

Application of Machine Learning for Partial Discharge Classification under DC Voltage

B. Schober & U. Schichler

*Institute of High Voltage Engineering and System Performance, Graz University of Technology, Graz, Austria
bernhard.schober@tugraz.at*

Abstract

Partial discharge measurement is one of the most important diagnosis methods and well investigated under AC voltage. Furthermore, machine learning is established and has been used successfully already many years for automated recognition of PD defects. For AC voltage, there are several diagnosis methods and interpretation tools. In the field of DC voltage this is not the case, so it needs significant tools to interpret the results. In this contribution typical PD defects of HVDC GIS/GIL are investigated, but the methods can be adopted to other HV equipment as well. The machine learning techniques were realized with MATLAB and WEKA. Statistical parameters, derived from the PD pulse sequences, were used as features. A hierarchical clustering of the features was performed to analyse the separability between the PD defects. Classification was done with three popular algorithms (SVM, k-NN, ANN). The parameters of these algorithms were varied and compared to each other's. SVM clearly outperformed the other classifiers.

1. Introduction

The rising demand for electrical energy in the last few decades, as well as a trend towards renewable energies, such as photovoltaic, hydropower and wind energy, makes it necessary to transport huge amounts of energy over large distances. Therefore, energy economy has been within the last few years and is still facing new challenges. To overcome these challenges, the usage of transmission with high-voltage direct current (HVDC) systems increased dramatically. The main advantages are higher transmission capacity, an enhanced power-flow and improved voltage regulation, transport of electrical energy over long distances and considerable proportion to the stability of the AC grid. For those reasons, also more and more integrations of medium-voltage direct current (MVDC) systems can be observed. The usage of already existing AC transmission lines, such as overhead lines and cables, for MVDC systems seems obvious, because of the higher efficiency and the lower costs compared to a new installation. Especially in outlying areas the rising energy supply can be ensured with DC transmission [1]. Consequently, the need for DC equipment increases and thereby also the need for reliable diagnostic tools for DC voltage arises.

One of the most important diagnostic methods are partial discharge (PD) measurements. It is very well established and approved under AC voltage. The interpretation of the measurement results is possible with PRPD patterns and in general the acceptance for this method is high as well. On the contrary, this is not the case for measurements under DC voltage. The corresponding standard IEC 60270 [2] is vague for

DC voltage and not very detailed in this regard. Furthermore, it lacks of significant and convincing diagnosis and evaluation tools to interpret the results. A promising method to facilitate the classification of PD measurements under DC voltage by a human expert is the NoDi* pattern (Normalized Differenced Pattern, Figure 1) and will be described in chapter 2.2 [3].

Machine learning can lead to even better classification results. This does not mean that human experts become obsolete, but that machine learning can find hidden patterns in the measurement data and facilitate PD monitoring purposes.

2. Interpretation of PD measurements

2.1 Fundamental quantities at DC voltage

As mentioned already in the introduction, interpretation by human experts is mostly done with PRPD patterns under AC voltage. Unfortunately, because of the lack of phase dependency of the PD pulses, this method cannot be used for DC voltage. The fundamental quantities of a PD pulse at DC voltage is the PD magnitude q_i , the occurrence time t_i and for the sake of completeness the voltage level v_i . Therefore, the pulse sequences consist of three different vectors.

Using this raw data for classification, regardless if it is done by a human expert or with machine learning, leads to insufficient results. For this purpose, classification is done using derived quantities, gained from the three main quantities. Derived parameters are e. g. the differences between the discharge magnitudes of two subsequent PD pulses Δq_i , the differences in time Δt_i , PD pulses per second, maximum, minimum and mean

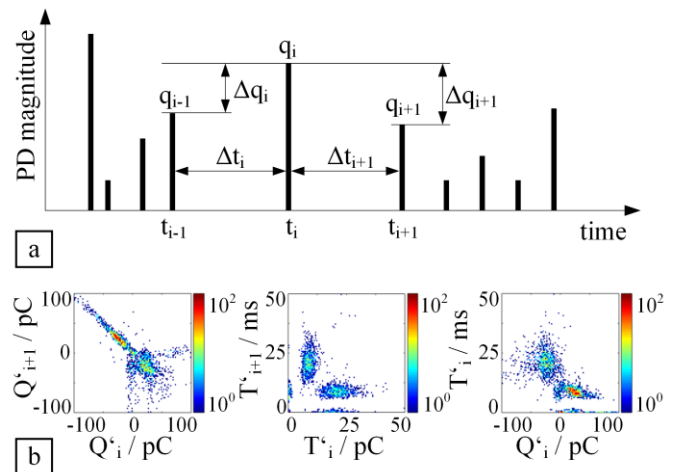


Figure 1. Pulse sequence of partial discharges and calculation of the differential values (a) and transformation to the NoDi* pattern exemplary for a bouncing particle (b)

values of the before mentioned differential values. The three possible correlation of the values of Δq_i and Δt_i leads to the previously mentioned NoDi* pattern.

A software tool named as “PDDC Analyzer”, programed with the goal to simplify the interpretation of PD pulse sequences with common diagrams, has been already mentioned and published [3, 4]. Since then, the former pre-processing program has already been extended with a live operating mode (“Live PDDC Analyzer”), which makes it possible to analyse pulse sequences in real time. This mode has, due to some limitations during live analysis, of course less functionality, but includes the most common and useful diagrams, such as several time plots and the NoDi* pattern for PD measurements under DC voltage and offers all relevant and important settings.

2.2 NoDi* pattern

The NoDi* pattern depicts the normalized differential values of q_i and t_i in different combinations, which allow an analysis regarding the differences in time Δt and in charge Δq between two discharges. The values are defined as:

$$\Delta q_i = q_i - q_{i-1} \quad (1)$$

$$\Delta t_i = t_i - t_{i-1} \quad (2)$$

The differential values are labeled with lowercase and the normalized values with capital letters. These two parameters can be correlated and subsequently all combinations can be computed and visualized in scatter plots: NoDi*_Q (density distribution of $H_n(\Delta q_i, \Delta q_{i+1})$), NoDi*_T ($H_n(\Delta t_i, \Delta t_{i+1})$) and NoDi*_{QT} ($H_n(\Delta q_i, \Delta t_i)$). Through the correlation of these values and mapping into a grid, clusters are formed and the intensity is depicted by a logarithmic color scale [3]. By investigating the resulting clusters, it is possible for a human expert to distinguish between the typical defects in HVDC GIS/GIL.

3. Machine learning for PD at DC voltage

3.1 Basics

In general machines do not need visualisations like the NoDi* pattern to interpret data and make predictions, yet there are other challenges to face. Decisions, which are made naturally and intuitively by humans, are extremely hard for machines. On the contrary, computations, which are very hard to solve for humans, are done easily by computers. Instead of programming all kind of decisions and possibilities to perform predictions, modern machine learning techniques take another path. As the name suggests, machines can be trained resp. learned to make the right decisions. For the described investigations, machine learning was done with the two popular programs MATLAB and WEKA. Both programs provide a high number of algorithms and have big machine learning libraries [5, 6].

The question arises, which data to put into the machine learning algorithms during learning and to predict classes afterwards. The input data are called features and are used to characterise the different PD defects, like a fingerprint.

3.2 Feature extraction and clustering

Feature extraction is a subarea of feature engineering with the goal to use the knowledge of the data to create significant

features as n-dimensional feature vectors. Significant means, that the features of the different classes are separable in the n-dimensional feature space. In the best case each feature of each defect has its own cluster. Without a significant feature extraction even the best machine learning algorithm will not be able to attain adequate results. For those reasons, it seems essential to spend time on observing the data and evaluate the extracted features.

1) Statistical parameters

Statistical analysis of the PD pulse sequence are useful to describe the distribution histograms of the several quantities of PD pulses. Characteristic parameters could be e. g. mean value and variance, the skewness, the kurtosis and the parameters of the Weibull distribution. In general, it seems to be a better choice to use parameters, which describe the shape of the distribution, rather than using absolute values, which depending on the real data like the mean value of the discharge magnitudes. The values considered to be characteristic for describing the shape of the distributions are as follows:

A) The mean value μ and variance σ^2 of a distribution:

$$\mu = \sum(x_i \cdot P_i) \quad (3)$$

$$\sigma^2 = \sum(x_i \cdot \mu)^2 \cdot P_i \quad (4)$$

The variance is a measure of a variable for the spreading from its mean value, whereas small values describe little spreading and big values high spreading.

B) The skewness S_K and kurtosis K_U are calculated by:

$$S_K = \frac{\sum(x_i - \mu)^3 \cdot P_i}{\sigma^3} \quad (5)$$

$$K_U = \frac{\sum(x_i - \mu)^4 \cdot P_i}{\sigma^4} - 3 \quad (6)$$

where x_i is the discrete value and P_i the probability for x_i [7]. The skewness describes the type and intensity of the asymmetry of the distribution and can range from positive values (right-skewness) to negative values (left-skewness) and zero (complete symmetry). The kurtosis is a measure of the gradient of the distribution. Positive kurtosis can be described as sharp and negative as flat graphs. Examples of these two statistical parameters can be found in Figure 2.

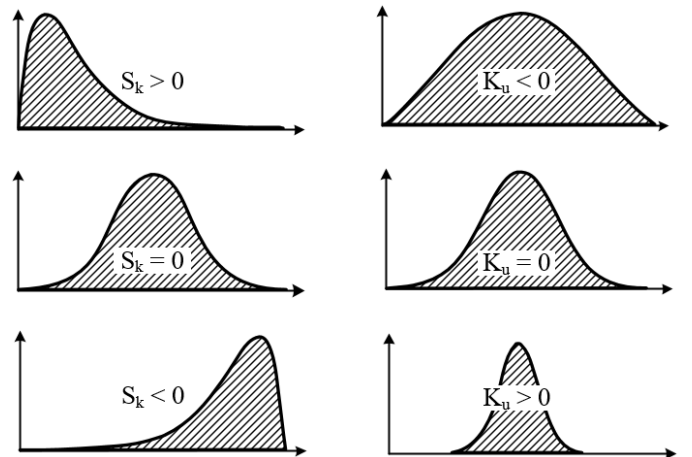


Figure 2. Illustration of different values for skewness (S_k) and kurtosis (K_u) of a distribution

C) The scale parameter a and shape parameter b of the fitted Weibull distribution f_W :

$$f_W(x) = \begin{cases} \frac{b}{a} \left(\frac{x}{a}\right)^{b-1} e^{-(x/a)^b}, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (7)$$

The Weibull distribution was fitted to the density distribution functions of Δq and Δt with the MATLAB function `wblfit()` using the maximum likelihood (MLE) estimate (Figure 3), like published by other researchers before [8, 9]. Next, the two parameters a and b were determined from the fitted distributions and used as features. To separate between the parameters of Δq and Δt , the features are labelled with the corresponding letters.

As can be seen in Figure 3, the histograms depend on the chosen number of bins. If the number of bins resp. the bin width is chosen too small, important information in the distribution could get suppressed. On the contrary, if it is chosen too high, the distribution could change significantly compared to the real distribution. Thus, it is a better choice to use the empirical cumulative distribution function (CDF) F_n rather than the density distribution function (Figure 4):

$$F_n(x) = \frac{1}{n} \sum_{j=1}^n \mathbf{1}_{\Delta q \leq x} \quad (8)$$

This graph has some advantages compared to the histogram. Independency of the bin width, thus easier detection of outliers, identification of clusters by a decreasing gradient in the graph, more robustness against misinterpretations because of unlucky axes scaling. Comparing the histogram and CDF in Figure 4, the Weibull distribution with the estimated parameters almost perfectly matches the empirical distribution in the CDF graph.

In total 14 statistical features (Table 1) were used to describe the differences in the distributions and to distinguish between them. The focus was on features, which do not depend on absolute values, which can differ between different PD measurement systems and methods. Some of the reported features had been used before to classify and predict different PD defects with machine learning [10]. In Figure 5 the box-and-whisker plot of the Weibull parameters $b_{\Delta q}$ and $b_{\Delta t}$ is shown. It can be derived, that *Bouncing Particle* and *Protrusion on HV* can be separated with this parameter from the others. Contrary to this, the Weibull parameter $b_{\Delta t}$ partly overlaps among the different defects and therefore has a lower separability.

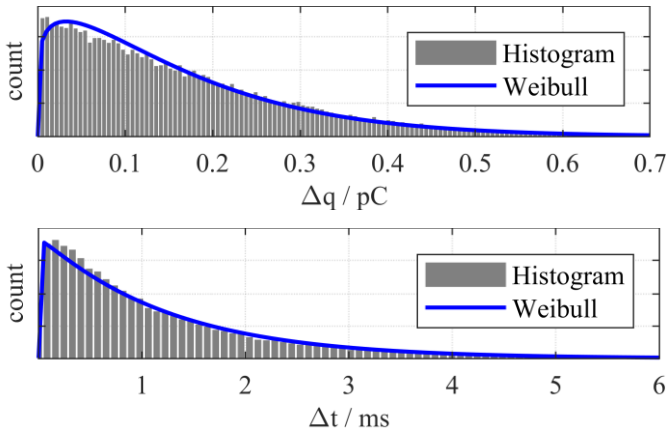


Figure 3. Density distribution $H_n(\Delta q)$ and $H_n(\Delta t)$ for the defect *Protrusion on HV* and the corresponding fitted Weibull distributions

2) Cluster analysis

Cluster analysis belongs to unsupervised learning and is a useful tool to examine hidden patterns in the data and is often used to explore the extracted features and to check whether clusters are built by the different defect classes. The most common methods are k-means clustering, where the data gets partitioned into k clusters, and hierarchical clustering, which is based on a measurement of distance between clusters. Hierarchical clustering is divided into agglomerative and divisive, whereas in hierarchical agglomerative clustering every single data set (instance) of e. g. PD measurements is considered as an own cluster in the n -dimensional space and subsequently merged into a new cluster with the cluster next to it. This procedure is repeated until one single cluster is formed. Divisive clustering starts with one cluster and splits the data into more clusters. In this contribution, hierarchical agglomerative (“bottom-up”) clustering with single-linkage and Euclidean distance was used as distance measurement. The results are visualized in a dendrogram, which shows the dissimilarity between the formed clusters as a tree (Figure 6).

The y-axis of the dendrogram describes this dissimilarity. Consequently, clusters which are closer to each other’s (Cluster a to Cluster b and Cluster c to Clusters d) are connected earlier in this graph. It is obvious, that the scale of the different features must be in the same range, because of the metric distance measurements, otherwise huge numbers would dominate the connection heights in the dendrogram. Therefore, different methods can be used such as standardising (also z-score), that means rescaling the data to zero-mean and unit-variance (Equation 9), and min-max normalisation, which rescales the data between $[0, 1]$ or $[-1, 1]$ (Equation 10).

$$x'_{Stand} = \frac{x - \mu}{\sigma} \quad (9)$$

$$x'_{Norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (10)$$

This process is called feature scaling, where x' is the normalized value, x the original value, μ the mean value and σ the variance of x . Standardisation is a good choice for different types of features and was chosen to normalize the features. Another aspect which cannot be neglected is, that many machine learning algorithms work just properly with normalized data and convergence much faster during the training period [11].

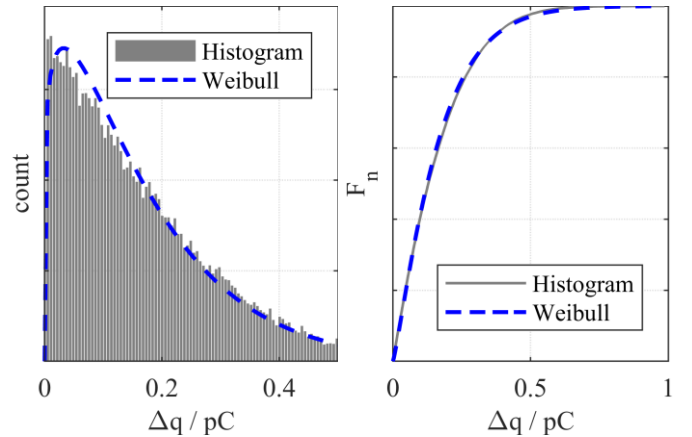
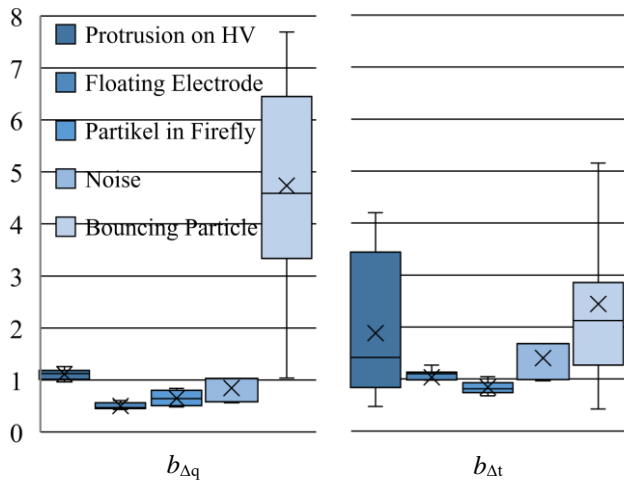


Figure 4. Comparison between the histogram and the cumulative distribution function (CDF) F_n with the corresponding Weibull fits

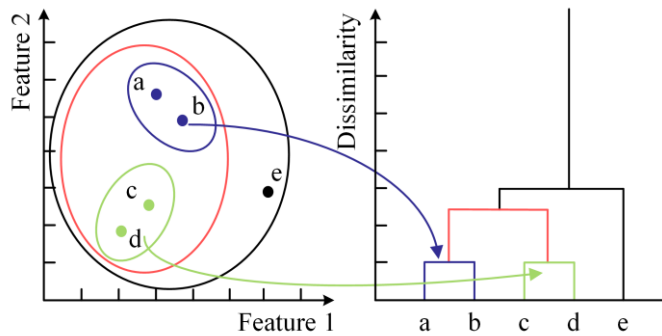
Table 1. Features extracted from the pulse sequences and used for training and classification

1	Variance of q
2	Skewness of q
3	Kurtosis of q
4	Weibull parameter $a_{\Delta q}$
5	Weibull parameter $b_{\Delta q}$
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	Weibull parameter $a_{\Delta t}$
13	Weibull parameter $b_{\Delta t}$
14	Numbers of discharges per second

**Figure 5.** Box-and-whisker plot of the Weibull shape parameters $b_{\Delta q}$ and $b_{\Delta t}$ of the fitted Weibull distributions for typical PD defects

3.3 Database used for training

The database used to train the machine learning algorithms consisted of 70 data sets (instances) of PD measurements of five typical defects which can occur in HVDC GIS/GIL. The defects were already described in previous publications [3, 10]. The bandwidth of the measurement system for these PD measurements was set to 3 MHz and the center frequency was chosen with 1.6 MHz. A hierarchical clustering of the standardised features was done to analyse the separability

**Figure 6.** Hierarchical agglomerative clustering and dendrogram of the formed clusters of the instances a to e

between them. The clustering was performed with MATLAB and WEKA, which, as expected, led to the same results.

Figure 7 shows the dendrogram of the 70 PD measurements. It can be seen, that the instances of *Bouncing Particle* are very similar to each other's. This defect is one of the most common ones in HVDC GIS/GIL and is responsible for about 20 % of all dielectric failures [12]. *Particle in Firefly* has a low dissimilarity as well. The dispersion of *Floating Electrode* and *Protrusion on HV*, on the other side, is higher and the separability between those can be harder. Disturbances such as noise floor from the amplifier and the measurement setup were also considered in the database and summarized as *Noise*. Some instances of *Noise* are very similar to *Particle in Firefly*, which could be problematic in terms of risk management, in the case that *Noise* will be classified as *Particle in Firefly* or the other way round.

3.4 Classification algorithms

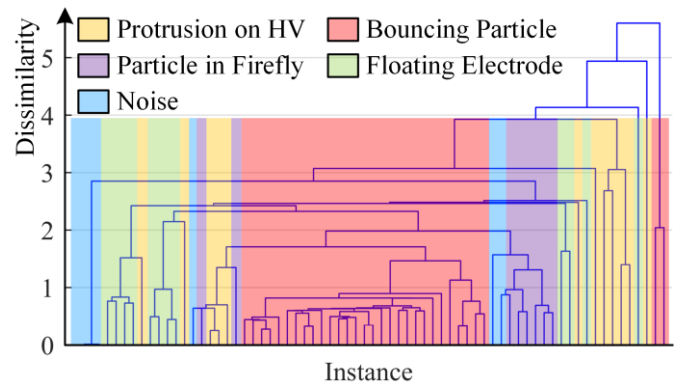
Classification was done with three popular and commonly used machine learning methods. The chosen methods (Figure 8) were support-vector machines (SVM), k-nearest neighbours (k-NN) and artificial neural networks (ANN).

1) Support-vector machines

SVM is an often used classification technique. The principle of SVM is to find a decision boundary, which separates the space into two halves. Usually, it needs just a few instances, more precisely the closest ones to the decision boundary to define it. These instances are called support-vectors (SV). Basically, SVM are binary linear classifiers, but can be modified in several ways. By combining multiple binary classification problems (One-vs-One, OvO) or by sequentially training always one classifier per class against all other classes (One-vs-All, OvA), SVM can also be used for multi-class classification. To apply SVM to non-linearly separable problems as well, kernel functions can be used, which projects the original data to a higher dimensional space to attain linear separability in the n-dimensional space (kernel trick) and consequently to find a hyperplane there (Cover's theorem). Common kernel functions are polynomial (exponent parameter d) or radial basis function (RBF) kernels (scale parameter γ) among others [11, 13].

2) k-nearest neighbours

This algorithm belongs to the so called lazy learning methods, in contrast to the other classifiers, which are eager

**Figure 7.** Dendrogram of the PD measurements of five different defect classes (agglomerative, Euclidean distance and single-linkage)

learning methods. It uses distance measurements to find the k closest samples and assigns the new sample to the most frequent one of those k samples. The difficulty is to find the right number for k . The advantage is, that classification can be described by a physical value (distance) and that this algorithm does not need any training. The classification is done with all samples (instance-based classification), which furthermore requires enough memory. Additionally, if the selected features are not the right ones or optimized to precisely describe the data, this method will attain a bad performance. Lazy learning is relatively slow compared to other learning methods, but redeems it with higher flexibility, because the reference data base can be easily replaced depended on the device under test [11, 14].

3) Artificial neural networks

Artificial neural networks are inspired by the human brain and biological neurons and consist of artificial neurons. ANN imitate the human brain by adjusting the strengths or weights of the neural connections between the artificial neurons. In general, ANN, similar to SVM, are used for binary classification of linearly separable input data. This very simple model of an ANN is called perceptron. The weights are adjusted during learning. ANN can be modified with hidden layers to learn non-linear functions. By extending the output neuron of the perceptron, a multi-class classification is possible. The number of input layers is defined by the number of features. Each neuron in the output layer represent one predicted class. The number of hidden layers and neurons per layer can be chosen [11].

4) k -fold cross-validation (CV)

The three machine learning algorithms were trained using stratified k -fold cross-validation. Therefore, the database is partitioned into k folds, which are randomly chosen that way to ensure, that each class is properly represented in each fold (stratified). The model is trained on $k-1$ folds and validated once with the remaining fold. This procedure is then repeated k times with different partitions. The result is the average test error over all repetitions. Cross-validation is a well-known and often used technique to protect the models against under- and overfitting [11]. In this work stratified 5-fold CV was chosen.

4. Results of classification

Based on the described PD measurements and derived features, the algorithms were training to realise PD identification under DC voltage. The input data consisted of the standardised 14 features (Table 1). The predicted classes were *Floating Electrode*, *Noise*, *Bouncing Particle*, *Particle in Firefly* and *Protrusion on HV*. The performances of the trained algorithms

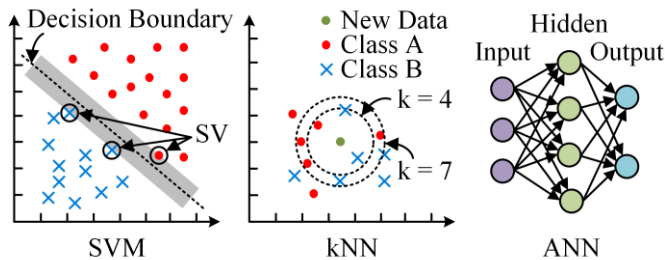


Figure 8. Classifiers and their principles used for prediction of PD measurements

are usually interpret with a confusion matrix, which depicts the amount of correctly classified instances in the major diagonal and the wrong predictions in the minor diagonal (Table 2).

Besides the confusion matrix, the accuracy is a helpful parameter to evaluate the classification performance. The accuracy can be calculated by:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (11)$$

For k -fold CV it is calculated by averaging over the k repetitions. Due to the random partition of k -fold CV, various repetitions can lead to varying results. For this purpose, 10 times stratified 5-fold CV were processed and the average calculated [11].

1) Support-vector machines

Table 3 shows the results attained from different SVM. As explained before, SVM are just binary classifiers, so different multi-class methods can be chosen. Another variable is the used kernel function. The worst and best result are highlighted in red and green in Table 3. It can be seen, that a Gaussian RBF kernel with a scale parameter of $\gamma = 0.0073$, a regulation parameter of $C = 1$ and OvO multi-class method performed the best and attained an accuracy of 96.3 %. In general, OvO performed better than OvA.

2) k -nearest neighbours

In Table 4 the performance of the different k -NN methods can be evaluated. The accuracy decreases for rising k . With a low number for k , the decision boundaries partition the classification region very fine and irregular. With a very high number, the classification is more robust against noise, but the boundaries are fuzzier. Especially in an unbalanced database, this can cause problems, so that a specific class dominates the classification. The best results with 85.1 % was achieved by using a 1-NN algorithm without distance weighting.

3) Artificial neural networks

The classification performance of the ANN can be found in Table 5. The network was training with back-propagation with a learning rate of 0.3 and momentum of 0.2 (standard values in

Table 2. Confusion matrix of a SVM with Medium Gaussian kernel and trained with One-vs-One method (parameters: $\gamma = 1/(3.7^2)$, $C = 1$)

Floating Electrode	12				
Noise		7			
Bouncing Particle			31		
Particle in Firefly			1	7	
Protrusion	1		1		10
True Class \ Predicted Class	Floating Electrode	Noise	Bouncing Particle	Particle in Firefly	Protrusion

WEKA). The best results were attained with one single hidden layer with accuracies from 91.4 % to 92.9 %. The usage of more than one hidden layer resulted in worse classification accuracies.

5. Ongoing investigation

The database will be extended to more instances of PD measurements in the future. The goal is to build a variable database of different kind of defects with variable feature selection and easy implementation of new features. This will simplify the process of feature engineering. This could be implemented in the existing software tool “Live PDDC Analyzer”.

It can be seen from the hierarchical clustering in Figure 6 that the dissimilarity between several defects like *Protrusion on HV* and *Floating Electrode* is low and in between these classes the dissimilarity is high. The other way around would be more desirable. More research needs to be done to extract more significant features to separate between these defects. In general, more features does not always automatically increases the

performance of the algorithm. Reducing the amount of features and consequently the redundancy in the data can be done with principal component analysis (PCA). In this studies, using PCA did not show better results. Next tasks and future work will also aim at optimizing parameters such as γ and C of the different algorithms (e. g. with grid-search).

6. Conclusion

This contribution provides a procedure to automatically classify and predict PD defects based on the fundamental quantities of PD pulse sequences under DC voltage. The challenging issue is to find the right features to significantly describe and distinguish between different defects. This research work was focused on statistical parameters. Especially the shape and scale parameter of a Weibull fit of the distribution seems promising. Based on these features, three machine learning methods were trained and the prediction accuracy compared. Support-vector machines have been proven already before to be very accurate in terms of PD identification, which also was the case in this research with an average accuracy of 96.3 %.

Table 3. Average accuracies of SVM with different Kernel functions and varying the multi-class method of 10 times 5-fold CV

Kernel function	Multi-class method	Accuracy
Linear $d = 1, C = 1$	One-vs-One	93.7 %
	One-vs-All	80.8 %
Quadratic $d = 2, C = 1$	One-vs-One	94.8 %
	One-vs-All	85.7 %
Cubic $d = 3, C = 1$	One-vs-One	95.0 %
	One-vs-All	85.4 %
Gaussian RBF $\gamma = 1/(0.94^2), C = 1$	One-vs-One	61.3 %
	One-vs-All	73.1 %
Gaussian RBF $\gamma = 1/(3.7^2), C = 1$	One-vs-One	96.3 %
	One-vs-All	84.6 %
Gaussian RBF $\gamma = 1/(15^2), C = 1$	One-vs-One	79.6 %
	One-vs-All	71.0 %

Table 4. Average accuracies of k-NN for Euclidean distance with varying k and distance weighting of 10 times 5-fold CV

Method for k	k	Weighting	Accuracy
Fine k-NN	1	-	85.4 %
	3	-	81.4 %
Medium k-NN	10	-	67.7 %
Coarse k-NN	100	-	44.3 %
Fine w-k-NN	3	squared inverse	85.1 %
Medium w-k-NN	10	squared inverse	84.1 %
Coarse w-k-NN	100	squared inverse	77.7 %

Table 5. Average accuracies of ANN with varying number of hidden layers (HL) and hidden neurons of 10 times 5-fold CV

No. of hidden layers	Neurons in HL	Accuracy
One	9	91.4 %
	14	92.9 %
	20	91.4 %
Two	9, 9	88.6 %
	9, 5	90.0 %
	14, 14	90.0 %
Three	9, 9, 9	71.4 %
	9, 7, 5	65.7 %
	14, 14, 14	74.3 %

References

- [1] J. Yu, A.T. Moon, K. Smith and N. M. Macleod, “Developments in the Angle-DC project; conversion of a medium voltage AC cable and overhead line circuit to DC”, *CIGRE Session*, Report B4-202, Paris, 2018.
- [2] IEC 60270, “High-voltage test techniques - Partial discharge measurements”, Edition 3.1, 2015.
- [3] A. Pirker and U. Schichler, “Partial discharge measurement at DC voltage - Evaluation and characterization by NoDi* pattern”, *IEEE Transactions on Dielectrics and Electrical Insulation*, Vol. 25, Issue 3, pp. 883-891, 2018.
- [4] A. Pirker and U. Schichler, “Partial Discharges at DC Voltage - Measurement and Pattern Recognition”, *2016 International Conference on Condition Monitoring and Diagnosis (CMD)*, Xi'an, China, 2016.
- [5] University of Waikato, “Machine Learning at Waikato University” [Online]. Available: <https://www.cs.waikato.ac.nz/~ml/index.html>. [Accessed: 15-Apr-2019].
- [6] The MathWorks, Inc., “MATLAB for Machine Learning” [Online]. Available: <https://www.mathworks.com/solutions/machine-learning.html>. [Accessed: 15-Apr-2019].
- [7] E. Gulski and F. H. Kreuger, “Computer-aided recognition of Discharge Sources”, *IEEE Transactions on Electrical Insulation*, Vol. 27, Issue 1, pp. 82-92, 1992.
- [8] R. Aldrian, G. C. Montanari, A. Cavallini and Suwamo, “Signal Separation and Identification of Partial Discharge in XLPE Insulation under DC Voltage”, *1st International Conference on Electrical Materials and Power Equipment (ICEMPE)*, Xi'an, China, 2017.
- [9] A. Cavallini, G. C. Montanari, M. Tozzi and X. Chen, “Diagnostic of HVDC Systems Using Partial Discharges”, *IEEE Transactions on Dielectrics and Electrical Insulation*, Vol. 18, Issue 1, pp. 275-284, 2011.
- [10] S. Kainaga, A. Pirker and U. Schichler, “Identification of partial discharges at DC voltage using machine learning methods”, *20th International Symposium on High Voltage Engineering*, Buenos Aires, Argentina, 2017.
- [11] I. H. Witten, E. Frank and M. A. Hall, *Data Mining - Practical Machine Learning Tools and Techniques*, Elsevier, Burlington, USA, 2011.
- [12] CIGRE JWG 33/23.12, “Insulation Co-ordination of GIS: Return of Experience - On Site Tests and Diagnostic Techniques”, *Electra* No. 176, 1998.
- [13] L. Hao and P. L. Lewin “Partial Discharge Source Discrimination using a Support Vector Machine”, *IEEE Transactions on Dielectrics and Electrical Insulation*, Vol. 17, Issue 1, pp. 189-197, 2010.
- [14] D. Aschenbrenner, “Computergestützte Teilentladungsfehler-Identifikation: Eine Grundlage für die Bestimmung des Betriebsrisikos von Betriebsmitteln in der elektrischen Energieversorgung”, PhD thesis, Bergische Universität Wuppertal, 2006.