012), pp. 118–125. ISSN: 18766102. DOI: 10.1016/j.egypro.2012.07.016.

[5] K. Meah and S. Ula. “Comparative evaluation of HVDC and HVAC transmission systems.” In: *2007 IEEE Power Engineering Society General Meeting.* IEEE, 2007, pp. 1–5.

[6] D. Jovcic. *High Voltage Direct Current Transmission.* John Wiley & Sons, 2019. 535 pp. ISBN: 9781119566540.

[7] G. Chen et al. “Review of high voltage direct current cables.” In: *CSEE Journal of Power and Energy Systems* 1.2 (June 2015). Conference Name: CSEE Journal of Power and Energy Systems, pp. 9–21. ISSN: 2096-0042. DOI: 10.17775/CSEEJPES.2015.00015.

[8] H. Koch. *Gas Insulated Transmission Lines (GIL).* John Wiley & Sons, Dec. 12, 2011. 385 pp. ISBN: 978-0-470-66533-6.

[9] H. Koch. *Gas Insulated Substations.* Ed. by H. Koch. IEEE Press and John Wiley & Sons, 2014. ISBN: 978-1-118-57072-2.

[10] M. Geske et al. “Assessment of typical defects in gas-insulated DC systems by means of Pulse Sequence Analysis based on UHF partial discharge measurements.” In: *VDE High Voltage Technology 2020; ETG-Symposium.* 2020, pp. 1–8. URL: `https://ieeexplore.ieee.org/abstract/document/9275509` (visited on 06/12/2021).

[11] A. Pirker, B. Schober, and U. Schichler. “PD Monitoring of HVDC GIS/GIL.” In: CIGRE Chengdu 2019 Symposium ; Conference date: 20-09-2019 Through 26-09-2019. Sept. 2019. URL: `https://e-cigre.org/publication/SYMP_CHEN_2019-symposium-chengdu-2019` (visited on 07/14/2021).

[12] M.-T. Nguyen et al. “Recurrent Neural Network for Partial Discharge Diagnosis in Gas-Insulated Switchgear.” In: *Energies* 11.5 (May 2018). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 1202. DOI: `10.3390/en11051202`.

[13] H.-G. Kranz. “Diagnosis of partial discharge signals using neural networks and minimum distance classification.” In: *IEEE Transactions on Electrical Insulation* 28.6 (1993), pp. 1016–1024.

[14] JWG D1/B3.57: C. Neumann et al. *DIELECTRIC TESTING OF GAS-INSULATED HVDC SYSTEMS.* Tech. rep. Working Group: JWG D1/B3.57. CIGRÉ, Oct. 5, 2020. In preparation.

[15] A. Abbasi et al. “Interim Report of WG D1.63: Progress on Partial discharge detection under DC voltage stress.” In: CIGRE COLLOQUIUM SCA2 /SCB2/ SCD1. CIGRE India. 2019.

[16] P. Wenger et al. “Combined Characterization of Free-Moving Particles in HVDC-GIS Using UHF PD, High-Speed Imaging, and Pulse-Sequence Analysis.” In: *IEEE Transactions on Power Delivery* 34.4 (Aug. 2019), pp. 1540–1548. ISSN: 1937-4208. DOI: `10.1109/TPWRD.2019.2909830`.

[17] L. D. Ott. “Statistische Analyse von Teilentladungen in gasisolierten Systemen unter hoher Gleichspannungsbeanspruchung.” German. MA thesis. Institut für Energie- und Automatisierungstechnik, TU Berlin, 2019.

[18] F. H. Kreuger, E. Gulski, and A. Krivda. “Classification of partial discharges.” In: *IEEE Transactions on Electrical Insulation* 28.6 (1993), pp. 917–931.

[19] S. Kainaga, A. Pirker, and U. Schichler. “Identification of Partial Discharges at DC Voltage using Machine Learning Methods.” In: 20th International Symposium on High Voltage Engineering, ISH 2017 (Aug. 27–Sept. 1, 2017). Buenos Aires, Argentina, Sept. 1, 2017. URL: `https://e-cigre.org/publication/ISH2017_528-identification-of-partial-discharges-at-dc-voltage-using-machine-learning-methods` (visited on 08/13/2021).

[20] B. Schober and U. Schichler. “Application of Machine Learning for Partial Discharge Classification under DC Voltage.” In: *Proceedings of the Nordic Insulation Symposium*. Issue: 26. 2019, pp. 16–21.

[21] B. Schober and U. Schichler. “Feature Extraction for Machine Learning Algorithms to Classify Partial Discharge under DC Voltage.” In: *2020 8th International Conference on Condition Monitoring and Diagnosis (CMD)*. 2020 8th International Conference on Condition Monitoring and Diagnosis (CMD). ISSN: 2644-271X. Oct. 2020, pp. 278–281. DOI: 10.1109/CMD48350.2020.9287260.

[22] S. Lu et al. “Condition Monitoring Based on Partial Discharge Diagnostics Using Machine Learning Methods: A Comprehensive State-of-the-Art Review.” In: *IEEE Transactions on Dielectrics and Electrical Insulation* 27.6 (2020), pp. 1861–1888. DOI: 10.1109/TDEI.2020.009070.

[23] M. Hering, H. Koch, and K. Juhre. “Direct Current High-Voltage Gas-Insulated Switchgear up to ±550 kV.” In: *Colloquium - Hakodate 2019*. CIGRE-IEC 2019 Conference on EHV and UHV (AC & DC). CIGRÉ, Apr. 2019. URL: https://e-cigre.org/publication/COLL_HAK_2019-colloquium-hakodate-2019 (visited on 07/13/2021).

[24] Siemens AG. *GIL structure*. Image URL: https://assets.new.siemens.com/siemens/assets/api/uuid:3d693de2c6ed84d7ce56e7cdb77d2789953999e0/width:640/quality:high/7097-gil-aufbau-en.jpg. 2020. URL: https://new.siemens.com/in/en/products/energy/high-voltage/power-transmission-lines/gas-insulated-lines.html (visited on 07/11/2021).

[25] A. Küchler. *Hochspannungstechnik. Grundlagen - Technologie - Anwendungen*. German. 4th ed. VDI-Buch. Springer Vieweg, 2017. ISBN: 978-3-662-54700-7. DOI: 10.1007/978-3-662-54700-7.

[26] C. Toigo et al. “Partial Discharge Behavior of Protrusion on High Voltage Conductor in GIS/GIL under High Voltage Direct Current: Comparison of SF_6 and SF_6 Alternative Gases.” In: *IEEE Transactions on Dielectrics and Electrical Insulation* 27.1 (2020), pp. 140–147. DOI: 10.1109/TDEI.2019.008358.

[27] S. Glomb, M. Göppel, and P. Pilzecker. *„Alternative Gase“ und Gasmischungen Teil 1*. German. White Paper. DILO Armaturen und Anlagen GmbH, 2020. URL: https://dilo.eu/fileadmin/dilo_de/9._Unternehmen/13._Downloads/1._Login/2._PDFs/DILO-White-Paper_2020_D.pdf (visited on 07/14/2021).

[28] M. Hering. “Überschlagsverhalten von Gas-Feststoff-Isoliersystemen unter Gleichspannungsbelastung.” German. PhD thesis. 2016. ISBN: 978-3-00-052735-7. URL: https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-200879 (visited on 06/04/2020).

[29] C. Neumann et al. "PD measurements on a DC gas-insulated transmission line (DC GIL) conducted in the frame of the Prototype Installation Test according to recommendation of CIGRE JWG D1/B3.57." In: *VDE High Voltage Technology 2020; ETG-Symposium.* VDE High Voltage Technology 2020; ETG-Symposium. Nov. 2020, pp. 1–7.

[30] T. Neidhart et al. "Long-Term Tests of a DC Gas Insulated Transmission Line (DC GIL) embedded in Temporally Flowable Backfill: Soil-Mechanical and Thermic Interaction." In: *VDE High Voltage Technology 2020; ETG-Symposium.* 2020, pp. 1–7.

[31] Y. Tu et al. "±100-kV HVDC SF6/N2 Gas-Insulated Transmission Line." In: *IEEE Transactions on Power Delivery* 35.2 (2020), pp. 735–744. DOI: `10.1109/TPWRD.2019.2925519`.

[32] K. Juhre et al. "Feasibility Study on the Applicability of Clean Air in Gas-insulated DC Systems." In: *VDE High Voltage Technology 2020; ETG-Symposium.* VDE High Voltage Technology 2020; ETG-Symposium. Nov. 2020, pp. 1–6. URL: `https://ieeexplore.ieee.org/abstract/document/9275521` (visited on 06/12/2021).

[33] High Voltage Testing and Engineering Commission. *Types of insulation fault on a GIS.* Image URL: `https://fkh.ch/wp-content/uploads/2018/10/Fehlertypen_GIS.png`. 2018. URL: `https://fkh.ch/en/dienstleistungen/pruefung-von-hs-betriebsmitteln-vor-ort/gis` (visited on 07/11/2021).

[34] T. Berg. "Leitfähige Partikel im koaxialen Rohrleiter unter hoher Gleichspannungsbeanspruchung." German. PhD thesis. 2014, p. 86. URL: `https://diglib.tugraz.at/leitfaehige-partikel-im-koaxialen-rohrleiter-unter-hoher-gleichspannungsbeanspruchung-2014` (visited on 06/04/2020).

[35] Working Group D1.03 (TF 11). *Gas Insulated System for HVDC: DC Stress at DC and AC Systems.* e-cigre. Technical Brochure 506. CIGRÉ, 2012. ISBN: 978-2-85873-198-5. URL: `https://e-cigre.org/publication/506-gas-insulated-system-for-hvdc-dc-stress-at-dc-and-ac-systems` (visited on 06/06/2021).

[36] T. Berg, H. Koch, and K. Juhre. "Free Moving Particles in Gas-Insulated Lines Under DC Conditions – Basic Properties, Specific Effects and Countermeasures." In: *Proceedings of the 21st International Symposium on High Voltage Engineering.* Ed. by B. Németh. Lecture Notes in Electrical Engineering. Cham: Springer International Publishing, 2020, pp. 1489–1499. ISBN: 978-3-030-31680-8. DOI: `10.1007/978-3-030-31680-8_141`.

[37] H. Parekh, K. D. Srivastava, and R. G. van Heeswijk. "Lifting Field of Free Conducting Particles in Compressed SF6 with Dielectric Coated Electrodes." In: *IEEE Transactions on Power Apparatus and Systems* PAS-98.3 (1979), pp. 748–758. DOI: `10.1109/TPAS.1979.319287`.

[38] U. Schichler et al. “Risk assessment on defects in GIS based on PD diagnostics.” In: *IEEE Transactions on Dielectrics and Electrical Insulation* 20.6 (2013), pp. 2165–2172. DOI: 10.1109/TDEI.2013.6678866.

[39] IEC 60270:2000. *High-voltage test techniques – Partial discharge measurement.* Standard. IEC, 2000.

[40] F. Rösch. “Analyse von Teilentladungen in gasisolierten Systemen bei hoher Gleichspannungsbelastung.” German. MA thesis. Institut für Energie- und Automatisierungstechnik, TU Berlin, 2017.

[41] Siemens AG. *Feldsonde für metallgekapselte gasisolierte Schaltanlagen - Betriebsanleitung.* Bereich Energieübertragung und -verteilung Geschäftsstelle Hochspannung. 1999.

[42] Working Group D1.25. *UHF Partial Discharge Detection System for GIS: Application Guide for Sensitivity Verification.* Technical Brochure 654. CIGRÉ, 2016. ISBN: 978-2-85873-357-6. URL: https://e-cigre.org/publication/654-uhf-partial-discharge-detection-system-for-gis-application-guide-for-sensitivity-verification (visited on 06/06/2021).

[43] M. Geske and R. Plath. *Teilentladungsdiagnostik und Monitoring DC CTL DBI AP1.6 und DC CTL AP9.* German. Poster presented at "Einweihung des Versuchsfeldes für erdverlegte gasisolierte Hochspannungs-Gleichstromleitungen". Griesheim: Technische Universität Berlin, Fachgebiet Hochspannungstechnik, Oct. 17, 2019.

[44] A. Pirker and U. Schichler. “Partial discharges at DC voltage - measurement and pattern recognition.” In: *2016 International Conference on Condition Monitoring and Diagnosis (CMD).* 2016 International Conference on Condition Monitoring and Diagnosis (CMD). Sept. 2016, pp. 287–290. DOI: 10.1109/CMD.2016.7757811.

[45] C. M. Bishop. *Pattern recognition and machine learning.* Springer Science+Business Media, LLC, 2006. ISBN: 978-0387-31073-2.

[46] L. E. Peterson. “K-nearest neighbor.” In: *Scholarpedia* 4.2 (2009), p. 1883. DOI: 10.4249/scholarpedia.1883. URL: http://www.scholarpedia.org/article/K-nearest_neighbor (visited on 04/02/2021).

[47] C. W. Tan et al. “Time series classification for varying length series.” In: (Oct. 9, 2019). arXiv: 1910.04341. URL: http://arxiv.org/abs/1910.04341 (visited on 11/05/2020).

[48] H. Ding et al. “Querying and mining of time series data: experimental comparison of representations and distance measures.” In: *Proceedings of the VLDB Endowment* 1.2 (Aug. 1, 2008), pp. 1542–1552. ISSN: 2150-8097. DOI: 10.14778/1454159.1454226.

[49] A. Bagnall et al. “The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances.” In: *Data Mining and Knowledge Discovery* 31.3 (May 1, 2017), pp. 606–660. ISSN: 1573-756X. DOI: 10.1007/s10618-016-0483-9.

[50] E. Bisong. “Logistic Regression.” In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Ed. by E. Bisong. Berkeley, CA: Apress, 2019, pp. 243–250. ISBN: 978-1-4842-4470-8. DOI: 10.1007/978-1-4842-4470-8_20.

[51] L. Breiman. “Random Forests.” In: *Machine Learning* 45.1 (Oct. 1, 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324.

[52] G. Ke et al. “Lightgbm: A highly efficient gradient boosting decision tree.” In: *Advances in neural information processing systems* 30 (2017), pp. 3146–3154.

[53] Z. Xing, J. Pei, and E. Keogh. “A brief survey on sequence classification.” In: *ACM SIGKDD Explorations Newsletter* 12.1 (Nov. 9, 2010), pp. 40–48. ISSN: 1931-0145. DOI: 10.1145/1882471.1882478.

[54] Y. Guo, Z. Wu, and Y. Ji. “A Hybrid Deep Representation Learning Model for Time Series Classification and Prediction.” In: *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*. 2017, pp. 226–231. DOI: 10.1109/BIGCOM.2017.13.

[55] H. Ismail Fawaz et al. “Deep learning for time series classification: a review.” In: *Data Mining and Knowledge Discovery* 33.4 (July 1, 2019), pp. 917–963. ISSN: 1573-756X. DOI: 10.1007/s10618-019-00619-1.

[56] Z. Geler et al. “Dynamic Time Warping: Itakura vs Sakoe-Chiba.” In: *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*. 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA). July 2019, pp. 1–6. DOI: 10.1109/INISTA.2019.8778300.

[57] P. Schäfer. “The BOSS is concerned with time series classification in the presence of noise.” In: *Data Mining and Knowledge Discovery* 29.6 (Nov. 1, 2015), pp. 1505–1530. ISSN: 1573-756X. DOI: 10.1007/s10618-014-0377-7.

[58] P. Schäfer and U. Leser. “Fast and Accurate Time Series Classification with WEASEL.” In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM ’17. New York, NY, USA: Association for Computing Machinery, Nov. 6, 2017, pp. 637–646. ISBN: 978-1-4503-4918-5. DOI: 10.1145/3132847.3132980.

[59] A. Bagnall et al. "Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles." In: *IEEE Transactions on Knowledge and Data Engineering* 27.9 (Sept. 2015). Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 2522–2535. ISSN: 1558-2191. DOI: 10.1109/TKDE.2015.2416723.

[60] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[61] D. Berrar. "Cross-validation." In: *Encyclopedia of bioinformatics and computational biology* 1 (2019). Publisher: Academic, pp. 542–545. DOI: 10.1016/B978-0-12-809633-8.20349-X.

[62] G. Wilson et al. "Good enough practices in scientific computing." In: *PLOS Computational Biology* 13.6 (June 22, 2017). Publisher: Public Library of Science, e1005510. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005510.

[63] M. Christ et al. "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)." In: *Neurocomputing* 307 (Sept. 13, 2018), pp. 72–77. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2018.03.067.

[64] M. L. Waskom. "seaborn: statistical data visualization." In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021.

[65] Microsoft Corporation. *Light Gradient Boosting Machine Classifier*. Version 3.2.1. URL: https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html (visited on 03/25/2021).

[66] F. Chollet et al. *Keras*. Version 2.4.2. 2015. URL: https://keras.io (visited on 07/11/2021).

[67] R. Tavenard et al. "Tslearn, A Machine Learning Toolkit for Time Series Data." In: *Journal of Machine Learning Research* 21.118 (2020), pp. 1–6. URL: http://jmlr.org/papers/v21/20-091.html.

[68] J. Faouzi and H. Janati. "pyts: A Python Package for Time Series Classification." In: *Journal of Machine Learning Research* 21.46 (2020), pp. 1–6. URL: http://jmlr.org/papers/v21/19-763.html.

[69] L. S. Shapley. *A Value for n-Person Games*. Pages: 307-318 Publication Title: Contributions to the Theory of Games (AM-28), Volume II Section: Contributions to the Theory of Games (AM-28), Volume II. Princeton University Press, Mar. 2, 2016. ISBN: 978-1-4008-8197-0. DOI: https://doi.org/10.1515/9781400881970-018.

[70] S. M. Lundberg and S.-I. Lee. "A unified approach to interpreting model predictions." In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., Dec. 4, 2017, pp. 4768–4777. ISBN: 978-1-5108-6096-4.

[71] 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.” In: *Electrical Engineering* 98.1 (2016). Publisher: Springer, pp. 29–42.

[72] X. Wan et al. “Pattern Recognition of Partial Discharge Image Based on One-dimensional Convolutional Neural Network.” In: *2018 Condition Monitoring and Diagnosis (CMD).* 2018 Condition Monitoring and Diagnosis (CMD). Sept. 2018, pp. 1–4. DOI: 10.1109/CMD.2018.8535761.

[73] L. Li, J. Tang, and Y. Liu. “Partial discharge recognition in gas insulated switchgear based on multi-information fusion.” In: *IEEE Transactions on Dielectrics and Electrical Insulation* 22.2 (Apr. 2015). Conference Name: IEEE Transactions on Dielectrics and Electrical Insulation, pp. 1080–1087. ISSN: 1558-4135. DOI: 10.1109/TDEI.2015.7076809.

[74] J. Lines, S. Taylor, and A. Bagnall. “Time Series Classification with HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles.” In: *ACM Transactions on Knowledge Discovery from Data* 12.5 (July 5, 2018), 52:1–52:35. ISSN: 1556-4681. DOI: 10.1145/3182382.

[75] A. Dempster, F. Petitjean, and G. I. Webb. “ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels.” In: *Data Mining and Knowledge Discovery* 34.5 (Sept. 1, 2020), pp. 1454–1495. ISSN: 1573-756X. DOI: 10.1007/s10618-020-00701-z.

[76] W. Jiang. “Time series classification: nearest neighbor versus deep learning models.” In: *SN Applied Sciences* 2.4 (Mar. 20, 2020), p. 721. ISSN: 2523-3971. DOI: 10.1007/s42452-020-2506-9.

[77] C. Tan et al. “A Survey on Deep Transfer Learning.” In: *Artificial Neural Networks and Machine Learning – ICANN 2018.* Ed. by V. Kůrková et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 270–279. ISBN: 978-3-030-01424-7. DOI: 10.1007/978-3-030-01424-7_27.

[78] H. Ismail Fawaz et al. “Transfer learning for time series classification.” In: *2018 IEEE International Conference on Big Data (Big Data).* 2018 IEEE International Conference on Big Data (Big Data). Dec. 2018, pp. 1367–1376. DOI: 10.1109/BigData.2018.8621990.

[79] W. Yijiang et al. “Partial Discharge Data Augmentation of High Voltage Cables based on the Variable Noise Superposition and Generative Adversarial Network.” In: *2018 International Conference on Power System Technology (POWERCON).* 2018 International Conference on Power System Technology (POWERCON). Nov. 2018, pp. 3855–3859. DOI: 10.1109/POWERCON.2018.8602223.

[80] W. L. Woon, A. El-Hag, and M. Harbaji. “Machine learning techniques for robust classification of partial discharges in oil–paper insulation systems.” In: *IET Science, Measurement &amp; Technology* 10.3 (May 1, 2016). Publisher: IET Digital Library, pp. 221–227. ISSN: 1751-8830. DOI: `10.1049/iet-smt.2015.0076`.

[81] L. Jinshu et al. “Gradient Boosting Decision Tree and Random Forest Based Partial Discharge Pattern Recognition of HV Cable.” In: *2018 China International Conference on Electricity Distribution (CICED).* 2018 China International Conference on Electricity Distribution (CICED). ISSN: 2161-749X. Sept. 2018, pp. 327–331. DOI: `10.1109/CICED.2018.8592402`.

[82] Y. Zang et al. “A Novel Partial Discharge Detection Method Based on the Photoelectric Fusion Pattern in GIL.” In: *Energies* 12.21 (2019). ISSN: 1996-1073. DOI: `10.3390/en12214120`.

[83] X. Li et al. “Partial Discharge Source Localization in GIS Based on Image Edge Detection and Support Vector Machine.” In: *IEEE Transactions on Power Delivery* 34.4 (2019), pp. 1795–1802. DOI: `10.1109/TPWRD.2019.2925034`.

[84] P. Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.” In: *Nature Methods* 17 (2020), pp. 261–272. DOI: `10.1038/s41592-019-0686-2`.