Modeling and Recognition of Smart Grid Faults by a Combined Approach of Dissimilarity Measures and One-Class Classification

Enrico De Santis∗†, Lorenzo Livi‡§, Alireza Sadeghian¶ and Antonello Rizzi∥

1Dept. of Information Engineering, Electronics, and Telecommunications, SAPIENZA University of Rome, Via Eudossiana 18, 00184 Rome, Italy
2Dept. of Computer Science, Ryerson University, 350 Victoria Street, Toronto, ON M5B 2K3, Canada

July 28, 2014

Abstract
Detecting faults in electrical power grids is of paramount importance, either from the electricity operator and consumer viewpoints. Modern electric power grids (smart grids) are equipped with smart sensors that allow to gather real-time information regarding the physical condition of the elements forming the whole infrastructure (e.g., cables and related insulation, voltages and currents, breakers status and so on). In real-world smart grid systems, usually, the operator collects additional information that are indirectly connected to the operating status of the grid itself, such as meteorological information. Designing a suitable recognition (discrimination) model of faults in a real-world smart grid system is hence a challenging task. This follows from the heterogeneity of the information that actually determine a typical fault condition. The second point is that, for synthesizing a recognition model, in practice only the conditions of observed faults are usually meaningful. Therefore, a suitable recognition model should be synthesized by making use of the observed fault conditions only. In this paper, we deal with the problem of modeling and recognizing faults in a real-world smart grid system, which supplies the entire city of Rome, Italy. Recognition of faults is addressed by following a combined approach of multiple dissimilarity measures and one-class classification techniques. We provide here an in-depth study related to the available data and the correlations with respect to the solutions found by the proposed one-class classifier. We offer also a comprehensive analysis of the fault recognition results by exploiting a fuzzy set based mechanism for the decisions of the classifier.

Index terms—Smart grid; Localized fault recognition; One-class classification; Clustering; Genetic algorithm; Fuzzy set.

1 Introduction
There are many definitions for a Smart Grid (SG). The SG European Technology Platform defines a SG as an “electricity network that can intelligently integrate the actions of all the connected users, generators, consumers and those that do both, in order to efficiently deliver sustainable economic and secure electricity supply” [3]. A SG employs innovative products and services together with intelligent monitoring, control,
communication, and self-healing technologies in order to (i) facilitate the connection and operation of generators of all sizes and technologies; (ii) allow consumers to play an active role in optimizing the operation of the system; (iii) significantly reduce the environmental impact of the whole electricity supply system; (iv) preserve or improve the level of system reliability, quality of service, and security.

SGs can be considered as an “evolution” rather than a “revolution” of the existing energy networks [2]. The evolution is leaded by the symbiotic exchange between power grid technologies and the Information Communication Technologies (ICT). ICT provide instruments, such as Smart Sensors (SS), to monitor the network status, wired and wireless communication network to collect and transport data, and powerful computational architectures to elaborate them. A SG can be framed, both in the applicative context and in a theoretical framework, as a complex non-linear and time-varying system [8, 16, 25, 28, 29, 45], where heterogeneous elements, including environmental factors, are extremely interconnected through the exchange of both energy and information. Computational Intelligence (CI) techniques offer solid modeling techniques and algorithmic solutions in the SG context [6, 46, 50]. Well-known CI techniques adopted in the SG context include approximate dynamic programming [9], neural networks and fuzzy inference systems for prediction and control [10, 30], and swarm intelligence and evolutionary computation for optimization [5, 12, 41].

An important key issue of SGs is the Decision Support System (DSS), which is an expert system that provides decision support for the commanding and dispatching systems of the power grid. Such a system analyzes the risk for damage of crucial equipments, assesses the power grid security, forecasts and provides warnings about the magnitude and location of possible faults, and timely broadcasts the early-warning signals through suitable communication networks [29]. The information provided by the DSS can be used for Condition Based Maintenance (CBM) in the power grid [35]. CBM is defined as “a philosophy that posits repair or replacement decisions on the current or future condition of assets.” The objective of CBM is thus to minimize the total cost of inspection and repair by collecting and interpreting (heterogeneous) data related to the operating condition of critical components. Through the use of CBM, advanced SS technology has the potential to help utilities to improve the power grid reliability by avoiding unexpected outages. Utilities may be able to reduce maintenance costs by providing “just in time” maintenance, gathering information remotely rather than dispatching personnel, and reducing hence the possibility of equipment replacement costs due to catastrophic failures. Asset management deals with the life cycle analysis of assets within technology in SGs, offering a complete and mutual management model – CBM is usually considered an instance of asset management. A discussion on how the changes in modern power grids have affected the maintenance procedures can be found in [51]; the importance of modern diagnostic techniques is treated in [42].

Collecting heterogeneous measurements in modern SG systems is of paramount importance. As an instance, the available measurements can be used for dealing with various important pattern recognition and data mining problems on SGs, such as fault recognition [7, 37, 53]. On the basis of the data at hand, different problem types could be formulated. In [19] the authors have established a relationship between environmental features and fault causes. A fault cause classifier based on the linear discriminant analysis (LDA) is proposed in [11]. Information regarding weather conditions, longitude-latitude information, and measurements of physical quantities (e.g., currents and voltages) related to the power grid have been taken into account. In [52], the authors proposed a system based on LDA, which processes phasor measurement unit data, with the aim of recognizing and locating faults on power lines. As concern fault diagnosis in power grids, in [49] is proposed a Support Vector Machine (SVM) based method to perform the recognition of faults related to high-voltage transmission lines. The One-Class Quarter-Sphere SVM algorithm is proposed [39] for faults classification in the power grid. The reported experimental evaluation is however performed on synthetically generated data only.

In this paper, we extend our previous work [13] on the problem of modeling and recognizing fault instances in the real-world SG system of ACEA by introducing several improvements – the considered SG feeds the entire city of Rome, Italy. Initially, we introduce the application’s context and the approach followed to implement the one-class classification system used to recognize conditions of fault. Since the ACEA data at hand is highly structured (i.e., it is formed by several heterogeneous information), we designed our own one-class classifier (OCC) that is suitable for this context. The first presented improvement consists in equipping
the designed OCC with the capability of producing also soft output decisions. This is implemented by interpreting the decision regions synthesized by the classifier as fuzzy sets with suitable membership functions [26, 27]. This fact allows us to provide also a measure of reliability concerning the already implemented hard classification mechanism. Finally, as concerns the experiments, we provide (i) several evaluations of the recognition systems on either synthetic and ACEA datasets, and (ii) a more in depth analysis of the informativeness of the solutions found by the classifier.

The paper is structured as follows. We offer a brief review on the one-class classification setting in Sec. 2. In Sec. 3 we describe the considered SG system and related data. Sec. 4 provides full details about the fault recognition system that we designed. In this section, we describe the system as a whole and also the new contributions introduced here in this study. In Sec. 5 we show and discuss the experiments. Finally in Sec. 6 we draw our conclusions.

2 Brief Overview on the One-class Classification Problem

The one-class classification problem can be considered as a particular instance of a standard n-class classification problem, where, during the training stage, patterns belonging to one class only are available. Such patterns are usually termed target or positive patterns. This particular scenario covers several interesting real-world situations [13, 15, 18, 22, 24, 33, 44]. Practically, OCCs define a decision rule on the basis of a model that is able to describe suitable boundaries pertaining the target patterns. Such boundaries define the decision regions/surface of the classifier. The aim of course is to synthesize effective models such that target patterns are recognized while non-target patterns are rejected.

Khan and Madden [24] provided a recent survey on the subject of one-class classification. One important class of OCCs has been elaborated from the well-known SVM [38, 43, 48]. Tax and Duin [43] defined a system called Support Vector Data Description (SVDD), which aims to recognize the target class with respect to any other possible non-target patterns. The classification model is defined in terms of hyper-spheres, which are placed over the training set through an SVM-like optimization problem (the minimization of the sphere radiuses is enforced). Such methods exploit the possibility of using positive definite kernel functions.

Schölkopf et al. [38] proposed an alternative approach to SVDD that employs a hyperplane, like in the conventional SVM case. The hyperplane is synthesized towards the aim of separating the region of the input space containing patterns form the region containing no data. Also this approach has the capability of using kernel functions. Other more recent approaches include algorithms based on the minimum spanning tree [21] and on Gaussian processes [23].

3 The Considered Power Grid

The ACEA power grid is the electrical distribution grid of Rome, the capital city of Italy. The grid is constituted of backbones of uniform section, exerting radially with the possibility of counter-supply if a branch is out of order. Each backbone of the power grid is supplied by two distinct Primary Stations (PS) and each half-line is protected against faults through the breakers. Along each line, there is a breaking point and the position is chosen with respect to the preassigned performance of the electric current that flow in each half-line. The Medium Voltage (MV) power grid consists in lines in which the nominal voltage is 20 kV, with the presence of few “legacy” lines that still work at 8.4 kV. The MV part of the network covers 10,490 km while the Low Voltage (LV) section covers 11.120 km. Cables can be on air or underground and their sections can vary along the backbone with the presence of bottlenecks. The MV section has 1.565 lines in service and it is supplied with 76 PSs, while LV section is supplied with 13.292 Secondary Stations (SS).

We deal with the problem of modeling and recognizing a particular type of grid fault, which is commonly termed as localized fault (LF) [4]. Before providing a precise definition of a LF, it is important to discriminate among outages and faults, according to the CEI 5160 normative [4]. An outage (i.e., an interruption of the service) is the condition in which the voltage on the access point to the electrical energy of a user is less than 5% of the declared voltage on all phases of supply [4]. Three types of outage are considered: (i) long, if the
duration is more than three minutes (long outages); (ii) short, if the duration is more than one second and less than three minutes (short outages); (iii) transient, if the duration does not exceed one second (transient outages).

A fault, instead, is related to the failure of the electrical insulation (e.g., cables insulation) that compromises the correct functioning of (part of) the grid. Therefore, a LF is effectively a fault in which a physical element of the grid is permanently damaged causing long outages.

## 4 The Proposed One-class Classification System for Smart Grid Fault Detection

### 4.1 Representation of a Fault Pattern

Instances of fault patterns (FPs) describing LFs occurred in the SG have been elaborated from a historical database provided by ACEA. The considered period spans across 2009–2012. Faults are characterized by heterogeneous data, including weather conditions, spatio-temporal data (i.e., longitude-latitude pairs and timestamps), physical data related to the state of power grid and its electric equipments (e.g., measured currents and voltages), and finally meteorological data. As a consequence, a FP is effectively defined by features of various types, containing categorical (nominal) data, quantitative data (i.e., data having a well-defined ordering), and also Time Series (TS) describing short outages occurred before a LF. A detailed description of the considered features characterizing a FP is provided in Tab. 1.

| Feature                      | Data typology and features space label | Description                                                                 |
|------------------------------|----------------------------------------|------------------------------------------------------------------------------|
| Day start                    | Quantitative (Integer) \( F_D \)       | Day in which the LF was detected                                            |
| Time start                   | Quantitative (Integer) \( F_T \)        | Time stamp (minutes) in which the LF was detected                           |
| Primary Station (PS) code    | Categorical (String) \( F_C^1 \)        | Unique backbone identifier                                                  |
| Voltage line                 | Categorical (String) \( F_C^2 \)        | Nominal voltage of the backbone                                            |
| Type of element              | Categorical (String) \( F_C^3 \)        | Element that caused the damage                                              |
| Location element             | Categorical (String) \( F_C^4 \)        | Element positioning (aerial or underground)                                 |
| Material                     | Categorical (String) \( F_C^5 \)        | Constituent material element (CU, AL)                                      |
| Primary station fault distance| Quantitative (Real) \( F^Q_1 \)         | Distance between the primary station and the geographical location of the LF|
| Median point                 | Quantitative (Real) \( F^Q_2 \)         | Fault location calculated as median point between two secondary stations    |
| # Secondary Stations (SS)    | Quantitative (Real) \( F^Q_3 \)         | Number of out of service secondary stations after the LF                    |
| Current out of bounds        | Quantitative (Integer) \( F^Q_4 \)      | The maximum operating current of the backbone is less than or equal to 60% of the threshold “out of bounds”, typically established at 90% of capacity |
| Max. temperature             | Quantitative (Real) \( F^Q_5 \)         | Maximum registered temperature                                              |
| Min. temperature             | Quantitative (Real) \( F^Q_6 \)         | Minimum registered temperature                                              |
| Delta temperature section    | Quantitative (Real) \( F^Q_7 \)         | Difference between the maximum and minimum temperature                      |
| Rain                         | Quantitative (Real) \( F^Q_8 \)         | Millimeters of rain calculated as the average three hours preceding to the LF|
| Cable section                | Quantitative (Real) \( F^Q_9 \)         | Section of the cable, if applicable                                          |
| Backbone electric current    | Quantitative (Real) \( F^{EC}_9 \)      | Function of electric current that flows in a given backbone of the considered power grid |
| Interruption (breaker)       | TS (Integer) \( F^{TS}_1 \)            | Outages caused by the opening of the breakers in the primary station         |
| Petersen alarms              | TS (Integer) \( F^{TS}_2 \)            | Alarms detected by the device called “Petersen’s coil” due to loss of electrical insulation on the power line |
| Saving intervention          | TS (Integer) \( F^{TS}_3 \)            | Decisive interventions of the Petersen’s coil which have prevented the LF    |

### 4.2 Data Preprocessing

Data normalization is a universally important aspect in pattern analysis, which becomes even more crucial when processing patterns characterized by many heterogeneous features. The available data provided by

---

Table 1: Description of the considered features describing a FP instance.
ACEA have been normalized using the affine normalization technique:

\[
v = \frac{c - m}{M - m} \in [0, 1].
\]  

\(c\) is the original (non-normalized) value; \(m\) and \(M\) are, respectively, the minimum/maximum values for the specific feature in the considered dataset.

### 4.2.1 Temporal Data

The “Day start” and “Time start” features (Tab. 1) have been encoded as integer values. The former ranges in \(\{0, 1, ..., 364\}\), and for the latter in \(\{0, 1, ..., 1439\}\), which corresponds to the number of minutes in a year. Normalization of those data follows straightforwardly.

### 4.2.2 Spatial Data

Three types of information regarding the geographical position of a LF are available: the absolute position of the PS where the LF has occurred, and the absolute position of the two SSs delimiting the section of power line where the revealing system detected the LF. The original coordinates of the geographical position of the LF have been expressed in WGS84 (decimal degrees), the same that it is used in the GPS geolocalization system. It is reasonable that the informations regarding the PSs positions and the absolute locations of the LFs can provide indirectly the information about the amount of electric current flowing in the power line. The main hypotheses that led us to that statement are: (i) the MV lines have a radial distribution with respect to the PSs and their extension is of the order of kilometers, (ii) the portion of power line between the two SSs have an extension of hundreds of meters. In addition the power grid has a meshed structure and it is exercised “radially”, so that every MV line is supplied through one PS only.

In Fig. 1 is depicted the typical scenario: a MV backbone composed by two lines supplied through two distinct PSs; the cutting point is situated in (roughly) the middle point. From the Kirchoff’s law, the current that flowing in the MV line between the PS denoted by A and the node denoted by the integer 2 is equal at the sum of subsequent currents:

\[
I_{PS\ A-2} = I_2 + I_3 + I_4 + I_5 + I_6.
\]  

The intensity of the electric current in the MV lines decreases as we move away from the PS, until the cutting point is reached.

Thus, those three features (the PS and the two SS positions) have been reduced to two features: (i) the distance between the PS location (“Primary Station fault distance”) and (ii) the middle point among the two SSs (“Median point”) – see Fig. 2.

![Figure 1: Scheme of a MV backbone.](image_url)

The maximum spatial resolution of the geographical localization of the LF is therefore defined by the two SS positions. The distance between two geographical locations is calculated through the Vincenty’s algorithm [47]. The normalization process of the position data is based on the calculation of the largest quadrilateral that includes the farther stations of voltage transform (see Fig. 3 for an example). Hence, applying the affine normalization (1), the spatial positions of the LFs result normalized in [0, 1]. The affine normalization is applied also for the distance values among PSs and the positions of the LFs.
4.2.3 Physical Data

The data describing the physical power grid is defined by both categorical and quantitative information. As concerns categorical data (“Primary station code”), the analyzed dataset has few missing values – less than 5%. However, the missing values of a feature have been substituted with the most frequent category for that feature. The normalization of quantitative data (i.e., “# Secondary station”, “Current out of bounds”, and “Cable section”) is implemented by means of (1).

An important feature that we consider is the electrical current value measured when the LF occurred. This was provided by ACEA in form of a long TS of electric current values, sampled every 10 minutes – data available for each SG backbone. In the dataset, we considered only a single scalar real-value for characterizing the whole TS, which is calculated by considering the data 3 hours before the LF. This time window is divided in two additional time windows, whereby finally computing the difference between the average values in each of those two time windows. This value is normalized with respect to the minimum and maximum values belonging to the considered backbone. This “compression” of the information conveyed by the whole TS describing the electrical current is performed with the aim to capture the average information about the “jumps” of the values of the electric current observed before the LF. Future research works will be focused to the study of the entire TS and its relation/causation with the observed LFs.
4.2.4 Meteorological Data

The meteorological data are acquired by suitable stations located in different areas of Rome. The “Rain” feature is calculated as the average millimeters of rain observed in the 24 hours antecedent the LF.

4.2.5 Short Outages Data

Here we describe the data related to the short outages observed before a LF. With an abuse of notation, we will refer to such sequences of events as time series, although formally such sequences are not sampled with a predefined constant period (those events are registered as they occur).

We consider three types of events that can be reconduted to the “short outages” types (see Sec. 3). The considered TS of events are: “Interruption (breaker)”, the “Petersen alarms”, and the “Saving intervention” (see Tab. 1). The short outages events are represented as variable-length sequences, which contain the temporal distances (expressed in seconds) from the subsequent LF event (see Fig. 4 for a visual example). The time window in which those events fall spans across three months (i.e., we search in the three months preceding a LF). A TS $S^i$ of $K$ outage events is defined as follows:

$$S^i = [\xi_{i1}, \xi_{i2}, ..., \xi_{iK_i(n)}],$$

where $\xi$ is the temporal distance from the LF event considered as the origin, $i \in \{1, 2, 3\}$ is the index distinguishing the three aforementioned types of outages, and $K_i(n)$ is the number of events for the $i$-th type of outage (that depends on the $n$-th pattern).

Figure 4: Representation of TSs of outages happened before a LF.

Normalization of TSs data is performed differently. Given a dissimilarity measure for TSs (see Sec. 4.3.1), we compute off-line, for each short outage feature, the dissimilarity matrix – a square matrix containing the pairwise dissimilarity values. Hence, we apply the normalization dividing the dissimilarity value between two TSs with the maximum value of the dissimilarity matrix.

4.3 The Proposed One-class Classifier

As a consequence of the difficulty of modeling useful (and meaningful) instances of non-faults in the considered SG, we designed an OCC for the purpose of recognizing LF s. Such a goal is implemented by building a OCC relying on clustering techniques. The idea of using a cluster for modeling a region of the idealized “fault space”, $F$, containing target patterns denoting LF s, is reasonable and also intuitive. The underlying assumption is that, similar statuses of the SG have similar chances of generating a LF, assumption that it is reflected by the cluster model.

A dataset of FPs is partitioned in $k$ (disjoint) clusters, where each cluster contains faults having similar features. Accordingly, the most important component of the OCC system is the core dissimilarity measure $d : F \times F \rightarrow \mathbb{R}^+$, which assigns dissimilarity values to a pair of FPs. The partition, as well as other parameters that will be described in the following, constitute the model of the OCC.
4.3.1 The Dissimilarity Measure Among FPs

A FP \( x \in S \) is described as:

\[
x = \{F_1, F_2, ..., F_m\},
\]

where the \( l \)-th feature, \( F_l \), \( 1 \leq l \leq m \), lies in its specific feature space \( F_l \). Hence, each \( x \) lies on the \( m \)-fold product feature space \( F = F_1 \times F_2 \times ... \times F_m \).

Given two FPs \( x, y \in S \subset F \), the proposed weighted dissimilarity measure reads as:

\[
d(x, y; w) = \sqrt{\sum_{j=1}^{m} w_j \times (x_j \odot y_j)^2},
\]

where the \( \odot \) operator represents a generic dissimilarity measure, and \( w_j \in [0, 1] \) is the weight related to the \( j \)-th feature. In practice, Eq. (5) computes the weighted \( l_2 \) norm of the vector containing the dissimilarity values calculated feature-wise. However, since not all dissimilarity measures implementing the \( \odot \) operator are metrics, the resulting dissimilarity measure \( (5) \) is not metric and hence it does not induce a metric space. This aspect is carefully taken in consideration into the design of our OCC System. The weights \( w \in [0, 1]^m \) are suitably optimized during the training phase of the OCC by means of a genetic algorithm (GA).

In the following paragraphs, we describe the implementations of \( \odot \), i.e., the specific dissimilarity measures tailored for each specific FP component. The nature of the feature, \( F_i \), will be denoted using the same notation of Tab. 1.

**Categorical Data** Categorical attributes, also referred to as nominal attributes, are data without an “intuitive” ordering (see Tab. 1 for the data treated as nominal). Let \( F^c = \{\eta_1, \eta_2, ..., \eta_n\} \) be the set of all categorical features of the entire dataset, each described by \( d \) categorical attributes: \( \nu_1, \nu_2, ..., \nu_d \). Let us define the domain of the attribute \( \nu_j \), \( \text{DOM}(\nu_j) = \{A_{j_1}, A_{j_2}, ..., A_{j_{n(j)}}\} \), where \( A_{ji} \ (1 \leq l \leq n(j)) \) is the set of possible values for the categorical attribute \( \nu_j \), and \( n(j) \) is its cardinality. Let us consider the well-known simple matching distance, defined as follows:

\[
\delta(x, y) = \begin{cases} 
0 & x = y, \\
1 & x \neq y.
\end{cases}
\]

Let \( x^c \) and \( y^c \) be the projections on the categorical feature space \( F^c \) of two generic patterns \( x, y \). The dissimilarity measure between the two categorical objects described by \( d \) categorical attributes is implemented as:

\[
d^c(x^c, y^c) = \frac{1}{d} \sum_{j=1}^{d} \delta(x^c_j, y^c_j).
\]

**Quantitative Data** As concerns the quantitative data (see Tab. 1) we distinguish between (i) “Normal” quantitative data and (ii) “Special” quantitative data. The former type includes both numerical and integer values (normalized in \([0, 1]\)); the operator \( \ominus \) is implemented by the absolute difference: \( d^N = |x - y| \).

As concerns integer values that rely to temporal information, such as the day in which the LF happened and the time-stamp within that day, it is defined a particular dissimilarity measure implementing \( \ominus \), called circular difference. Given an ordered set of integer numbers \( \{0, 1, ..., a\} \), the circular difference among any \( x, y \) in this set is given by:

\[
d^{CD}(x, y; a) = \min(|x - y|, a - |x - y|).
\]

For “Day start” and “Time start”, which are referred to the \( F^D \) and \( F^T \) features subspaces, the maximum value for \( a \) in \([8]\) is 364 and 1439, respectively. The implementation of the circular difference is designed to avoid that pairs of close days or timestamps give raise to high values of the dissimilarity function.

“Special” quantitative data are normalized in the range \([0, 1]\), but can assume also a special symbol \( \epsilon \), indicating the “not applicable” condition. It is the case for the “Cable section” feature, since for LFs not
related to cables is undefined. The dissimilarity measure \( d^S : ([0,1] \cup \epsilon) \times ([0,1] \cup \epsilon) \rightarrow [0,1] \) is defined as follows. Given two special quantitative values \( x, y \in F^S \), we have:

\[
\begin{align*}
    d^S &= \begin{cases} 
    |x - y| & x \neq \epsilon \land y \neq \epsilon, \\
    1 & x = \epsilon \lor y = \epsilon, \\
    0 & x = \epsilon \land y = \epsilon.
    \end{cases}
\end{align*}
\]

(9)

**Time Series Data** Dynamic time warping (DTW) is a well-known technique to find an optimal alignment between two sequences of variable length. The use of the DTW as dissimilarity measure for sequences of generic objects is well-established in many applications, such as biology, finance, multimedia, and image analysis [36, 40]. An in-depth description of DTW algorithm can be found in [32].

Following the notation introduced in Sec. 4.2.5, the data set consists in three types of TSs, \( S^i \), with \( i \in \{1, 2, 3\} \). Each one represents a vector belonging to the TSs feature vector subspace, \( F^{iTS} \), \( i \in \{1, 2, 3\} \).

Thus, given two TSs belonging to the subspace \( F^{iTS} \), the OCC computes the component-wise dissimilarity measure by applying the DTW algorithm.

4.3.2 Model Definition and the Classifier Decision Rule

The most important part of the OCC model is the partition \( P \) of \( S_{tr} \), which is obtained through a clustering algorithm – see Fig. 5 for an overview.

![Figure 5: The OCC model is defined as a partition of the training set \( S_{tr} \).](image)

A hard partition of order \( k \) is a collection of \( k \) disjoint and non-empty clusters, \( P = \{C_1, C_2, ..., C_k\} \). Each cluster \( C_i \in P \) is synthetically described by a representative element, which we denote as \( c_i = R(C_i) \); accordingly, let \( R(P) = \{c_1, c_2, ..., c_k\} \) be the set of representatives of the partition \( P \). The representative of \( C_i \) is computed as the element \( c_i \) that minimizes the sum of distances (MinSOD) [14]:

\[
    c_i = \arg \min_{x_j \in C_i} \sum_{x_k \in C_i} d(x_j, x_k).
\]

(10)

A cluster representative \( c_i \) is, in a sense, the prototype of a typical fault scenario individuated in \( S_{tr} \). As a consequence, the information provided by the cluster \( C_i \) as a whole is useful to conceive a region of the pattern space “around” \( c_i \), which describes similar fault scenarios. By defining \( \delta(C_i) \geq 0 \) as a measure of cluster extent, we construct the decision region associated to the cluster \( C_j \), which we use to implement the classification rule. The cluster extent can be computed as the average/maximum intra-cluster dissimilarity value or by considering their standard deviation, for instance. However, in addition to \( \delta(C_i) \) we consider also a tolerance parameter, \( \sigma_i \geq 0 \), for defining the decision region. The decision region derived from a cluster \( C_i \) is hence defined by the quantity \( B(C_i) = \delta(C_i) + \sigma_i \), which effectively defines the neighborhood of \( c_i \).

Fig. 6 provides a schematic overview of a cluster model and its use in the process of classifying a test pattern \( \bar{x} \). The classification rule for a test pattern \( \bar{x} \) operates in two stages. First, the nearest cluster
representative \( c^* \in R(P) \) is individuated according to the following expression:

\[
c^* = \arg \min_{c_j \in R(P)} d(\bar{x}, c_j).
\] (11)

The second step consists in comparing the dissimilarity value \( d(\bar{x}, c^*) \) with \( B(C_i) \). We define a binary-valued function \( h(\cdot) \) that performs the hard classification:

\[
h(x) = \begin{cases} 
1 & \text{if } d(\bar{x}, c^*) \leq B(C_i), \\
0 & \text{otherwise}.
\end{cases}
\] (12)

Along with the hard classification (12), we developed a mechanism based on fuzzy sets to provide the user with a measure of “reliability” associated to the decisions. This topic is discussed in the following section.

### 4.3.3 Evaluating the Reliability of the Classification

A Boolean decision on the fact that a new test pattern (i.e., a given SG status) is a fault or not, is operatively reasonable. However, it is important to provide the user also with an additional measure that quantifies the reliability of such a Boolean decision. This becomes even more appropriate in the particular OCC setting. For this purpose, we equip each cluster \( C_i \) with a suitable membership function \( \mu \), denoted in the following as \( \mu_{C_i}(\cdot) \). In practice, we generate a fuzzy set over \( C_i \). The membership function allows us to quantify the uncertainty (expressed by the membership degree in \([0, 1]\)) of a decision about the recognition of a test pattern. Fig. 7 depicts this idea by an intuitive illustration. Membership values close to either 0 or 1 denote “certain” and hence reliable decisions. When the membership degree assigned to a test pattern is close to 0.5, there no clear distinction about the fact that such a test pattern is really a fault or not (regardless of the correctness of the Boolean decision).

For this purpose, we used a parametric sigmoid model for \( \mu_{C_i}(\cdot) \), which is defined as follows:

\[
\mu_{C_i}(x) = \frac{1}{1 + \exp((-d(c_i, x) - b_i)/a_i)},
\] (13)

where \( a_i, b_i \geq 0 \) are two parameters specific to \( C_i \), and \( d(\cdot, \cdot) \) is the dissimilarity measure [5]. Notably, \( a_i \) is used to control the steepness of the sigmoid (the lower the value, the faster the rate of change), and \( b_i \) is used to translate the function on the input domain. If a cluster (that models a typical fault situation found in the training set) is very compact, then it describes a very specific fault scenario. Therefore, no significant variations should be accepted to consider test patterns as members of this cluster. The converse is also true. If a cluster is characterized by a wide extent, then we might be more tolerant in the evaluation of the membership. Accordingly, the parameter \( a_i \) is defined directly as \( \delta(C_i) \). On the other hand, we define \( b_i \) as \( b_i = \delta(C_i) + \sigma_i/2 \). This allows us to position the part of the sigmoid that changes faster right in-between the area of the decision region determined by the dissimilarity values falling in \([B(C_i) - \sigma_i, B(C_i)]\).
Finally, the soft decision function, $s(\cdot)$, is defined as

$$s(\bar{x}) = \mu_{C^*}(\bar{x}),$$  \hspace{0.5cm} (14)

where $C^*$ is the cluster satisfying Eq. 11.

The evaluation of Eq. 14 over a test set $S_{ts}$, $n = |S_{ts}|$, yields $n$ membership degrees, each assigned to a specific pattern of $S_{ts}$. We can evaluate the reliability of the decisions taken on $S_{ts}$ as a whole by considering a fuzzy set, say $\mathcal{M}$, characterized by those $n$ membership degrees. We can implement such an evaluation measure by calculating the fuzzy entropy [26] of $\mathcal{M}$ according to the following expression,

$$\frac{\text{card}(\mathcal{M} \cap \mathcal{M}^c)}{\text{card}(\mathcal{M} \cup \mathcal{M}^c)} \in [0, 1],$$  \hspace{0.5cm} (15)

where $\cap$ and $\cup$ are defined as the minimum and maximum operators, respectively, and the cardinality of the resulting fuzzy set is taken as the sum of the membership degrees. Fuzzy entropy values close to zero would indicate that, overall, the decisions are reliable. Conversely, if the fuzzy entropy is one (this case is possible when all membership degrees are equal to 0.5), then the decisions are highly unreliable.

It is worth stressing that the reliability measures herein discussed should not be confused with the measures of correctness of the recognition (i.e., the evaluation of the correctness of the discrimination among target and non-target patterns).

### 4.3.4 Training of the OCC by k-means

We propose a learning strategy to synthesize the OCC model that is based on the well-known $k$-means [20]. This clustering procedure depends on an integer parameter, $k$, defining a priori the partition order. The dissimilarity measure described in Sec. 4.3.1 depends on a vector of weights, $\bar{w}$. Moreover, the decision regions - Sec. 4.3.2 - are based on the thresholds $\sigma_i$. Setting those parameters, denoted $p_j = [w_j, \sigma_j]$, is of utmost importance, and of course it has a significative influence on the results yielded by $k$-means.

For this reason, a GA is employed to find optimal values for $p_j$ parameters, maximizing the following objective function:

$$f(p_j) = \alpha A(S_{vs}) + (1 - \alpha) \sum_{i=1}^{k} 1 - \sigma_i.$$  \hspace{0.5cm} (16)

In (16), $A(S_{vs})$ is the performance measure achieved on $S_{vs}$ (we specify the nature of such a measure in the experiments section). The second term in (16) defines a constrain for the (average) cluster extension. The GA is in charge to find the parameters, $p_j$, that minimize the $l_1$ norm of the tolerances used to define the decision regions, while at the same time provide effective performance in terms of recognition. Fig. 8 shows a diagram illustrating the optimization stage.
As concerns the learning phase, it is well-known that the $k$-means algorithm is sensible to the adopted cluster initialization strategy; here we used a fast randomized initialization. To compensate this fact, the current version of the OCC takes as external parameter the $k$ value and a classification model is synthesized for each $k$ in a given user-defined range $k_{\text{min}}, k_{\text{max}}$. For each $k$ in this range, we synthesize three models with different random initializations. The fitness (16) associated to a candidate solution $p_j$ is hence the average of the fitness calculated for those three models. In the test phase, we use a majority voting scheme to decide if a given test pattern falls in one of the synthesized decision regions or not.

5 Experimental Evaluation

In Sec. 5.1 we present the main experimental setting adopted in this paper. Successively, we show the obtained results in terms of quality of the recognition, respectively on synthetically generated data (Sec. 5.2) and the data provided by ACEA (Sec. 5.3).

5.1 Experimental setting

A (one-class) classification problem instance is defined as a triple of disjoint sets, namely training set ($S_{\text{tr}}$), validation set ($S_{\text{vs}}$), and test set ($S_{\text{ts}}$). Given a specific parameters setting, a classification model is synthesized on $S_{\text{tr}}$ and it is validated on $S_{\text{vs}}$. The generalization capability of the model is computed on $S_{\text{ts}}$.

The proposed OCC produces both hard (12) and soft decisions (14). In the hard decision case, we evaluate the recognition performance of the classifier by exploiting the confusion matrix. In particular, we consider the false positive rate (FPR), recall, precision, and accuracy [17]. On the other hand, in the soft decision case, we quantify the correctness of the classifier by computing the area under the ROC curve (AUC) [17] generated by interpreting the membership degrees (14) as suitable “scores” assigned by the classifier to the test patterns.

The OCC parameters defining the model are optimized by means of a GA, which is guided by the objective in Eq. 16. In (16), we implement the accuracy term as the accuracy elaborated from the confusion matrix. In the voting scheme used during the cross-validation, it is possible to obtain more than one model scoring the same highest value of accuracy on $S_{\text{vs}}$. In such a case, we chose the model characterized by the soft decisions denoting the lowest fuzzy entropy (15). This would help in choosing a model that discriminates and it is also more reliable. The GA performs stochastic uniform selection, Gaussian mutation, and scattered crossover (with crossover fraction of 0.8). The GA implements a form of elitism that imports the two fittest individuals in the next generation; the population size is 50. The stop criteria is defined by considering a maximum number of iterations (250) and a check that controls the variations of the best individual fitness.
5.2 Tests on synthetic data

Fig. 9 shows the first synthetic test that we discuss just to show the functioning of the proposed OCC. The target patterns used for training the OCC are distributed in three Gaussian shaped, well-separated, clusters. Patterns used for testing are clearly highly recognizable. The \( k \)-means is executed with \( k = 3 \) and by synthesizing three different models to be used for the majority voting mechanism. The accuracy obtained on the test set with the \( k \)-means is equal to one for all three models. In Fig. 10 we report the soft decisions on the test set of the three best-performing models. While the hard decisions are all correct, we note that with the first model we obtain more reliable decisions. In fact, the computed fuzzy entropy is much lower than the other two (the fuzzy entropy is almost zero). As a consequence, the OCC chooses the results of this model as the final output.

![Figure 9: Synthetic test.](image)

We now move to a synthetic test designed to provide a justification for the generation of the non-fault (i.e., non-target) patterns that we used within the ACEA data (discussed later in Sec. 5.3). Fig. 11(a) illustrates the considered setting. Training, validation, and test target patterns are still grouped in three well-separated Gaussian clusters. Non-target patterns used for validating and testing a model are distributed uniformly over the \([0, 1]^2\) domain; those patterns outnumber the target patterns. By testing the OCC over such data, we expect to observe an “implicit” FPR that is proportional to the number of (training set) target patterns. In other terms, we expect to observe \( \text{FPR} \approx |S_{tr}|/|\hat{S}_{ts}| \), where \( \hat{S}_{ts} \) is the subset of test patterns belonging to the non-target class. We considered 150 target patterns for the training, validation, and test sets, while we used 1500 non-target patterns in the validation and test sets. As expected, the FPR of the best model is \( \approx 0.1082 \), with an overall accuracy of 0.8953. By means of this interpretation, we could safely affirm that the “true” FPR is only \( \approx 0.0082 \), since \( |S_{tr}|/|\hat{S}_{ts}| = 0.1 \). The AUC is 0.9884; Fig. 11(b) shows the calculated membership values for the test patterns. Although the discrimination is very good, the OCC necessarily commits some mistakes, due the uniform distribution of the non-target patterns.

To demonstrate the reliability of this interpretation of the test, we repeated this experiment by increasing the ratio \( |S_{tr}|/|\hat{S}_{ts}| \) from 0.1 to 0.475, and, accordingly, increasing also the spread of the target patterns over the domain. In Fig. 12 we show the linear correlation between the increments of the \( |S_{tr}|/|\hat{S}_{ts}| \) ratio and the calculated FPR over the respective test set. As it was expected, there is a strong linear relationship among those two quantities (correlation coefficient is \( \approx 0.96 \)), which demonstrates that the implicit FPR obtained with this method of generation of non-target patterns is predictable.
5.3 Results on the ACEA data

The ACEA dataset available for our experiments does not contain instances of non-fault situations. This fact creates some difficulty in evaluating any data-driven inference mechanism. To generate instances of non-fault patterns, we use the method discussed in the second experiment presented in the previous section.
Non-target (i.e., non-fault) patterns are formed by generating, with a uniform distribution, each feature value characterizing the FP (see Sec. 4.1 for details on the considered features). In Fig. 13 we show the first two components of the PCA calculated over the dissimilarity matrix, $D_{ij} = d(x_i, x_j), \forall x_i, x_j \in S$, generated for the entire available ACEA dataset $S$, containing either fault and non-fault instances; $d(\cdot, \cdot)$ in Eq. 5 is computed by using unitary weights. The herein considered dataset is divided in training, validation, and test sets according to the following splits. The training set is composed of 532 fault patterns; in the validation set we have 470 fault and 500 non-fault patterns; finally for the test we have 82 fault and 500 non-fault patterns. The number of non-faults is chosen by considering also the typical running time of the OCC synthesis, which is of the order of hours.

![Figure 13: First two components of the PCA elaborated over the dissimilarity matrix constructed from the ACEA dataset containing either fault and non-fault (uniformly generated) patterns.](image)

Tab. 2 shows the obtained results. We tested the OCC for five different values of $k$ in the $k$-means algorithm. For each $k$, we repeated the test five different times by changing the random seed; hence the results are intended as averages with related standard deviations. In the table we show the false positive rate (FPR), the recall (R), the accuracy (A), the area under the ROC curve (AUC), the fuzzy entropy (FE) of the winning model, and finally the mutual information (MI) among the fitness of the best individual and the estimated entropy of the related weights. Let us focus now on the first five columns. Results in terms of recognition, considering both hard and soft decisions, are in general very good. The best overall result is
obtained with $k = 7$ (please note a somewhat clear 7-cluster structure of the fault patterns appearing from the first two components of the PCA in Fig. [13]). Notably, FPRs are always very low, demonstrating the capability of the proposed OCC of synthesizing well-defined and effective decision regions. Notwithstanding, non-fault patterns are generated by using a uniform distribution over each feature describing a FP (please see Fig. [13]), the system is able to isolate such patterns correctly – FPR results are better than expected. On the same line of thoughts, both accuracy and AUC are nearly one, denoting an almost perfect recognition of FPs. The FE of the best models (see Sec. 4.3.3) is almost negligible (in accord with the very high AUC), which tells us that the soft decisions are also highly reliable.

Table 2: Average test sets results on the ACEA data. Results are reported for five different values of $k$ for the $k$-means algorithm.

| $k$ | FPR | RA | AUC | FE | MI |
|-----|-----|----|-----|----|----|
| 4   | 0.0049±0.003 | 0.9420±0.027 | 0.9880±0.005 | 0.9960±0.002 | 0.0041±0.002 | 0.7059±0.059 |
| 5   | 0.0059±0.001 | 0.9570±0.012 | 0.9890±0.001 | 0.9950±0.001 | 0.0048±0.002 | 0.6168±0.082 |
| 6   | 0.0049±0.002 | 0.9360±0.020 | 0.9850±0.002 | 0.9950±0.001 | 0.0049±0.001 | 0.6652±0.029 |
| 7   | 0.0049±0.003 | 0.9570±0.015 | 0.9900±0.003 | 0.9960±0.002 | 0.0015±0.018 | 0.7878±0.072 |
| 8   | 0.0049±0.003 | 0.9570±0.007 | 0.9960±0.003 | 0.9950±0.001 | 0.0016±0.014 | 0.7268±0.131 |

Let us discuss now the information conveyed by the MI column in Tab. 2. As it is clear from Eq. 5 each feature is weighted by a specific $w_j \in [0,1]$. Such weights, $w_j$, are calculated by exploiting the fitness function in Eq. [16] which searches for the best-performing parameters configuration on a suitable validation set. It is important to assess the importance of the considered features describing a fault pattern with respect to the classification problem at hand. Notably, we expect to find features that are, in average, more relevant than others, thus contributing with more impact in the discrimination process. To verify such a hypothesis, in Fig. [14] we show the estimated density (kernel-based estimator) of the weights related to the best-performing individual; $k = 7$ is considered here. From the figure it is possible to deduce that the features are not uniformly weighted.

It is worth noting that, regardless of the value of $k$, there is a subset of features that is always associated with high weights (details not shown). Such features are: “Time start”, “Primary station fault distance”, “Median point”, “Max temperature”, “Rain”, “Interruption (breaker)”, and “Petersen alarms”. Such features confirm what the expertise of the ACEA company indicates as most important factors congruent to a LF. The time of the day (“Time start”) is a sensible variable due to the changing on energy demand that normally presents two peaks, one at the middle of the day and one in the later evening. The distance from the PS (“Primary station fault distance”) and the absolute position of the LF (“Median point”) is also a characterizing property, confirming the hypothesis on the amount of electric current that flow along a backbone – see Sec. [4.2.2] Other important indicators advised by the ACEA company experts are the weather conditions, specially the millimeters of rains and the maximum temperature in a day. In fact, it is well-known (and it is also reasonable) that the events of heavy rain are strongly correlated with black-outs in the grid. Accordingly, a strong discrimination is conveyed by the sequences of automatically registered events. In particular, the interruptions registered in the PSs due to the opening of the prevention breakers caused by short circuits on the power line and the Petersen alarms, which are registered as soon as a loss of the dielectric capacity of the power equipments is detected. This phenomena can be physically characterized by overheating (due to the high currents) in which the dielectric of power equipments changes his properties for a short period. The presence of bursts of these events could be indicative of an imminent LF. This last perspective will be studied in detail in future research studies.

The OCC system is trained by cross-validation, exploiting the fitness function in Eq. [16] Here we aim to demonstrate that the solutions found by the OCC become more informative as the system finds better solutions, i.e., with solutions that improve the discrimination of faults/non-faults. This result helps us in justifying the claim that the final solutions found by the proposed OCC are informative (the considered features have different importance in the discrimination process). As previously described, the synthesis of the OCC consists in performing a certain number of iterations until the stop criterion in reached. An important observation is that, during the iterations characterizing the optimization, the fitness of the best-
performing individual has a non-decreasing trend (this is obtained since we use a form of elitism in our GA). Accordingly, we expect to observe a non-decreasing trend also for what concerns the estimated entropy of the distribution underlying the weights $w_k$ related to the best-performing individual of each iteration. To demonstrate such a claim, we calculated the (non-linear) correlation among the sequence of fitness values and the one of the entropy estimations, both related to the best-performing solutions found at each iteration/evolution. Fig. 15 shows those two sequences for a specific test performed with $k = 7$. Although at the beginning the two series are not very correlated, they stabilize to a similar trend that is captured via the estimation of the mutual information $[31]$. The MI column in Tab. 2 reports the average mutual information estimated between those two series for each $k$; MI values fall within the $[0, 1]$ range. As it is possible to observe, the non-linear correlation is in general good, especially in the $k = 7$ case, where also the MI reaches its maximum score.

6 Conclusions

Predicting faults in real-world smart grid systems is a challenging task. This is due to the high variability of those types of systems and also to the heterogeneity of the data that actually characterize a fault situation. In our study, we modeled localized faults in the ACEA smart grid by means of several heterogeneous features. The proposed one-class classifier is based on an interplay among clustering and multiple, ad-hoc, dissimilarity measures, each one specialized to a particular feature type (e.g., categorical, metric, and TS). The classifier synthesis is guided by a genetic algorithm, in charge to optimize the weighting parameters of the dissimilarity measure adopted in the input domain, as well as the decision region boundaries. The proposed system is able to provide both hard (i.e., Boolean) and soft decisions regarding the recognition of a test pattern. Soft decisions are introduced also to offer a measure of reliability concerning the decisions; non reliable decisions can be recognized by calculating the fuzzy entropy of the resulting membership function. Experimental evaluations performed on the ACEA data demonstrate the effectiveness of the proposed solution in a real-world setting.

Future directions include the possibility to evaluate other clustering algorithms and different global optimization schemes. In addition, we will focus our study also on the series of electrical current measured in the ACEA grid. The aim is to characterize the underlying system generating such series with particular interest in finding a correlation/causation rule with respect to the observed faults.
Figure 15: Sequences of fitness values and entropy estimations on the weights corresponding to the best individual solution found at each evolution of the OCC model optimization.

References

[1] CEI - comitato elettrotecnico italiano. URL http://www.ceiweb.it/it/

[2] International energy outlook 2011 - energy information administration. URL http://www.eia.gov/forecasts/ieo/index.cfm

[3] The SmartGrids european technology platform | SmartGrids. URL http://www.smartgrids.eu/ETPSmartGrids

[4] La road map per la continuità del servizio di distribuzione elettrica di roma, acea distribuzione spa, aracne editrice, città ducale (it), 2010.

[5] A. Y. Abdelaziz, F. M. Mohammed, S. F. Mekhamer, and M. A. L. Badr. Distribution systems reconfiguration using a modified particle swarm optimization algorithm. Electric Power Systems Research, 79(11):1521–1530, 2009.

[6] M. A. Abido, E.-S. M. El-Alfy, and M. Sheraz. Computational intelligence in smart grids: Case studies. In Computational Intelligence for Decision Support in Cyber-Physical Systems, pages 265–292. Springer, 2014.

[7] M. Afzal and V. Pothamsetty. Analytics for distributed smart grid sensing. In Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES, pages 1–7, 2012. doi: 10.1109/ISGT.2012.6175733.

[8] S. M. Amin and B. F. Wollenberg. Toward a smart grid: power delivery for the 21st century. IEEE Power and Energy Magazine, 3(5):34–41, 2005.

[9] R. Anderson, A. Boulanger, W. Powell, and W. Scott. Adaptive stochastic control for the smart grid. Proceedings of the IEEE, 99(6):1098–1115, 2011. ISSN 0018-9219. doi: 10.1109/JPROC.2011.2109671.

[10] A. Anvari Moghaddam and A. R. Seifi. Study of forecasting renewable energies in smart grids using linear predictive filters and neural networks. Renewable Power Generation, IET, 5(6):479–480, 2011. ISSN 1752-1416. doi: 10.1049/iet-rpg.2010.0104.

[11] Y. Cai and M.-Y. Chow. Exploratory analysis of massive data for distribution fault diagnosis in smart grids. In Power Energy Society General Meeting, 2009. PES ’09. IEEE, pages 1–6, 2009. doi: 10.1109/PES.2009.5275689.

[12] E. De Santis, A. Rizzi, A. Sadeghian, and F. M. Frattale Mascioli. Genetic optimization of a fuzzy control system for energy flow management in micro-grids. In 2013 Joint IFSA World Congress and NAFIPS Annual Meeting, pages 418–423. IEEE, 2013.
[13] E. De Santis, L. Livi, A. Sadeghian, and A. Rizzi. Fault recognition in smart grids by a one-class classification approach. In Proceedings of the 2014 International Joint Conference on Neural Networks, pages --. IEEE, 2014. pages will be available soon.

[14] G. Del Vescovo, L. Livi, F. M. Frattale Mascioli, and A. Rizzi. On the Problem of Modeling Structured Data with the MinSOD Representative. International Journal of Computer Theory and Engineering, 6(1):9–14, 2014. ISSN 1793-8201. doi: 10.7763/IJCTE.2014.V6.827.

[15] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire. An experimental evaluation of novelty detection methods. Neurocomputing, 135(0):313 – 327, 2014. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2013.12.002.

[16] F. Dörfler, M. Chertkov, and F. Bullo. Synchronization in complex oscillator networks and smart grids. Proceedings of the National Academy of Sciences, 110(6):2005–2010, 2013.

[17] T. Fawcett. An Introduction to ROC Analysis. Pattern Recognition Letters, 27(8):861–874, June 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.10.010.

[18] A. Gambardella, G. Giacinto, M. Migliaccio, and A. Montali. One-class classification for oil spill detection. Pattern Analysis and Applications, 13(3):349–366, 2010. ISSN 1433-7541. doi: 10.1007/s10044-009-0164-z.

[19] S. D. Guikema, R. A. Davidson, and L. Haibin. Statistical models of the effects of tree trimming on power system outages. IEEE Transactions on Power Delivery, 21(3):1549–1557, 2006. ISSN 0885-8977. doi: 10.1109/TPWRD.2005.860238.

[20] A. K. Jain. Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8):651–666, June 2010. ISSN 0167-8655. doi: 10.1016/j.patrec.2009.09.011.

[21] P. Juszczak, D. M. J. Tax, E. Pękalska, and R. P. W. Duin. Minimum spanning tree based one-class classifier. Neurocomputing, 27(7–9):1859 – 1869, 2009. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2008.05.003.

[22] M. Kemmler, E. Rodner, P. Rösch, J. Popp, and J. Denzler. Automatic identification of novel bacteria using Raman spectroscopy and gaussian processes. Analytica Chimica Acta, 794:29 – 37, 2013. ISSN 0003-2670. doi: http://dx.doi.org/10.1016/j.aca.2013.07.051.

[23] M. Kemmler, E. Rodner, E.-S. Wacker, and J. Denzler. One-class classification with gaussian processes. Pattern Recognition, 46(12):3507 – 3518, 2013. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/j.patcog.2013.06.005.

[24] S. S. Khan and M. G. Madden. A survey of recent trends in one class classification. In L. Coyle and J. Freyne, editors, Artificial Intelligence and Cognitive Science, volume 6206 of Lecture Notes in Computer Science, pages 188–197. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-17079-9. doi: 10.1007/978-3-642-17080-5_21.

[25] W. Pedrycz and F. Gomide. An Introduction to Fuzzy Sets: Analysis and Design. Complex Adaptive Systems. MIT Press, 1998. ISBN 9780262161718.
[35] D. Raheja, J. Llinas, R. Nagi, and C. Romanowski. Data fusion/data mining-based architecture for condition-based maintenance. *International Journal of Production Research*, 44(14):2869–2887, July 2006. ISSN 0020-7543. doi: 10.1080/00207540600654509.

[36] A. Rizzi, L. Livi, H. Tahayori, and A. Sadeghian. Matching general type-2 fuzzy sets by comparing the vertical slices. In *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, pages 866–871, 2013. doi: 10.1109/IFSA-NAFIPS.2013.6608514.

[37] S. Saha, M. Alddeen, and C. P. Tan. Fault detection in transmission networks of power systems. *International Journal of Electrical Power & Energy Systems*, 33(4):887 – 900, 2011. ISSN 0142-0615. doi: http://dx.doi.org/10.1016/j.ijepes.2010.12.026.

[38] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Neural Information Processing Systems*, pages 582–588, 2000.

[39] N. Shahid, S. Aleem, I. Naqvi, and N. Zaffar. Support vector machine based fault detection amp; classification in smart grids. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 1526–1531, 2012. doi: 10.1109/GLOCOMW.2012.6477812.

[40] A. P. Shanker and A. Rajagopalan. Off-line signature verification using DTW. *Pattern Recognition Letters*, 28(12):1407 – 1414, 2007. ISSN 0167-8655. doi: 10.1016/j.patrec.2007.02.016.

[41] G. Storti, F. Possemato, M. Paschero, A. Rizzi, and F. Mascioli. Optimal distribution feeders configuration for active power losses minimization by genetic algorithms. In *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, pages 407–412, 2013. doi: 10.1109/IFSA-NAFIPS.2013.6608435.

[42] C. L. Sweetser. The importance of advanced diagnostic methods for higher availability of power transformers and ancillary components in the era of smart grid. In *Power and Energy Society General Meeting, 2011 IEEE*, pages 1–3, 2011. doi: 10.1109/PES.2011.6039785.

[43] D. M. J. Tax and R. P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11–13):1191 – 1199, 1999. ISSN 0167-8655. doi: http://dx.doi.org/10.1016/S0167-8655(99)00087-2.

[44] L. V. Utkin. Fuzzy one-class classification model using contamination neighborhoods. *Advances in Fuzzy Systems*, 2012:22, 2012. doi: 10.1155/2012/984325.

[45] G. K. Venayagamoorthy. Potentials and promises of computational intelligence for smart grids. In *Power Energy Society General Meeting, 2009. PES ’09. IEEE*, pages 1–6, July 2009. doi: 10.1109/PES.2009.5275224.

[46] G. K. Venayagamoorthy. Dynamic, stochastic, computational, and scalable technologies for smart grids. *IEEE Computational Intelligence Magazine*, 6(3):22–35, 2011. ISSN 1556-603X. doi: 10.1109/MCI.2011.941588.

[47] T. Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, XXII, April 1975.

[48] C.-D. Wang and J. Lai. Position regularized support vector domain description. *Pattern Recognition*, 46(3):875 – 884, 2013. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/j.patcog.2012.09.018.

[49] Z. Wang and P. Zhao. Fault location recognition in transmission lines based on support vector machines. In *Computer Science and Information Technology, 2009. ICICSIT 2009. 2nd IEEE International Conference on*, pages 401–404, 2009. doi: 10.1109/ICCSIT.2009.5234528.

[50] P. J. Werbos. Putting more brain-like intelligence into the electric power grid: What we need and how to do it. In *Proceedings of the International Joint Conference on Neural Networks*, pages 3356–3359, 2009. doi: 10.1109/IJCNN.2009.5179088.

[51] J. Wetzer. Maintaining future (electrical) power systems. In *Future Power Systems, 2005 International Conference on*, pages 6 pp.–6, 2005. doi: 10.1109/FPS.2005.204286.

[52] Y. Zhang, Y. Liu, X. Wang, and Z. Wang. Fault pattern recognition in power system engineering. In *Industrial Mechatronics and Automation, 2009. ICIMAC 2009. International Conference on*, pages 109–112, 2009. doi: 10.1109/ICIMAC.2009.5156572.

[53] Y.-G. Zhang, Z.-P. Wang, J.-F. Zhang, and J. Ma. Fault localization in electrical power systems: A pattern recognition approach. *International Journal of Electrical Power & Energy Systems*, 33(3):791 – 798, 2011. ISSN 0142-0615. doi: http://dx.doi.org/10.1016/j.ijepes.2011.01.018.