VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Computer Science

ANALYSIS OF TIME SERIES DATA

ING. TOMÁŠ VANTUCH

Field of study: Computer Science and Applied Mathematics
Supervisor: Prof. Ing. Ivan Zelinka, Ph.D.

Ostrava, 2018

PREFACE

Nowadays emergence of the innovative approaches in data science and machine learning, enforces their use in modeling of the real world physical problems, even when they have been already explored and modeled with solid results. The evaluation of a partial discharge activity, as a phenomenon implying malfunction on an observed system, is one of such problems. The motivation of its reexamination and use of new data science approaches is motivated to increase the relevance of extracted knowledge which will be beneficial for the overall detection performance. The original data obtained by a patented metering device, deployed in the real environment, only underlines this need.

This thesis deals with an analysis and feature extraction from the time series data in order to design a robust fault detection mechanism. The robustness means the ability to correctly process an input data with various defects and interferences while focusing only on what is relevant and to gather as much valuable information about it as possible.

The entire work is a set of experimental models and analyses interconnecting a fundamental knowledge of the observed data with modern bio-inspired and soft-computing based machine learning algorithms and optimization approaches. The referential solution inspired by a state-of the art knowledge is designed with adjustable feature extraction process which parameters are further optimized making use of an swarm based optimization. Another models using evolutionary based feature synthesis, wavelet based signal decomposition or denoising driven by weighted singular values serve as the competitors in order to reveal other possibilities in studied problem.

The estimation of entropy, complexity and chaos in the data was supposed to increase the set of applicable features for the detection. The separability of several complexity indicators, like sample entropy, approximate entropy, 0-1 test for chaos and correlation dimension, was examined on data containing all kinds of measured malfunctions. Gathered results were accompanied with a discovery of a significant instability of one testing indicator, which has been found and reported for the first time.

The author's other proposals to represent the partial discharge pattern as a complex network are also novelties and they brought superior results in comparison with the state-of the art based classification models. They solid reasoning and simplicity offered multiple optimizations and evaluations which are documented in this work.

ACKNOWLEDGMENTS

I wish to thank to all who helped me during my studies and during the time I have been working on this dissertation. I would like to thank to my supervisor, prof. Ing. Ivan Zelinka, Ph.D. for his countless advices that shaped my path through the research and also to my boss, prof. Ing. Stanislav Mišák, Ph.D. for his leadership working in the ENET Centre and the opportunities he gave me.

CONTENTS

1 INTRODUCTION 1
- 1.1 Problem description . . . 2
- 1.2 Signal data description . . . 4
- 1.3 Goals and motivation . . . 9
- 1.4 Thesis outline . . . 10

2 STATE OF THE ART 11
- 2.1 Denoising and signal preprocessing . . . 12
- 2.2 Feature extraction and relevancy estimation . . . 13
- 2.3 Applications of machine learning and evolutionary optimization models 14

3 DENOISING AND SIGNAL PREPROCESSING 17
- 3.1 Discrete Wavelet Transform . . . 17
- 3.2 Denoising based on Optimized Singular Values . . . 20
- 3.3 Dimensionality Reduction based Signal Preprocessing . . . 23
- 3.4 PD Pattern Selection . . . 29

4 FEATURE EXTRACTION 32
- 4.1 Fundamentally Based Feature Extraction . . . 33
- 4.2 Features synthesized by symbolic regression . . . 36
- 4.3 Chaos estimation . . . 39
- 4.4 Complex Network Based Signal Representation . . . 50

5 OPTIMIZATION, TRAINING AND EVALUATIONS 58
- 5.1 Brief review of applied algorithms . . . 58
  - 5.1.1 Machine learning models . . . 58
  - 5.1.2 Hyper-parameter optimization . . . 64
  - 5.1.3 Swarm intelligence based optimization . . . 64
  - 5.1.4 Multi objective optimization . . . 66
  - 5.1.5 Evaluation metrics . . . 67
- 5.2 Results review . . . 68
  - 5.2.1 Denoising based on weighted singular values . . . 68
  - 5.2.2 Fundamentally based classification . . . 69
  - 5.2.3 Relevancy of synthesized features . . . 70
  - 5.2.4 Classification of complex networks . . . 71

6 CONCLUSIONS 80

LIST OF FIGURES

Figure 1.1 The scheme of measuring the pulse component of voltage signal including an analogue low-pass filter. 3
Figure 1.2 Measured signal snapshots in a raw state. Failure-free signal with corona discharge pulses (left) possess annotation 0 and fault indicating signal with higher EBN (right) possess annotation 6. 5
Figure 1.3 The sections of the sinusoidal shape (1,2) with the statistically highest occurrence of PD pulses. 6
Figure 1.4 Histogram of pulse amount distribution on the failure-free signals containing at least 1000 pulses. 7
Figure 2.1 Different PD activity representation. The time-resolved data (left) depicted with phase-resolved data (right) (source [1]). 11
Figure 3.1 Block diagram of filter analysis 18
Figure 3.2 Example of the signal snapshot in its raw (up) and denoised (down) state. 20
Figure 3.3 Fault indicating PD pattern signal (left) decomposed into its singular values (right) by SVD. 21
Figure 3.4 UML diagram of false-hit pulse suppression and PD-pattern pulse extraction approach. 22
Figure 3.5 Correlation coef. (left) and Granger causality (right) between original time set and its reconstructed version 28
Figure 3.6 Mutual information (left) and Euclidean distances (right) between original time set and its reconstructed version 29
Figure 3.7 UML diagram of the preprocessing steps that are able to extract the denoised PD patterns. 30
Figure 3.8 Signal processed by the Butterworth filtering. Raw signal (A), the filtered sinusoidal shape from the raw signal to proceed the synchronization (B), synchronized signal snapshot (C,D), filtered sine shape from D (E) and filtered signal by their difference (D - E = F). 31
Figure 4.1 UML diagrams of the entire experiment (left) and the feature extraction combined with the random forest training as a SOMA fitness function (right). 35
Figure 4.2 Corona discharge pulse followed by a dumped oscillation. 36
Figure 4.3 The scatter plot of the i_1 and i_3 indicator's values. 39
Figure 4.4 Plot of signals $1a$ (Left) and $7a$ (Right). 43
Figure 4.5 Plot of p_c versus q_c for $c = 2.6$ and signals $1a$ (Left) and $7a$ (Right). 43

Figure 4.6 Plot of K_c versus c for $N/n_{cut} = 2$ and signals $1a$ (Left) and $7a$ (Right). 43
Figure 4.7 Plot of ApEn, SampEn and CorrDim in dependence of embedding dimension of signals $1a$ (Left) and $7a$ (Right). 44
Figure 4.8 Plot of K versus $\lfloor N/n_{cut} \rfloor$ for signals $1a$ (Left) and $1b$ (Right). . . . 48
Figure 4.9 Medians of maximal values of Approximate Entropy (left) and Sample Entropy (right) calculated on dimensions [2,15] with all fault indicating classes with four different wavelets applied for pre-processing. 49
Figure 4.10 Scatter plot of entropy values of signals with annotation an2 and an6 on embedding dimension 10 to visualize the performed clusters. Bior3.1 (right) and Coif3 (left) waves are not performing visually different clustering solution, neither by calculated silhouette score. 51
Figure 4.11 Signals (left) and their network based representations (right). The failure-free signal with high appearance of corona pulses (up) and the fault indicating PD pattern signal (down). 54
Figure 4.12 Decomposition of signal into smaller overlapped windows (1–8), e.g., window no. 1 is marked along horizontal axis 0 to 1 and window 2 is marked 0.5 to 1.5 allowing windows 1 and 2 to overlap (share) signal portion 0.5 to 1. Examples of four complex networks for four non-overlapping windows 1 , 3, 5, and 7 of the segmented signal are shown, which are obtained using the method mentioned in Table 4.11. 54
Figure 5.1 The tree interpretation of the GP's individual 59
Figure 5.2 Graphical interpretation of Optimal Separating Hyperplane between separated observations of SVM classification (source [2]) . . 62
Figure 5.3 A decision tree with three input variables (b1, b2 and b3). At each of the root and internal nodes (splits), a statistical measure is applied. The values a, b, c and d are thresholds for splitting. A dataset is split into smaller subsets until the terminal nodes (leaves) return the class labels (A, B and C). (source [3]) 63
Figure 5.4 Graphical comparison of classification performance based on application of different threshold levels. 73
Figure 5.5 UML diagram of the entire signal processing and detector model. 74
Figure 5.6 MOO's Pareto front - the set of optimal solutions' multidimensional fitness values. 76

LIST OF TABLES

Table 1.1 Annotations of 7 typical signals of PD patterns. 6
Table 1.2 Number of selected signals according to the kind of the signals, or its annotation. 8
Table 4.1 Estimated mutual information values of the extracted features . . 34
Table 4.2 False-hit pulse suppression parameters defined by a human expert and their SOMA optimization ranges 34
Table 4.3 Operations supported by the adjusted grammar for the GP 38
Table 4.4 Description of 7 typical signals with possible occurrence of the PD pattern applied in this analysis. 40
Table 4.5 Maximum values of ApEn and SampEn in dependence on embedding dimension D, ordered from the biggest to the smallest value. 45
Table 4.6 Correlation dimension calculated on embedding dimensions $2, 3, \cdots 14$ on all selected typical signals. 46
Table 4.7 Sample Entropy calculated on embedding dimensions $2, 3, \cdots 14$ on all selected typical signals. 46
Table 4.8 approximate entropy calculated for embedding dimensions $2, 3, \cdots 14$ on all selected typical signals 47
Table 4.9 Chaos 0-1 calculated for $\lfloor N/n_{cut} \rfloor$ and $2, 3, \cdots 14$ on all selected typical signals. 47
Table 4.10 Silhouette score for the selected clustering solutions 50
Table 4.11 Features extracted from complex networks [4] applied in following experiments. 57
Table 5.1 Performance of classification algorithm according to the applied pre-processing method on the selected subset 69
Table 5.2 Random Forest and SOMA setting in case of fundamentally based denoising experiment 70
Table 5.3 Performance of random forest classification algorithm according to the several different adjustments of the fundamentally processing of the PD pattern signal. 71
Table 5.4 Adjustment of GE for evolutionary based synthesis of non-linear features 72
Table 5.5 Performance of the classification algorithm trained on evolutionary synthesized features. 72
Table 5.6 Tabular comparison of classification performance based on application of different threshold levels. 73
Table 5.7 Settings applied for NSGA-II in MOO's feature selection procedure. 75

Table 5.8 Results obtained from CV of predictors optimized by GSO on all 14 candidate datasets obtained from the Pareto front (PF) selected from case where a dataset represent only signals with high appearance of pulses, i.e., a set of 290 signals. Evaluation measures for each predictor's mean accuracy a, precision p, recall r, and F-score s. 79

1

INTRODUCTION

The dissertation work described in the following document is a set of proposals and applications of the machine learning and soft-computing models on a time series (TS) data. To be more specific, the main topic of the work is the evaluation of a partial discharge pattern, so called PD-pattern detection, examining the measured signal snapshots. These snapshots possess the sinusoidal shape and contain an information about the current state of an observed system - an insulation system of a medium voltage overhead lines. In times of a system malfunction, the partial discharge activity, as an impuls component of a current or voltage signal, occurs in the places of a degradation or rupture of the insulation system.

This is a well known and established research field where many high quality proposals for PD-pattern detection were already designed and evaluated. Their nature incorporates the mathematical and statistical modeling, digital image and signal processing, machine learning, artificial and swarm intelligence as well as evolutionary based approaches. The tasks they are solving are not necessary the fault detection only. There are several subtasks possessing almost equal importance such as the signal preprocessing and noise suppression, feature extraction with the relevancy estimation and the various kinds of optimization for problems like hyper-parameter optimization, variable selection, etc.

Application of a patented metering device, developed by the team in VŠB-TU Ostrava, increases the impact of this work by setting the need for a complex detection system that will incorporates all of the aspects of data processing in order to perform the detection of a highest possible performance. Also the deployment of the metering device is in the forested places and inaccessible terrain which implies several additional requirements carried by the project such as the external background noise suppression, clarity and computational effectiveness of the solutions, low cost and high precision with ability to adapt on different areas where solution can be deployed.

Not all of those aspects are covered by this dissertation work, however several innovative concepts are designed and evaluated with comparison to their ancestors. However, the work looks like having only highly applicational character, the truth is that several unconventional approaches were able to br examined and described with interesting results even when their application is not yet reasonable or possible. Such approaches includes the application of a complex network based feature extractions, a recurrent quantification analysis or the chaos estimation.

PROBLEM DESCRIPTION

Today's technologically developed times are implying a need for a reliable supply of electrical energy to households and the transport, industrial and other sectors. New, advanced technologies and tools of timely fault detection help to faster omitting of faults, thus an increase of reliability of the electrical energy supplies. A system of overhead lines with covered conductors (hereinafter CCs) has been developed in Finland in 1976 [5, 6].

Such overhead lines were built in Norway and Sweden as well, and gradually appeared in other countries across the European Union. Employing CCs in the construction of these overhead lines is not much different from application of the bare wires in outdoor overhead lines; the only difference is in using of the XLPE insulation in the former [7, 8]. By using the insulation, a fault rate of these overhead lines is reduced and it is possible to build the lines in not easily accessible places, e.g. in densely forested areas. Unlike typical outdoor overhead lines with AlFe conductors, when a tree branch falls on the lines with CCs, an immediate interphase shortcircuit does not occur; therefore, the risk of disconnection from the electrical energy supply is significantly reduced [9, 10].

However, in case of CC rupture with subsequent downfall of the line, it is not possible to detect the fault by standard digital relays because the earth fault does not arise [11]. The low-energy current passes through the fault point, which implies that standard digital relays working on current principle cannot detect this fault. Nevertheless, it is possible to detect a partial discharge (PD) activity in fault point, which generates inhomogeneous electric field around a degradation of insulation system.

Two basic methods can be used for indirect evaluation of PD activity:

1. Evaluation of PD activity in the insulation system of CC by measuring the current signal in CC by a Rogowski sensor [6].

2. Evaluation of PD activity in the insulation system of CC by measuring the voltage signal of electrical stray field along the CC.

Both methods use for evaluation of PD activity a pulse component. The pulse component is generated by PD activity in the insulation system and its occurrence is characterized by a frequency domain of hundreds of kH to MHz. The first method evaluates the pulse component generated by PD activity from the current measured by a Rogowski sensor. The main advantage of this method is a high selectivity of evaluation of PD activity. However, this high selectivity of PD evaluation requires both high sensitivity and accuracy of the Rogowski sensor over a wide frequency range because the pulse component is low-energetic and the measurement period is in the order of microseconds. These high requirements on the Rogowski sensor increase a price of the metering device, which is a great disadvantage of this method (1) [6].

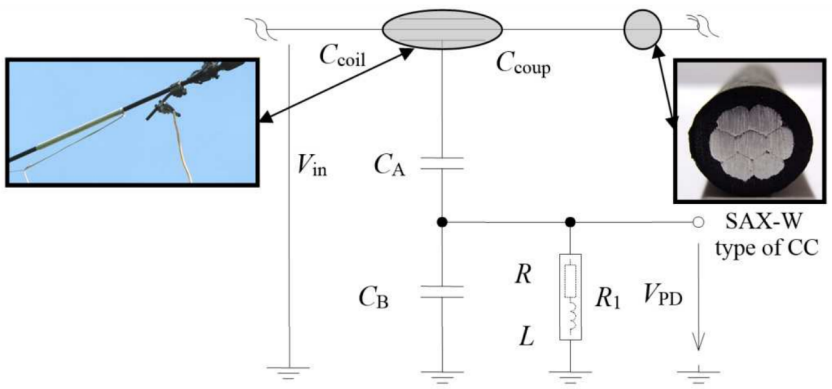


Figure 1.1: The scheme of measuring the pulse component of voltage signal including an analogue low-pass filter.

The metering device applied in these experiments is based on the second method that evaluates the pulse component generated by PD activity as a voltage signal of the electrical stray field measured along an insulation system, in this case the insulation system of CC. The requirements for application of this method were: high selectivity of fault detection, high reliability and a low price of the prototype CC fault detector.

As it was stated in the previous section, the pulse component generated by the PD activity in an insulation system of CC is measured as the voltage signal of electrical stray field along the CC. This voltage signal was measured by a capacitor voltage divider ($C_A - C_B$) from a circular metal sensor (a circular coil; see C_{coil} in Fig. 2) through its coupling capacitor C_{coup} to the CC. As a sensor it is possible to use, for example, an inductor wound on the CC surface, where Ccoup will be given by the number of turns (see Fig. 1.1).

The main requirement for a fault-detecting system is the oscilloscopic measurement of varying the PD-pattern voltage of electrical stray fields around medium-voltage overhead lines with CCs. We used a measuring card connected via a PCI interface for this purpose.

In addition to measuring the PD-pattern, the fault detector measures other conditions, such as temperature, pressure, humidity and global radiation (exposure), since the discharge activity is affected by these factors. These conditions affect, for example, the frequency of PD occurrence, the PD amplitude shape and size .

All measured and processed data are sent by the GSM network to an external computer, where they are processed and an overhead line fault condition is analysed. If the threshold values of indicators are exceeded, the fault detector sends a warning signal to the CC operator. The fault detector also includes additional control electronic circuits which are used to switch the PC to automatic control, and to measure temperature in-

side the switchboard (in the case of unnecessary heating). The device is powered by a battery which is charged by photovoltaic panels. Thus, it is possible to use the detector anytime on overhead lines with CCs, even when there is no possibility to get energy from the distributed system power supply.

A different metering device has been designed and deployed in [rel1]. It is based on measurements of radio waves emitted from all three phases and captured by one anthena device placed below them. Signals obtained by this measurements are under the study, but without results presented in this work.

SIGNAL DATA DESCRIPTION

As it was mentioned previously, the raw signals from a real MV overhead line contain high amount of uncertain information. The reason is an interference of a unique background noise that is almost impossible to simulate in a laboratory and is absolutely specific for each deployment location. Besides the PD-pattern itself, every other signal in the impulse component of a raw signal is considered as a background noise. According to [12], there are several sources of background noise:

- Discrete spectral interference DSI (radio emissions).
- Repetitive pulses interference (power electronics).
- Random pulses interference RPI (lightning, switching operations, corona).
- Ambient and amplifier noise.

The overhead line acts as a long wire antenna, so long-wave transmitters are the most significant permanent source of the DSI. Sometimes, more than 100 radio stations can be identified in the raw signal. Most of them are used for radio broadcasting, some for other purposes (time signals, communication). These sources of DSI can be easily recognizable with Fast Fourier transform (FFT) [13] based on their modulation [14]. Broadcasting of the radio waves is variable and depends on many factors [15, 16]. That is why some of the radio stations are not present in the measured raw signal all day long or their signal is variable (note that not all of them transmit 24/7). Usually, DSI in the examined band is more significant during the night because of the ionosphere condition. A high level of DSI can cover the PD-pattern in a raw signal.

Another significant source of interference on MV overhead lines with CC is RPI, which is most often represented by a corona discharge. Generally, RPI creates false hit peaks in a time domain of a raw signal, which can be misdetected as a PD-pattern.

On the other hand, the applied metering device is capable to detect the PD-patterns of such quality that the various kinds of the system malfunctions should be distinguishable. The central database, that downloads the signals in an hourly based schedule, is

maintained by an expert that annotates the obtained signals by his visual experience or by an additional personal check in a place of deployment. This supervised approach extends the data by the most important information, the fault detection annotation.

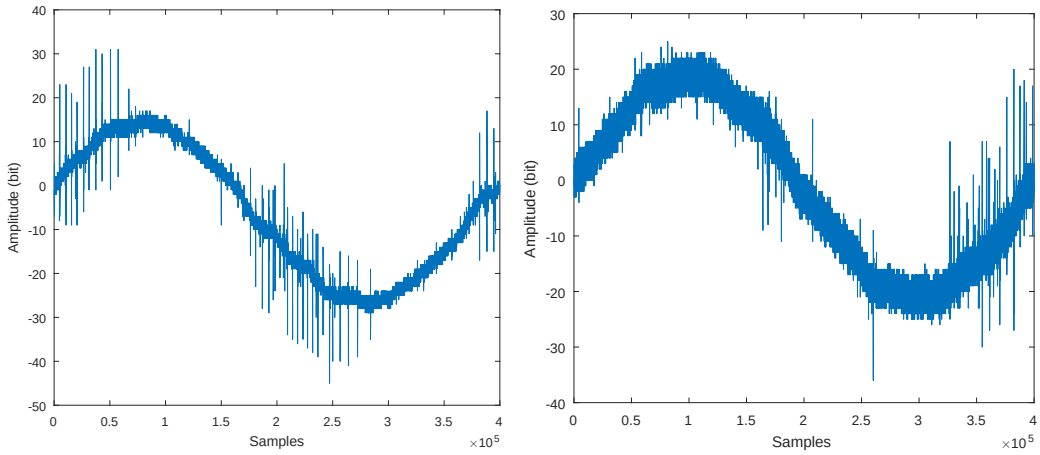


Figure 1.2: Measured signal snapshots in a raw state. Failure-free signal with corona discharge pulses (left) possess annotation 0 and fault indicating signal with higher EBN (right) possess annotation 6.

There are 7 different annotations across the entire obtained dataset, where the only one represents the untouched state of the CC while the rest describe several different states of the system when the isolation is some contact with tree or a branch (see Table 1.1).

The annotations represent various interactions with an ambient objects (trees, branches, etc.) but not all of them represents necessary the fault state. In case of all experiments described in this thesis, it was performed only the binary classification between failure-free and fault state of the system, but also the smallest interactions of CCs with their surroundings (annotation 1 and 2) were considered as a fault states of the system.

An illustrative example of the measured signals can be seen in Fig. 1.2 where two differently annotated signals are depicted.

PD pattern selection

The signal is a snapshot reflecting the entire sinusoidal shape. The one of the most common fundamental features of the PD pattern is the place-related occurrence of the discharge pulses mostly in places close to the extremes of the sinusoidal curve (see Figure 1.3). This allows us to omit the rest of the data to make the experiment more time and data effective while the relevancy of the inputs is kept or even higher. The

Table 1.1: Annotations of 7 typical signals of PD patterns.

Annotation	Annotation description
0	signal indicating no fault on the covered conductor (CC)
1	weak appearance of the event when a tree or branch is in contact with CC and through that the CC is connected to the ground
2	weak appearance of the event when a branch is in contact with multiple CCs and through that the phases are interconnected
3	interruption of the CC fault
4	incorrect behavior due to degradation of the CC insulation system
5	strong appearance of the event when a tree or branch is in contact with CC and through that the CC is connected to the ground
6	strong appearance of the event when a branch is in contact with multiple CCs and through that the phases are interconnected

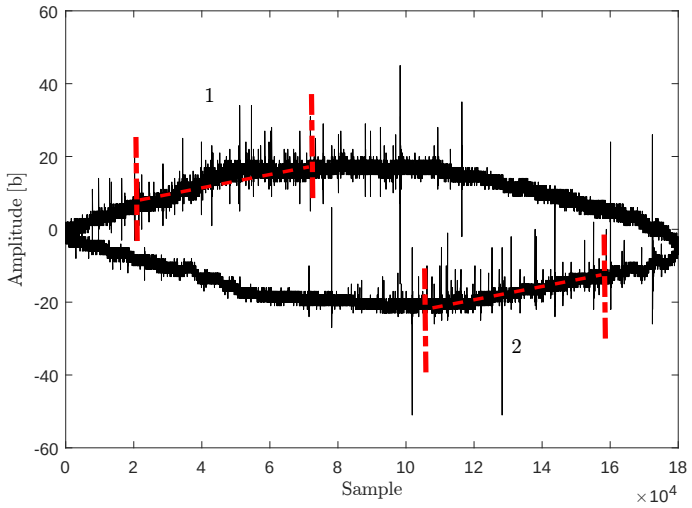


Figure 1.3: The sections of the sinusoidal shape (1,2) with the statistically highest occurrence of PD pulses.

placement of the PD pattern is not strictly on the given indexes of the time set, its position varies with the whether and time condition, kind and distance of the isolation fault, etc.

Subset selection

The necessity to select the representative subset for the following data-mining based experiments relied on fact that the entire dataset contained more than 10.000 annotated signals measured on the one selected location (Hoštálkovy, the north-west area from the town Krnov). Metering devices were placed on more than 10 different places over Czech Republic, but in that time, not everywhere the fault-annotated signals were obtained. The failure of the insulation system on MV overhead lines is a very rare event due to the system's reliability, therefore the number of signals containing PD activity is significantly smaller than the number of failure-free signals (94 to more than 10.000 in that time). This phenomenon is entitled as imbalanced dataset problem and it leads the machine learning algorithms to converge into a solution that predicts the dominant annotation for all given signals. The high amount of signals also rapidly increases the computational time because the dataset contains a lot of redundant informations that are able to be omitted (most of the failure-free signals). The imbalanced dataset problem was addressed in many available studies [17, 18, 19] and it can be solved by proper design of a representative subset (over-sampling or under-sampling method [17]) which is lately applied as training dataset during the experiment.

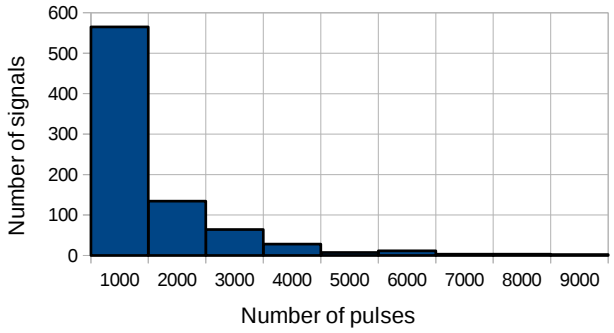


Figure 1.4: Histogram of pulse amount distribution on the failure-free signals containing at least 1000 pulses.

In case of this work, the simple under-sampling method was applied on failure-free signals, because they represented the highest amount of redundancy in the dataset. The number of detected pulses in these signals plays an important role. By simple summation of values above the adjusted threshold (details see in section 4.1) represented the number of pulses in the signal (without separation of false-hit pulses). The histogram

Table 1.2: Number of selected signals according to the kind of the signals, or its annotation.

annotation	0	1	2	3	4	5	6	failure-free	fault
amount of signals	406	10	69	0	0	3	14	406	94

of the pulse count distribution can be seen in Fig. 1.4. It plots only amounts of signals having pulse amount higher than 1000, because number of signals below this threshold is higher than 9000.

The vast majority of the measured signals contained no pulses, which implies the signal indicating a failure-free state of the system. The selected subset took all available fault annotated signals (96) and all failure-free signals with higher amount of pulses (more than 3000), which was additional 118 signal. Another 100 signals were selected visually to cover several kinds of corona discharges and noise interferences. The rest (186) was selected randomly from the three groups, '0 - 1000 pulses' (62), '1000 - 2000 pulses' (62) and '2000 - 3000 pulses' (62). The entire selected subset contained 500 signals with annotation distribution depicted in Table 1.2.

The low or zero amounts of signals annotated by number 3, 4 and 5 were another reason to apply only binary-classification to ensure only separation between fault and failure-free signals. As the dataset increased during the time, the fault indicating annotations were added in the overall dataset but the selected subset was not changed to keep the consistency in all experiments. Only the chaos estimation was executed on all kinds of annotations which required an involvement of these additional signals.

Several variations of this dataset was created in cases when experiments needed so. In case of denoising based on weighted singular values, the dataset was lowered in half by random selection of 50% of each kind of the signal. The evolutionary based synthesis applied only 161 signal equally selected from each class with some additional signals to cover corona discharges and various types of noise. This lowering of the dataset affected the results in a small scale which is also concluded in the summary of this work.

GOALS AND MOTIVATION

The character of this dissertation work is purely computer science, especially the data science and its motivation aims on several aspects of the signal processing data. The necessity of re-examination of the basic approaches based on a fundamental knowledge is obvious, because only by the statistical and experimental evaluation, we are able to confirm its validity on our original data. Their complex nature, as it was stated before, requires to perform a valid data processing of multiple aspects such as the signal preprocessing, noise suppression, feature extraction, variable selection and the classification. All of the mentioned aspects are reflected by experiments described in this work and author paid a special attention to bring some innovative proposal for each of this area.

The proposals are based on evolutionary or machine learning algorithms application and unconventional approaches for feature extraction and noise suppression. The following highlights are stated as the main goals of the dissertation.

- To research a signal preprocessing and feature extraction procedures from the available literature in order to develop a state-of the art detection model
- To interconnect evolutionary, swarm-intelligence and machine learning approaches to increase the overall performance of the developed detector
- To design, develop and evaluate a way of graph representation of the PD pattern data to extend the set of applicable features
- To evaluate the relevancy of applicable features on the given problem by the chaos and complexity based indicators use

THESIS OUTLINE

This work is organized in the same order as the signals are processed. It starts by the review of an available literature partitioned into its main aspects mentioning the main trends and future aims. The following chapters describes only the algorithms and approaches that were used or designed and evaluated. The chapter focused on the signal processing describes the connection between noise suppression algorithms and dimensionality reduction techniques.

The feature extraction procedures are kept in chapter that follows. This is the most extensive chapter of this work, because the extraction of the valuable features is considered as the most critical part in this field and its processing can be inspired by the wast amount of approaches and areas of modeling. The last chapter containing the applied machine learning approaches encloses the work-flow of the performed experiments by listing and describing of the obtained results.

The innovative contributions with references to their deeper description, that are presented in this work, are enumerated in a following list:

- The comparison of the Dimensionality Reduction algorithms in preserving the statistical dependencies, especially the Granger causality was ensured in our study that was published in Procedia Computer Science [11] (see section 3.3).
- The innovative denoising model based on weighted singular values applied for PD-pattern detection is described in section 3.2. It was presented at international conference in Environment and Electrical Engineering (EEEIC) [rel2].
- The set of relevant features synthesized by an approach entitled as symbolic regression was ensured on signal data with achievement of high distinguish ability. This experiment was published in an international conference Intelligent Data Analysis and Applications [rel3], see in section 3.2.
- The first time application of PD-pattern representation by complex network was presented in The Euro-China Conference on Intelligent Data Analysis and Applications [rel3] (see section 4.4).
- The chaos estimation on a signal data with applied noise suppression method revealed the dynamical properties of the PD patterns. Experiment equipped with statistical relevancy evaluations of the measured levels of chaos was published in an international journal and conference.

2

STATE OF THE ART

As it was stated in the previous chapter, the PD pattern detection is a well established field of study for several decades while, during its early beginnings, the detection was maintained by an experienced human expert mostly by a visual examination of the obtained patterns. Nowadays trends aim on the development of an autonomous, algorithmic based models able to correctly detect the insulation malfunction with possible identification of its kind or source. The shift in publishing reflects these trends and the majority of high quality publications are more focused on the methods of the sophisticated algorithms for digital data storing and processing, than the new metering devices. Such algorithms comes from all areas of study including artificial intelligence, signal processing, data analysis, statistic or applied math which will be briefly reviewed in later section.

Throughout the times since PD pattern detection attempts to be automatized, two major signal representation had occurred and gathered the most attention and they are the phase-resolved and time-resolved data representation [20]. Each of them brings different set of features and possible work-flows.

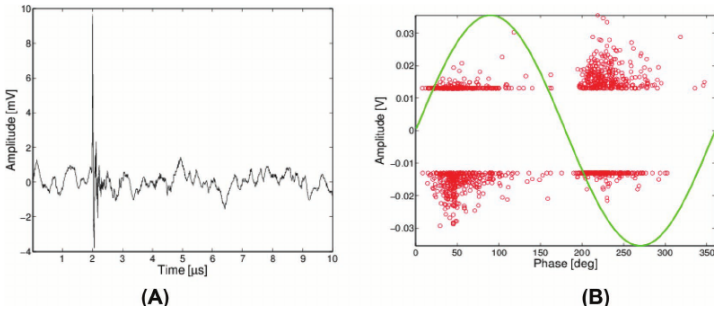


Figure 2.1: Different PD activity representation. The time-resolved data (left) depicted with phase-resolved data (right) (source [1]).

The phase-resolved data are obtained in the relation to the waveform of the ac test voltage, which is held constantly and it is split into suitable number of smaller bins. The PD pulses are quantified by their discharge magnitude (q), the corresponding phase angle or discharge epoch (φ) at which they appear, or their number of densities or discharge rates (n) over some chosen interval of time. The visualizations of these data possess univariate (function of the phase position φ) or bivariate (such as $\varphi \sim q \sim n$)

forms depending on the relation that is examined. In case of univariate forms, several options is available:

- $(q_a \sim \varphi)$ average charge in each phase window vs. φ
- $(q_p \sim \varphi)$ peak charge in each phase window vs. φ
- $(n \sim \varphi)$ number of PD pulses in each phase window vs. φ
- $(I \sim \varphi)$ average discharge current in each phase window vs. φ

Visualization of bivariate data representation results into the 3D plot which is considered as more information containing representation of PD pattern distribution [21, 22, 23].

The time-resolved data, on the other hand, are simply the discharge magnitude level related to time. This brings interesting benefits like plotting the true shapes of the discharge pulses, observing the correlation between PD's signal shape and the nature of its source issue, etc. The quality and features observed from the shape of the PD pulses depends on the sampling rate of the metering device.

All experiments described in this work dealt with the time-resolved data only which is the reason of the further sections aim.

DENOISING AND SIGNAL PREPROCESSING

To suppress the noise interference inside of the given data means, in general, to lower amount of the uncertainty while to preserve the information content relevant to the solved problem as high as possible [24]. All of the existing experiments performed in a real environment faced more or less to one common problem. It is the presence of an external background noise interference that degrades the quality of the observed data, which negatively implies the performance of the detection model [25]. The laboratory based models add a digitally generated noise to test their detectors for a pseudo-real conditions or to create more robust solution for the real conditions.

One of the easiest way how to handle noise is to adjust a threshold value of the maximum discharge amplitude. The pulses lower to this threshold will be ignored [26] while the rest will be considered as the PD pattern pulses. The statically adjusted threshold value is not suitable in applications that require a higher accuracy. In case of a too low threshold value, model falsely considers the noise data as the PD pulses while too high threshold value can ignore them [27, 28]. The thresholding as a denoising technique has not been applied only directly on the signal, but also on the signal describing coefficients obtained by a wavelet based decompositions.

This way of signal preprocessing has been confirmed as the best performing technique due to its variable adjustments and robustness [29]. The discrete wavelet transform (DWT) is more preferred and more frequently used over a continuous wavelet

transform (CWT), because of its simplicity and higher relevancy of the returned data. DWT comes is many adjustments and interconnection with other algorithms for adaptive thresholding of decomposed signals [29] or evolutionary based wavelet selection. The application of dimensionality reduction on an approximation coefficients [30] or a deeper examination of the mother wavelet selection has been also carried out by several studies [31, 32]. These make the DWT even more suitable for such a complex task as the PD pattern detection.

The modifications or extensions of the wavelet transformation technique underline their validity. The extensions like a wavelet package transform (WPT) [33], a second generation wavelet transform (SGWT) [34] or a complex wavelet transform [35] are worthy to mention while todays competitors comes also from different fields of studies.

The denoising autoencoder as the noise suppression technique based on an artificial neural network requires the noisy and noise-free version of each training signal, but after this phase, it is able to be comparable with state of the art techniques while computational costs are reduced [36]. The other data science based approaches with demonstrative success were obtained making use of the support vector machine (SVM) [37] or dimensionality reduction techniques [38]. In case of their linear of statistical character (like SVD or PCA), they are working with the information distributed in coefficients obtained from the data decomposition. Their slight suppression or amplification directly affects the quality of the reconstructed data which was experimentally proven in one of the author's proposal [rel2] and it may also be considered as denoising.

The success rate of the signal's denoising can be measured by signal to noise ratio (SNR) or puls shape distortion [39]. The quality of the denoising model can be also evaluated through the performance of the entire fault detection solution as a implication of higher quality of the features extracted from the denoised data.

FEATURE EXTRACTION AND RELEVANCY ESTIMATION

In case of time-resolved data representation, which is used in all experiments described in this work, the fundamentally based features extractable from PD pattern are based on the true shape of the PD pulse in the time domain. The width, height, position on a sine wave and time that the pulse requires to rise or decay into some percentage of its magnitude has been used in many studies [40, 41, 42]. Features related to the rise and decay of the pulse possess high relevance towards the PD pattern occurrence therefore they are considered as the valuable features. Their measurement requires higher sampling resolution of the metering device which directly increase the final cost of the solution. This is the true reason, of their absence in many detections where they are substituted by other sophistic approaches attempting to gain the comparable relevance. The set of the derived features from the basic PD pattern data may contain

the average discharge current, quadratic rate, discharge power, repetition rate, peak discharge magnitude and average discharge magnitude [20].

Other approaches, not demanding an expert level knowledge, apply an automated feature extraction based on an evolutionary, genetic-like computation that are able to synthesize a set of features from a given set of signals. A symbolic regression especially, like genetic programming [43] or grammatical evolution [44], is considered as ideal technique meant for this purpose. It has been applied in author's study as an experiment inspired by success in similar domains [45, 46]. The evolutionary based synthesized features from a raw or preprocessed signals is necessary to be driven by some information criteria measures like Gini index [47], information gain [48], mutual information [49] etc. as well as it may be executed on a raw signal data or a signal coefficients obtained by an applied wavelet based decomposition [rel6].

The study of the PD pattern detection based on a recurrent quantification analysis [50] performed the evaluation of the signal data based on a systems dynamic modeled from its state recurrences [51]. These are obtained by a reconstruction of the signal phase space. Similarly, such reconstruction may be applied to evaluate the hidden dynamic, the so called chaos, inside of the time set [52, 53]. Such experiments were performed also by the author, describing wider set of the chaos and complexity indicators with statistical evaluation of their relevance [rel7, rel8].

The relevance estimation is a vital step in every data-mining task in order to properly evaluate the information value of the obtained features and those having the highest relevance, apply in further data driven modeling [54]. There are several ways how to estimate the relevancy of a given feature, all of them expect the presence of an fault-indication annotated signals. The relevance is then a level of a similarity between these annotations and the examined feature or several features. The correlation coefficient is considered as one of the simplest linear evaluations [55]. On the other hand, one of the non-linear evaluation representatives is definitely the mutual information which is possible to be estimated in various ways [56, 57]. The transfer entropy, as a purely non-linear evaluation, is another option that may be considered [58]. The mentioned criteria could be find almost in all available studies [59, 60], however it is not necessary to apply the mentioned criteria in order to evaluate the entire set of features for their relevancy. The final performance of the modeled detector applying the examined set of features may serve to this purpose as well.

APPLICATIONS OF MACHINE LEARNING AND EVOLUTIONARY OPTIMIZATION MODELS

The detection models derived from the extracted features are mostly based on the concept of supervised learning due to the presence of fault indicating annotations. They play the role of an expected outcome of the detector.

The competitive learning in self-organize maps (SOM) is a typical representative of an unsupervised learning [61]. This concept does not require the fault indicating annotations, the inputs perform an excitation of an output layer which is clustered in a clearly separable groups. Each group can represent different part of an examined dataset, ideally possessing the same annotation. SOM is frequently applied as an additional layer of a feature processing [62], but there are also many successful proposals using this concept as the fault detection model [63, 64].

The supervised based artificial neural networks form probably the largest group of an applied machine learning models in this field. The basic n-layered networks (MLP - multi layered perceptron) trained by a back-propagation learning algorithm [65] serve as classifier from early nineties [66, 67]. Their simplicity in learning from the given set of annotations and decent performance are the reasons of a stable interest from researchers [68, 69, rel6]. The learning vector quantization [70, 41], counter propagation neural networks [71] or more modern extreme learning machine [72] or deep learning [73] were also successfully applied in the PD pattern detection.

The disadvantage of ANN based detectors is a difficult representation of the systems' constants, the so called "back box concept", when predictability of the model is rather difficult and unclear [74]. The support vector machine is a typical competitor of ANNs in various classification and regression tasks. The linear modeling of an optimal hyperplane, separating the classified observation with their transition into a higher dimension through the kernel function, is the basic description of this model. It also has been applied uncounted times in the fault detection problems [75].

A simpler, condition based class of the machine learning models is the decision tree (DT) [76] with its various kinds of implementations [77]. DT offers a simple but well performing and robust classification model which can be optimized due to several options to handle outliers and keeping the final solution unbiased and general. Combined with genetic optimization [78], fuzzy logic [79] or clustering [80] it was able to create well performing solution.

An approach to make the classification model more robust and precise is to train several models, each of them on different part of dataset, and combine their final decision through a voting mechanism. It may be simply the averaging of their output values or a multiplication by adjustable weights can be used. This entire concept is called boosting mechanism [81] and, for the first time, it was performed as the multiplication of DTs into a random forest solution [82]. Since than, almost every possible machine learning model has been multiplied into an ensemble of itself with an increase of its performance [83, 84, 85].

All of the mentioned models are designed to be trained by some given heuristic or gradient based algorithm. This means to optimize their structure and inner set of weights or constants to the given data to perform the closest possible approximation of the expected values (annotations). This process is driven by a defined set of hyper-parameters.

Such a parameters can be, for example, the number of hidden layers of a network, the number of trees in a random forest, the tolerance constant C in the support vector machine etc. The setting of these hyper-parameter is the user's responsibility and it may be achieved in several ways. It can be adjusted by a human expert with a deeper knowledge about the applied algorithm or it can be adjusted just arbitrary in repetitive attempts keeping the best solution. This trial-and-error or expert based adjustment methods are not very recommended, because not everyone can assume how the algorithm will work on a given data and the trial-and-error method does not guarantee a proper statistical testing. The grid search or randomized search are on the other hand well described, frequently used and pretty reliable methods how to optimize a given model [86, 87]. Some researchers attempt to substitute these approaches by an evolutionary based optimizations or swarm intelligence inspired models [88, 89]. They perform good convergence towards the solution with simple adjustment and were also applied in some of the author's studies.

Concluding this chapter, it is worthy to mention that today's trends are still using the fundamentally based features due to their high relevance as well as seeking for a new one. The signal decomposition, in attempt to properly handle the multiple PD sources or the noise interference is also one of the main directions of todays research. The single source PD classification, according to the review studies has still a space to improve [20, 90]. In case of the machine learning models, the hybridization of models like ANN trained by PSO [91], SVM trained by PSO [92] or firefly based optimization [93] can increase the performance in this field as it is performing in others.

3 DENOISING AND SIGNAL PREPROCESSING

Described preprocessing models, in the following sections, were applied or entirely designed by the author. All of them, as a preprocessing parts, were also presented on an international conferences or published in an academic journals. Due to their occurrence in the following chapters, I am briefly describing the models and encouraging you, the reader, to follow the references for more informations.

DISCRETE WAVELET TRANSFORM

Wavelet transform (WT) [94, 95] has been widely applied in many engineering fields for solving various real-life problems, not only monitoring of power quality disturbance [96, 97] but in EEG signal classification [98] or image processing [99].

In this area of study, the wavelet transform comes with better results compared to Fourier transform [100, 101] due to the more flexible way of time–frequency representation of a signal by allowing the use of variable sized analysis windows.

The wavelet transform is the projection of a discrete signal into two spaces: the set of approximation coefficients and the set of detail coefficients. There is an effective implementation for obtaining these coefficients, developed by Mallat [102] and it is working by passing the signal through the set of low-pass and high-pass filters, as it is show in fig. 3.1.

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n-k] \tag{3.1}$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n-k] \tag{3.2}$$

Various high quality software packages are available for wavelet analysis and by this experiment the MATLAB wavelet toolbox was used.

Level and wavelet estimation

In this procedure, two adjustable parameters are critical. It is the applied mother wavelet w and the maximal level of decomposition j. Their adjustment has been performed in many studies experimentally or by a designed heuristic. Several approaches for the DWT parameters estimation has been set and successfully applied in tasks related to signal decomposition.

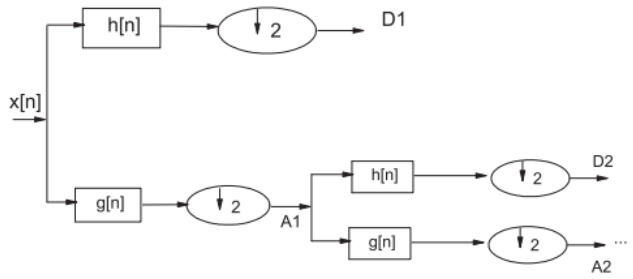


Figure 3.1: Block diagram of filter analysis

There are two main directions to evaluate the selection of decomposition level and mother wavelet. The first represents the evaluation by comparison of the reconstructed signal to its raw version and the second is performed by a calculation of the qualitative parameters on the decomposed coefficients.

The comparison of the reconstructed signal X^r to its raw version X has been performed by calculation of the mean square error (MSE) [103] or by estimation of their correlation coefficient (*cor*) [32, 104].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_i^r)^2 \tag{3.3}$$

$$corr = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(x_i^r - \bar{x^r})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2(x_i^r - \bar{x^r})^2}} \tag{3.4}$$

An alternative to those methods is an adaptive Energy based wavelet selection method (EBWS) presented in [104] which computes an energy on each decomposition level from its coefficients.

$$E_{aj} = \frac{\sum_k (a_{j,k})^2}{\sum_k (a_{j,k})^2 + \sum_i^j (d_{i,k})^2} \tag{3.5}$$

All of these methods are valid for the adjustment of the signal decomposition, however a new approaches still appears in improvement of the noise suppression or the simplicity of estimation [31]. In general, it is always worthy to rely on more metrics during the optimal wavelet and level selection. This approach was used during adjusting of DWT on our data as well. The results can be seen in Table ....

Threshold estimation

The threshold λ, as it was mentioned previously, may present the back-bone of the denoising procedure. There are several ways how to estimate its value according to available literature [105, 106]

- A universal threshold value: $2\sqrt{log(n)}$
- A Stein unbiased risk estimate (SURE) [107]: $\sqrt{2log_e(n * log_2(n))}$
- Minimax estimation based on MSE [108].
- An extensions of SURE, minimax, etc.

As everyone would suggest, there is also an opportunity to adjust the threshold value by an evolutionary based approach [109]. Once the threshold value is set, its application has to be decided. There are two general ways to apply the threshold value, the soft and hard thresholding.

$$T_{hard}(x) = \begin{cases} x & \text{if } |x| > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

$$T_{soft}(x) = \begin{cases} x - \lambda & \text{if } x < \lambda \\ 0 & \text{if } |x| \leq \lambda \\ x + \lambda & \text{if } x < -\lambda \end{cases} \tag{3.7}$$

Denoising procedure

The entire denoising procedure based on DWT must involve all previously mentioned steps. Author's experiments, that used wavelet based denoising, applied the hard thresholding on detail and approximation coefficients mostly. The threshold value calculation was inspired by SURE but extended [110]. The Haar wave as the mother wavelet and only one level of decomposition were selected as an optimal solution. The complete steps of the denoising procedure are listed below.

1. Perform a discrete wavelet transformation (DWT) to obtain wavelet coefficients $w_j(x)$ on level j from signal x which consists of approximation c_a and detail c_d coefficients.
2. Estimate the threshold T_d at decomposition level j via (MAD stands for Mean Absolute Deviation):

$$\sigma = 1/0.6745\,\mathrm{MAD}(|c_d|) \tag{3.8}$$

$$T_d = \sigma\sqrt{2\log(n)}\,. \tag{3.9}$$

3. Perform a hard threshold operation on detailed coefficients and truncate approximation coefficient which will lead to suppression of the sine shape.

4. Reconstruct a denoised version of the original signal from the thresholded coefficients.

An example of a reconstructed signal can be seen in Fig. 3.2.

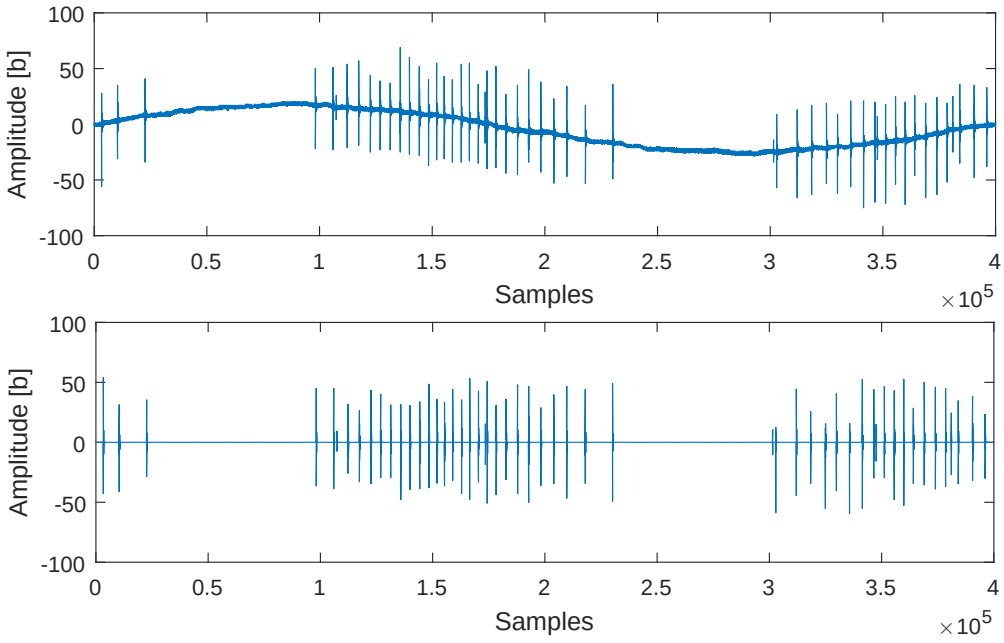


Figure 3.2: Example of the signal snapshot in its raw (up) and denoised (down) state.

DENOISING BASED ON OPTIMIZED SINGULAR VALUES

The novel denoising oriented algorithm was proposed by author's study [rel2]. The algorithm combines two widely known machine learning techniques, it is the Singular Value Decomposition (SVD) for decomposing the matrix data into its singular values [111] and the Particle Swarm Optimization [112] for their supervised driven optimization.

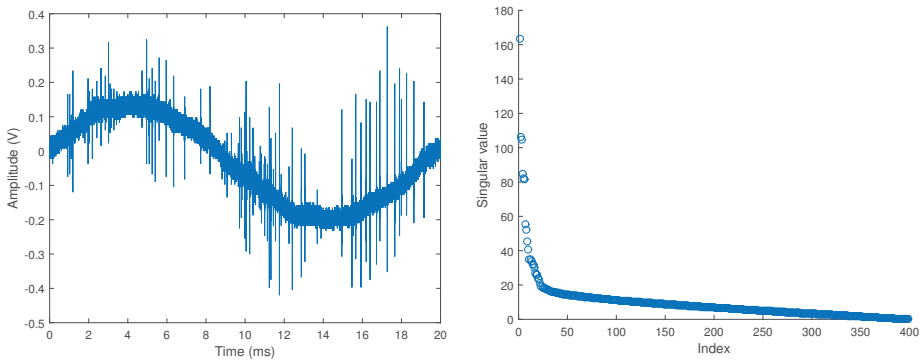


Figure 3.3: Fault indicating PD pattern signal (left) decomposed into its singular values (right) by SVD.

For every given matrix A, an orthogonal matrices U, Σ and V exist where $A = U\Sigma V^T$. The matrix Σ is a diagonal with nonzero diagonal entries, called singular values. The square roots of the nonzero eigenvalues of matrices AA^T and A^TA are equal to singular values ($\sigma_1 \cdots \sigma_k$) of the matrix A. Singular values are decreasingly ordered with a usually exponential decay where the highest amount of information is stored in the first few singular values. A common approach is to use only theses k greatest singular values to perform k-reduced singular value decomposition of A. The rest of the singular values are normally insignificant in relation to the information content and possess mostly an uncertainty or noise. Several research proposals, inspired by this fact, were able to derive an image denoising algorithm based on singular values [113, 114].

In case of PD pattern analysis, especially in case of our data, the situation is rather different. According to our analysis, the most of the information content is related to the shape and placement of the sine wave and the dominant part of noise interference. The PD pulses do not represent, in case of insulation fault, more than 5% of the signal values.

To perform a valid denoising of PD pattern signals, we can not rely on the highest singular values but rather the small singular values spread at the tail of the graph (see Fig 3.3). The author's proposals, as it was mentioned before, employs a PSO for this optimization. The details about the PSO algorithm may be found in section 5.1.3.

The algorithm runs in several simple steps that are listed below

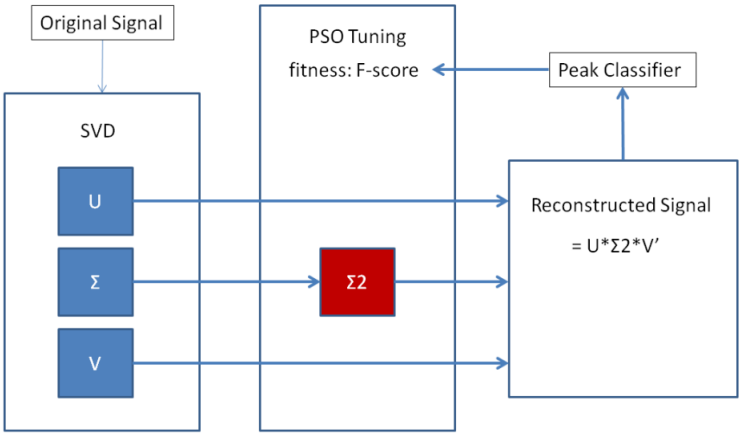


Figure 3.4: UML diagram of false-hit pulse suppression and PD-pattern pulse extraction approach.

1. Signal s of size n is transformed into matrix A by Simplified Trajectory Matrix approach:

$$A_{j,f} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,f} \\ a_{2,1} & a_{2,2} & \dots & a_{2,f} \\ a_{j,1} & a_{j,2} & \dots & a_{j,f} \end{bmatrix} \tag{3.10}$$

   where f is the frame size and j is equal to n/f.

2. The singular values $(\delta_1, \delta_2, \dots, \delta_f)$ are extracted from A by SVD.

3. The random vector of weights $W = w_1, w_2, \dots, w_f$ is initiated as a weights for singular values. The denoising singular values δ^d for obtaining of the reconstructed denoised signal are computed by multiplication with their weights where each i-th δ_i^d is equal to $\delta_i \times w_i$.

4. PSO driven optimization is ensured to optimize the weights W according to the adjusted fitness function (the best possible performance of the classification algorithm applied on the reconstructed signals).

The diagram of the PSO-SVD algorithm is depicted in Fig. 3.4.

The entire process of the optimization is driven by quality of the performance of the classification algorithm. This algorithm may be chosen arbitrary and may contain additional steps of feature extraction. In this case, only number, average height, width and position of pulses were extracted and applied as input values for the machine learning model. In [rel2], a simple threshold based classifier was applied as a detection model. The amount of found pulses was the only input variable considered for its decision. Its

simplicity and low computational complexity served as its advance while lower classification performance is considered as its weak spot. The experiment was, on the other hand, focused on improvement of the overall performance by proper signal preprocessing and the results as reviewed in final section 5.2.3.

DIMENSIONALITY REDUCTION BASED SIGNAL PREPROCESSING

The experiment with similar nature, as the previously described denoising, was ensured by the author in study [rel4]. The experiment was focused on dimensionality reduction (DR) algorithms and their ability to preserve the statistical dependencies between the raw signal and its reconstructed version. The hypothesis was almost similar, as the DR algorithm reduces the given data into the smaller dimension, the higher amount of information is lost. The estimations of dependencies between reconstructed and raw versions of the signals revealed, when the noise is reduced and when we lost the information content.

The time series from the stock market titles were applied for this experiment. All of them formed the matrix of an input data, which was further reduced and reconstructed.

The representatives of DR's algorithms were chosen on purpose to cover the most typical kinds of them. It was the Principal Component Analysis [115], Non-Negative Matrix Factorization [116], Neighborhood Preserving Embedding [116] and the Autoencoder [117]. Those algorithms were reducing the matrix containing the time series representing the market titles. Iterative reductions and backward reconstruction revealed the levels when the amount of information inside of the signals vanished and relevancy dropped to zero. This study revealed also which of the DR's method is more suitable for keeping the most of the statistical dependencies inside of the time series data during higher number of reduction and backward reconstruction.

BRIEF OVERVIEW OF DIMENSIONALITY REDUCTION METHODS

To reduce the dimension of a given dataset means to obtain dataset with lower dimension. The given and reduced datasets are possessing a matrix form, which one dimension is being reduced. Each of the reduced observations is than described by the lower amount of un-correlated features (variables). These extracted features are usually some linear or non-linear combinations of the input variables [118, 119].

Principal Component Analysis

(PCA) is a linear DR technique [115] that reduces the data by finding a few orthogonal linear combinations of the original variables with a largest variance. The number of PCs is equal to the amount of the original variables, while only few of them hold the

maximum variance. It is the reason why the rest of the principal components can be disregarded with minimum loss of information [120].

The mapping matrix M is found by equation $cov(X)M = \lambda M$, where $cov(X)$ is a covariance matrix of the input data and the matrix λ contains the eigenvalues (PCs) of the covariance matrix on a diagonal. The columns of the matrix M are sorted according to the levels of eigenvalues of the matrix λ. Reduced representation of the input data (matrix Y) is therefore computed from the first d principal components by the equation $Y = XM_d$.

Non-Negative Matrix Factorization

(NMF) is a linear, non-negative matrix DR approach [116]. The idea of this method is to estimate two matrices W and H as a decomposition of a given matrix V of N vectors. The matrix W is $N \times M$ matrix of the basis vectors and H is the obtained new low-dimensional representation of the given matrix V.

The Alternating Least Squares (ALS) algorithm represents one of the simplest mathematical ways to obtain such $W \times H$ representation for the given matrix [121]. The back-bone of this approach is the switching between two phases - once the H is fixed and W is found by a non-negative least squares solver and later W is fixed while H is solved analogously. This methodology is based on the knowledge that NMF optimization function is not convex in both W and H properties, but it is convex in either W or H. It is worthy to mention that $W \times H$ representation is never equals to V and the matrix of residuals is always produced. Their absolute value will also affect the results of the experiment.

Autoencoder

The autoasociative neural network encoder (autoencoder) is the DR model based on Multi-layered perceptron [117]. Based on the adjusted activation function of the networks' neurons, this model may be linear (the solution is strongly related to PCA) or non-linear (in case of commonly used sigmoid activation function) [122].

The structure of the ANN consists of input and output layer, which possess the same size, and a variable number of the hidden layers with adjustable amounts of neurons. The input is set to the given signal, while the output is expected to be the same. The middle part - the bottleneck is made of smaller dimension and occurs in one or more hidden layers. The output values of this hidden layer are the low-dimensional representation of the input values.

The autoencoder may be trained by the Backpropagation learning algorithm [65], which is based on a gradient descent method.

Neighborhood Preserving Embedding

(NPE) is a linear approximation to the Local Linear Embedding (LLE) [123], which is the manifold oriented non-linear DR method [124]. NPE shares some aspects also with the Locality Preserving Projection [125]. Both of them are focused on the projection of a local structure of the manifold, but the objective functions of these approaches are different.

The LLE algorithm comes through three steps. The first step constructs the adjacency graph. There are two conditions which implies the edges between the nodes. It could be decided by kNN (if the point i is the near neighbor of the point j) or by maximal distance adjusted by a threshold value (i and j are connected if $||x_j - x_i|| \leq \epsilon$).

The weights of the connections are computed in the second step. Let W denote the weight matrix where W_{ij} represents the weight of the edge between points i and j. The computation of the weights is achieved by the following objective function:

$$min \sum_i ||x_i - \sum_j W_{ij} x_j||^2 \tag{3.11}$$

where $\sum_j W_{ij} = 1, j = 1, 2, ...m$.

The last step is the computing of the projection, which is performed by solving of the generalized eigenvector problem.

$$\begin{gathered} XMX^T a = \lambda XX^T a \\ where \;\; X = (x_1, ...x_m); \;\; M = (I - W)^T (I - W); \;\; I = diag(1, ..., 1) \end{gathered} \tag{3.12}$$

The column vectors $a_0, a_1, ...a_{d-1}$ are the solution of equation (3.12), ordered according to their eigenvalues ($\lambda_0 \leq \lambda_1 ... \leq \lambda_{d-1}$) and it is compound into transformation matrix $A = (a_0, a_1, ...a_{d-1})$ so the formula of the reduction can be written as $y_i = A^T x_i$.

INFORMATION CRITERIA MEASURES

The dependency measures represented by the information criteria were aimed to estimate the information loss after the reductions. The first simple comparison metric was the Euclidean distance, which simply computes the similarity between two given coordinates in their N-dimensional space.

The other widely known measurements compare these time series by their statistical dependencies. The correlation coefficient as the covariance between X and Y divided by product of their standard deviation is very familiar linear dependency evaluation and was already described in Eq 3.1. The significance of the dependency is considered by the absolute difference of the resulted value from zero. The negative value of the correlation coefficient indicates the anti-correlation meaning the progress of time series demonstrates the opposite moves.

Mutual Information

(MI) is the non-linear, widely known, evaluation of the dependency between two random variables [126]. In other words, MI reveals how the presence of signal X decrease the level of uncertainty of the signal Y. This estimation is frequently applied for feature selection in the fields of data mining and machine learning.

In case of the discrete variables, the MI is defined by joint probability ($p(x,y)$) and marginal probabilities ($p(x)$,$p(y)$) of X and Y.

$$I(X;Y) = \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{3.13}$$

In case of continuous variables, when the probability distribution function (PDF) is unknown, the MI's estimation is not an easy task and it can be performed by various ways [127].

In this paper, it was used the Kernel Density estimation (KDE) [128], which can return more precise results with lower amount of adjustment.

Granger causality

(GC) is a proposal of Clive Granger [129] for the evaluation of causal interaction between two time series. His procedure quantifies how the variable Y helps to predict the variable X ($\mathcal{F}_{Y->X}$) in a linear manner by an estimation of their vector-autoregressive (VAR) model and computing the covariance matrices of their residuals.

In this paper there was employed the simplest unconditional, time-domain form. The calculation of G-causality follows this simple steps provided by a Matlab toolbox [130] entitled as Multivariate Granger causality (MVGC). Let's suppose we have two stationary and trend-free variables X and Y and their VAR model possess a following form

$$X_t = \sum_{k=1}^{p} A_{xx,k} \cdot X_{t-k} + \sum_{k=1}^{p} A_{xy,k} \cdot Y_{t-k} + \varepsilon_{xx,t} \tag{3.14}$$

The Acaice Information Criteria (AIC) [131] performs the proper adjustment of a number of lagged values of X and Y. To obtain a stable VAR model [132], it has to be tested for co-linearity, stationarity, heteroscedasticity, etc. In this equation, the dependency of X on Y is captured in its coefficient matrix A_{xy}. If all of these are equal to zero, we can decide that X is independent from Y and the following regression of X has an equal forecasting performance

$$X_t = \sum_{k=1}^{p} A'_{xx,k} \cdot X_{t-k} + \varepsilon'_{xx,t} \tag{3.15}$$

By this regression (3.15), it is written that variable X depends only on its past values and the added white noise. The unconditional, time-domain GC is than computed by the following equation

$$\mathcal{F}_{Y->X} = ln\frac{|\Sigma'_{xx}|}{|\Sigma_{xx}|} \tag{3.16}$$

where Σ'_{xx} is the covariance of the residuals of the regression model (3.15) and Σ_{xx} is the covariance of the residuals of the VAR model (3.14). Finally it is computed the p-value that that could reject the null hypothesis of zero causality, which is the following:

$$H_0 : A_{xy,1} = A_{xy,2} = ... = A_{xy,p} = 0 \tag{3.17}$$

By going back to the experiment, it has to be mentioned that the experiment was focused on measuring changes in dependencies between the time series [rel4]. It was achieved by computing dependency between the random time set x_i from the original dataset and its reconstructed version x_i^r from the reduced dataset.

The DR's methods implementation was provided by Matlab Toolbox for Dimensionality Reduction [133], except the implementation of NMF, which is already included in Matlab. The PCA is a non-parametric DR, but every other methods needs to be properly adjusted. The Autoencoder was used with one hidden layer, which number of units inside reflect the dimension of mapped matrix. The activation function was the sigmoid and then number of learning iterations was 5000 for each testing. The NPE method obtains the k value (number of nearest neighbors) equal to 6 and the ALS algorithm for NMF obtained the number of iterations equals to 300.

Reconstruction of the dataset

Reconstruction of the dataset's matrix was provided differently for each of the DR's method according to its character. The PCA's reconstruction was performed in two steps. The first one was simply the computation of the product of the mapping (matrix of eigenvalues) and mapped matrices and the second step was adding a matrix of the mean values of the original dataset.

The NMF's reconstruction was computed as a product of W and H matrices as it is defined in the method's description.

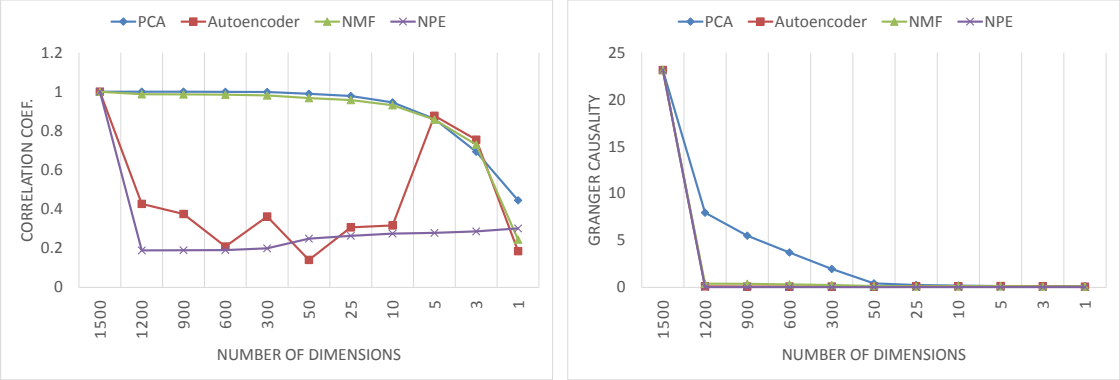
The reconstruction of mapped data from Autoencoder is performed during each of the learning's iteration, because the neural network attempts to recreate the input sequence on the output layer. To perform the reconstruction, it is necessary to apply the learned network weights and put the mapped data as input values into the hidden layer. The output layer of the network will return the reconstructed data.

In case of NPE, the reconstruction was performed the same linear way as it was computed in case of PCA. The product of eigenvalues and mapped matrix is summed with the vector of means of the original matrix. This results into the lower quality of reconstruction, because the eigenvalues does not reflect the statistical behavior of the dataset, but the neighbor connections (adjagency matrix).

Results

There was taken 100 random time series of the original dataset compared to their reconstructed versions and the results are the medians of the measured values. The measurement of the correlation (Fig. 3.5) reveals that in case of PCA and NMF the reduction of dimension does not significantly affect the level of correlation until the reduction reaches the level of 95%. These DR's methods were dealing with this feature reasonably well, but on the other hand the reconstructed time series from Autoencoder and NPE were absolutely uncorrelated with their original versions.

Figure 3.5: Correlation coef. (left) and Granger causality (right) between original time set and its reconstructed version



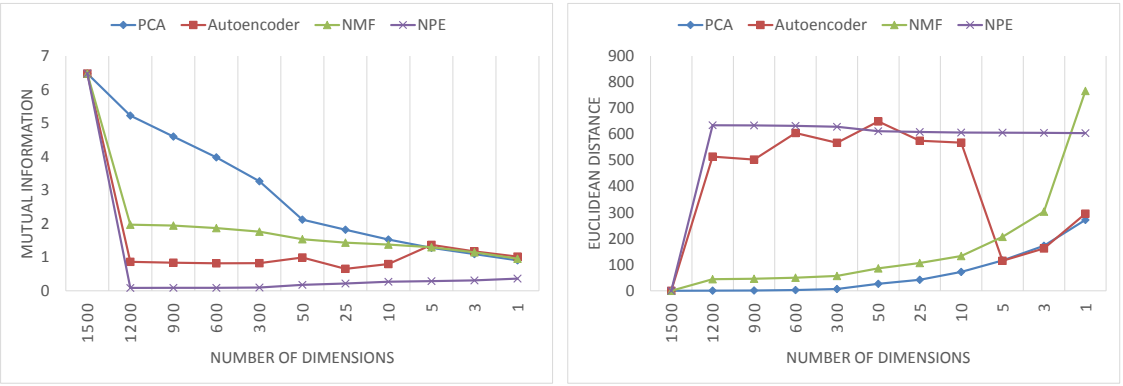
From the compared methods, the PCA also seems to be the ideal choice in times of keeping the maximal amount of Granger causality (see in Fig. 3.5). The focus on the variance through the eigenvalues was able to capture the behavior of the time series mostly until the reduction reaches the border near to 4% of original data set. The area between 50-25 was the most frequent place where the p-value drops under the level of significance - so there was no more presence of causality between compared time series.

In cases of other DR algorithms, the return values of GC were obtained very low and p-values were mostly confirming the null hypothesis of zero causality during all of the iterations.

The Fig. 3.6 describes the amount of MI between original and reconstructed time series. The DR methods like PCA and NMF were more successful than the rest of

the methods in this criteria. The success of the Autoencoder strictly depends on its learning ability by its adjusted structure. This is the reason why most of its results are so inconsistent.

Figure 3.6: Mutual information (left) and Euclidean distances (right) between original time set and its reconstructed version



The values of Euclidean distance (Fig. 3.6) reveal that the maximal similarity (lowest values) was obtained again only by PCA and NMF algorithms. These methods obtain the smallest reconstruction error during the most of the reduction levels. The Autoencoder was not able to learn the given dataset for correct reduction, even if the number of its iterations was 5000. The NPE method deals with very high reconstruction errors, but as it was mentioned previously, it does not indicate the low quality of reduction, only different aim.

PD PATTERN SELECTION

This chapter served as an overview of the tested and applied methods for the signal preprocessing and the most of them brought valuable increase of performance of the entire fault detection model. To conclude this chapter, the general UML diagram is depicted in Fig. 3.7 to visualize the signal processing steps responsible for the noise suppression and maximization of the data relevance. Some of the following steps do not possess any research-like nature, they are rather simple, but in the fundamental view of the problem, they are necessary.

The metering device taking the snapshots of the voltage sinusoidal curve is not synchronized with any timing device and the measured sine waves are therefore not the same for all the measurements. They starts from any arbitrary place of the sine wave and in order to find the PD pattern in the signal correctly (see section 1.2), the signal needs to start from zero and continue by an increase towards one.

Figure 3.7: UML diagram of the preprocessing steps that are able to extract the denoised PD patterns.

To obtain such shape from any randomly time-shifted sine wave, we can simply locate the places where sine is crossing the x axis from its negative side and split the signal into two parts at this point. The part of the wave from right side will be switched to the left and sub-signals may be simply concatenated, because occurrence of the PD and noise patterns is almost repetitive in the closest sine waves.

In order to clearly detect pulses in the signal time set, the suppression of the sine wave may be helpful. Pulses in the time set of a zero mean are easily separable as a deviation from zero. This process has to be done after the sine synchronization, because it will be not possible otherwise. In both operations, the simple Butterworth filter may be used. The sine wave can be filtered by cut-off frequency 50Hz. The filtered sine wave is still noisy and imperfect, because of strong external background interference and the true shape of the wave. This true shape depends on the load which can alter the frequency in the range between 49.5Hz and 50.5Hz. Although, the filtered wave serves as indicator of its time-shift and later it is subtracted from the synced sine wave to perform its suppression (see Fig. 3.8).

Sine shape suppression

The Butterworth filter (BF) [134], commonly used in gait analysis applications, may be applied in the signal preprocessing step in order to suppress the sine shape. The only parameter which needs to be set by the user is a cut-off frequency, assuming that the order and number of passes are constant. The effect of BF application with defined cut-off frequency 50Hz is depicted on Fig. 3.8. On the other hand, there are also other ways to obtain comparable signal modification, wavelet decomposition followed by truncation of approximative coefficients and signal reconstruction can perform similarly.

After these procedures, the signal is denoised and two relevant parts may be cut out to decrease amount of processed data, which will positively affect effectiveness and precision of the detection model.

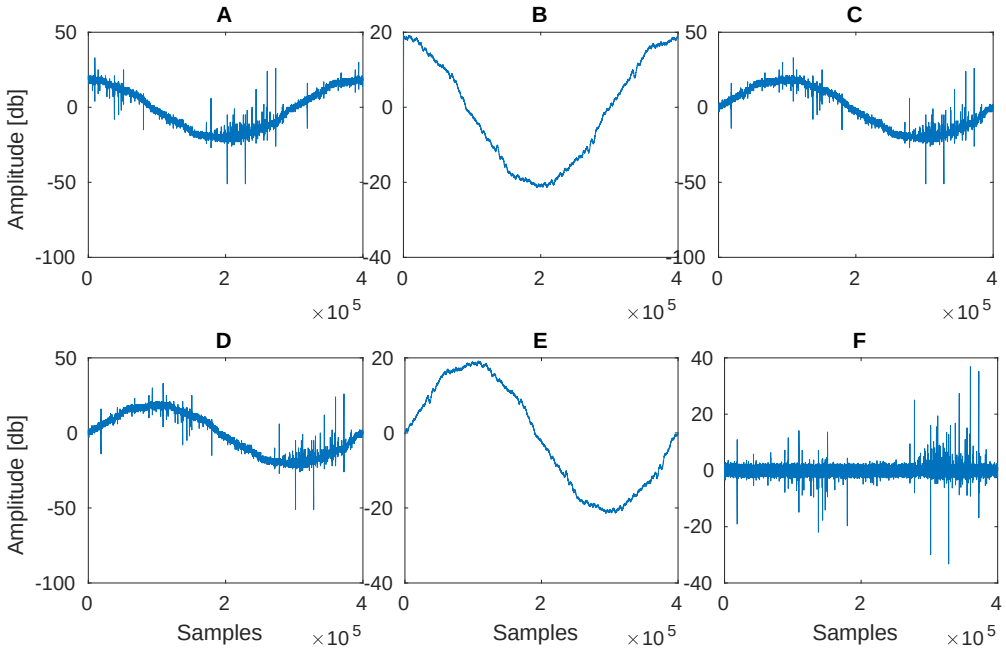


Figure 3.8: Signal processed by the Butterworth filtering. Raw signal (A), the filtered sinusoidal shape from the raw signal to proceed the synchronization (B), synchronized signal snapshot (C,D), filtered sine shape from D (E) and filtered signal by their difference (D - E = F).

4

FEATURE EXTRACTION

This chapter describes the extracted features, the reasoning of their extraction and the examination of their relevance. In order to estimate the relevancy of the entire set, the previously mentioned mutual information estimation method may be applied. There are several ways how to estimate MI and according to [135] the Kraskov's estimation possess the highest accuracy.

The Kraskov algorithm is one of many approaches to estimate MI among two random variables. The Kraskov algorithm is widely used as it is considered as one of the most effective and accurate. It is able to evaluate the dependency among two multivariate variables in the same way as in case of univariate variables. Kraskov estimation is based on Kozachenko–Leonenko entropy estimation [49]:

$$\hat{H}(X) = -\psi(K) + \psi(N) + log(c_d) + \frac{d}{N}\sum_{n=1}^{N} log(\epsilon_X(N,K)) \tag{4.1}$$

where N means the number of samples in X, d is the dimensionality of samples x, c_d means the volume of a d-dimensional unitary ball, and $\epsilon_X(N,K)$ is twice the distance (usually chosen as the Euclidean distance) from x_i to its k-th neighbor. The most known derivate of MI estimation has the following form (Eq. (2)) [49]:

$$\hat{I}(X;Y) = \psi(K) + \psi(N) - \frac{1}{K} - \frac{1}{N}\sum_{i=1}^{N}(\psi(\tau_{x_i}) - \psi(\tau_{y_i})) \tag{4.2}$$

where τ_{x_i} is the number of points whose distance from x_i is not greater than $0.5 * \epsilon_X(i) = 0.5 * max(\epsilon_X(i), \epsilon_Y(i))$. As we can see, the Kraskov's MI does not require the computation of underlying probability distributions of the given variables, but simply estimates the dependency by its neighbor based clustering, which simplifies the entire approach.

It is necessary to mention, that the MI values are very relative. They depend on the size of examined dataset and also on the kind of target variable. It can be the binary variable representing only the fault and failure-free states or it can be a multi-class variable representing all defined annotations (see Tab. 1.1). The results of the MI estimation should be taken more as a ranking values that are upper unbound. This lowers the information content of the MI outcome, but still its nonlinear character and simplicity of computation, makes it a valuable test. The final confirmation of the relevance still relies on the quality of the detection model.

FUNDAMENTALLY BASED FEATURE EXTRACTION

The experiment performed on PD pattern data relating mostly on the fundamental features of the PD pattern pulses was published in [rel5]. This experiment returned several outcomes (see Fig. 4.1). It was the balanced dataset, estimation of the features' relevance, the definition of the PD pulse shape and its optimization, and the evaluation of performance of the applied classification algorithm.

The low occurrence of the insulation faults makes the obtained data, in the view of the fault to failure-free signals ratio, imbalanced. This was necessary to solve (see Fig. 4.1 - left (2)) by standard under-sampling method, when several signals from the most representative groups were taken to form the new dataset, which was much more balanced and applied also in other experiments for the comparison. This dataset contains 500 different signals and it was split into three parts for all detection prototypes, the training, validation and testing part.

According to our available literature, we took 15 statistical features and performed their relevancy estimation by MI to keep only those that possessed the highest information value. The computation of mean value, standard deviation (std), skewness (measure of the asymmetry of the data around the sample mean) and kurtosis was trivial. The entropy of the signal, energy of the decomposition and entropy of the detailed coefficients was inspired by [136]. The fractal dimension, as the feature representing the fractality of the PD pattern has been applied in [137]. The number of pulses, their minimal and maximal widths and heights were taken as required. The estimated relevancies can be seen in Table 4.1.

As it was expected, the highest relevancies were observed in cases of the pulse related features, while the rest was rather irrelevant and could be omitted. The additional motivation of the experiment was to define a valid shape of the PD pattern pulse able to be extracted from a denoised signal while the other pulses, not matching the defined criteria, were ignored as false-hit pulses. This selective procedure also served as an additional layer of denoising. The diagram of the false-hit suppression is in Fig 4.1 (right).

False-hit pulse

Statistically significant part of false-hit pulse reaches a much higher amplitude than the PD pulse, which was observed during the data processing as well as confirmed by an human expert. A false hit is also very often followed by another one with the opposite polarity, creating a symmetric pair. We used this knowledge to find these pulses and to cancel them. A significant source of the false hit pulses comes from a corona discharges. These discharges create typical "pulse trains" which can be easily recognizable in the time domain of a signal.

Table 4.1: Estimated mutual information values of the extracted features

Name	MI value
Mean value	0.3787
Standard deviation	0.0516
Skewness	0.0483
Kurtosis	0.0311
Entropy of signal	0.0708
Energy of decomposition	0.0708
Entropy of detail coeffs.	0.0721
Fractal dimension	0.2820
Number of peaks	1.7676
Mean width of peaks	1.8311
Mean height of peaks	0.7818
Max. width of peaks	1.9384
Max. height of peaks	0.9907
Min. width of peaks	1.8822
Min. height of peaks	0.9474

Table 4.2: False-hit pulse suppression parameters defined by a human expert and their SOMA optimization ranges

Name of the parameter	Experts setting	SOMA range
maxDistance (ticks)	10	$\langle 4, 10 \rangle$
maxHeightRatio (%)	0.25	$\langle 0.05, 0.5 \rangle$
maxHeight (%)	100	$\langle 80, 140 \rangle$
maxTicksRemoval	500	$\langle 50, 500 \rangle$
Threshold coef.	1	$(0, 5\rangle$
Mother wavelet	db4	all members of the wavelet families db, sym, coif, bior, rbio, gaus, cgous.
Level of decomposition	1	1, ...6

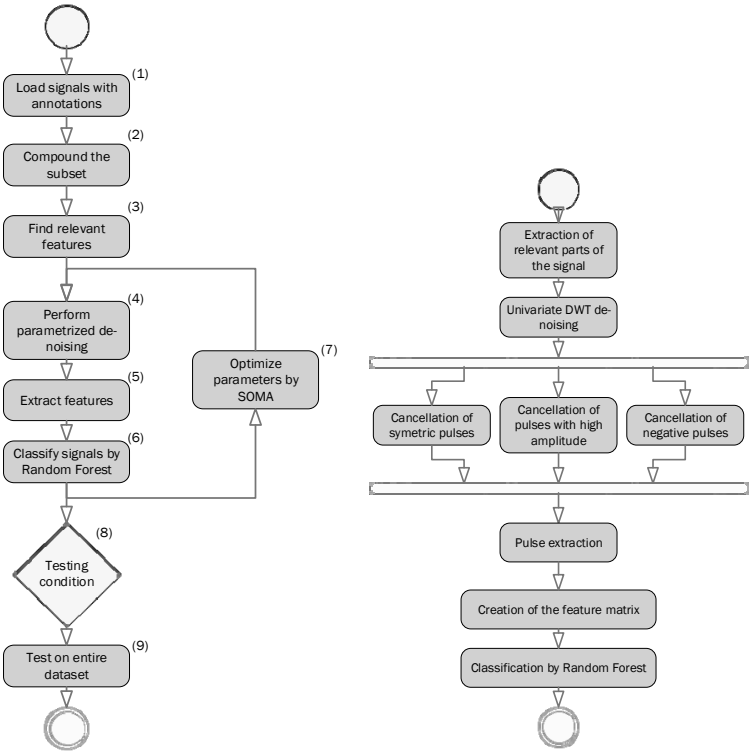


Figure 4.1: UML diagrams of the entire experiment (left) and the feature extraction combined with the random forest training as a SOMA fitness function (right).

The pulses that remains in the signal after denoising is examined to find and remove the symmetric peak pairs. In this process, each pulse is compared to its follower on the opposite site of the signal. If their distance is under a defined limit *maxDistance*, then the ratio of their amplitudes is calculated. When it is higher than the defined limit *maxHeightRatio*, the pulse pair is considered symmetric. A symmetric pair of pulses is very often followed by dumped oscillations, as it can be seen in Figure 4.2, which is the case of the corona discharge. When the first symmetric pair is removed from the signal, the following dumped oscillations can be misdetected as a PD-pattern. The length of this oscillation cannot be predicted because of the variable dump factors. Therefore, all the peaks in a defined distance (*maxTicksRemoval*) behind the symmetric pair need to be also cancelled. In another step, all the peaks with a higher amplitude than the defined limit *maxHeight* are removed from the extracted set of peaks.

As we can see, this fundamentally based false-hit removal is driven by several adjustable parameters. They were firstly adjusted by a human expert, but in the following part of the experiment, they were optimized by an evolutionary driven optimization al-

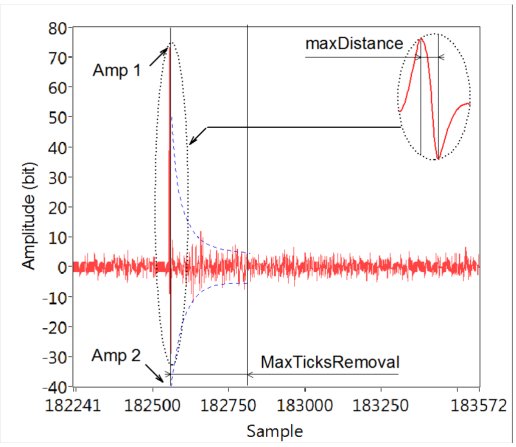


Figure 4.2: Corona discharge pulse followed by a dumped oscillation.

gorithm. The Self-organizing migrating algorithm was applied in this case with fitness function driven by the evaluation of the trained classification model.

This algorithm iteratively adjusted the false-hit suppression mechanism with the DWT denoising procedure which affected the quality of the obtained PD pulses. The range of the settings can be seen in Table 4.2. The evaluation of the adjustment was performed by training and testing of the random forest algorithm on the extracted features of the PD patterns which will be described in the following chapter.

FEATURES SYNTHESIZED BY SYMBOLIC REGRESSION

The approach, able to derive a set of relevant features without any fundamental knowledge of an observed system, could offer a promising way to extend our view about the problem. In cases, when input variables are unknown, when system's relations are uncertain and not mathematically defined, the evolutionary based approaches may be used to synthesize a solution. The grammatical evolution (see section 5.1.1) was applied in this experiment.

The evolutionary based synthesis was applied to produce a set of polynomials acting as the synthesized PD indicators. Their values formed a set of input variables of a trained classification algorithm. The definition of their form, objective function and process of evaluation was necessary to design [rel6]. In case of this experiment, the signal parts went trough the low and high pass filters of DWT to obtain three levels of the coefficients (approximate and detail). Each of these coefficient series was used as the inputs for the next phase of the experiment, the genetic programming driven synthesis of the nonlinear features.

The symbolic regression is a family of approaches able to synthesize the complex solution from simple user-defined building blocks. They could be a mathematical operations, a programming commands, etc. and the result may be, as in this case, the polynomial.

```
1 S → (sum⟨exp⟩)|(prod⟨exp⟩)|(min⟨exp⟩)|(max⟨exp⟩)
2 exp → (+⟨exp⟩⟨exp⟩)|(−⟨exp⟩⟨exp⟩)|(∗⟨exp⟩⟨exp⟩)|(%⟨exp⟩⟨exp⟩) —
3 (e⟨exp⟩)|(sqrt⟨exp⟩)|(sin⟨exp⟩)|(cos⟨exp⟩)|⟨v⟩
4 v → (x⟨index⟩)|(sum⟨index⟩⟨exp⟩)|(prod⟨index⟩⟨exp⟩)
5 index → (i, ⟨count⟩)
6 count → random integer
```

Algorithm 1: Grammar definition for grammatical evolution algorithm

The implementation of this algorithm was based on ECJ toolkit [138]. The defined grammar contained operators like $+$, $-$, $\times$, $\div$, $sin(x)$, $cos(x)$, e^x, etc. The behavior of some of the operations was necessary to be modified in order to ensure the mathematical validity of all obtained polynomials (avoid division by zero, logarithm of negative number, etc.). The set of correction was defined and can be seen in Table 4.3. The absolute values serve as the inputs for the square root, as well as for the logarithm. The division by zero simply returned the zero value. All this restrictions were used to make the process of GE more stable.

The obtained polynomials performed the calculations on the signal coefficients returning a single float value that served as a non-linear feature. Its relevance was evaluated by the Information Gain (IG) [139]. The IG criteria is used in creating C4.5 Decision tree [140] where the attribute with the highest IG is taken as splitting attribute. In this experiment the IG was used as a fitness function of the synthesized polynomial features.

The information gain is calculated as a difference of the entropy and average entropy (information). Its formula is defined below.

$$E(S) = -\sum_{j=1}^{n} P(s_j) log(P(s_j)) \tag{4.3}$$

$$E(S, A) = \sum_{i} P(S_i) \times E(S_i) \tag{4.4}$$

$$Gain(S, A) = E(S) - E(S, A) \tag{4.5}$$

All of the signals were split into 7 sub-signals with 50% overlap to reflect the fact that some parts of the signal are more relevant than others. There were obtained A_1, A_2, D_1, D_2 coefficient vectors by the DWT first and second level of decomposition from all of the sub-signals. Parts of the signals and their coefficient vectors served as the inputs of GE and because of that, more than ten non-linear synthesized features was

Table 4.3: Operations supported by the adjusted grammar for the GP

Operation	Symbol	Protection
Addition	$+$	N/A
Subtraction	$-$	N/A
Multiplication	$\times$	N/A
Division	$\div$	Output 0 when denominator input is zero
Square root	$\surd$	Apply an absolute value operator before radical
Natural logarithm	log	Output zero for an argument of zero; and apply an absolute value operator to negative arguments
Sine	sin	N/A
Cosine	cos	N/A
Natural exponential	e^x	N/A
Maximum	max	N/A
Minimum	min	N/A
Summation	$\sum$	N/A
Product	$\prod$	N/A

created with non-zero IG. The top five indicators with highest IG (one indicator mean the polynomial form and the part of signal which was applied for its synthesis) were chosen to build the input data set for the classification model. The artificial neural network (ANN) was applied for this purpose. The examples of the synthesized features are shown below as polynomial 4.6 and polynomial 4.7.

$$i_1 = max(\sqrt{\sum_{n=0}^{4} sin(sin(x_n))}, \sqrt{\sum_{n=1}^{5} sin(sin(x_n))}, .., \sqrt{\sum_{n=N-4}^{N} sin(sin(x_n))}) \tag{4.6}$$

$$i_3 = \frac{\sqrt{min(x_0, x_1, x_2, ...x_N)}}{log(\sum_{n=0}^{N} sin(x_n))} \tag{4.7}$$

In the figure 4.3, there is a visualization of the separability according to these synthesized features. They are transforming the searched space of signals into the recognizable zones due to their significant IG value (IG(i_1) = 0.757131, IG(i_3) = 0.607786). The adjustment, training and evaluation of the ANN model is described in the following chapter.

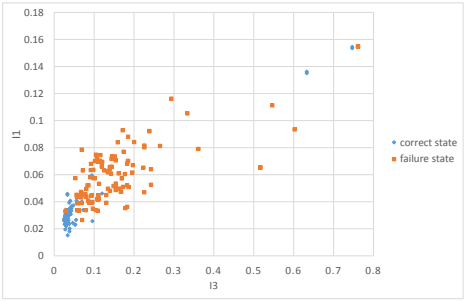


Figure 4.3: The scatter plot of the i_1 and i_3 indicator's values.

CHAOS ESTIMATION

The dynamical properties of the PD pattern data were also studied making use of the most profound chaos based indicators. Comparing to the available literature, in [141] Chelai Yin et al. were able to apply these indicators as the nonlinear dynamics parameters to distinguish different phases of PD pattern occurrence and suppress the periodic EBN. The PD-induceded degradation process of oil-paper insulation was studied by the indicators of chaos in study of Jun Gao et al. [142] as well. In 2014, Lyuboslav Petrov et al. brought wider study of Partial Discharge analysis, which pointed out several hidden patterns that can be detected in the data [143]. Application of chaotic measures also showed a strong classification ability of non-linear PD pattern data.

The motivation of the author's studies [rel7, rel8] was to perform an evaluation of the PD pattern data behavior by the most modern chaos tests. As a main result, indicators such as correlation dimension, sample entropy, approximate entropy, and 0-1 test for chaos were applied to the denoised PD pattern signals to examine their intrinsic behavior. Moreover, values of some of the indicators were examined for their relevance towards the fault indicating annotations.

It was selected only 7 representative signals for the performed analysis (indexed from 1 to 7). They differ in the amount and type of EBN, occurrence of the corona discharge pulses and the PD pattern type. These 7 representative signals were denoised and their relevant parts (according to 3.4) were selected and marked as *a* and *b* part of the signal. The basic annotations are listed in Table 1.1. The description of the selected representative signals are described in Table 4.4.

The main aim was to show the dynamical properties (more precisely, to quantify the change of complexity) of PD patterns using methods like approximate entropy, sample entropy, correlation dimension, and 0-1 test for chaos. It is shown that almost every

Table 4.4: Description of 7 typical signals with possible occurrence of the PD pattern applied in this analysis.

Index	CC fault	Description
1	no	Signal with high occurrence of Corona discharge pulses.
2	no	Signal with low level of noise.
3	no	Signal with high level of noise.
4	yes	Signal with starting CC-fault, an1, see in Table 1.1.
5	yes	Signal with starting CC-fault, an2, see in Table 1.1.
6	yes	Signal with advanced CC-fault, an5, see in Table 1.1.
7	yes	Signal with advanced CC-fault, an6, see in Table 1.1.

typical signal, described in Table 4.4, is chaotic. More methods is involved to examine how they differ and if these values are able to distinguish the examined classes. In the following, the applied methods are summarized.

Approximate entropy

The technique of determining system changing complexity was investigated by many authors in the past decades. One of the methods, so called approximate entropy was introduced by Pincus [144] (see also [145]) and is defined as follows.

For the time-series data $\phi(j)$ for $j \in \{1, 2, \cdots N\}$ form a sequence of vectors $x(1), x(2), \cdots, x_{N-m+1}$ in $\mathbb{R}^m$, defined by

$$x(i) = (\phi(i), \phi(i+1), \cdots, \phi(i+m-1)) \tag{4.8}$$

for each $1 \leq i \leq N - m + 1$. Now, for such i define

$$C_i^m(r) = \frac{\#\{j : d(x(i), x(j)) \geq r\}}{N - m + 1} \tag{4.9}$$

where # stands for the number of elements of a given set and distance between $x(i)$ and $x(j)$ is defined by

$$d(x(i), x(j)) = \max_{k=1,2,\cdots,m} \{|\phi(i+k-1) - \phi(j+k-1)|\}$$

motivated by Takens [146].

Finally, for a given m and r the approximate entropy is defined by

$$\mathrm{ApEn}(m, r) = \lim_{N \to \infty} [\Phi^m(r) - \Phi^{m+1}(r)] \tag{4.10}$$

where

$$\Phi^m(r) = \frac{\Sigma_{i=1}^{N-m+1} \log C_i^m(r)}{N-m+1}.$$

Sample entropy

The sample entropy is also considered as a measure of complexity of time-series data $\phi(j)$ for $j \in \{1, 2, \cdots N\}$ (see e.g. [147]). It is defined as the negative natural logarithm of the conditional probability that two sequences $x(i)$ and $x(j)$ of data points of dimension m are similar (have distance lower than r) and remain similar at the next point $(m+1)$, excluding self-matches that is $i \neq j$. The system of equations to describe its computation is as follows

$$\text{SampEn}(m, r, N) = -\ln\left[\frac{\Phi^{m+1}(r)}{\Phi^m(r)}\right]. \tag{4.11}$$

A lower value for the sample entropy therefore corresponds to a higher probability indicating more self-similarity.

Correlation dimension

The correlation dimension is a measure of geometric structural complexity of strange attractors, by description of their static nature. The Grassberger-Procaccia algorithm was applied for calculating this measure [148], which can be described as follows. For time-series $\phi(j)$ of N samples, it reconstructs the attractor dynamics by using delay coordinates to form multiple state space vectors, $Y_i = (x_i, x_{i+\tau}, \ldots, x_{i+(m-1)+\tau})$ where $i = 1, 2, \ldots, N_m$ and $N_m = N - (m-1)\tau$. Y_i is the reconstructed state space vector and m represents the embedding dimension with time lag τ.

The correlation integral $C_m(r)$ is defined in m-dimensional reconstructed space as the probability of finding a pair of vectors the distance of which is not larger than the given threshold r

$$C_m(r) = \frac{2}{N_m(N_m-1)} \sum_{i,j=1}^{N_m} H(r - r_{i,j}), r \neq j \tag{4.12}$$

where $H(x)$ is the Heaviside function and $r_{i,j}$ is the distance between two reconstructed vectors calculated using the Euclidean norm.

Chaos 0-1

The 0-1 test for chaos was introduced in [149] to distinguish between the regular and chaotic dynamics in deterministic dynamical systems. An output of the test 0 stands for

the regular movement and 1 for chaotic patterns that is denoted by K. As opposed to the computational methods of the Lyapunov exponent, this method is direct on tested data, does not require any preprocessing of the data, and needs only minimal computational effort. This method was originally stated as regression one, and later on in [150] it was improved as correlation that is faster and qualitatively gives better results; it is faster in terms of convergency. This correlation method works as follows for a given set of observations $\phi(j)$ for $j \in \{1, 2, , 3, \cdots N\}$.

Firstly, compute the translation variables for a suitable choice of $c \in (0, 2\pi)$:

$$p_c(n) = \sum_{j=1}^{N} \phi(j) \cos(jc), \quad q_c(n) = \sum_{j=1}^{N} \phi(j) \sin(jc).$$

The plot of the dependence of p_c on q_c for suitable choices of the parameters are shown in Figure 4.5 showing nontrivial dynamics. Then the mean square displacement

$$M_c(n) = \lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} [p_c(j+n) - p_c(j)]^2 + [q_c(j+n) - q_c(j)]^2,$$

where the limit is confident by calculating $M_c(n)$ only for $n \leq n_{cut}$ where $n_{cut} \ll N$, and is standardly set $n_{cut} = N/10$. Now, let us estimate modified mean square displacement

$$D_c(n) = M_c(n) - \left(\lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} \phi(j) \right)^2 \frac{1 - \cos(nc)}{1 - \cos(c)}.$$

Put $\xi = (1, 2, \cdots n_{cut})$, $\Delta = (D_c(1), D_c(2), \cdots D_c(n_{cut}))$. Finally, we get the output of the 0-1 test as the correlation coefficient of ξ and Δ for fixed parameter c

$$K_c = \mathrm{corr}(\xi, \Delta) \in [-1, 1]. \tag{4.13}$$

Obviously, K_c is dependent on the choice of c and as it was pointed out in [150] it is enough to get K, as the output of the 0-1 test, as limiting value of all K_c. Our tests confirm experience of [150] that it is sufficient to introduce

$$K = \mathrm{median}(K_c). \tag{4.14}$$

To avoid the resonances distorting the statistics, parameter c is chosen from the restricted interval $(\pi/5, 4\pi/5)$ for all computations, see [150]. In these tests, summarized in Table 4.9, 101 samples from 0.63 to 2.51 by 0.0188 for each $n_{cut} = \lfloor N/i \rfloor$ and $i = 2, 3, \cdots, 14$ were done, here $\lfloor x \rfloor$ rounds x to the nearest integer less than or equal to x. In Figure 4.6 the graphs of K_c depending on c of $1a$ and $7a$ signals are shown, these cases, 620 tests were done from 0.01 to 6.2 by 0.01.

The output parameter of the 0-1 test for chaos can acquire only one of the values 0 or 1, which corresponds to the regular and chaotic motions, respectively. More details can be found in [149].

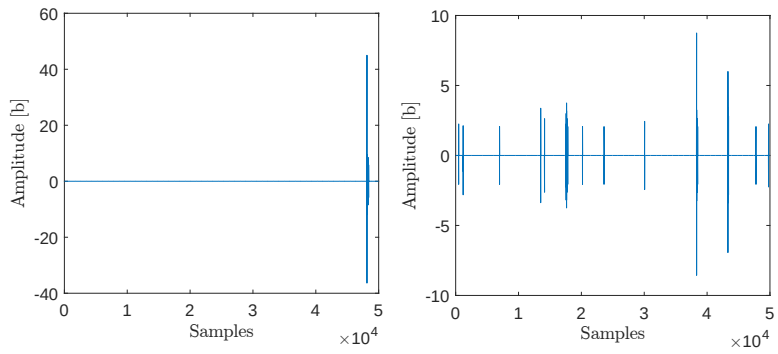


Figure 4.4: Plot of signals 1*a* (Left) and 7*a* (Right).

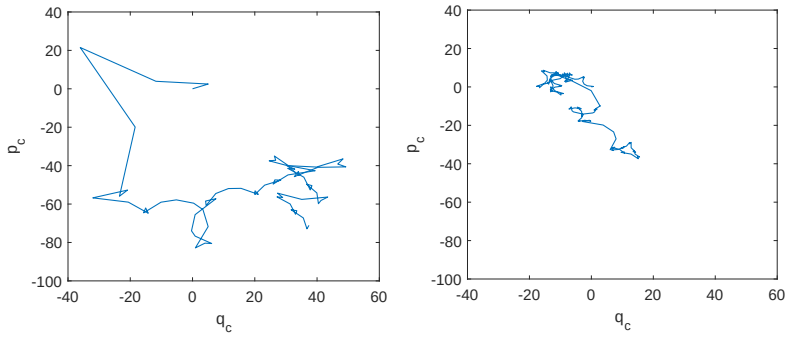


Figure 4.5: Plot of p_c versus q_c for $c = 2.6$ and signals 1*a* (Left) and 7*a* (Right).

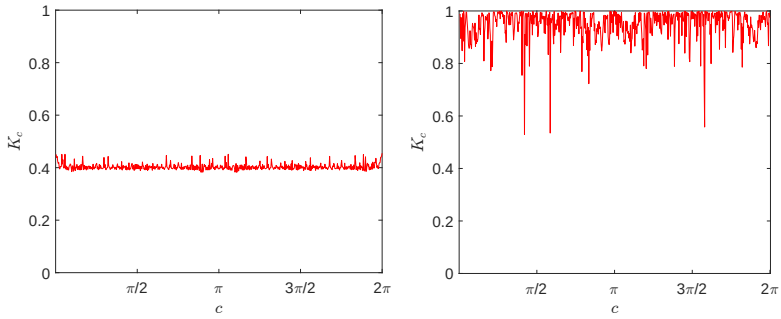


Figure 4.6: Plot of K_c versus c for $N/n_{cut} = 2$ and signals 1*a* (Left) and 7*a* (Right).

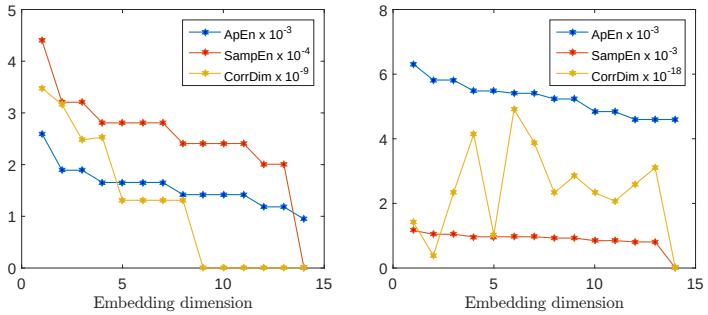


Figure 4.7: Plot of ApEn, SampEn and CorrDim in dependence of embedding dimension of signals 1*a* (Left) and 7*a* (Right).

Dynamical properties of PD patterns

The measure of complexity of researched signals form 1*a* up to 7*b* was computed under the above mentioned techniques. The output of these tests are in Tables 4.8, 4.7, and 4.6 with respect to the adjusted embedding dimension. The maximum resulting values, as the principal part, are extracted and ordered in the Table 4.5. The values of correlation dimension approached very close to the zero, hence the test was not accepted as significant. Therefore, the approximate entropy and sample entropy were only considered as relevant, the outputs of both tests are giving comparable results, consequently the complexity measure is visible.

As a consequence of the previous results, a natural question arose, whether the studied signals perform the chaotic patterns or not. For this purpose the 0-1 test of chaos was applied. The chaotic behavior was confirmed on all examined signals and on each value of the applied n_{cut} parameter. All results of the 0-1 tests for chaos are summarized in Table 4.9. Only in case of the 1*a* signal, the necessity for more precise examination was revealed. In the case, the default value of $n_{cut} = N/10$ gave $K = 8.87E-01$ the value that should not be declared as chaos of the studied signal since it is more then 10% from the required value 1. Therefore, the stress test was performed on 0-1 chaos parameter n_{cut}. The outputs of this test are depicted in Fig. 4.8 (Left) detecting decreasing value of K while n_{cut} is approaching 50% of the time-series dataset.

This instability (decreasing value of K) happened since the block of nonzero values of signal 1*a* was moving to the left, see Figure 4.4 (Left) while n_{cut} increased. That means the test is not applicable due to its instability, indicating randomness of 1*a* signal. The rest of the signals from 1*b* up to 7*b* performed stability under this stress test, the 7*a* case is depicted in Figures 4.8 (Right) and 4.4 (Right).

Table 4.5: Maximum values of ApEn and SampEn in dependence on embedding dimension D, ordered from the biggest to the smallest value.

ApEn			SampEn		
idx	max	D	idx	max	D
4a	2.76E-02	2	4a	6.70E-03	2
1b	2.46E-02	2	6b	4.59E-03	2
6b	2.45E-02	2	1b	4.02E-03	2
7b	2.12E-02	2	7b	3.89E-03	2
5a	1.38E-02	2	5a	2.74E-03	2
5b	1.16E-02	2	5b	2.17E-03	2
4b	9.53E-03	2	4b	1.87E-03	14
7a	6.31E-03	2	7a	1.16E-03	2
2a	6.07E-03	2	2a	1.12E-03	2
6a	5.29E-03	2	6a	1.05E-03	10
3a	4.80E-03	2	3a	1.01E-03	14
2b	3.50E-03	2	2b	6.41E-04	2
1a	2.60E-03	2	3b	5.21E-04	6
3b	2.55E-03	2	1a	4.41E-04	2

An examination of separability

The further examination published in [rel8] was aimed to evaluate the distinguishing ability of the sample and approximate entropy under four different adjustments of DWT based denoising. For this purpose an ANOVA test and cluster analysis evaluated by Silhouette score were used.

Analysis of variance (ANOVA), developed by Ronald Fisher, is a set of tools aimed to test a variation in the means of several independent variables. The most important point in traditional ANOVA is a test of significance of the difference among the means μ_i of variables. This test permits us to conclude whether the differences among the means of several variables are too deviated to be attributed as a sampling error or a significant difference [151, 152]. Based on that, the design of the adjusted hypothesis is defined as follows: $H0 : \mu_1 = \mu_2 = \cdots = \mu_r$, which means that under the zero hypothesis, there is no significant difference among the given means. The rejection of the zero-hypothesis will imply a presence of at least one variable with a significantly different mean from the rest of the variables.

The cluster analysis is a grouping mechanism of samples into the clusters according to their similarity that depends on a distance function and a given representation of the samples [153]. Such process was found helpful in various applications and, in this case, the process is used to examine the relevancy of the estimated entropies. The clear

Table 4.6: Correlation dimension calculated on embedding dimensions $2, 3, \cdots 14$ on all selected typical signals.

	2	3	4	5	6	7	8	9	10	11	12	13	14
1*a*	3.48E-09	3.16E-09	2.48E-09	2.53E-09	1.31E-09	1.31E-09	1.31E-09	1.31E-09	1.66E-18	1.11E-18	1.85E-18	1.48E-18	3.70E-18
1*b*	1.95E-06	1.82E-06	1.52E-06	1.41E-06	1.17E-06	1.05E-06	9.78E-07	9.47E-07	8.67E-07	8.21E-07	7.17E-07	6.54E-07	6.28E-07
2*a*	7.90E-08	7.91E-08	5.63E-08	5.63E-08	3.20E-08	3.20E-08	3.16E-08	3.16E-08	2.64E-08	2.64E-08	1.82E-08	1.82E-08	1.82E-08
2*b*	1.30E-08	1.30E-08	8.88E-09	8.89E-09	5.37E-09	5.37E-09	3.38E-09	3.38E-09	3.38E-09	3.38E-09	2.61E-09	2.61E-09	1.84E-09
3*a*	-2.30E-19	-2.30E-19	-3.44E-19	-5.74E-19	-9.18E-19	6.89E-19	-1.61E-18	-1.15E-18	1.45E-17	-2.30E-19	6.89E-19	9.18E-19	-1.38E-18
3*b*	-5.30E-20	-1.59E-19	-1.06E-19	1.38E-18	6.36E-19	1.38E-18	7.42E-19	1.27E-18	6.36E-19	-1.06E-18	6.36E-19	1.70E-18	0.00E+00
4*a*	4.54E-06	4.33E-06	4.42E-06	3.98E-06	4.21E-06	4.23E-06	4.20E-06	4.16E-06	4.07E-06	4.03E-06	3.78E-06	3.37E-06	3.92E-06
4*b*	2.05E-18	1.53E-18	-5.11E-19	1.02E-18	5.11E-18	2.56E-18	4.60E-18	-5.11E-19	5.11E-18	-3.58E-18	-1.48E-17	-1.53E-18	1.53E-18
5*a*	4.01E-07	4.02E-07	3.45E-07	3.46E-07	3.25E-07	3.26E-07	2.93E-07	2.94E-07	2.95E-07	2.96E-07	2.96E-07	2.97E-07	2.73E-07
5*b*	2.51E-07	2.51E-07	2.19E-07	2.20E-07	2.05E-07	2.05E-07	1.92E-07	1.93E-07	1.87E-07	1.87E-07	1.88E-07	1.88E-07	1.88E-07
6*a*	2.15E-09	2.15E-09	2.15E-09	2.15E-09	2.15E-09	2.16E-09	2.16E-09	2.16E-09	2.16E-09	2.17E-09	1.86E-09	1.86E-09	1.86E-09
6*b*	1.24E-06	1.24E-06	1.15E-06	1.15E-06	1.14E-06	1.14E-06	1.13E-06	1.14E-06	1.14E-06	1.15E-06	1.12E-06	1.12E-06	1.11E-06
7*a*	1.42E-18	3.88E-19	2.33E-18	-4.14E-18	1.03E-18	4.91E-18	3.88E-18	2.33E-18	2.85E-18	2.33E-18	2.07E-18	2.59E-18	3.10E-18
7*b*	9.02E-07	9.04E-07	8.06E-07	8.09E-07	8.00E-07	8.03E-07	7.66E-07	7.69E-07	7.14E-07	7.16E-07	6.14E-07	6.16E-07	5.83E-07

Table 4.7: Sample Entropy calculated on embedding dimensions $2, 3, \cdots 14$ on all selected typical signals.

	2	3	4	5	6	7	8	9	10	11	12	13	14
1*a*	4.41E-04	3.21E-04	3.21E-04	2.81E-04	2.81E-04	2.81E-04	2.81E-04	2.41E-04	2.41E-04	2.41E-04	2.41E-04	2.01E-04	2.01E-04
1*b*	4.02E-03	3.18E-03	3.18E-03	2.49E-03	2.49E-03	2.37E-03	2.38E-03	2.18E-03	2.18E-03	1.85E-03	1.85E-03	1.52E-03	1.53E-03
2*a*	1.12E-03	8.84E-04	8.84E-04	7.24E-04	7.24E-04	6.44E-04	6.44E-04	5.64E-04	5.64E-04	4.84E-04	4.84E-04	4.84E-04	4.84E-04
2*b*	6.41E-04	5.21E-04	5.22E-04	4.01E-04	4.01E-04	3.21E-04	3.21E-04	3.21E-04	3.21E-04	2.81E-04	2.81E-04	2.41E-04	2.41E-04
3*a*	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.01E-03	1.01E-03	1.01E-03	1.01E-03	1.01E-03	1.01E-03	1.01E-03	1.01E-03
3*b*	5.21E-04	5.21E-04	5.21E-04	5.21E-04	5.21E-04	4.81E-04	4.81E-04	4.82E-04	4.82E-04	4.82E-04	4.82E-04	4.82E-04	4.82E-04
4*a*	6.70E-03	6.23E-03	6.25E-03	6.14E-03	6.16E-03	6.18E-03	6.20E-03	6.13E-03	6.15E-03	6.09E-03	6.11E-03	6.04E-03	6.06E-03
4*b*	1.85E-03	1.85E-03	1.85E-03	1.86E-03	1.86E-03	1.86E-03	1.86E-03	1.86E-03	1.86E-03	1.87E-03	1.87E-03	1.87E-03	1.87E-03
5*a*	2.74E-03	2.58E-03	2.59E-03	2.51E-03	2.51E-03	2.44E-03	2.44E-03	2.44E-03	2.44E-03	2.45E-03	2.45E-03	2.37E-03	2.37E-03
5*b*	2.17E-03	1.89E-03	1.90E-03	1.82E-03	1.82E-03	1.74E-03	1.74E-03	1.70E-03	1.70E-03	1.71E-03	1.71E-03	1.71E-03	1.71E-03
6*a*	1.04E-03	1.04E-03	1.04E-03	1.05E-03	1.05E-03	1.05E-03	1.05E-03	1.05E-03	1.05E-03	1.01E-03	1.01E-03	1.01E-03	1.01E-03
6*b*	4.59E-03	4.44E-03	4.45E-03	4.42E-03	4.43E-03	4.39E-03	4.40E-03	4.41E-03	4.42E-03	4.39E-03	4.40E-03	4.37E-03	4.38E-03
7*a*	1.16E-03	1.04E-03	1.05E-03	9.65E-04	9.66E-04	9.66E-04	9.67E-04	9.27E-04	9.27E-04	8.47E-04	8.47E-04	8.07E-04	8.08E-04
7*b*	3.89E-03	3.73E-03	3.74E-03	3.71E-03	3.71E-03	3.64E-03	3.65E-03	3.53E-03	3.54E-03	3.21E-03	3.22E-03	3.18E-03	3.19E-03

Table 4.8: approximate entropy calculated for embedding dimensions $2, 3, \cdots 14$ on all selected typical signals

	2	3	4	5	6	7	8	9	10	11	12	13	14
1*a*	2.60E-03	1.89E-03	1.89E-03	1.66E-03	1.66E-03	1.66E-03	1.66E-03	1.42E-03	1.42E-03	1.42E-03	1.42E-03	1.18E-03	1.18E-03
1*b*	2.46E-02	1.96E-02	1.83E-02	1.65E-02	1.48E-02	1.37E-02	1.35E-02	1.27E-02	1.21E-02	1.13E-02	1.10E-02	9.54E-03	8.97E-03
2*a*	6.07E-03	5.17E-03	4.87E-03	4.57E-03	4.30E-03	3.81E-03	3.75E-03	3.70E-03	3.37E-03	3.17E-03	3.00E-03	2.99E-03	2.98E-03
2*b*	3.50E-03	3.11E-03	2.92E-03	2.54E-03	2.54E-03	2.20E-03	1.98E-03	1.95E-03	1.90E-03	1.90E-03	1.85E-03	1.50E-03	1.50E-03
3*a*	4.80E-03	4.80E-03	4.80E-03	4.80E-03	4.80E-03	4.80E-03	4.80E-03	4.80E-03	4.80E-03	4.80E-03	4.80E-03	4.80E-03	4.80E-03
3*b*	2.55E-03	2.55E-03	2.55E-03	2.55E-03	2.55E-03	2.44E-03	2.44E-03	2.44E-03	2.44E-03	2.44E-03	2.44E-03	2.44E-03	2.44E-03
4*a*	2.76E-02	2.61E-02	2.58E-02	2.54E-02	2.53E-02	2.51E-02	2.51E-02	2.48E-02	2.47E-02	2.43E-02	2.43E-02	2.43E-02	2.42E-02
4*b*	9.53E-03	9.53E-03	9.53E-03	9.53E-03	9.53E-03	9.53E-03	9.53E-03	9.53E-03	9.52E-03	9.52E-03	9.52E-03	9.52E-03	9.52E-03
5*a*	1.38E-02	1.33E-02	1.33E-02	1.30E-02	1.30E-02	1.28E-02	1.28E-02	1.26E-02	1.26E-02	1.25E-02	1.25E-02	1.23E-02	1.23E-02
5*b*	1.16E-02	1.02E-02	1.02E-02	9.60E-03	9.60E-03	9.29E-03	9.29E-03	9.10E-03	9.10E-03	9.02E-03	9.02E-03	8.95E-03	8.95E-03
6*a*	5.29E-03	5.27E-03	5.27E-03	5.27E-03	5.27E-03	5.27E-03	5.27E-03	5.27E-03	5.27E-03	5.10E-03	5.10E-03	5.07E-03	5.07E-03
6*b*	2.45E-02	2.35E-02	2.35E-02	2.33E-02	2.33E-02	2.30E-02	2.30E-02	2.29E-02	2.29E-02	2.28E-02	2.28E-02	2.25E-02	2.25E-02
7*a*	6.31E-03	5.82E-03	5.82E-03	5.48E-03	5.48E-03	5.41E-03	5.41E-03	5.23E-03	5.23E-03	4.84E-03	4.84E-03	4.60E-03	4.60E-03
7*b*	2.12E-02	2.00E-02	2.00E-02	1.97E-02	1.96E-02	1.94E-02	1.94E-02	1.88E-02	1.88E-02	1.75E-02	1.75E-02	1.71E-02	1.71E-02

Table 4.9: Chaos 0-1 calculated for $\lfloor N/n_{cut} \rfloor$ and $2, 3, \cdots 14$ on all selected typical signals.

	2	3	4	5	6	7	8	9	10	11	12	13	14
1*a*	4.01E-01	4.86E-01	5.55E-01	6.13E-01	6.63E-01	7.07E-01	7.46E-01	7.81E-01	8.12E-01	8.40E-01	8.65E-01	8.87E-01	9.07E-01
1*b*	9.75E-01	9.88E-01	9.95E-01	9.98E-01	9.98E-01	9.99E-01	9.99E-01	9.99E-01	9.99E-01	9.99E-01	1.00E+00	1.00E+00	1.00E+00
2*a*	8.81E-01	9.74E-01	9.99E-01	1.00E+00	9.99E-01	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
2*b*	9.80E-01	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
3*a*	9.82E-01	9.96E-01	9.97E-01	9.96E-01	9.96E-01	9.97E-01	9.97E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.99E-01	9.99E-01
3*b*	9.77E-01	9.92E-01	9.96E-01	9.95E-01	9.96E-01	9.97E-01	9.96E-01	9.97E-01	9.98E-01	9.97E-01	9.98E-01	9.98E-01	9.99E-01
4*a*	9.58E-01	9.69E-01	9.78E-01	9.90E-01	9.92E-01	9.94E-01	9.93E-01	9.94E-01	9.93E-01	9.92E-01	9.91E-01	9.91E-01	9.90E-01
4*b*	9.54E-01	9.89E-01	9.90E-01	9.87E-01	9.86E-01	9.90E-01	9.90E-01	9.91E-01	9.93E-01	9.94E-01	9.95E-01	9.96E-01	9.95E-01
5*a*	9.77E-01	9.91E-01	9.95E-01	9.96E-01	9.97E-01	9.97E-01	9.97E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
5*b*	9.00E-01	9.23E-01	9.60E-01	9.83E-01	9.93E-01	9.98E-01	9.98E-01	9.99E-01	9.99E-01	9.99E-01	9.99E-01	9.98E-01	9.98E-01
6*a*	8.82E-01	9.85E-01	9.98E-01	9.99E-01	9.99E-01	9.99E-01	9.99E-01	9.99E-01	9.99E-01	9.99E-01	9.99E-01	1.00E+00	9.99E-01
6*b*	9.76E-01	9.91E-01	9.94E-01	9.98E-01	9.97E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.99E-01	9.99E-01	9.99E-01	9.99E-01
7*a*	9.67E-01	9.82E-01	9.94E-01	9.93E-01	9.95E-01	9.97E-01	9.98E-01	9.99E-01	9.99E-01	1.00E+00	1.00E+00	1.00E+00	1.00E+00
7*b*	9.80E-01	9.90E-01	9.91E-01	9.96E-01	9.97E-01	9.97E-01	9.95E-01	9.96E-01	9.96E-01	9.97E-01	9.97E-01	9.98E-01	9.98E-01

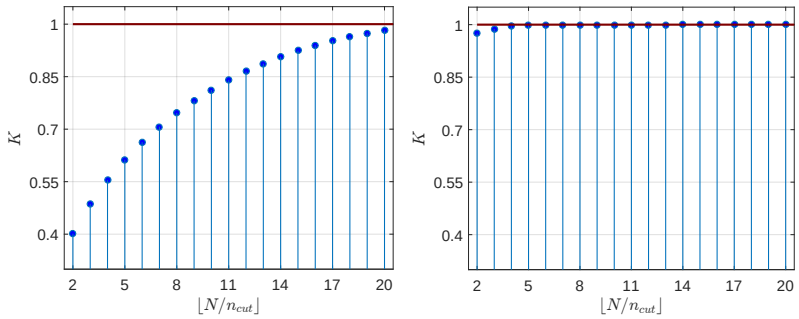


Figure 4.8: Plot of K versus $\lfloor N/n_{cut} \rfloor$ for signals $1a$ (Left) and $1b$ (Right).

clusters with minimum overlap can imply lower noisiness and uncertainty, while the inseparable highly overlapped clusters imply the presence of irrelevant features with minimal separability.

The number of clusters can be determined by gap statistic [154]. It compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The number of clusters remained on the value of 2, due to attempt of separability only between fault and failure-free state of the system.

The Silhouette score may be calculated using the mean intra-cluster distance a and the mean nearest-cluster distance b for each sample [155] in order to evaluate the clarity of the solution. The Silhouette Coefficient of the solution is than defined as $(b - a)/max(a, b)$. To be more specific, b is the distance between a sample and the nearest cluster of which the sample is not a part.

The maximal value across all dimensions for all signals is taken, and the mean and standard deviation of these values are depicted on Figure 4.9. As we can see the difference in values among applied wavelets is not significant. The splitting ability appears only in signals annotated as an5, which makes them separable from signals annotated as an2 and an6.

Because signals with annotation an1 coves almost entire range of obtained values, the splitting ability by these features on this annotation is minimal. To perform deeper analysis of these conclusions, we applied also a multiple comparison test (post-hoc analysis) [156] between subparts of all annotated signals, which could lead to higher splitting possibility. The p-value of ANOVA was lower than the level of significance (0.05) which rejects the null hypothesis saying that the average value is the same for all compared types of signal. The further multiple comparison test revealed which types of signal are different under the adjusted level of significance. The results were very similar among all applied wavelets as well as between both extracted entropies. The p-value matrix for sample entropy in case of db4 wave application is depicted below.

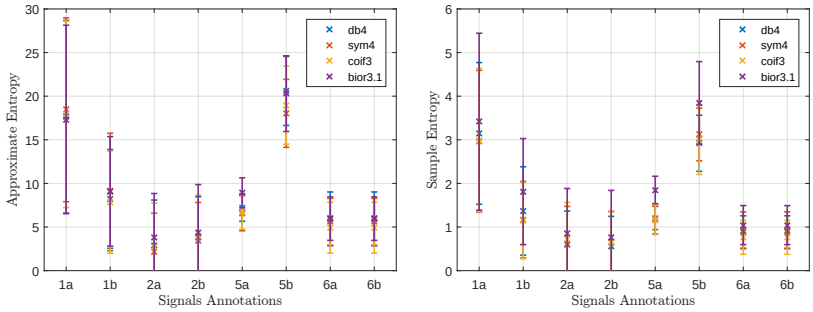


Figure 4.9: Medians of maximal values of Approximate Entropy (left) and Sample Entropy (right) calculated on dimensions [2,15] with all fault indicating classes with four different wavelets applied for pre-processing.

$$p-value = \begin{pmatrix} - & \mathbf{0.044} & \mathbf{0.000} & \mathbf{0.000} & \mathbf{0.047} & 0.999 & \mathbf{0.000} & \mathbf{0.000} \\ - & - & 0.524 & 0.507 & 0.992 & 0.082 & 0.657 & 0.657 \\ - & - & - & 1.000 & 1.000 & \mathbf{0.001} & 1.000 & 1.000 \\ - & - & - & - & 1.000 & \mathbf{0.001} & 1.000 & 1.000 \\ - & - & - & - & - & 0.053 & 1.000 & 1.000 \\ - & - & - & - & - & - & \mathbf{0.001} & \mathbf{0.001} \\ - & - & - & - & - & - & - & 1.000 \end{pmatrix} \tag{4.15}$$

As we can see, this adjustment generates the extracted feature which is able to distinguish the signal type $1a$ from $1b$, $2a$, $2b$, $5a$, $6a$ and $6b$ types. Also types $2a$ and $2b$ are statistically different from $5b$ as well as $5b$ is different from $6a$ and $6b$ too.

Compared to the different adjustment, in the case of the approximate entropy feature computed after preprocessing with bior3.1 wave, the results are very similar. The only change is visible in statistical difference of $1a$ type, which is no longer different from $1b$ and $5a$ types.

$$p-value = \begin{pmatrix} - & 0.243 & \mathbf{0.000} & \mathbf{0.002} & 0.644 & 0.977 & \mathbf{0.003} & \mathbf{0.003} \\ - & - & 0.649 & 0.902 & 1.000 & 0.141 & 0.862 & 0.862 \\ - & - & - & 0.999 & 0.958 & \mathbf{0.003} & 1.000 & 1.000 \\ - & - & - & - & 0.994 & \mathbf{0.008} & 1.000 & 1.000 \\ - & - & - & - & - & 0.341 & 0.987 & 0.987 \\ - & - & - & - & - & - & \mathbf{0.008} & \mathbf{0.008} \\ - & - & - & - & - & - & - & 1.000 \end{pmatrix} \tag{4.16}$$

In the rest of the adjustment, as we can see in Figure 4.9, the different waves did not bring very different results. After the applied tests, we can conclude that Sample and Approximate entropy as the complexity based features were not different among all of

Table 4.10: Silhouette score for the selected clustering solutions

Selected wavelet	Silhouette score
db4	0.127
sym4	0.084
bior31	0.131
coif3	0.097

the applied fault states of the PD patterns. These results were also very similar in all cases of applied preprocessing adjustments.

Figure 4.10 is an example of results obtained from the cluster analysis. Clusters are highly overlapped and not separated which is proven visually and also by value of the silhouette score close to the zero in all of the cases (see Table 4.10). The presence of correlation between the entropy values is the only visible result from this test. The results are similar for all of the applied waves.

The presence of chaotic behavior in PD pattern data was confirmed and several aspects of the complexity were examined but the separability of different PD pattern activities was not confirmed. This is the main reason why the chaos based features were not used as the input vectors for any detection algorithm.

COMPLEX NETWORK BASED SIGNAL REPRESENTATION

According to the available literature, the complex network based representation of the PD pattern data has not been performed yet, however this approach with ability to represent the relations among pulses is reasonable in several ways. The real environment affects the signals by high presence of external background noise interference, which creates a lot of false-hit pulses in the signals. Such pulses appears similar to the PD pattern pulses but they do not implying any kind of damage on CC. The differences can be observed only in small number of features and in their distribution inside of the signal. These assumptions lead us to apply the modeling of similarities between all of the pulses as a complex network. The features related to the clusters, connectivity and their distributions may possess different forms describing differently annotated signals.

The study of *complex networks* (CN) is based on two main areas of studies: graph theory and statistics. The research on CN, due to its a large variety of applications and result's interpretations, is considered as an interdisciplinary research. The main aim of CNs' application is to observe and study the systems by their topological features instead of their observable samples that do not hold any contextual information of the problem being considered [157].

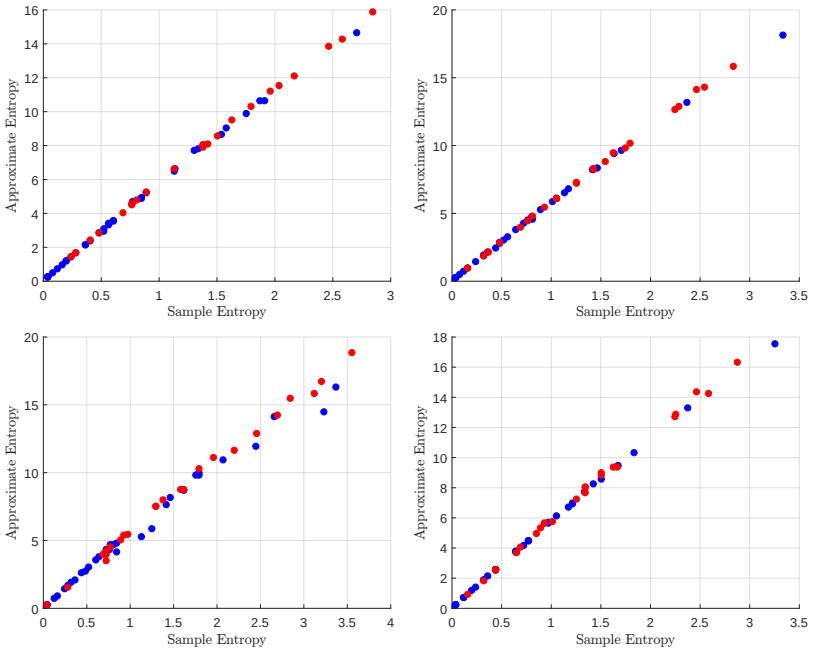


Figure 4.10: Scatter plot of entropy values of signals with annotation an2 and an6 on embedding dimension 10 to visualize the performed clusters. Bior3.1 (right) and Coif3 (left) waves are not performing visually different clustering solution, neither by calculated silhouette score.

The correct application of CN always relies on a reasonable transfer or interpretation of the observed data into the CN, except the cases when the model being studied already fulfills this assumption (e.g., world-wide-web, social networks, the spread of disease). Additionally, there are several transferring approaches of signals (time-series) [158, 159, 160] and matrix-based datasets [161, 162] into CNs. There are several studies of CN analysis [163, 164, 165], which deals with CN's features as an indicators of the system's inner behavior.

In author's experiments [rel3, rel9, rel10], after the applied wavelet based denoising, only the pulse related features were processed due to their highest confirmed relevancy. These 5 following features formed each pulse describing vector:

- the width, the height, and the position of the pulse on a sinusoidal curve (count to 3 pulse features);
- the closeness of its symmetric pulse (number of observations between them) (count to 4 features);

- the ratio between pulse's amplitude and the amplitude of its symmetric pulse (count to 5 features).

Each signal, therefore, returned a matrix of pulses' feature vectors that were later transformed into the CN.

The CN's acquisition was based on the computation of similarities between pulses' feature vectors. Each of them represented a node of the network. The similarity between two nodes was estimated by the *Euclidean distance* metric, and the obtained distance indicated the closeness of two pulses in n-dimensional feature space and thus, a link between two nodes was created. Such a method is described in [rel3] and many other works that were performed in previous studies [162, 161] transforming the data matrix into a CN by using similarity function and an adjustable threshold value. This kind of a network was also entitled as a similarity based network [166]. The threshold value is mostly adjusted experimentally. Its level implies overall connectivity of the network. In cases of low threshold value, the network is too sparse and does not posses enough information while on the other side, when the threshold level is too high, the over-connected network contain also noisy and uncertain connections. In [rel3], the threshold level was adjusted experimentally after several evaluations and in later studies [rel9, rel10] it remained unchanged. The detailed description of the CN acquisition approach may be seen in Alg 2.

In the first study [rel3], the hypothesis of CN features relevance was confirmed. The networks possessed different forms for differently annotated signals (see Fig. 4.11) which was also reflected by the feature set calculated on the obtained networks. To increase the amount of the obtained features, to increase the precision of the entire experiment by reflection of the signal's relevant parts, the later experiment [rel10] involved the dynamical analysis of the signals. In general, the split of the signals with overlapped window was applied to examine the changes of networks obtained from a single signal.

A total m segmented windows from signal x were considered, each of which had a 50% overlap. That is, each denoised signal x was decomposed (split) into m smaller time-series (windows $W = \{w_1, w_2, \ldots, w_m\}$). Each of the CN feature extracted during this analysis was represented by a vector instead of a scalar value because of windowed evaluations for each of the signal (see Fig. 4.12). The calculated network features are described as follows.

Basic CN's features

In case of both experiments, a few basic features were extracted from the CNs.

The density of the network was taken as a number of edges divided by the number of all potential connections. The other feature was global clustering coefficient which was introduced and described in study of Newman [167]. It can be calculated as a number of triangles divided by number of connected triples multiplied by three.

Input: Denoised signal split into m windows with 50% overlap $W = \{w_1, w_2, \ldots, w_m\}$ ($m = 8$), threshold value λ, and denoising hard threshold $\theta = 0$.

Output: Matrix M of extracted features from dynamic CN analysis

```
// A matrix:  m × 22 of CN features
M ← array(m, 22)
for k ← 1 to m do
    // Find all pulses greter than θ
    P ← find(w_k > θ)
    // A matrix:  size(P) × 5 of pulses features
    PM ← array(size(P), 5)
    for i ← 1 to size(P) do
        PM[i] ← extractPulseFeatures(P_i)
    end
    // Create nodes of CN for window k
    CN ← addNodes(size(P))
    // Create links between nodes of CN
    for i ← 1 to size(PM) do
        for j ← i + 1 to size(PM) do
            // compare feature tuples of pulses
            if d(PM_i, PM_j) > λ then
                CN ← addEdge(i, j)
            end
        end
    end
    // Extract features from constructed CN
    M[k] ← extractCNFeatures(CN)
end
return M
```

Algorithm 2: Dynamic complex network (CN) analysis based feature extraction

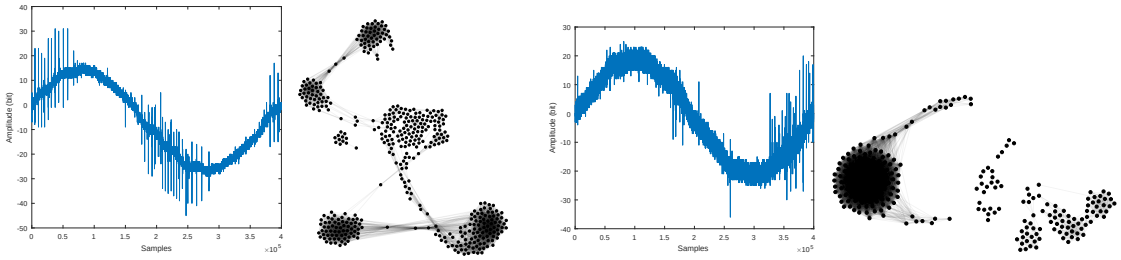


Figure 4.11: Signals (left) and their network based representations (right). The failure-free signal with high appearance of corona pulses (up) and the fault indicating PD pattern signal (down).

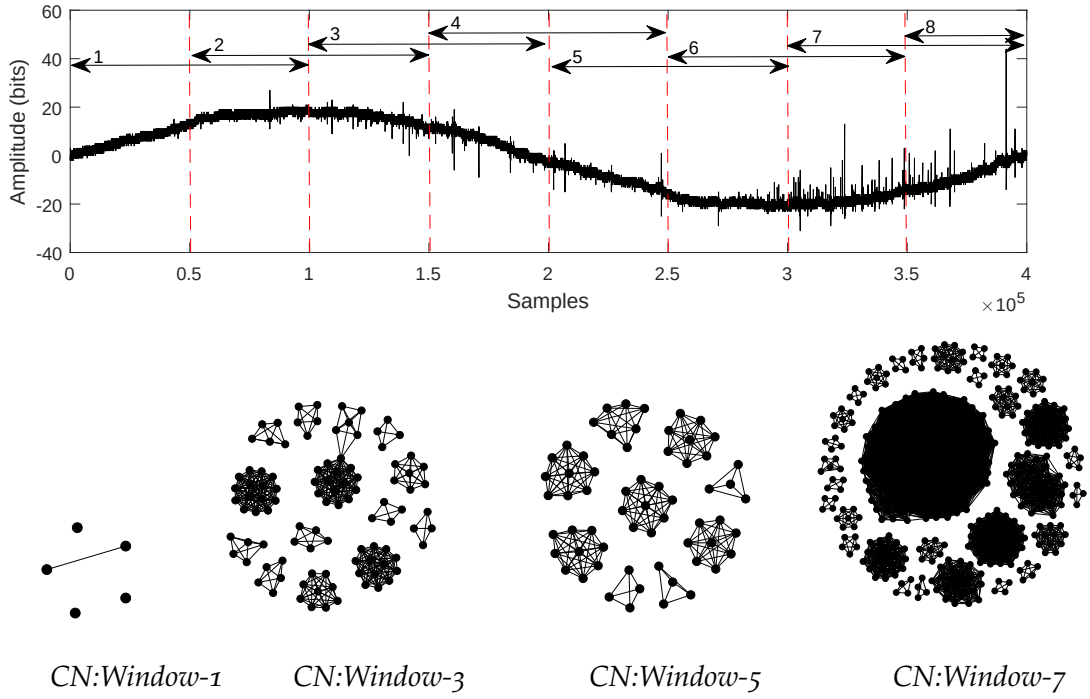


Figure 4.12: Decomposition of signal into smaller overlapped windows (1–8), e.g., window no. 1 is marked along horizontal axis 0 to 1 and window 2 is marked 0.5 to 1.5 allowing windows 1 and 2 to overlap (share) signal portion 0.5 to 1. Examples of four complex networks for four non-overlapping windows 1 , 3, 5, and 7 of the segmented signal are shown, which are obtained using the method mentioned in Table 4.11.

On the other hand the local clustering coefficient was introduced much sooner [168] and it determines whether a graph is a small-world network. This coefficient is computed for each node and the averaged value of all nodes was taken as another feature for classification in this experiment.

Aggregated CN's features

As it was concluded and compared in study of Aliakbary et. al. [169] the degree distribution can be evaluated by several methods. The most widely applied techniques for comparison are the Kolmogorov-Smirnov (KS) test [170], comparison based on distribution percentiles [171], and comparison based on fitted power-law exponent [172]. The mentioned techniques are very sensitive to presence of outliers and comparisons of different sized networks, which was actually the nature of these experiments.

The feature extraction method proposed by Aliakbary [169] process the degree distribution into quantification feature vector (Eq. (4.24)). It contains eight Interval Degree Probability values (IDP) (Eq. (4.23)), which computes the probability that the selected interval I (Eq. (4.22)) contains the degree of randomly chosen node.

$$P_G(d) = P(D(v) = d); v \in V(G) \tag{4.17}$$

$$\mu_G = \sum_{d=\min_G(D(v))}^{\max_G(D(v))} d \times P_G(d) \tag{4.18}$$

$$\sigma_G = \sqrt{\sum_{d=\min_G(D(v))}^{\max_G(D(v))} P_G(d) \times (d - \mu_G)^2} \tag{4.19}$$

$$R_G(r) = \begin{cases} [\min_G(D(v)), \mu_G - \sigma_G]; & r = 1 \\ [\mu_G - \sigma_G, \mu_G]; & r = 2 \\ [\mu_G, \mu_G + \sigma_G]; & r = 3 \\ [\mu_G + \sigma_G, \max_G(D(v))]; & r = 4 \end{cases} \tag{4.20}$$

$$|R_G(r)| = \max(right(R_G(r)) - left(R_G(r)), 0) \tag{4.21}$$

$$I_G(i) = \begin{cases} [left(R_G(\lceil \frac{i}{2} \rceil)), left(R_G(\lceil \frac{i}{2} \rceil)) + \frac{|R_G(\lceil \frac{i}{2} \rceil)|}{2}]; & i \text{ is odd} \\ [left(R_G(\lceil \frac{i}{2} \rceil)) + \frac{|R_G(\lceil \frac{i}{2} \rceil)|}{2}, right(R_G(\lceil \frac{i}{2} \rceil))]; & i \text{ is even} \end{cases} \tag{4.22}$$

$$IDP_G(I) = P(left(I) \geq D(v) < right(I)); v \in V(G) \tag{4.23}$$

$$Q(G) = \langle IDP_G(I_G(i)) \rangle_{i=1..8} \tag{4.24}$$

Such quantification feature vectors were calculated on distribution of degree, node betweenness, and edge betweenness. This brings 24 new feature values into the final classification.

Injected CN's features

The third approach of data-mining from the complex network was based on fundamental knowledge about the PD-pulses. Based on the most relevant features of the pulse and the annotated dataset of the signals, it was selected 10 most typical (visual selection of signals with lowest amount of noise and highest amount of PD pulses) signals containing PD activity.

All extracted pulse vectors from these signals were clustered by kNN clustering into two clusters and the centroid points of each cluster were taken as a typical PD-pulses. These typical pulses were injected into each of the evaluated networks with assumption, that those injected pulses (new nodes of the network) will obtain different features in fault signals and different features in failure-free signals. The features considered in this group were the degree and betweenness of the node.

The experiment in [rel10] increased the computational requirements by involving the windowed analysis (see Fig. 4.12) therefore the number of computed CN features was necessary to be reduced. From the obtained CNs, the features were calculated as described in Table 4.11. In addition to the essential 15 CN's features (see Table 4.11), the following non-trivial features were also included: 1) the number of connected components, 2) the number of nodes in the biggest connected component, 3) the number of edges, 4) the maximal independence set, and 5) the degree of Pearson correlation coefficient as the associativity coefficient of degree between pairs of linked nodes [173]. Apart from the mentioned non-trivial features, the modularity as a ratio of intra-community connections to inter-communities connections [174] was also calculated based on the estimated community structure using Louvain method [175].

The total count of the features per network was 22. This number increased by multiplying with the number of signal windows. It resulted into more than 200 extracted features per signal. Such a high number of feature implied a necessity for an additional optimization with aim to select the set of features forming the input dataset of the classification algorithm. Such a input dataset should possess only the most relevant features while the less relevant or redundant variables need to be omitted. Such optimization was ensured making use of the Multi objective optimization to reflect all of the defined criteria. Details with the adjustments and results will be described in the following chapter.

Table 4.11: Features extracted from complex networks [4] applied in following experiments.

Feature	Definition	Interpretation
1. Density	$D = \frac{L}{N(N-1)}$	Ratio between the number of links (L) and the number of all possible links N.
2. Degree*	$k_j = \sum_i w_{ij} a_{ij}$	Weighted count of links a_{ij} of a given node k_j.
3. Average neighbor degree*	$P(k) = \sum_{k' \leq k} p(k_j)$	Average value of degree of j-th node k_j neighbors .
4. Betweenness centrality*	$B_u = \sum_{ij} \frac{\sigma(i,u,j)}{\sigma(i,j)}$	Fraction of $\sigma(i,u,j)$ (number of shortest paths between i and j that comes through u) and $\sigma(i,j)$ (number of all shortest paths between i and j).
5. Closeness centrality*	$C_i = \frac{n-1}{\sum_{j \in N, i \neq j} d_{ij}}$	Fraction of number of possible connections n of node i with the sum of all of its shortest paths d_{ij} to all other nodes j of the network.
6. Average clustering (AC) coefficient	$C = \frac{1}{N} \sum_{i \in N} C_i = \frac{1}{N} \sum_{i \in N} \frac{2t_i}{k_i(k_i-1)}$	AC is computed over all nodes N of the network, where t_i is the number of triangles from i-th node and k_i is its degree.
7. Diameter	$Max_{i,j \in N}(d_{ij})$	Diameter is the longest shortest path of the graph.

Note: Each feature marked $*$ had its min, max, and average values. Hence, a total $4 \times 3 + 3 = 15$ features were computed.

5 OPTIMIZATION, TRAINING AND EVALUATIONS

Following chapter describes the applied machine learning, evolutionary and swarm intelligence based optimization models. The review and deeper description of the applied algorithms is followed by results obtained in the author's experiments.

BRIEF REVIEW OF APPLIED ALGORITHMS

This is devoted to bring insides about the applied machine learning and optimization models applied in the mentioned experiments. As it was stated as the goals of this work, the application of these soft-computing data driven approaches obtained the highest priority.

Machine learning models

Grammatical Evolution

Evolutionary based approaches, in general, are inspired by Darwin's Theory Of Evolution and through a set of iterations, they evolve a desired solution.

A candidate of a population represents a building block of this process. Its form may be a binary string, but it can also possess various representations such as a tree or a list of arrays. The entire population of these candidates evolves through set of generations and during each of them, the candidates are evaluated according to defined criteria. They are the indicators of solution's quality, so called objective functions. The candidates with the highest objective function values are selected for crossover, which is a procedure aimed to produce a new population for further generation. Before the new candidate becomes a member of new generation, the mutation randomly alters some of its part. This process ensures stochasticity of the entire process and keeps divergence in every population. After several generations, the evolutionary algorithm should converge into a solution.

GE is the evolutionary based algorithm inspired by biological evolution and genetic programming proposed by Koza [176]. Both of these approaches are very popular in various applications [177, 178]. From defined set of individuals based on a defined grammar (in Backus-Naur form) there is a synthesized solution for the defined problem [179]. The grammar defined in this experiment can be seen in Alg. 1.

The process of GE comes through several steps. First the individual is represented by a binary vector that is converted into the codons, the integers forming a vector. The member of the population turns itself into a polynomial by systematic substitution of these codons by the terminal symbols of the grammar. The mapping is performed by modulus of the codon value by the amount of terminals of the replaced non-terminal symbol. The mapping starts from the starting non-terminal symbol S of the grammar and ends until all non-terminals are replaced by the terminals. An example of the final solution can be seen in Fig. 5.1.

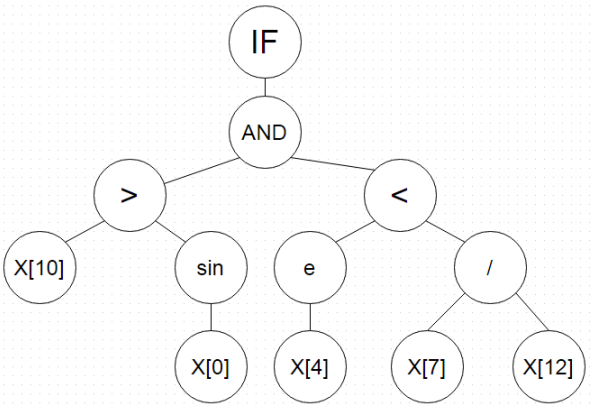


Figure 5.1: The tree interpretation of the GP's individual

When the mapping process substituted all codons and the individual still contains some non-terminal symbols, the vector of codons can be loaded again from its beginning. This can happen for an adjustable number of times or until the valid polynomial is not obtained. In cases, when none of these condition is reached, the member of the population is marked as invalid, it is no longer evaluated and it will be omitted for the next generation.

The next phase is the evaluation of all valid individuals by an adjusted fitness function.

Artificial neural network

The artificial neural network (ANN) represents a group of algorithms structurally and functionally inspired by the human brain [180]. The most common form of the ANN is a feed-forward multilayer neural network, which is often employed in the tasks of the approximation of non-linear functions [181], pattern recognition [182] or signal classification.

Optimization of this kind of model may be provided by several ways, but mostly it is done by supervised learning with a training set of the predetermined knowledge.

Structure of the ANN is represented by a directed graph of neurons organized into the recognizable layers. The directed graph of a network contains at least 3 recognizable layers: one required input layer, one or more hidden layers and one required output layer. The number of the hidden layers is optional and depends on the complexity of designed problem. Neurons of two connected layers are fully linked by weighted connections in a way that each neuron in the previous layer has a connection with each neuron of the following layer. The feed-forward propagation of the input signal starts on the input layer, where the signal is presented as an input and exposed by the synapsis (connections) to the following layer until the output layer is not reached. Each neuron of the hidden and output layer proceeds the summation of its all input signals x_i multiplied by the particular connection's weight coefficient w_i as it is show in the equation:

$$z = \sum_{i}^{n} w_i x_i. \tag{5.1}$$

The excitation value of the neuron is computed by its sigmoid activation function presented in equation:

$$y = \frac{1}{1 + e^{-\lambda z}}, \tag{5.2}$$

where λ is the slope of the sigmoid function. Final output values of the neural network are the excitations of neurons of the output layer.

Due to supervised learning mechanism, there is an comparison of gained and desired outputs by the error function, which leads to a kind of optimization (ANN's learning). The learning of the feed-forward neural network is processed by updating of the connection weights between neurons based on the amount of the error between desired and computed errors. The most popular method for neural network adaptation is a backpropagation.

Backpropagation learning algorithm

The backpropagation (BP) algorithm is based on a gradient descent method. Adaptation of weights is processed by propagation of output errors back through the network from the upper layers to the lower layers. The goal is to minimize the error function:

$$E = \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{M} (y_j - d_j)_p^2, \tag{5.3}$$

where error between real output from network y_j and desired output d_j is summed for P patterns of training set and for all m output neurons in the output layer. Error minimization is done by adaption of the weights between neurons. Change of weight is done by equation:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}, \tag{5.4}$$

where η is a learning coefficient. Then partial derivation of error E based on connection weight w_{ij} is obtained:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y}.\frac{dy}{dz}.\frac{\partial z}{\partial w_{ij}}. \tag{5.5}$$

Based on equations 5.1 and 5.2 previous equation is simplified by:

$$\frac{\partial z}{\partial w_{ij}} = x_i \qquad \text{and} \qquad \frac{dy}{dz} = y(1-y)\lambda. \tag{5.6}$$

More details about solution of partial derivations are described in [183, 184].

Support Vector Machine

The SVM was introduced by Vapnik in 1995 [185, 186] as a classifier for two-class (binary) problems and from this time it was applied in many studies [187, 188].

In its basics, SVM applies kernel based mapping for input vectors X into the high dimensional feature space where all of the training observations are separable by designed Optimal Separating Hyperplane (OSH). OSH should maximize the distances (so called margins) between the nearest observations and the OSH itself. The observations of one class should be ideally on one side of the space divided by OSH and observations of the second class should be on the opposite side. The training observations, which are the closes to the OSH are called the support vectors.

In case of the observations are linear separable, the solution (OSH) is given by following equation

$$Y = sign(\sum_{i=1}^{N} y_i \alpha_i (x \cdot x_i) + b) \tag{5.7}$$

where Y is the output value, y_i is the target value and $\cdot$ represents the product of input vector x_i and N support vectors. The α and b represent the parameters of the hyperplane.

In other (non-linear) cases, the SVM applies a kernel transformation function (K) which transforms the inputs into high-dimensional feature space, which gives the ability to find suitable OSH.

$$Y = sign(\sum_{i=1}^{N} y_i \alpha_i K(x, x_i) + b) \tag{5.8}$$

There are several known kernel functions which are able to by applied such as the polynomial kernel function ($K(x, x_i) = (x \cdot x_i + 1)^d$), Gaussian radial bases kernel function ($K(x, x_i) = exp(-1/\delta^2 (x - x_i)^2)$), etc.

Training of the SVM means to solve a linearly constrained quadratic problem (QP) where number of variables is equal to the number of input vectors of the training dataset (see in Fig. 5.2).

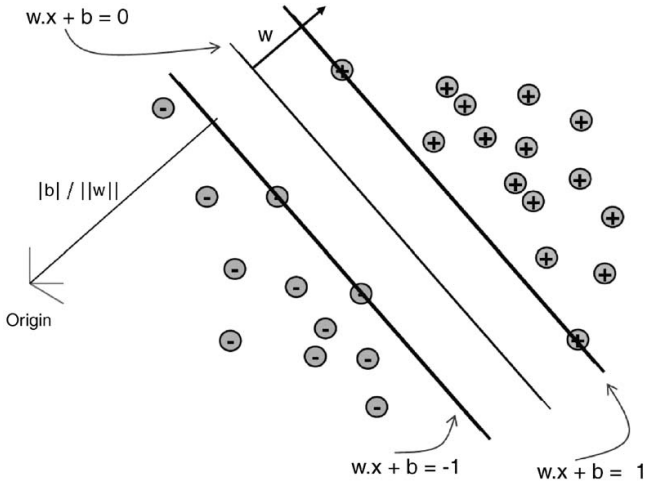


Figure 5.2: Graphical interpretation of Optimal Separating Hyperplane between separated observations of SVM classification (source [2])

Adaboost

Adaboost is the typical representative of approaches labeled as the boosting algorithms [189]. The idea is to develop an ensemble of low performing classifiers, so called weak learner. They are able to decompose the complex mapping required by the problem. Each of these trained weak learners may reflect only some aspect of the solved problem, while the entire ensemble performs a decision of much higher complexity by the voting mechanism [190]. This ensures the lower bias of the entire solution. Adaboost applies a weighed threshold classifier (WTC) as the weak-learner, and through multiple iterations, the algorithm adds new WTCs trained on incorrectly classified samples of the previously added model.

Random Decision Forest

RF is a general title for ensemble based machine learning model which was proposed by Breiman in 2001 [191]. This model was succesfuly applied in many machine learning studies [192, 193]. The core idea of the algorithm is focused on the application of an ensemble of CART-like tree classifiers (boosting) and their learning performed on the boosted-aggregated observations (bagging).

The decision tree (DT) is a tree-like structure of conditions with binary output values [140]. These conditions represent the nodes and leaves of the tree and serve as conditions for classification of the observation. Each condition make a single decision on one chosen attribute from the dataset and such an attribute is called splitting criteria. The

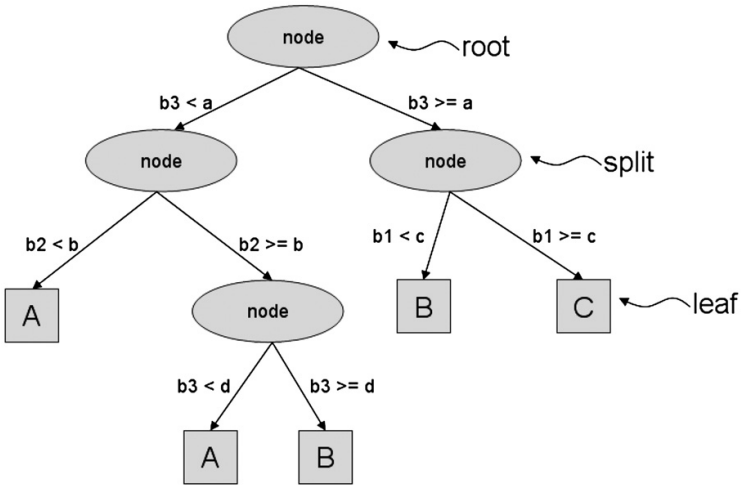


Figure 5.3: A decision tree with three input variables (b1, b2 and b3). At each of the root and internal nodes (splits), a statistical measure is applied. The values a, b, c and d are thresholds for splitting. A dataset is split into smaller subsets until the terminal nodes (leaves) return the class labels (A, B and C). (source [3])

attribute becomes the splitting criteria, when his information gain value (see Eq. 4.5) is the highest on the particular subset of observations. The structure of such tree is depicted in Fig 5.3.

Learning of the ensemble of trees means to train the set of trees where each of them obtain different random subset of the observations and different random subset of variables. This process minimizes the correlation between the trees, which increase the robustness of the model and decrease the possible amount of over-fitting. The final classification is derived from voting mechanism where votes from all of the trees are taken into account and final class is assigned to the observation by votes of the majority of the ensemble.

The bootstrapping mechanism comes from statistic and it is also know as random sampling with replacement [194]. This mechanism in context of RF algorithm produces balanced subset of observations for each of the tree. They are trained on resampled observations, which can handle the imbalanced problem or the problem of inability to learn some specific observations.

The other useful feature of RF algorithm is the possibility to compute the importance of the dataset's variables. The ranking value is derived from averaged value of information gain of the variable across all of the learned trees. This feature was reviewed and applied in many studies [195, 196].

eXtreeme Gradient Boosting

(XGB) is another example of the boosting models [197]. It is a special implementation of a Gradient Boosting model [198]. XGB applies the decision trees as a weak learners trained separately one after another, when each of them is trained on the residuals r_{im} of an ensemble extended by its ancestor. This learning process iterates M times and during every iteration it minimizes the one-dimensional optimization problem as it is stated below.

$$\gamma_m = \arg_\gamma \min \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \tag{5.9}$$

where the γ_m is an optimized weight for m-th classifier $h_m(x)$ trained to approximate the residuals of a current ensemble $F_m(x_i)$ as

$$r_{im} = -\frac{\partial L(y_i, F(x_i))}{\partial F_m(x_i)} \tag{5.10}$$

once the m-th classifier is trained and its weight γ_m is optimized, it is added into the ensemble as

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{5.11}$$

This gradient based approach represents the main difference from the random forest model.

Hyper-parameter optimization

Grid search hyper-parameter optimization (GSO) is an optimisation technique frequently applied for fine tuning of the machine learning model by optimisation of their hyper-parameter values [86]. This process comes through all combinations of parameter values (defined by their ranges) and according to the performance of the classifier, it chooses the best combination of settings.

On the other hand, there is also a well established Random search hyper-parameter optimization which performs a tests on the arbitrary selected combinations of adjustments [86]. This is much more suitable when adjustable values of hyper-parameter are not categorical variable, therefore they offer too much combinations. Their comparison has been brought for several times [87].

Swarm intelligence based optimization

The necessity to optimize a set of model coefficients in order to increase the quality of the obtained features (see sections 4.1 and 4.2) creates an opportunity to apply the swarm intelligence based models that iteratively searches for an optimal solution.

Self-organizing migrating algorithm

(SOMA) entitled as stochastic evolutionary algorithm was proposed by Zelinka [199]. Ideologically, this algorithms stands right between purely swarm optimization driven PSO and evolutionary-like DE. It employees the entire nature of migrating particles across the search-space making use a set of stochastic evolutionary operators to manipulate with the particles of the population.

The pertubation represents the mutation of the particles' movement. The operation creates pertubation binary vector by the adjusted PRT parameter and the given formula

$$v_j^{prt} = \begin{cases} 1, & \text{if } r_j < PRT \\ 0, & \text{otherwise} \end{cases}, (j = 1, 2, \cdots, d) \tag{5.12}$$

The V^{prt} drives the individual moves only into allowed dimensions (marked as true) and neglects moves in the rest (marked as false).

The crossover operator aims to keep the direction of the particles' moves towards the leader (particle with the best fitness value). This operator involves the pertubation vector (see Eq. 5.13) which is supposed to control the stochasticity of the populations' moves as well as the diversity of possible solutions.

$$x_i^{g+1} = x_i^g + (x_L^g - x_i^g) v_i^{prt} \tag{5.13}$$

During each migration loop, each particle performs n steps according to the adjusted step size and the path length. If the path length is higher than one, particle will travel longer distance, than is his distance towards the leader.

Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an swarm intelligence based optimization technique that drives its particles towards the leader of the population (the particle with highest value of the fitness function) ($lbest^g$) with respect to the personal best position of the particle from its previous moves ($pbest^g$) [200]. The influence of these vectors is weighted by adjustable constants (c_1, c_2) and two random variables r_1 and r_2 from range $\langle 0, 1 \rangle$. These variables with current particle's velocity $V_i^g(j)$ complete the equation that computes the following particle's velocity and the further position as well.

$$v_i^{t+1}(j) = v_i^t(j) + c_1 r_1 (nbest_i^t(j) - x_i^t(j)) + c_2 r_2 (lbest_i^t(j) - x_i^t(j)) \tag{5.14}$$

where index t represents the iteration of the swarm, j is the index of particle of the swarm and i is its i-th real value. The computation of the following coordinates for the $X(j)$ particle comes as follows.

$$X^{t+1}(j) = X^t(j) + V^{t+1}(j) \tag{5.15}$$

The fitness function of the PSO is performance of the machine learning based detection algorithm. It is trained and evaluated during every evaluation of the particle's fitness function.

Multi objective optimization

In general, the MOO problem is posed as an optimization problem of several (mostly conflicting) objective functions [201]. According to the defined equality and inequality constraints, the input value vector X defines the search space (often called as a feasible design space) for solutions. MOO is mostly based on search-engine optimization, but there is one major difference from single-objective optimization. It is the number of solutions, which in case of MOO is defined as a set of feasible solutions (called a Pareto Front). Each of its candidates is called Pareto Optimal and together, they form a so-called trade-off curve in a chart of objective values. The Pareto optimal solutions are equally distributed on this curve towards all of the optimization functions and the solutions placed in the middle of the curve ideally hold the optimization trade-off towards the applied cost functions.

Non Dominated Sorting Genetic Algorithm II

(NSGA-II) is a very popular evolutionary driven optimization technique used for MOO problems [202]. It has a low complexity, explicitly preserves the diversity of solutions and its elitism approach ensures the preservation of already found Pareto-front members. The main idea derives from standard evolutionary algorithms which makes it very convenient and simple to use in a wide range of studies [203, 204], where each new solution comes from crossover of its ancestors and (or) random mutation. NSGA-II starts with random initialization of the first main generation individuals P^g_{NSGAII} and auxiliary generation Q^g_{NSGAII} where $g = 1$. During each of the iterations Q^g_{NSGAII} is evaluated by functions F and merged with P^g_{NSGAII} to create R^g_{NSGAII}. Next, a new main generation P^{g+1}_{NSGAII} is selected from merged R^g_{NSGAII} by ranking and crowding procedures. Finally, a new auxiliary generation Q^{g+1}_{NSGAII} is derived from P^{g+1}_{NSGAII} by selection, crossover and mutation. This population is in the end of iteration merged with the previous one. The n candidates are selected by ranking and crowding procedures and they form the new population for next iteration. The algorithm stops after the defined number of iterations. The result of the algorithm is a Pareto-front final generation P_{NSGAII}.

Fuzzy decision making

The result of MOO is the Pareto front set where a single feasible solution needs to be found. The selection of the most suitable candidate is ensured by a fuzzy decision

making process having calculated the linear membership function for all members of the Pareto Front [205, oth6].

In [205, oth6], the objective function to be minimized, the membership function follows below:

$$\mu_i^r = \begin{cases} 1 & f_i^r \leq f_i^{min} \\ \frac{f_i^{max} - f_i^r}{f_i^{max} - f_i^{min}} & f_i^{min} \leq f_i^r \leq f_i^{max} \\ 0 & f_i^r \geq f_i^{max} \end{cases} \tag{5.16}$$

Accordingly, the objective function to be maximized, the definition is given below:

$$\mu_i^r = \begin{cases} 0 & f_i^r \leq f_i^{min} \\ \frac{f_i^r - f_i^{min}}{f_i^{max} - f_i^{min}} & f_i^{min} \leq f_i^r \leq f_i^{max} \\ 1 & f_i^r \geq f_i^{max} \end{cases} \tag{5.17}$$

where f_i^{min} and f_i^{max} are the minimal and the maximal value of objective function f_i from the payoff table (Φ), f_i^r is the value of ith objective function of rth Pareto Front member and μ_i^r is its membership value for ith objective function. During calculation of the total membership value μ^r of the rth Pareto Front member, we are able to apply membership weights w^m for each of the objective function as it is defined below:

$$\mu^r = \frac{\sum_{i=1}^{m} w_i^m \mu_i^r}{\sum_{i=1}^{m} w_i^m} \tag{5.18}$$

which enables us to control the importance of the membership values for each of the objective functions separately.

The payoff table is a squared matrix containing the normalized values of objective functions when each of them was optimized separately by some single objective approach. The best objective function values represent the position of U, so called utopia point - the best possible solution and the worst objective function values represent N - nadir point, or the worst possible solution. The best trade-off solution is placed the nearest to the U. In some cases, the single objective optimizations to obtain payoff table does not have to be involved. In order to reduce the computational complexity, the utopia and nadir points are estimated as the best ($U = \{0, 0\}$) and worst ($N = \{1, 1\}$) possible values from the normalized objective function values of Pareto front.

Evaluation metrics

All the testing of the mentioned models was performed by Cross-validation technique (CV) [206] and performance of the ML algorithms were evaluated by four statistical

criteria (accuracy, precision, recall and f-score) computed according to the correct or incorrect classifications of positive and negative classes. From the confusion matrix (CM), all of the compared metrics were computed, where CM encodes decisions of the classification algorithm according to their correctness. We defined four kinds of classified observations:

- tp (true positive) – samples annotated and classified by positive label Y_p^{tp}.
- tn (true negative) – samples annotated and classified by negative label Y_p^{tn}.
- fp (false positive) – samples annotated as negative but classified as positive Y_p^{fp}.
- fn (false negative) – samples annotated as positive but classified as negative Y_p^{fn}.

The accuracy, precision, recall and f-score are simply defined as it is stated below:

$$Y_p = \{Y_p^{tp}, Y_p^{tn}, Y_p^{fp}, Y_p^{fn}\} \tag{5.19}$$

$$Accuracy = \frac{Y_p^{tp} + Y_p^{tn}}{Y_p} \tag{5.20}$$

$$Precision = \frac{Y_p^{tp}}{Y_p^{tp} + Y_p^{fp}} \tag{5.21}$$

$$Recall = \frac{Y_p^{tp}}{Y_p^{tp} + Y_p^{fn}} \tag{5.22}$$

$$F = 2 \cdot \frac{recall \cdot precision}{recall + precision} \tag{5.23}$$

RESULTS REVIEW

Denoising based on weighted singular values

The simple threshold based classifier was applied in this experiment (see section 3.2) where number of pulses was the only one input variable taken for the fault detection. Experiment was aimed to reveal if PSO algorithm (see details in 5.1.3) can optimize the weights in order to expose the relevant pulses while the noise will be suppressed.

As we can see in Table 5.1 all of the denoising methods were able to improve results of the pulse based classifier compare to the scenario without application of any pre-processing (”No denoise” column). The highest f-score and accuracy was achieved by model based on optimized singular values.

Table 5.1: Performance of classification algorithm according to the applied pre-processing method on the selected subset

Pre-processing	No denoise	DWT	WPD	SVDPSO
Accuracy	65.5	75.5	74.5	81
Precision	51	62	62.5	68
Recall	56.5	76	74	70
F-score	53.6	68.28	67.76	68.98

It is worthy to mention the high computational requirements in order to perform this experiment. The simple pulse-based classifier was applied in purpose to lower this computational complexity, but on the other hand, it made the results very poor and incomparable with other experiments. The only outcome of this experiment was that optimized singular values can perform valuable preprocessing compare to wavelet based models.

Fundamentally based classification

Following section describes the adjustments and results of the experiment mentioned in section 4.1. It was a proposal of fundamentally based suppression of false-hit pulses in order to increase the quality of the data. This algorithm was divided in three different modules as it is depicted in Fig. 4.1 (right). These modules were separately maintained and several tests were performed to examined various combinations of these modules. The shortcuts in Tab 5.3 mean the names of the modules which are as follows.

- **RA** - relevant areas selection - module selects the PD pulses only in sine wave parts where they are supposed to statistically appear
- **SP** - symmetric pulse - the pulse in the negative side of the signal and in the close distance to the examined pulse
- **HA** - high amplitude - pulses with amplitude higher that the sine wave mostly comes from the corona discharge activity

The entire algorithm was developed, optimized and tested in the Matlab programming language. The setting of the RF classification model and SOMA based optimization of false-hit pulse suppression can be seen in Table 5.2.

The tuning of the denoising parameters (Table 4.2) was performed in order to maximize the precision of the final detection model. The precision was also adjusted as an objective function for SOMA algorithm. The expert based results and the SOMA opti-

Table 5.2: Random Forest and SOMA setting in case of fundamentally based denoising experiment

Model	Parameter	Value
RF	Weak learner	Decision tree
	Size of ensemble	200
	Input variables per tree	all
	Samples per tree	all
	Splitting criteria	Information Gain
SOMA	Number of individuals	15
	Number of migrations	100
	Path length	0.09
	Step size	0.02
	Pertubation	0.25

mized approach were compared to classification without additional denoise represented by false-hit suppression (**ND**).

Table 5.3 contains the performance parameters measured on a training subset (I.) and testing subset (II.). The experiment confirmed that PD detection performed significantly better when the method was focused on the sine's relevant parts, which could potentially contain a PD-pattern pulses. On the other hand, the performance of the detection dropped in case of processing the irrelevant parts or even the entire period of the raw signal.

The module cancelling the pulses with high amplitude brought the less increase of the performance, however it was based on fundamental knowledge. The SOMA based optimization was able to increase the overall performance, which represents a promising result in order to be applicable in different metering locations.

Relevancy of synthesized features

In section 4.2 it was described the process of the evolutionary based synthesis of non-linear features able to serve as the input variables for further classification. The adjustment of GE's hyper-parameters is depicted in Table 5.4.

ANN (see details in section 5.1.1) was applied as the classification model. The number of neurons of the hidden layer was adjusted to an intrinsic dimension of the training input data set. This intrinsic dimension is a number of principal components needed to capture 80% of the variance in the input data set [207]. The number of input neurons faced the number of input variable. The data set was then divided into three parts. The first part consisted of 70% of the data set for training phase, the second part of 15% for

Table 5.3: Performance of random forest classification algorithm according to the several different adjustments of the fundamentally processing of the PD pattern signal.

	Results (%)	Expert				SOMA	ND
		RA	RA SP	RA HA	RA SP HA	RA SP HA	
I.	Accuracy	85.8	89.7	86.1	89.3	93.2	84.8
	Precision	71.9	84.3	72.7	84.1	95.2	65.7
	Recall	43.6	57.7	46.1	55.6	68.4	43.4
	F-score	53.2	67.8	55.3	66.1	79.1	51.4
II.	Accuracy	99.9	99.9	99.9	99.9	99.8	98.8
	Precision	70.7	81.6	72.7	80	83.6	57.8
	Recall	43.7	58.1	46.1	58.1	66.7	37.4
	F-score	52.7	66.7	55.2	65.7	72.8	44.2

validation phase and third part of 15% for testing phase. Validation set was used during training phase to avoid over-fitting by early stopping of training [208]. The learning coefficient η and the slope of the sigmoid function λ were experimentally set to 0.4 and 0.6 respectively. The number of training epochs was adjusted to 5000 with an option of early stopping in cases when model did not converge enough.

The performance of the classifications in all the ANN's phases is shown in table 5.5 below.

The results confirmed high relevancy of the applied features due to high performance of the model on the given dataset. What is necessary to mention, is the higher computational requirement necessary for the entire experiment. This was one of the reasons of making use of the smaller dataset containing only 161 signals (see section 1.2).

Classification of complex networks

The following section summarizes the setting and results of the experiments based on the complex network representation and described in section 4.4. The first part is dedicated to experiment published in [rel3]. The settings of the RF classifier were adjusted experimentally with respect to the previous research and those settings were kept during entire experiment (100 trees in ensemble, random subset of variables and random subset of observations for each tree to prevent over-fitting) - for all levels of the threshold value.

Table 5.4: Adjustment of GE for evolutionary based synthesis of non-linear features

Parameter	Meaning	Value
size of population	100	number of candidates in population
length of genome	400	size of binary vector of each candidate
number of generations	5000	number of iterations of the entire algorithm
crossover type	one-point	split of genome in one point for crossover
crossover rate	0.8	probability of the genome being crossed
mutation rate	0.1	probability of the genome being mutated
initial breeding	random	initialization of the first population

Table 5.5: Performance of the classification algorithm trained on evolutionary synthesized features.

(%)	Training	Validation	Testing
accuracy	90.47	95.83	91.30
precision	86.44	88.88	88.09
recall	96.22	100	94.87
f-score	91.07	94.11	91.35

The value of the threshold T varied from 5 to 40% of the distance values in the similarity matrix (see Alg. 2). It was necessary to examine different levels of these values to compare if the obtained CN contains enough information for best classification performance. On the other hand, in case when the value T was too high, new edges of the CN brought only noise or uncertainty.

The cross-validation method (see in section 5.1.5) was applied for adjustment and testing of classification performance and the results covered the calculated values of accuracy, precision, recall and basic F-score. The best results in our experiment were obtained on 30% value of the threshold (see Fig. 5.4 and Table 5.6).

This experiment was performed on a smaller but very much balanced dataset containing only 290 signals. The performance has increased for all measured metrics compared to the previous experiment [rel3] (see in Table 5.6), which confirms the relevancy of application of complex network based features.

The extended experiment proposed in [rel10] processed each signal into several networks based on a signal window size. This results into a wider spectrum of obtained features which was not reasonable to use directly for classification anymore. To perform a valid modeling, the relevant subset of input features had to be made.

The proposed model, shown in Fig. 5.5, applied a feature extraction and classification phases. The feature optimization was inspired by the feature optimization methods ap-

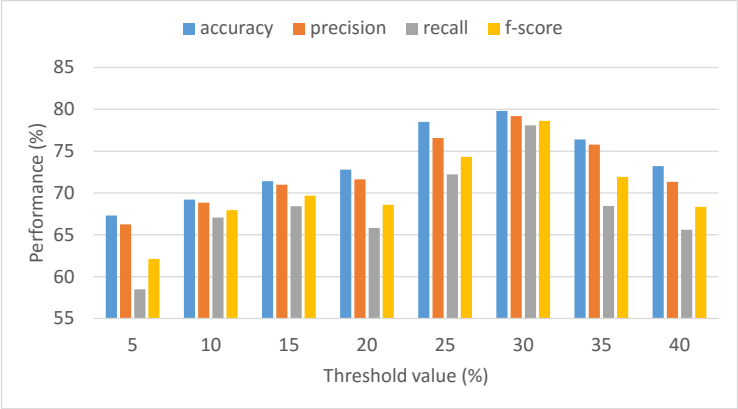


Figure 5.4: Graphical comparison of classification performance based on application of different threshold levels.

Table 5.6: Tabular comparison of classification performance based on application of different threshold levels.

Threshold value (%)	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
5	67.3	66.2	58.5	62.1
10	69.2	68.8	67.1	67.9
15	71.4	71.0	68.4	69.7
20	72.8	71.6	65.8	68.6
25	78.5	76.6	72.2	74.3
30	**79.8**	**79.2**	**78.1**	**78.6**
35	76.4	75.8	68.4	71.9
40	73.2	71.3	65.6	68.3

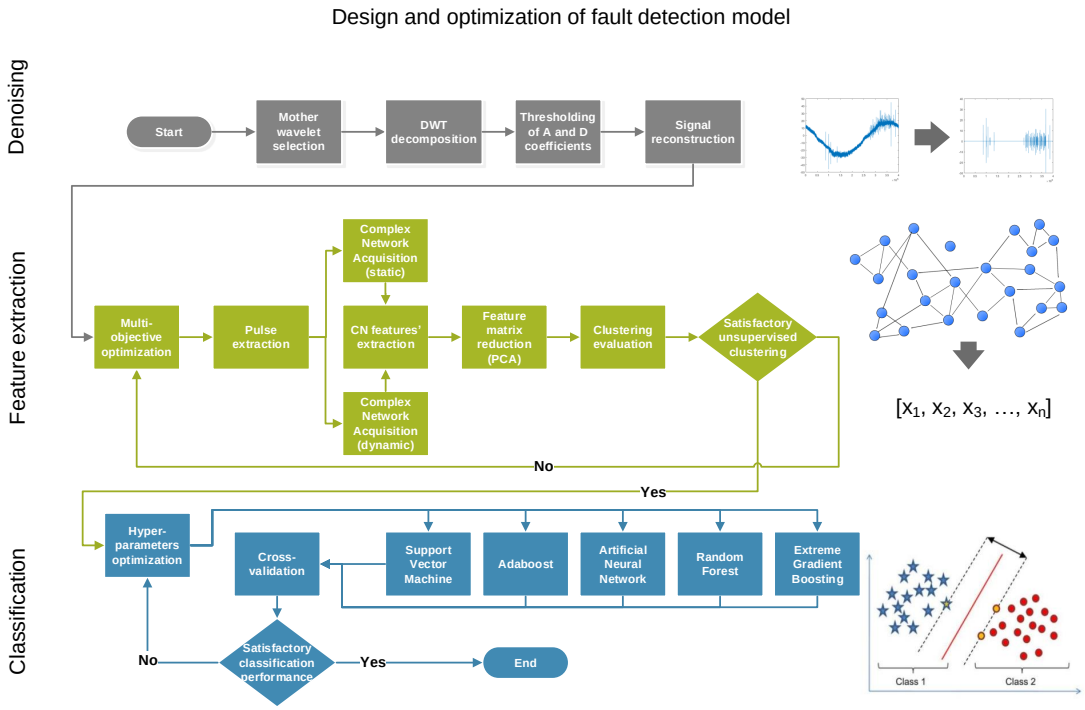


Figure 5.5: UML diagram of the entire signal processing and detector model.

plied in [209, 210]. Those studies used evolutionary way for the synthesis of solution that can meet the given criteria (objective functions). In our work, each evaluated candidate (solution vector in MOO) is defined by the number of feature extracted from the CN based feature extraction, i.e., a binary vector of size 22 indicating presence and absence of each CN feature lead to the reduction of feature dimension size. Hence, MOO aims to select which features are necessary and which are not.

In an iteration of MOO, the dimensionality of the selected CN features for the dataset were further reduced by using *principal component analysis* (PCA) [115]. The sole purpose of the application of PCA at this stage was to reduce the computational overhead and to remove the components that may have low impact on the fault detection. Since we cannot be sure at the beginning as to which CN feature are unnecessary, PCA was only applied after a preliminary feature selection by MOO.

The evaluation (fitness function in NSGA-II) of the candidate is based on the performance of clustering over the selected dataset (PCA-based reduced dimension of the binary vector-based selected features). A *k-means* clustering algorithm [211] was applied on the selected datasets, where the number of clusters in k-means was adjusted iteratively from 2 to 8 and the input matrix was provided by the candidate (solution vector).

Table 5.7: Settings applied for NSGA-II in MOO's feature selection procedure.

Parameter's name	Value
Number of candidates	100
Number of generations	25000
Parent selection	tournament selection of two
Crossover	uniform
Mutation rate	0.05

The performance of the k-means clustering was estimated by the *normalized mutual information* (*NMI*) and the *clustering variance ratio*. The clustering solution with the highest performance indicates the fitness value of the examined candidate. The metric *NMI* evaluates the nonlinear dependency between two given objects as per expression [212]:

$$MI(C,Y) = \sum_{c_i \in C, y_i \in Y} p(c_i, y_i) \log_2 \frac{p(c_i, y_i)}{p(c_i)p(y_i)} \tag{5.24}$$

where *MI* is the mutual information between $p(c_i)$ and $p(y_i)$, which represent the probability of the selected observation belongingness to the cluster c_i or it is labeled by signal y_i, respectively. Normalization of the *MI* is is performed computing ration of two given entropies: the entropy of cluster indexes ($H(C)$) and the entropy of annotation labels ($H(Y)$). Hence, *NMI* is given as:

$$NMI(C,Y) = \frac{MI(C,Y)}{\max(H(C), H(Y))} \tag{5.25}$$

which varies between 0 and 1, where the higher value means higher dependency between cluster index c_i and the signal's labeled y_i and therefore better clustering performance.

The clustering variance ratio was considered as a second objective function. It is the ratio between *within-cluster variance* and *between-cluster variance* [209]. The within-cluster variance is defined as per:

$$V_w(X) = \frac{\sum_{j=1}^{N_C} \sum_{i=1}^{N_X} m_{ij} d_E(X_i, X_j)}{N_X} \tag{5.26}$$

where N_C is the number of clusters, N_X is the number of samples, d_E is the Euclidean distance and m_{ij} is binary value indicating if the given observation X_j belongs to the i-th cluster or not.

On the other hand, the between-cluster variance is NOT calculated on the entire dataset of observations, but only on the centroids of the given clusters. This variance is calculated as per:

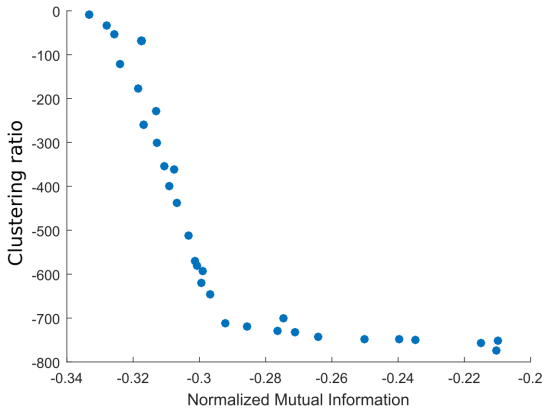


Figure 5.6: MOO's Pareto front - the set of optimal solutions' multidimensional fitness values.

$$V_b(X) = \sum_{i=1}^{N_C} d_E(C_i, C_j); j = i+1, i+2, ...N_C \tag{5.27}$$

where C_i and C_j are the clusters. Our motivation, in this work, is to *maximize* the between-cluster variance and *minimize* the within-cluster variance to achieve highest possible homogeneity in all of the clusters. The clustering ratio CR was, therefore, calculated as per:

$$CR = V_w(X)/V_b(X) \tag{5.28}$$

and it has to be used as denominator against -1 to return the lowest possible value for second MOO cost function.

These two cost functions NMI and CR are the x-axis (horizontal) and y-axis (vertical) dimensions of the Pareto front (see Fig. 5.6). The number of parameters optimized by NSGA-II was 24 (22 - binary values for feature selection, the size of reduced dimension and the selected window size).

None of the solution, therefore, can be said to be the best compared to any other candidate in the PF. However, the candidates in the region closer to origin (intersection of vertical and horizontal axis in Fig. 5.6) can be considered as a more "trade-off solutions" than the candidates far from the origin in both vertical and horizontal directions.

Since several candidates had tuned out to have a same value of objective function, in Fig. 5.6 the dot may be a representation of overlapping two or more candidates. A candidate is, thus, indicate a derived dataset, where for each signal, features were extracted using CN analysis and further optimized (reduced) by MOO.

To make a fair comparison with the results in [rel3], we randomly selected 15 candidates from the PF, and the selected candidates were used for evaluating classifiers performance.

The hypeparameters of the predictors was optimized by using the Grid search hyperparameter optimization (GSO) mentioned in [86]. The hyperparameter optimization is associated with the searching (finding) the best possible configuration of a predictor. Hence, every time the GSO was applied to find a configuration of a predictor, it executes both training phase followed by the cross-validation phase. As per the assessed training and cross-validation performance of the predictors, the best configuration of the predictor was chosen for the final setting. In this work, the following predictors and their configuration were assessed:

In case of SVM, GSO optimized the selection of *kernel functions*. In our experiments GSO was allowed to choose a function from a set containing functions such as the radial-basis function, polynomial function, and a linear kernel function. Each of these function contain a penalization constant C that was chosen from a set $\{1, 10, 100, 1000\}$ and a gamma value was chosen from the set $\{0.01, 0.001, 0.0001\}$. Whereas, for MLP, a proper number of *hidden neurons* was chosen from the set $\{10, 15, 20, 25, 30, 35\}$, and the *learning rate* was chosen from the set $\{1, 0.5, 0.2, 0.1\}$. However, *sigmoid function* was set fixed as an activation function and the number of training iteration was set fixed to 5000.

Other three predictors were ensemble methods. The predictor *AdaBoost* applies weight threshold classifier (WTC) as a weak learning model for its ensemble learning [189], the *random decision forest* (RF) [191] uses an ensemble of CART-like tree classifiers [140], and the *extreme gradient boosting* (XGB) adopts a RF like ensemble learning approach [197].

In the configuration selection of AdaBoost , the number of weak learners (WTC) in ensemble were varied from 50, 60, 70, 80, 90, 100, to 110, and the AdaBoost learning rate was selected from the set $\{1, 0.5, 0.2, 0.1\}$. The hyperparameters adjusted for RF contained only a number of trees in the ensemble were varied from 50, 60, 70, 80, 90, 100, to 110, and the post-pruning mechanism was either set to true or false. On the other hand, for XGB the number of trees were varied between 50, 60, 70, 80, 90, 100, to 110, learning rate η (also called shrinkage of learning weights of features) that prevent over-fitting was selected from the set $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, and the max depth of trees were selected from the set $\{2, 3, 4, 5, 6\}$.

In Table 5.8, the cross-validation performance of the predictors is shown in three ways, which are the average (mean), the max (highest), and the min (lowest) accuracy, precision, recall, and F-score values of the predictor for the selected dataset from the Pareto Front. Table 5.8 shows the results for the experiments where dataset contained 290 signals which only included signals with high appearance of pulses, 500 signals where more failure-free signals were involved.

The comparison of the obtained results in this study with the results presented in [rel3] reveals that the application of CN's features has significantly improved the per-

formance of the selected classification algorithms. The average performance of XGB predictors on the randomly selected candidates (datasets) from the Pareto Front gave a mean precision about 81% and the predictor RF gave max precision about 83.78% (see Table 5.8). The significance of improvement in the precision of the predictor is vital since it can reduce (minimize) the chance of incorrect detection of Faults in PD pattern. Hence, reduce the chance of false call on emergency maintenance of covered conductor (CC).

The representation of signal pulses as the nodes in a CN brought advantage in terms of signals pre-processing and optimization operations. According to available literature, the CN's approach for PD pattern classification was applied for the first time, and it was advantageous since the pulses are the most relevant phenomenon for a PD pattern classification and only its proper representation can lead to reasonable results. The correct PD pattern pulses mostly appear in similar patterns that mainly forms dense clusters on the repeatable places of sinusoidal signal's signature. However, the exact pattern, the cluster's size, and its place on a sine wave vary according to circumstances, such as the weather conditions, the metering device, the source of the CC fault. Therefore, the CN application brought a very different viewpoint to PD pattern classification by extending and exploiting a large number possible PD pattern features.

Table 5.8: Results obtained from CV of predictors optimized by GSO on all 14 candidate datasets obtained from the Pareto front (PF) selected from case where a dataset represent only signals with high appearance of pulses, i.e., a set of 290 signals. Evaluation measures for each predictor's mean accuracy a, precision p, recall r, and F-score s.

		290 signals			500 signals		
Predictor	(%)	mean PF	max PF	min PF	mean PF	max PF	min PF
SVM	a	85.43 ± 0.66	87.58	84.13	93.85 ± 0.15	94.32	93.50
	p	75.51 ± 1.55	77.97	70.01	75.52 ± 1.50	78.71	71.33
	r	82.97 ± 2.95	91.31	77.49	58.23 ± 2.02	64.00	53.35
	s	78.67 ± 1.28	81.89	74.35	65.49 ± 1.21	68.85	62.39
MLP	a	76.14 ± 3.80	86.20	66.20	93.85 ± 0.15	94.32	93.50
	p	69.02 ± 11.51	75.77	41.88	68.55 ± 9.02	74.74	42.92
	r	72.73 ± 12.36	94.48	32.89	50.94 ± 7.72	63.53	24.93
	s	65.76 ± 10.09	82.67	32.03	55.37 ± 7.41	67.38	30.71
RF	a	85.47 ± 1.05	87.93	82.06	93.88 ± 0.20	94.32	93.28
	p	79.83 ± 1.69	**83.78**	73.91	79.92 ± 1.48	**83.47**	76.15
	r	75.62 ± 2.68	81.19	66.67	52.98 ± 1.85	56.97	47.87
	s	77.06 ± 1.90	80.70	70.37	63.37 ± 1.57	66.65	58.99
XGB	a	86.00 ± 3.93	88.27	84.13	93.98 ± 0.85	94.47	93.58
	p	**81.44** ± 1.62	83.60	78.64	**81.44** ± 1.62	83.60	78.64
	r	76.03 ± 6.26	80.90	70.92	53.46 ± 4.34	57.00	49.88
	s	78.13 ± 5.92	81.92	74.55	64.23 ± 5.10	67.53	61.04
AdaBoost	a	80.83 ± 1.55	85.17	77.58	92.86 ± 0.31	93.65	92.16
	p	71.16 ± 2.77	78.21	64.88	71.28 ± 2.24	75.81	65.90
	r	71.52 ± 3.45	79.27	64.13	49.95 ± 2.49	56.55	44.48
	s	70.67 ± 2.54	76.60	64.81	58.30 ± 2.19	64.10	53.52

6 CONCLUSIONS

In order to summarize the outcomes of this work, the conclusion chapter is divided to face the goals proposed in the beginning. The time series analysis served as the main topic, due to the signals, carrying the partial discharge activity and having the time series representation in the most of experiments. Unconventional computational models, machine learning and data mining approaches were applied as the main tools in this work.

The processing of the signal data has been examined in several author's studies. The experiment comparing four ideologically different methods of dimensionality reduction revealed the modifications in the information content while noise and uncertainty were reduced. These changes were detected measuring statistical dependencies between the time series of the original and reconstructed dataset.

Described results revealed that algorithm based on manifold learning was not able to keep the information content even when the dimensionality reduction was minimal. This problem could be caused by the fact that the way of reduction did not encode the features relevant for the statistical testing. A nonlinear DR's representative, the Autoencoder, which is basically DR by ANN, did not bring the best results and was the most difficult to adjust due to its higher number of parameters affecting its learning ability.

The statistical dependencies were kept in a reasonable quality by Principal Component analysis, due to its focus on statistical properties of the input dataset. The reasonably well results were also obtained by Non-negative matrix factorization, this method was able to keep correlation between compared time series. On the other hand the values of causality and mutual information were lower than in case of PCA.

In a similar manner, the data decomposition into its singular values, later optimized by particle swarm optimization was designed as a denoising procedure. This experiment compared two different denoising concepts applicable for the problem of classification of signals containing partial discharge activity. Obtained signal data were buried in the noise interference which implied the necessity of such preprocessing procedures. The applied models covered area of machine learning and signal denoising. DWT and WPT denoising algorithms gave very satisfactory improvement for the simple threshold based classifier. The proposed combination of SVD and PSO was able to compete with wavelet based algorithms and according to all of the criteria, the evolutionary optimized singular values performed better.

The feature synthesis based on grammatical evolution brought interesting results and statistically relevant set of features for a given dataset. The entire concept of polynomial synthesis was robust, valid and offered a promising way for a future work. Wider and more representative dataset with properly adjusted fitness functions may increase the overall relevancy as well as the performance of the model. The weak spot of the experiment appeared in the limited data set as well as higher computational requirements of the entire concept.

On the other hand, the fundamentally based, pulse-like features possess also very high relevancy and their extraction is rather simple. In cases, when good performing wavelet based denoising is combined with false-hit pulse suppression, the quality of the fault detection model is increased. The false-hit pulse suppression model was successfully designed to be modular and parametrically adjustable, which brought an opportunity of its optimization. The human expert based adjustment was compared with SOMA based optimization. While performance of both models was satisfactory, the swarm intelligence based optimization was able to find a combination of parameters leading to even higher performance.

Later experiments were covering the very first application of a complex network based representation of PD pattern activity for detection of fault behavior on covered conductors. The complex network, created from extracted pulses, reflected valuable features for PD-pulses classification. This hypothesis was confirmed by increase of performance of the classification model also compared to author's previous research. Model based on CN's representation may be used as a supportive tool for classification of signals with higher noise interference, because pulse-free signals may be automatically considered as the failure-free signals.

The complex network (CN) and multi-objective optimization (MOO) based signal processing model, proposed in the last experiment, contained three phases of signal processing in order to classify PD pattern behavior properly. The CN constructed from the signal's pulses were found to be sufficiently representative to improve precision of the predictors. Such an improvement was due to the extraction of a large number of features using CN's, which showed the ability to address signals with a high noise interference. MOO for feature selection lead to the reduction of the dataset dimension that lowered a computational cost by an automatic recognition of the relevant features. It may be concluded that the CN based features had significantly improved the prediction of correct and incorrect PD pattern and the high precision of the trained predictor may reduce the number of detection false alarms. Hence, it minimizes false calls on emergency services.

The chaos examination of the PD activity was not enclosed by an application of extracted features as the input values of a detection algorithm, however these experiments were still able to bring some valuable results. Author's contributions extended the area of studies making use the chaos based indicators on PD pattern data [141, 142, 143] by

several significant differences. First of all, the data came from a natural environment by monitoring the PD activity on medium voltage overhead lines, which was ensured making use of an original metering device. All of those facts implied the huge variety of signals (PD pattern changes according to weather conditions, time of the day, type and distance of the damage, etc. The application of proper pre-processing was therefore considered as necessary.

The main motivation was to examine several chaos based indicators on the PD pattern data in order to test the complexity of the preprocessed signals and to reveal whether some of the tested features are able to serve as a splitting criteria according to several fault indicating annotations. Further results aimed on a description of the dynamical properties of PD patterns described in Table 4.4. More precisely, the complexity of signals was properly evaluated using approximate entropy, sample entropy, and correlation dimension, see Figure 4.4.

Finally, the 0-1 test for chaos confirmed the presence of chaotic behavior in almost all signals and in the case of $1a$ signal, randomness was detected using the newly applied stress test on the n_{cut} parameter, see Figure 4.8. This instability was considered as a discovery of unexpected behavior of the chaos 0-1 test. It is therefore another part of contribution of the entire work, because according to the available literature, such a finding was published for the first time.

The limitation of the first work relied on fact that only the representative signals of each class were examined and compared. In the following chaos based experiment, several adjustments were applied to achieve differently preprocessed signals by discrete wavelet transformation. This process was followed by estimation of complexity by approximate and sample entropy calculation.

The visualization of the results and further ANOVA testing confirmed that some of the PD-pattern types are statistically different from others by results of applied complexity measures. The most subparts of annotated signals were not significantly different in pairwise comparison, which shows high similarity of measured entropies among the PD pattern fault signals. The difference in results among the applied preprocessing adjustments was also not significant.

This work fulfilled adjusted goals and all results were published in the international conferences and journals. The set of described experiments covered the major phases required for the correct signal processing for the fault detection model. The chaos indicating features or the signal representation based on complex network brought an innovative aspect of the work and will be beneficial for the future applications.

As a future work, it is required to recompute and refine the results on a growing database to develop one final model able to be deployed in the real environment. The mentioned models, like the weighted singular based denoising or the evolutionary based feature extraction, need to be optimized in order to lower their computational costs. This

may be achieved by making use of CUDA programming or by application of a more balanced dataset.

AUTHOR'S BIBLIOGRAPHY - RELATED PUBLICATIONS

[rel1] Stanislav Misak, Jan Fulnecek, Tomas Jezowicz, Tomas Vantuch, and Tomas Burianek. Usage of antenna for detection of tree falls on overhead lines with covered conductors. *Advances in Electrical and Electronic Engineering*, 15(1):21–27, 2017.

[rel2] Stanislav Mišák, Tomáš Ježowicz, Jan Fulneček, Tomáš Vantuch, and Tomáš Buriánek. A novel approach of partial discharges detection in a real environment. In *Environment and Electrical Engineering (EEEIC), 2016 IEEE 16th International Conference on*, pages 1–5. IEEE, 2016.

[rel3] Tomas Vantuch, Jan Gaura, Stanislav Misak, and Ivan Zelinka. A complex network based classification of covered conductors faults detection. In *The Euro-China Conference on Intelligent Data Analysis and Applications*, pages 278–286. Springer, 2016.

[rel4] Tomas Vantuch, Vaclav Snasel, and Ivan Zelinka. Dimensionality reduction method's comparison based on statistical dependencies. *Procedia Computer Science*, 83:1025–1031, 2016.

[rel5] S Misák, J Fulnecek, Tomáš Vantuch, Tomáš Buriánek, and T Jezowicz. A complex classification approach of partial discharges from covered conductors in real environment. *IEEE Transactions on Dielectrics and Electrical Insulation*, 24(2):1097–1104, 2017.

[rel6] Tomas Vantuch, Tomas Burianek, and Stanislav Misak. A novel method for detection of covered conductor faults by pd-pattern evaluation. In *Intelligent Data Analysis and Applications*, pages 133–142. Springer, 2015.

[rel7] Marek Lampart, Tomáš Vantuch, Ivan Zelinka, and Stanislav Mišák. Dynamical properties of partial-discharge patterns. *International Journal of Parallel, Emergent and Distributed Systems*, pages 1–16, 2017.

[rel8] Tomas Vantuch, Marek Lampart, and Michal Prilepok. An examination of an entropy based features on partial discharge pattern. In *International Conference on Intelligent Information Technologies for Industry*, pages 265–275. Springer, 2017.

[rel9] Tomas Vantuch and Ivan Zelinka. Covered conductors fault behavior studied by features of complex networks. In *AIP Conference*, volume 1863, page 070029. AIP Publishing, 2017.

[rel10] Tomas Vantuch, Varun Kumar Ojha, and Stanislav Misak. Complex network and multi-objective optimization based signal processing for covered conductor faults detection. *Pattern analysis and applications*, 2017 (submitted).

AUTHOR'S BIBLIOGRAPHY - OTHER PUBLICATIONS

[oth1] Jindrich Stuchly, Stanislav Misak, Tomas Vantuch, and Tomas Burianek. A power quality forecasting model as an integrate part of active demand side management using artificial intelligence technique-multilayer neural network with backpropagation learning algorithm. In *Environment and Electrical Engineering (EEEIC), 2015 IEEE 15th International Conference on*, pages 611–616. IEEE, 2015.

[oth2] Michal Prilepok and Tomas Vantuch. Partial discharge pattern classification based on fuzzy signatures. In *International Conference on Intelligent Information Technologies for Industry*, pages 254–264. Springer, 2017.

[oth3] Stanislav Misak, Jindrich Stuchly, Jakub Vramba, Tomas Vantuch, and David Seidl. A novel approach to adaptive active relay protection system in single phase ac coupling off-grid systems. *Electric power systems research*, 131:159–167, 2016.

[oth4] Tomas Vantuch and Ivan Zelinka. Evolutionary based arima models for stock price forecasting. In *ISCS 2014: Interdisciplinary Symposium on Complex Systems*, pages 239–247. Springer, 2015.

[oth5] Stanislav Misak, Jindrich Stuchly, Tomas Vantuch, Tomas Burianek, David Seidl, and Lukas Prokop. A holistic approach to power quality parameter optimization in ac coupling off-grid systems. *Electric Power Systems Research*, 147:165–173, 2017.

[oth6] Tomáš Vantuch, Stanislav Mišák, Tomáš Ježowicz, Tomáš Buriánek, and Václav Snášel. The power quality forecasting model for off-grid system supported by multiobjective optimization. *IEEE Transactions on Industrial Electronics*, 64(12):9507–9516, 2017.

[oth7] Tomáš Vantuch, Stanislav Mišák, and Jindřich Stuchlỳ. Power quality prediction designed as binary classification in ac coupling off-grid system. In *Environment and Electrical Engineering (EEEIC), 2016 IEEE 16th International Conference on*, pages 1–6. IEEE, 2016.

[oth8] Tomas Vantuch, Jindrich Stuchly, Stanislav Misak, and Tomas Burianek. Data mining application on complex dataset from the off-grid systems. In *Mendel 2015*, pages 63–75. Springer, 2015.

[oth9] Tomas Burianek, Tomas Vantuch, Jindrich Stuchly, and Stanislav Misak. Off-grid parameters analysis method based on dimensionality reduction and self-organizing map. In *Mendel 2015*, pages 235–244. Springer, 2015.

[oth10] Tomáš Vantuch. Impact of hurst exponent on indicator based trading strategies. In *Nostradamus 2014: Prediction, Modeling and Analysis of Complex Systems*, pages 337–345. Springer, 2014.

[oth11] Tomas Vantuch, Ivan Zelinka, and Pandian Vasant. An algorithm for elliott waves pattern detection. *Intelligent Decision Technologies*, (Preprint):1–10.

[oth12] Tomáš Vantuch, Jan Fulneček, Michael Holuša, Stanislav Mišák, and Jan Vaculík. An examination of thermal features' relevance in the task of battery-fault detection. *Applied Sciences*, 8(2):182, 2018.

[oth13] Lumir Kojecky, Ivan Zelinka, Awadhesh Prasad, Tomaš Vantuch, and Lukas Tomaszek. Investigation on unconventional synthesis of astroinformatic data classificator powered by irregular dynamics. *IEEE Intelligent Systems*, (Accepted).

[oth14] Tomáš Vantuch, Aurora González Vidal, Alfonso P. Ramallo-González, Antonio F. Skarmeta, and Stanislav Misak. Machine learning based electric load forecasting for short and long-term period. In *IEEE World Forum on Internet of Things 2018*. IEEE, 2018.

[oth15] Tomáš Vantuch and Michal Prilepok. An ensemble of multi-objective optimized fuzzy regression models for short-term electric load forecasting. In *IEEE Symposium Series on Computational Intelligence 2017*. IEEE, 2017.

BIBLIOGRAPHY

[1] Min Wu, Hong Cao, Jianneng Cao, Hai-Long Nguyen, Joao Bartolo Gomes, and Shonali Priyadarsini Krishnaswamy. An overview of state-of-the-art partial discharge analysis techniques for condition monitoring. *IEEE electrical insulation magazine*, 31(6):22–35, 2015.

[2] Carlos H Caldas and Lucio Soibelman. Automating hierarchical document classification for construction management information systems. *Automation in Construction*, 12(4):395–406, 2003.

[3] Jonathan Cheung-Wai Chan and Desiré Paelinckx. Evaluation of random forest and adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. *Remote Sensing of Environment*, 112(6):2999–3011, 2008.

[4] L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and Paulino Ribeiro Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in physics*, 56(1):167–242, 2007.

[5] HK Agarwal, K Mukherjee, and P Barna. Partially and fully insulated conductor systems for low and medium voltage over head distribution lines. In *Condition Assessment Techniques in Electrical Systems (CATCON), 2013 IEEE 1st International Conference on*, pages 100–104. IEEE, 2013.

[6] Esmaeil Hemmati and S Mohammad Shahrtash. Evaluation of unshielded rogowski coil for measuring partial discharge signals. In *Environment and Electrical Engineering (EEEIC), 2012 11th International Conference on*, pages 434–439. IEEE, 2012.

[7] Mohammad Hamed Samimi, Arash Mahari, Mohammad Ali Farahnakian, and Hossein Mohseni. A review on the rogowski coil principles and applications. *measurements*, 4:5, 2013.

[8] A Martinez Nóbrega, MLB Martinez, and Alvaro Antonio Alencar de Queiroz. Analysis of the xlpe insulation of distribution covered conductors in brazil. *Journal of materials engineering and performance*, 23(3):723–735, 2014.

[9] Stanislav Misak and Stefan Hamacek. Utilization of the finite element method for optimizing of overhead covered conductors. *Annals of DAAAM & Proceedings*, 2010.

[10] Irina Makhkamova, Philip C Taylor, JR Bumby, and Khamid Mahkamov. Cfd analysis of the thermal state of an overhead line conductor. In *Universities Power Engineering Conference, 2008. UPEC 2008. 43rd International*, pages 1–4. IEEE, 2008.

[11] G.M. Hashmi, M. Lehtonen, and M. Nordman. Modeling and experimental verification of on-line pd detection in mv covered-conductor overhead networks. *Dielectrics and Electrical Insulation, IEEE Transactions on*, 17(1):167–180, February 2010.

[12] Hao Zhang, TR Blackburn, BT Phung, and D Sen. A novel wavelet transform technique for on-line partial discharge measurements. 1. wt de-noising algorithm. *IEEE Transactions on Dielectrics and Electrical Insulation*, 14(1), 2007.

[13] Randolph B Randall. *Frequency analysis*. Brül & Kjor, 1987.

[14] European Communication Committee. The european table of frequency allocations and applications in the frequency range 8.3 khz to 3000 ghz (eca table). *ECO 2015*, 15.1.2016.

[15] Christopher Haslett. *Essentials of radio wave propagation*. Cambridge University Press, 2008.

[16] Les Barclay. *Propagation of radiowaves*, volume 502. Iet, 2003.

[17] Julio Hernandez, Jesús Ariel Carrasco-Ochoa, and José Francisco Martínez-Trinidad. An empirical study of oversampling and undersampling for instance selection methods on imbalance datasets. In *Iberoamerican Congress on Pattern Recognition*, pages 262–269. Springer, 2013.

[18] Vaishali Ganganwar. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47, 2012.

[19] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 875–886. Springer, 2009.

[20] NC Sahoo, MMA Salama, and R Bartnikas. Trends in partial discharge pattern classification: a survey. *IEEE Transactions on Dielectrics and Electrical Insulation*, 12(2):248–264, 2005.

[21] T Hucker and H-G Krantz. Requirements of automated pd diagnosis systems for fault identification in noisy conditions. *IEEE transactions on dielectrics and electrical insulation*, 2(4):544–556, 1995.

[22] H-G Kranz. Diagnosis of partial discharge signals using neural networks and minimum distance classification. *IEEE Transactions on Electrical Insulation*, 28(6):1016–1024, 1993.

[23] RE James and BT Phung. Development of computer-based measurements and their application to pd pattern analysis. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2(5):838–856, 1995.

[24] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[25] JIN JUN. *Noise reduction and source recognition of partial discharge signals in gas-insulated substation*. PhD thesis, 2006.

[26] E Gulski. Computer-aided measurement of partial discharges in hv equipment. *Electrical Insulation, IEEE Transactions on*, 28(6):969–983, 1993.

[27] Ivy Shim, John J Soraghan, and WH Siew. Detection of pd utilizing digital signal processing methods. part 3: Open-loop noise reduction. *IEEE Electrical Insulation Magazine*, 17(1):6–13, 2001.

[28] Mehdi Allahbakhshi and Asghar Akbari. A method for discriminating original pulses in online partial discharge measurement. *Measurement*, 44(1):148–158, 2011.

[29] S Sriram, S Nitin, KMM Prabhu, and MJ Bastiaans. Signal denoising techniques for partial discharge measurements. *IEEE Transactions on Dielectrics and Electrical Insulation*, 12(6):1182–1191, 2005.

[30] Ronggen Yang and Mingwu Ren. Wavelet denoising using principal component analysis. *Expert systems with Applications*, 38(1):1073–1076, 2011.

[31] Caio FFC Cunha, André T Carvalho, Mariane R Petraglia, and Antonio CS Lima. A new wavelet selection method for partial discharge denoising. *Electric Power Systems Research*, 125:184–195, 2015.

[32] X Ma, Chengke Zhou, and IJ Kemp. Automated wavelet selection and thresholding for pd detection. *IEEE Electrical Insulation Magazine*, 18(2):37–45, 2002.

[33] CS Chang, J Jin, C Chang, Toshihiro Hoshino, Masahiro Hanai, and Nobumitsu Kobayashi. Separation of corona using wavelet packet transform and neural network for detection of partial discharge in gas-insulated substations. *IEEE Transactions on power Delivery*, 20(2):1363–1369, 2005.

[34] Xiaodi Song, Chengke Zhou, Donald M Hepburn, Guobin Zhang, and Matthieu Michel. Second generation wavelet transform for data denoising in pd measurement. *Dielectrics and Electrical Insulation, IEEE Transactions on*, 14(6):1531–1537, 2007.

[35] Xu Zhongrong, Tang Ju, and Sun Caixin. Application of complex wavelet transform to suppress white noise in gis uhf pd signals. *Power Delivery, IEEE Transactions on*, 22(3):1498–1504, 2007.

[36] Lukun Wang, Xiaoying Zhao, Jiangnan Pei, and Gongyou Tang. Transformer fault diagnosis using continuous sparse autoencoder. *SpringerPlus*, 5(1):448, 2016.

[37] Maheswari Ramasamy Velayutham, Subburaj Perumal, Vigneshwaran Basharan, and Willjuice Iruthayarajan Maria Silluvairaj. Support vector machine-based denoising technique for removal of white noise in partial discharge signal. *Electric Power Components and Systems*, 42(14):1611–1622, 2014.

[38] Mohsen Bakhshi Ashtiani and S Mohammad Shahrtash. Partial discharge denoising employing adaptive singular value decomposition. *IEEE Transactions on Dielectrics and Electrical Insulation*, 21(2):775–782, 2014.

[39] L Satish and B Nazneen. Wavelet-based denoising of partial discharge signals buried in excessive noise and interference. *Dielectrics and Electrical Insulation, IEEE Transactions on*, 10(2):354–367, 2003.

[40] Amira A Mazroua, MMA Salama, and R Bartnikas. Pd pattern recognition with neural networks using the multilayer perceptron technique. *IEEE Transactions on Electrical Insulation*, 28(6):1082–1089, 1993.

[41] Amira A Mazroua, R Bartnikas, and MMA Salama. Discrimination between pd pulse shapes using different neural network paradigms. *IEEE Transactions on Dielectrics and Electrical Insulation*, 1(6):1119–1131, 1994.

[42] Amira A Mazroua, R Bartnikas, and MMA Salama. Neural network system using the multi-layer perceptron technique for the recognition of pd pulse shapes due to cavities and electrical trees. *IEEE transactions on power delivery*, 10(1):92–96, 1995.

[43] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[44] Michael O'Neil and Conor Ryan. Grammatical evolution. In *Grammatical Evolution*, pages 33–47. Springer, 2003.

[45] Binh Tran, Bing Xue, and Mengjie Zhang. Genetic programming for feature construction and selection in classification on high-dimensional data. *Memetic Computing*, 8(1):3–15, 2016.

[46] Ling Guo, Daniel Rivero, Julián Dorado, Cristian R Munteanu, and Alejandro Pazos. Automatic feature extraction using genetic programming: an application to epileptic eeg classification. *Expert Systems with Applications*, 38(8):10425–10436, 2011.

[47] Robert I Lerman and Shlomo Yitzhaki. A note on the calculation and interpretation of the gini index. *Economics Letters*, 15(3-4):363–368, 1984.

[48] John T Kent. Information gain and a general measure of correlation. *Biometrika*, 70(1):163–173, 1983.

[49] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[50] Norbert Marwan, M Carmen Romano, Marco Thiel, and Jürgen Kurths. Recurrence plots for the analysis of complex systems. *Physics reports*, 438(5):237–329, 2007.

[51] I Candel, A Digulescu, A Şerbănescu, and E Sofron. Partial discharge detection in high voltage cables using polyspectra and recurrence plot analysis. In *Communications (COMM), 2012 9th International Conference on*, pages 19–22. IEEE, 2012.

[52] Xiaoxing Zhang, Song Xiao, Na Shu, Ju Tang, and Wei Li. Gis partial discharge pattern recognition based on the chaos theory. *IEEE Transactions on Dielectrics and Electrical Insulation*, 21(2):783–790, 2014.

[53] Alan Wolf, Jack B Swift, Harry L Swinney, and John A Vastano. Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16(3):285–317, 1985.

[54] What Is Data Mining. Data mining: Concepts and techniques. *Morgan Kaufinann*, 2006.

[55] Richard Boddy and Gordon Smith. *Statistical methods in practice: for scientists and technologists*. John Wiley & Sons, 2009.

[56] Janett Walters-Williams and Yan Li. Estimation of mutual information: A survey. *Rough Sets and Knowledge Technology*, pages 389–396, 2009.

[57] Jorge R Vergara and Pablo A Estévez. A review of feature selection methods based on mutual information. *Neural Computing and Applications*, 24(1):175–186, 2014.

[58] Raul Vicente, Michael Wibral, Michael Lindner, and Gordon Pipa. Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of computational neuroscience*, 30(1):45–67, 2011.

[59] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[60] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Aaai*, volume 2, pages 129–134, 1992.

[61] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1):1–6, 1998.

[62] Raymond Wong and Jeen Keen. *Partial discharge classification on xlpe cable joints under different noise levels using artificial intelligence techniques*. PhD thesis, University of Malaya, 2016.

[63] Yu Han and YH Song. Using improved self-organizing map for partial discharge diagnosis of large turbogenerators. *IEEE Transactions on Energy Conversion*, 18(3):392–399, 2003.

[64] KX Lai, BT Phung, and TR Blackburn. Partial discharge analysis using pca and som. In *Power Tech, 2007 IEEE Lausanne*, pages 2133–2138. IEEE, 2007.

[65] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. In James A. Anderson and Edward Rosenfeld, editors, *Neurocomputing: Foundations of Research*, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[66] L Satish and BI Gururaj. Partial discharge pattern classification using multilayer neural networks. *IEE Proceedings A (Science, Measurement and Technology)*, 140(4):323–330, 1993.

[67] H.-G. Kranz. Diagnosis of partial discharge signals using neural networks and minimum distance classification. *Electrical Insulation, IEEE Transactions on*, 28(6):1016–1024, Dec 1993.

[68] Yasin 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. *Electrical Engineering*, 98(1):29–42, 2016.

[69] Abdullahi Abubakar Mas'ud, Ricardo Albarracín, Jorge Alfredo Ardila-Rey, Firdaus Muhammad-Sukki, Hazlee Azil Illias, Nurul Aini Bani, and Abu Bakar Munir. Artificial neural network application for partial discharge recognition: Survey and future directions. *Energies*, 9(8):574, 2016.

[70] Edward Gulski and A Krivda. Neural networks as a tool for recognition of partial discharges. *IEEE transactions on electrical insulation*, 28(6):984–1001, 1993.

[71] Martin Hoof, Bernd Freisleben, and Rainer Patsch. Pd source identification with novel discharge parameters using counterpropagation neural networks. *IEEE Transactions on Dielectrics and Electrical Insulation*, 4(1):17–32, 1997.

[72] Jing Liang Zhou Sha. Pattern recognition of partial discharge based on moment features and probabilistic neural network. *Power System Protection and Control*, 44(3):98–102, 2016.

[73] VM Catterson and B Sheng. Deep neural networks for understanding and diagnosing partial discharge data. In *Electrical Insulation Conference (EIC), 2015 IEEE*, pages 218–221. IEEE, 2015.

[74] Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, Håkan Hjalmarsson, and Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

[75] Hilton de Oliveira Mota, Leonardo Chaves Dutra da Rocha, Thiago Cunha de Moura Salles, and Flávio Henrique Vasconcelos. Partial discharge signal denoising with spatially adaptive wavelet thresholding and support vector machines. *Electric Power Systems Research*, 81(2):644–659, 2011.

[76] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[77] Matthew N Anyanwu and Sajjan G Shiva. Comparative analysis of serial decision tree classification algorithms. *International Journal of Computer Science and Security*, 3(3):230–240, 2009.

[78] KX Lai, BT Phung, and TR Blackburn. Descriptive data mining of partial discharge using decision tree with genetic algorithm. *Australian Journal of Electrical and Electronics Engineering*, 6(3):249–259, 2009.

[79] TK Abdel-Galil, RM Sharkawy, Magdy MA Salama, and R Bartnikas. Partial discharge pattern classification using the fuzzy decision tree approach. *IEEE Transactions on Instrumentation and Measurement*, 54(6):2258–2263, 2005.

[80] Zhou Zhou, Gangquan Si, Jiaxi Chen, Kai Zheng, and Wenmeng Yue. A novel method of transformer fault diagnosis based on k-mediods and decision tree algorithm. In *Electrical Materials and Power Equipment (ICEMPE), 2017 1st International Conference on*, pages 369–373. IEEE, 2017.

[81] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[82] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[83] Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim, and Sung Yang Bang. Constructing support vector machine ensemble. *Pattern recognition*, 36(12):2757–2767, 2003.

[84] Imran Maqsood, Muhammad Riaz Khan, and Ajith Abraham. An ensemble of neural networks for weather forecasting. *Neural Computing & Applications*, 13(2):112–122, 2004.

[85] Francisco Herrera, Francisco Charte, Antonio J Rivera, and María J del Jesus. Ensemble-based classifiers. In *Multilabel Classification*, pages 101–113. Springer, 2016.

[86] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[87] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011.

[88] Yuan Ren and Guangchen Bai. Determination of optimal svm parameters by using ga/pso. *Journal of computers*, 5(8):1160–1168, 2010.

[89] Yukun Bao, Zhongyi Hu, and Tao Xiong. A pso and pattern search based memetic algorithm for svms parameters optimization. *Neurocomputing*, 117:98–106, 2013.

[90] Wong Jee Keen Raymond, Hazlee Azil Illias, Hazlie Mokhlis, et al. Partial discharge classifications: Review of recent progress. *Measurement*, 68:164–181, 2015.

[91] M Geethanjali, S Mary Raja Slochanal, and R Bhavani. Pso trained ann-based differential protection scheme for power transformers. *Neurocomputing*, 71(4):904–918, 2008.

[92] Tsair-Fwu Lee, Ming-Yuan Cho, Chin-Shiuh Shieh, and Fu-Min Fang. Particle swarm optimization-based svm application: Power transformers incipient fault syndrome diagnosis. In *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, volume 1, pages 468–472. IEEE, 2006.

[93] Lanre Olatomiwa, Saad Mekhilef, Shahaboddin Shamshirband, Kasra Mohammadi, Dalibor Petković, and Ch Sudheer. A support vector machine–firefly algorithm-based model for global solar radiation prediction. *Solar Energy*, 115:632–644, 2015.

[94] Paulo F Ribeiro. Wavelet transform: an advanced tool for analyzing non-stationary harmonic distortions in power systems. *Proceedings IEEE ICHPS VI*, pages 365–369, 1994.

[95] Haitao Guo C. Sidney Burrus, Ramesh A. Gopinath. *Introduction to Wavelets and Wavelet Transforms*. Prentice Hall; 1 edition, August 24 1997.

[96] T.B. Littler and D.J. Morrow. Wavelets for the analysis and compression of power system disturbances. *Power Delivery, IEEE Transactions on*, 14(2):358–364, Apr 1999.

[97] Suresh K.Gawre, N.P.Patidar, and R. K. Nema. Article: Application of wavelet transform in power quality: A review. *International Journal of Computer Applications*, 39(18):30–36, February 2012. Full text available.

[98] Ling Guo, Daniel Rivero, Julián Dorado, Cristian R. Munteanu, and Alejandro Pazos. Automatic feature extraction using genetic programming: An application to epileptic {EEG} classification. *Expert Systems with Applications*, 38(8):10425 – 10436, 2011.

[99] Zhu Xizhi. The application of wavelet transform in digital image processing. In *MultiMedia and Information Technology, 2008. MMIT '08. International Conference on*, pages 326–329, Dec 2008.

[100] Valdomiro Vega, Nelson Kagan, Gabriel Ordonez, and Cesar Duarte. Automatic power quality disturbance classification using wavelet, support vector machine and artificial neural network. In *Electricity Distribution - Part 1, 2009. CIRED 2009. 20th International Conference and Exhibition on*, pages 1–4, June 2009.

[101] S. Santoso, W.M. Grady, E.J. Powers, J. Lamoree, and S.C. Bhatt. Characterization of distribution power quality events with fourier and wavelet transforms. *Power Delivery, IEEE Transactions on*, 15(1):247–254, Jan 2000.

[102] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674–693, Jul 1989.

[103] Usman Seljuq, Faraz Himayun, and Haroon Rasheed. Selection of an optimal mother wavelet basis function for ecg signal denoising. In *Multi-Topic Conference (INMIC), 2014 IEEE 17th International*, pages 26–30. IEEE, 2014.

[104] Jian Li, Tianyan Jiang, Stanislaw Grzybowski, and Changkui Cheng. Scale dependent wavelet selection for de-noising of partial discharge detection. *IEEE Transactions on Dielectrics and electrical insulation*, 17(6), 2010.

[105] Pantelis D Agoris, Sander Meijer, Edward Gulski, and Johan J Smit. Threshold selection for wavelet denoising of partial discharge data. In *Electrical Insulation, 2004. Conference Record of the 2004 IEEE International Symposium on*, pages 62–65. IEEE, 2004.

[106] G Suganya, S Jayalalitha, K Kannan, and S Venkatesh. Survey of de-noising techniques for partial discharge interferences. 2006.

[107] David L Donoho and Iain M Johnstone. Threshold selection for wavelet shrinkage of noisy data. In *Engineering in Medicine and Biology Society, 1994. Engineering*

Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE, volume 1, pages A24–A25. IEEE, 1994.

[108] David L Donoho, Iain M Johnstone, et al. Minimax estimation via wavelet shrinkage. *The annals of Statistics*, 26(3):879–921, 1998.

[109] RV Maheswari, B Vigneshwaran, and L Kalaivani. Genetic algorithm based automated threshold estimation in translation invariant wavelet transform for denoising pd signal. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 34(4):1252–1269, 2015.

[110] Guomin Luo and Daming Zhang. Recognition of partial discharge using wavelet entropy and neural network for tev measurement. In *Power System Technology (POWERCON), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.

[111] Virginia Klema and Alan Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, 25(2):164–176, 1980.

[112] James Kennedy and Russell C Eberhart. Particle swarm optimization. In *Proc. of the IEEE International Conference on Neural Networks*, pages 1942–1948. 1995.

[113] Hasan Demirel, Cagri Ozcinar, and Gholamreza Anbarjafari. Satellite image contrast enhancement using discrete wavelet transform and singular value decomposition. *IEEE Geoscience and remote sensing letters*, 7(2):333–337, 2010.

[114] Ajit Rajwade, Anand Rangarajan, and Arunava Banerjee. Image denoising using the higher order singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):849–862, 2013.

[115] I.T. Jolliffe. *Principal component analysis*. Springer series in statistics. Springer-Verlang, 1986.

[116] Ioan Buciu. Non-negative matrix factorization, a new tool for feature extraction: Theory and applications.

[117] Geoffrey E Hinton, Peter Dayan, and Michael Revow. Modeling the manifolds of images of handwritten digits. *Neural Networks, IEEE Transactions on*, 8(1):65–74, 1997.

[118] Anand Rajaraman, Jeffrey D Ullman, Jeffrey David Ullman, and Jeffrey David Ullman. *Mining of massive datasets*, volume 77. Cambridge University Press Cambridge, 2012.

[119] Mehmed Kantardzic. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[120] Jonathon Shlens. A tutorial on principal component analysis. In *Systems Neurobiology Laboratory, Salk Institute for Biological Studies*, 2005.

[121] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001.

[122] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. *Unsupervised and Transfer Learning Challenges in Machine Learning, Volume 7*, page 43, 2012.

[123] Xiaofei He, Deng Cai, Shuicheng Yan, and Hong-Jiang Zhang. Neighborhood preserving embedding. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1208–1213 Vol. 2, Oct 2005.

[124] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. cited By 5799.

[125] X Niyogi. Locality preserving projections. In *Neural information processing systems*, volume 16, page 153. MIT, 2004.

[126] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002.

[127] Janett Walters-Williams and Yan Li. Estimation of mutual information: A survey. In Peng Wen, Yuefeng Li, Lech Polkowski, Yiyu Yao, Shusaku Tsumoto, and Guoyin Wang, editors, *Rough Sets and Knowledge Technology*, volume 5589 of *Lecture Notes in Computer Science*, pages 389–396. Springer Berlin Heidelberg, 2009.

[128] Young-Il Moon, Balaji Rajagopalan, and Upmanu Lall. Estimation of mutual information using kernel density estimators. *Physical Review E*, 52(3):2318–2321, 1995.

[129] C. W. J. Granger. Essays in econometrics. chapter Investigating Causal Relations by Econometric Models and Cross-spectral Methods, pages 31–47. Harvard University Press, Cambridge, MA, USA, 2001.

[130] Lionel Barnett and Anil K. Seth. The mvgc multivariate granger causality toolbox: A new approach to granger-causal inference.

[131] Hirotugu Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[132] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

[133] L.J.P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review, 2008.

[134] Stephen Butterworth. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.

[135] Gauthier Doquire, Michel Verleysen, et al. A comparison of multivariate mutual information estimators for feature selection. In *ICPRAM (1)*, pages 176–185, 2012.

[136] Wei-min BI, Ju Tang, Chen-guo YAO, and Sheng-li SONG. Simulation and experiment study on wavelet packet decomposition based on entropy threshold for dsi rejection of pd. *Proceedings of the CSEE*, 5:028, 2003.

[137] Xiaotian Bi, Ang Ren, Simeng Li, Mingming Han, and Qingquan Li. An advanced partial discharge recognition strategy of power cable. *Journal of Electrical and Computer Engineering*, 2015:48, 2015.

[138] David R White. Software review: the ecj toolkit. *Genetic Programming and Evolvable Machines*, 13(1):65–67, 2012.

[139] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[140] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[141] Chelai Yin, Lixing Zhou, and Yini Luo. Applications of chaos theory on partial discharge detection and character analysis. In *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*, pages 1–4. IEEE, 2008.

[142] Jun Gao, Youyuan Wang, Ruijin Liao, Ke Wang, Lei Yuan, and Yiyi Zhang. Investigation on oil-paper degradation subjected to partial discharge using chaos theory. *Journal of Electrical Engineering and Technology*, 9(5):1686–1693, 2014.

[143] LA Petrov, PL Lewin, and Tadeusz Czaszejko. On the applicability of nonlinear time series methods for partial discharge analysis. *IEEE Transactions on Dielectrics and Electrical Insulation*, 21(1):284–293, 2014.

[144] Steven M Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301, 1991.

[145] Ki H Chon, Christopher G Scully, and Sheng Lu. Approximate entropy for all signals. *IEEE engineering in medicine and biology magazine*, 28(6), 2009.

[146] Floris Takens. Invariants related to dimension and entropy. *Atas do*, 13:353–359, 1983.

[147] Joshua S Richman and J Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049, 2000.

[148] Peter Grassberger and Itamar Procaccia. Characterization of strange attractors. *Physical review letters*, 50(5):346, 1983.

[149] Georg A Gottwald and Ian Melbourne. A new test for chaos in deterministic systems. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 460(2042):603–611, 2004.

[150] Georg A Gottwald and Ian Melbourne. On the implementation of the 0–1 test for chaos. *SIAM Journal on Applied Dynamical Systems*, 8(1):129–145, 2009.

[151] Ronald R Hocking. *Methods and applications of linear models: regression and the analysis of variance*. John Wiley & Sons, 2013.

[152] Martin G Larson. Analysis of variance. *Circulation*, 117(1):115–121, 2008.

[153] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[154] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[155] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[156] James Jaccard, Michael A Becker, and Gregory Wood. Pairwise multiple comparison procedures: A review. *Psychological Bulletin*, 96(3):589, 1984.

[157] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006.

[158] Norbert Marwan, Jonathan F Donges, Yong Zou, Reik V Donner, and Jürgen Kurths. Complex network approach for recurrence analysis of time series. *Physics Letters A*, 373(46):4246–4254, 2009.

[159] Reik V Donner, Michael Small, Jonathan F Donges, Norbert Marwan, Yong Zou, Ruoxi Xiang, and Jürgen Kurths. Recurrence-based time series analysis by means of complex network methods. *International Journal of Bifurcation and Chaos*, 21(04):1019–1046, 2011.

[160] Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nuno. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13):4972–4975, 2008.

[161] Giuliano Armano and Marco Alberto Javarone. Clustering datasets by complex networks analysis. *Complex adaptive systems modeling*, 1(1):1, 2013.

[162] Giuliano Armano and Marco Alberto Javarone. Datasets as interacting particle systems: a framework for clustering. *arXiv preprint arXiv:1202.0077*, 2012.

[163] Pablo Kaluza, Andrea Kölzsch, Michael T Gastner, and Bernd Blasius. The complex network of global cargo ship movements. *Journal of the Royal Society Interface*, 7(48):1093–1103, 2010.

[164] Lei Lin, Qian Wang, and Adel Sadek. Data mining and complex network algorithms for traffic accident analysis. *Transportation Research Record: Journal of the Transportation Research Board*, (2460):128–136, 2014.

[165] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069, 2010.

[166] G. Caldarelli. *Complex Networks*. EOLSS Publications, 2010.

[167] Mark EJ Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[168] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[169] Sadegh Aliakbary, Jafar Habibi, and Ali Movaghar. Feature extraction from degree distribution for comparison and analysis of complex networks. *The Computer Journal*, page bxv007, 2015.

[170] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[171] Jeannette Janssen, Matt Hurshman, and Nauzer Kalyaniwalla. Model selection for social networks using graphlets. *Internet Mathematics*, 8(4):338–363, 2012.

[172] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, volume 29, pages 251–262. ACM, 1999.

[173] Mark EJ Newman. Assortative mixing in networks. *Physical review letters*, 89(20):208701, 2002.

[174] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[175] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[176] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[177] Lourdes Araujo, Juan Martinez-Romo, and Andrés Duque. Grammatical evolution for identifying wikipedia taxonomies. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1345–1346. ACM, 2015.

[178] Michael Fenton, Ciaran McNally, Jonathan Byrne, Erik Hemberg, James McDermott, and Michael O'Neill. Automatic innovative truss design using grammatical evolution. *Automation in Construction*, 39:59–69, 2014.

[179] Michael O'Neill and Conor Ryan. *Grammatical evolution: evolutionary automatic programming in an arbitrary language*, volume 4. Springer Science & Business Media, 2003.

[180] Frank. Rosenblatt. *Principles of neurodynamics; perceptrons and the theory of brain mechanisms*. Spartan Books Washington, 1962.

[181] A spiking neural network architecture for nonlinear function approximation. *Neural Networks*, 14(6–7):933 – 939, 2001.

[182] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[183] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In David E. Rumelhart, James L. McClelland, and CORPORATE PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[184] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. In James A. Anderson and Edward Rosenfeld, editors, *Neurocomputing: Foundations of Research*, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[185] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[186] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 1995.

[187] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1):307–319, 2003.

[188] Sayan Mukherjee, Edgar Osuna, and Federico Girosi. Nonlinear prediction of chaotic time series using support vector machines. In *Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop*, pages 511–520. IEEE, 1997.

[189] Raúl Rojas. Adaboost and the super bowl of classifiers a tutorial introduction to adaptive boosting. *Freie University, Berlin, Tech. Rep*, 2009.

[190] Leo Breiman. *Bias, variance, and arcing classifiers*. Tech. Rep. 460, Statistics Department, University of California, Berkeley, CA, USA, 1996.

[191] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[192] Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6):1947–1958, 2003.

[193] Anita Prinzie and Dirk Van den Poel. Random forests for multiclass classification: Random multinomial logit. *Expert systems with Applications*, 34(3):1721–1732, 2008.

[194] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[195] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. In *Advances in Neural Information Processing Systems*, pages 431–439, 2013.

[196] Yi-Wei Chen and Chih-Jen Lin. *Feature Extraction: Foundations and Applications*, chapter Combining SVMs with Various Feature Selection Strategies, pages 315–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[197] Tianqi Chen and Tong He. xgboost: extreme gradient boosting. *R package version 0.4-2*, 2015.

[198] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[199] Ivan Zelinka. Soma—self-organizing migrating algorithm. In *New optimization techniques in engineering*, pages 167–217. Springer, 2004.

[200] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.

[201] Abdullah Konak, David W Coit, and Alice E Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[202] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[203] M Fadaee and MAM Radzi. Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: A review. *Renewable and Sustainable Energy Reviews*, 16(5):3364–3369, 2012.

[204] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[205] Jamshid Aghaei, Nima Amjady, and Heidar Ali Shayanfar. Multi-objective electricity market clearing considering dynamic security by lexicographic optimization and augmented epsilon constraint method. *Applied Soft Computing*, 11(4):3846–3858, 2011.

[206] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[207] Z. Boger and H. Guterman. Knowledge extraction from artificial neural network models. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 4, pages 3030–3035 vol.4, Oct 1997.

[208] Lutz Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761 – 767, 1998.

[209] Hao Tieng, Haw-Ching Yang, Min-Hsiung Hung, and Fan-Tien Cheng. A multi-objective optimization approach for selecting key features of machining processes. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 899–904. IEEE, 2014.

[210] Deng Cai, Chiyuan Zhang, and Xiaofei He. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 333–342. ACM, 2010.

[211] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[212] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Advances in neural information processing systems*, pages 507–514, 2005.