Classification and Calibration Techniques in Predictive Maintenance: A Comparison between GMM and a Custom One-Class Classifier

Enrico de Santis[a], Antonino Capillo[b], Fabio Massimo Frattale Mascioli[c] and Antonello Rizzi[d]
Department of Information Engineering, Electronics and Telecommunications,
University of Rome "La Sapienza", Rome, Italy

Keywords: Predictive Maintenance, Machine Learning, Gaussian Mixture Models, Faults Modeling, One-Class Classification.

Abstract: Modeling and predicting failures in the field of predictive maintenance is a challenging task. An important issue of an intelligent predictive maintenance system, exploited also for Condition Based Maintenance applications, is the failure probability estimation that can be found uncalibrated for most standard and custom classifiers grounded on Machine learning. In this paper are compared two classification techniques on a data set of faults collected in the real-world power grid that feeds the city of Rome, one based on a hybrid evolutionary-clustering technique, the other based on the well-known Gaussian Mixture Models setting. While the former adopts directly a custom-based weighted dissimilarity measure for facing unstructured and heterogeneous data, the latter needs a specific embedding technique step performed before the training procedure. Results show that both approaches reach good results with a different way of synthesizing a model of faults and with different structural complexities. Furthermore, besides the classification results, it is offered a comparison of the calibration status of the estimated probabilities of both classifiers, which can be a bottleneck for further applications and needs to be measured carefully.

1 INTRODUCTION

Low-cost smart sensors and cloud technologies, boosted with powerful and efficient communication networks, enable new tool-boxes, grounded on AI, to face challenges in predictive maintenance programs, specifically in modern power grids (Smart Grids). In fact, leveraging AI models to identify the abnormal behavior in Medium Voltage (MV) feeders (i.e. faults and outages) turns equipment sensor data into meaningful, actionable insights for proactive asset maintenance, preventing downtime or accidents, meeting present-day time-to-market requirements. The choice of the specific predictive model is not straightforward, especially in real-world applications where they are adopted in production environments. One of the challenges is the synthesis of a low structural complexity model able to act as a gray-box - enabling knowledge discovery tasks - useful even as a building block for more complex programs within business strategic plans, such as estimating the impact of environmental conditions on power grid devices. The measure of the impact together with the probability of failure can drive risk analysis programs on the entire power grid, where technicalities turn into long-term huge investments programs, hence in decisions taken by high-level managers. Thus, from the output point of view, the ML system should provide an interpretable model with not only a Boolean decision over an event but even with a *calibrated* probability (Martino et al., 2019) of occurrence, because the latter quantity plays a decisive role in the downstream decision-making process. As an example, the classifiers proposed in this study will be adopted for measuring the failure rate derived from the probability of fault. This information will be part of a long-term real-world Condition Based Maintenance program. From the input side, a real-world application, such as a data-driven predictive maintenance task in Smart Grids (De Santis et al., 2013), is likely to deal with heavily structured patterns (Zhang et al., 2018) requiring many efforts in feature engineering. Specifically, it consists of building a set of features and a suitable kernel, where stan-

[a] https://orcid.org/0000-0003-4915-0723
[b] https://orcid.org/0000-0002-6360-7737
[c] https://orcid.org/0000-0002-3748-5019
[d] https://orcid.org/0000-0001-8244-0015

de Santis, E., Capillo, A., Mascioli, F. and Rizzi, A.
Classification and Calibration Techniques in Predictive Maintenance: A Comparison between GMM and a Custom One-Class Classifier.
DOI: 10.5220/0010109905030511
In *Proceedings of the 12th International Joint Conference on Computational Intelligence (IJCCI 2020)*, pages 503-511
ISBN: 978-989-758-475-6

dard ML algorithms – designed to be fed by n-tuples of real-valued numbers – can safely operate.

The current study was born within the "Smart Grids intelligence project" (ACEA, 2014; Possemato et al., 2016; Storti et al., 2015), with the aim of equipping the power grid that feeds the entire city of Rome – managed by the ACEA (Azienda Comunale Energia e Ambiente) company – with a poly-functional Decision Support System, able to recognize in real-time power failures estimating the probability of fault depending on environmental conditions and data related to the power grid devices. Specifically, the paper offers a multi-level comparison between two different approaches to classification of faults – the evolutionary based One-Class classification system (OCC_system) (De Santis et al., 2015; De Santis et al., 2018b) along with several improvements and the well known Gaussian Mixture models (GMMs) – starting from a complex representation of the power grid status involving different type of real-world data, such as time data, weather data, power grid structural data, load data and unevenly spaced time-series data related to micro-interruption occurring due to, for example, partial discharges. The two levels of comparison grounds on the specific feature engineering techniques suited to feed the two different classification algorithms (input side) and the quality of the output obtained in terms of classification performances and calibration of probabilities (output side). In fact, the modeling and prediction of faults in power grids is a wide research area (Zhang et al., 2018) where failure causes are debated (Guikema et al., 2006; Cai and Chow, 2009) and modeled within the ML setting (Wang and Zhao, 2009) even in extreme environmental conditions (Liu et al., 2005).

The current study investigates an embedding technique for representing complex structured data (within the family of metric recovery techniques) allowing the adoption of a standard GMM, beside an evolutionary classification technique (the already-cited OCC_System) in charge of learning a suitable metric for a custom based dissimilarity measure, where the predictive model is grounded on a clustering technique, offering the possibility of synthesizing a gray-box interpretable model. Within this setting, as a novelty, the output soft decision – the score values of the classifiers – are evaluated for assessing the usability as calibrated probabilities associated to a power grid status.

The following paper is organized as follows. In Sec. 2 is provided a description of the data set and the problem setting within the field of predictive maintenance and fault recognition with a description on how to measure the calibration of output probabilities. Sec. 3 offers a synthetic survey on classifications techniques, specifically the GMM family and the OCC_System. The experimental setting and the results are discussed in 4, while conclusion are drawn in Sec. 5.

2 THE REAL-WORLD DATA SET

The power grid managed by ACEA consists of a series of MV lines equipped with smart sensors collecting faults data for storing and processing tasks. We refer to a fault as the failure of the electrical insulation (e.g., cables insulation) that compromises the correct functioning of the grid. Therefore, what we call Localized Fault (LF) is actually a fault in which a physical element of the grid is permanently damaged causing long outages. The available real-world data set consists in data patterns describing the power grid states that are classified into standard functioning states (SFSs) and LFs, that is, to each pattern ζ it is associated a label $y(\zeta) : y = \{\texttt{LF}, \texttt{SFS}\}$. These data patterns have been organized together with ACEA field-experts and are structured in several features. Basically a power grid state is composed of two main components or group of features, that is one to constitutive parameters of power grid devices and a second group related to external causes, intended as "forces", with a fast-changing dynamic, that influence the power grid state. The former are, for example, the cable section, the constituent material, etc., while the latter are the weather and the load condition (De Santis et al., 2017b; Bianchi et al., 2015). A detailed description of the selected features can be found in (De Santis et al., 2015). The features belong to different data types: categorical (nominal), quantitative (i.e., data belonging to a normed space) and times series (TSs). The last one describes the sequence of short outages that are automatically registered by the protection systems (known as "Petersen" alarm system) as soon as they occur. Hence, LFs on MV feeders are characterized by heterogeneous data, including weather conditions, spatio-temporal data (i.e. longitude-latitude pairs and time), physical data related to the state of the power grid and its electric equipment (e.g., measured currents and voltages). Thereby, the starting patterns space is structured and non-metric and, as will be explained in detail, a suitable embedding needs to be adopted to deal with ML algorithms designed to work with real-valued tuples. The data set was validated by cleaning it from human errors and by completing in an appropriate way missing data, as explained in (De Santis et al., 2015; De Santis et al., 2017a).

3 THE CLASSIFICATION PROCEDURE

The classification task consists of learning a model $\mathcal{M}$ of a specific oriented process $\mathcal{P}$. This means synthesizing a classifier – a predictive model – where the underlying free parameters are learned feeding a set of $\langle x, y \rangle$ pairs to a training algorithm. In other words, the training process allows learning a decision function f that, given an input x, returns a predicted class label $\hat{y}$, that is $\hat{y} = f(x, \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ is a set of free parameters of the model $\mathcal{M}$. Finally, $\mathcal{M} = \mathcal{M}(\langle x, y \rangle_{i=1}^{n}, \boldsymbol{\Theta})$, that is, an instance of the model, in general, will depend on the training pairs $\langle x, y \rangle_{i=1}^{n}$ and the set of free parameters $\boldsymbol{\Theta} = [\boldsymbol{\theta}, \boldsymbol{\Phi}]$, where $\boldsymbol{\theta}$ are the learning parameters (model parameters) and $\boldsymbol{\Phi}$ is a set of hyperparameters, which define the structural complexity of the model. The latter need a suitable search procedure to be set. If the model is learned over only one class – namely the target class, because the others are not available for some reason – we have a One-class classification problem (Khan and Madden, 2010).

Furthermore, it is possible to distinguish the hard classification task, where the classifier outputs the label $\hat{y}$, and the soft classification one, where it outputs a score value – i.e. a real-valued number s – providing roughly the likeliness that a data pattern belongs to a suitable class. Probabilistic classifiers returns the posterior probability $P(Y|X)$ of an output $\hat{y}$ given an input x. P will depend even on some model parameters $\boldsymbol{\Theta}$, not highlighted in the expression above. In general, the hard decision on a class label can be obtained letting:

$$\hat{y} = \arg\max_{y} P(Y = y|X), \tag{1}$$

that is, for a given input pattern $x \in X$, the decision rule assigns the output label $y \in Y$ to the one corresponding to the maximum posterior probability.

Albeit not all classifiers are probabilistic classifiers, some classifiers such as Support Vector Machine (SVM) (Cortes and Vapnik, 1995) or Naïve Bayes may return a score $s(x)$ which roughly states the "confidence" in the prediction of a given data pattern x. A typical decalibrated classifier produces a model that predicts examples of one or more classes in a proportion which does not fit the original one, i.e., the original class distribution. In the binary case it can be expressed as a mismatch between the expected value of the proportion of classes and the actual one (Bella et al., 2010). Intuitively, calibration means that whenever a forecaster assigns a probability of 0.8 to an event, that event should occur about 80% of the time (Kuleshov et al., 2018). A plain methodology adopted to explore the calibration of a classifier is the Reliability diagram (Murphy and Winkler, 1977) where on the x-axis are reported the scores (or probability for a probabilistic classifier), whereas on the y-axis are reported empirical probabilities $P(y|s(x) = s)$, namely the ratio between the number of patterns in class y with score s and the total number of patterns with score s. If the classifier is well-calibrated, then all points lie on the bisector straight line of the first and third quadrant, meaning that the scores are equal to the empirical probabilities. Due to the real-valued nature of the scores and the fact that it is quite impossible to quantify the number of data points sharing the same score, a binning procedure is adopted. Moreover, in literature can be found a series of calibration techniques, some of which allow estimating a calibration function (adopting a supervised learning framework) making scores similar to the empirical probabilities. More details can be found in (Martino et al., 2019; Kuleshov et al., 2018). Hence, a well-suited set of probabilities related to the classification task of data patterns requires either a well-calibrated classifier or some additional downstream processing. To quantify the goodness of the calibration, i.e. how the probability estimates are far from the empirical probabilities, two methods have been proposed in literature: the Brier score (Brier, 1950; DeGroot and Fienberg, 1983) and the Log-Loss score. Given a series of N known events and the respective probability estimates, the Brier score is the mean squared error between the outcome o (1 if the event has been verified and 0 otherwise) and the probability $p \in [0, 1]$ assigned to such event. In the context of binary classification, the Brier score BS is defined as

$$BS = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left(T(y_i = 1|x_i) - P(y_i = 1|x_i) \right)^2 \tag{2}$$

where $T(\hat{y}_i = 1|x_i) = 1$ if $\hat{y}_i = 1$ and $T(\hat{y}_i = 1|x_i) = 0$ otherwise and $P(\hat{y}_i = 1|x_i)$ is the estimated probability for pattern x_i to belong to class 1. Likewise the MSE, the lower the BS value, the better.

The Log-Loss for binary classification is defined as follows:

$$LL = -\frac{1}{N} \sum\nolimits_{i=1}^{N} \left[y_i \log p_i + (1 - y_i) \log(1 - p_i) \right] \tag{3}$$

and, as per the Brier score, the lower, the better. The Log-Loss index matches the estimated probability with the class label with logarithmic penalty. Hence, for small deviations between $\hat{y}_i$ and p_i the penalty is low, whereas for large deviations the penalty is high.

In standard ML problems the input to the learning algorithm is often a real-valued data pattern of some dimension, while in real-world applications it is likely disposing of structured data pattern, where not all attributes lie in a normed space. For example, some of that can be graphs, time series, categorical variables, etc. At the same time, some classifiers, such as SVM or the herein adopted OCC_System, are grounded on a custom-based kernel, in turn, designed on a custom-based dissimilarity measure able to face each structured pattern through a suitably designed sub-dissimilarity measure. In other words, indicating with $\zeta_i = [o_{i1}, o_{i2}, ..., o_{iR}]$ the i-th structured data pattern composed by R structured objects o, a custom-based dissimilarity measure between ζ_i and ζ_j can be formally expressed as:

$$d(\zeta_i,\zeta_j;\overline{\mathbf{w}}) = f^d(f_r^{sub}(o_{ir},o_{jr});\overline{\mathbf{w}}) \quad r=1,2,...R, \quad (4)$$

where $f_t^{sub}(\cdot)$ is a sub-dissimilarity tailed to the specific data type (underlying the r-th structured attribute o_r), f^d is a compositional relation, with suitable properties, that applies on sub-dissimilarities and can depend on a set of weights parameters $\overline{\mathbf{w}}$. If the latter are subject to learning, the problem of the dissimilarity definition is framed in a metric learning framework (Bellet et al., 2013) and weights can help to interpret models, driving knowledge discovery tasks. As stated, some ML algorithms can face directly with custom-based kernels while others (e.g. GMM), where working with structured data domains, need some embedding procedure, hence a methodology that allows embedding structured data patterns in a well-suited algebraic space, such as the Euclidean space (De Santis et al., 2018a). This procedure can start from the dissimilarity values, computed by means of expression (4), collected in a dissimilarity matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$, where $\mathbf{D}_{ij} = d(\zeta_i,\zeta_j;\overline{\mathbf{w}})$. Among the main embedding techniques, it is worth to cite the possibility of adopting *directly* the dissimilarity matrix as a data matrix (hence, the rows as real-valued data patterns in $\mathbb{R}^n$), eventually reducing the number of dimensions with some heuristics, such as clustering (the technique is known as *dissimilarity representation*) (Pękalska and Duin, 2005). It is noted that, in this case, the data domain is inherently endowed with the standard Euclidean norm. Another way to obtain an embedding is reconstructing the well-behaved Euclidean space, starting from the dissimilarity matrix, being careful to the fact that dissimilarity functions, such as custom-based dissimilarities, could not fulfill all metric or Euclidean properties (e.g. the Dynamic Time Warping for unevenly spaced sequences). In this case, it is required a more general mathematical space for the embedding: the Pseudo Euclidean (PE) space (Pękalska and Duin, 2005). The embedding procedure is similar to the metric space recovery procedure known as Multidimensional Scaling, with the difference that the involved Gram matrix deriving from the kernel matrix is indefinite. In summary, the PE embedding procedure leads to obtaining a data matrix $X = Q_{k_{emb}} \left|\Lambda_{k_{emb}}\right|^{\frac{1}{2}}$, where $|\Lambda|^{\frac{1}{2}}$ is a diagonal matrix of which diagonal elements are the square roots of the absolute value of eigenvalues organized in decreasing order, and $Q_{k_{emb}}$ are the k_{emb} eigenvectors of the kernel (Gram matrix), obtained by a suitable decomposition procedure from $\mathbf{D}$. Specifically, this procedure embeds the dissimilarity matrix in the so-called Associated Euclidean Space (AES) (Duin et al., 2013). More details can be found in (De Santis et al., 2018c). In this work the data matrix X obtained from the PE embedding is adopted for training the GMM, while for the OCC_System the training procedure is grounded on a custom-based kernel (see (4)), as will be explained in details in Sec. 3.2.

3.1 Gaussian Mixture Models

GMM is a well-known technique both in the unsupervised and supervised learning setting. The rationale behind mixture models is that data are generated by a linear combination of a certain number of Gaussian models, i.e. components, described by a set of suitable unknown parameters. In other words, given this set of Gaussian models, the generation process involves i) primarily picking up one of the models and ii) successively generating a data pattern according to its parameters. Hence, giving a sampling of the underlying process generating data, which particular component generates data is unknown and it is considered a *latent variable* to be estimated together with the model statistical parameters.

Given a set of training instances $X = \{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x} \in \mathbb{R}^d$, the GMM statistical distribution can be written as:

$$f(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma},w) = \sum_{i=1}^{k} w_i \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i), \quad (5)$$

where $\mathbf{x}$ is a data pattern, k is the number of the Gaussian components, w_i is the weight of each of the k components, such that $\sum_{i=1}^{k} w_i = 1$ and $w_i \geq 0 \forall i$. In Eq. (5) $\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$ is the normal (multivariate) distribution, with $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ as the mean vector and the covariance matrix, respectively. The training procedure of a GMM consists in the maximum likelihood estimation of model parameters – through the minimization of a maximum likelihood function L – adopting a heuristic known as Expectation-Maximization algorithm (EM) (Dempster et al., 1977).

The number of Gaussian components is an hyper-parameter, likewise k in the k-means. Among a number of criteria for estimating the hyper-parameter, such as the Principal Component Analysis (PCA), two statistical criteria are commonly adopted: i) the minimization of the $AIC = 2 \cdot k - 2L$, where AIC states for *Akaike Information Criterion* (Akaike, 1974), k is the number of components and L is the likelihood function of the model; ii) the minimization of $BIC = \log(n) \cdot k - 2 \cdot \log(L)$, where BIC states for *Bayesian Information Criterion* (Schwarz et al., 1978) and n is the number of observations (i.e. training data patterns).

Within the supervised learning setting, it is possible to learn a GMM model – described by the set of parameter θ – for each class y and to compute its output for any new instance $\boldsymbol{x}^{test}$. The class assignment is based on the maximum likelihood, choosing the target class y^* according to:

$$y^* = \arg\max_{y} L(\boldsymbol{x}^{test}, \theta_y), \tag{6}$$

hence, y^* is the label assigned to the new instance $\boldsymbol{x}^{test}$.

It is worth to note that for the purpose of numerical stability, because the GMM model involves the computation of the inverse of the covariance, i.e. $\mathbf{\Sigma}^{-1}$, a small regularization factor λ can be added on diagonal elements, such as $\mathbf{\Sigma}_{reg} = \mathbf{\Sigma} + \lambda \boldsymbol{I}$.

3.2 The OCC_system

The OCC_system instantiates a (One-Class) classification problem, on a data set $\mathbf{X}$, defined as a triple of disjoint sets, namely training set ($\mathcal{S}_{tr}$), validation set ($\mathcal{S}_{vs}$), and test set ($\mathcal{S}_{ts}$). Given a specific parameters setting (of which a description is provided below), a classification model is built on $\mathcal{S}_{tr}$ and it is validated on $\mathcal{S}_{vs}$. The generalization capability of the optimized model is computed on $\mathcal{S}_{ts}$,

The main idea in order to build a model of structured data patterns, such as LF patterns in the ACEA power grid, is to use a clustering-evolutionary hybrid technique. The main assumption is that similar states of the power grid have similar chances of generating a LF, reflecting the cluster model. Hence, the core of the recognition system is a custom-based dissimilarity measure, within the family of ones described formally by (4), computed as a weighted Euclidean distance, i.e. $d(\zeta_i, \zeta_j; \overline{\mathbf{W}}) = \left[(\zeta_i \ominus \zeta_j)^T \overline{W}^T \check{\overline{W}} (\zeta_i \ominus \zeta_j)\right]^{1/2}$, where ζ_i, ζ_j are specifically two LF patterns and $\overline{W}$ is a diagonal matrix (it could be even a full matrix with some properties, such as the "symmetry") whose elements are generated through a suitable vector of weights $\overline{w}$ (in the case of a diagonal matrix). The dissimilarity measure is component-wise, therefore the $\ominus$ symbol represents a generic dissimilarity measure, tailored on each pattern subspace, that has to be specified depending on the semantic of data at hand.

For quantitative data it's worth to make the difference between integer values describing temporal information and real-valued data related to other information, such as the physical power grid status or the weather conditions. As concerns the former, the dissimilarity measure is the circular difference of the temporal information, because faults that occur on the last day of the year must be temporally near to the faults that occur close to the first day of the next year; real-valued data correctly normalized, instead, can be treated with the standard arithmetic difference. Categorical data in our LF data set are of nominal type, thus they do not have an intrinsic topological structure and therefore the well-known simple matching measure is adopted. The dissimilarity measure among the unevenly spaced Time Series data is performed by means of the Dynamic Time Warping (DTW) algorithm. The DTW is a well-known algorithm born in the speech recognition field that, using the dynamic programming paradigm, is in charge of finding an optimal alignment between two sequences of objects of variable lengths (Müller, 2007). It is well-known that DTW does not respect the triangle inequality property for a metric space manifesting, consequently, a non metric behavior (Duin et al., 2013).

The rationale behind the OCC_System is obtaining a partition $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{k_{occ}}\}$ such that $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ if $i \neq j$ and $\cup_{i=1}^{k_{occ}} \mathcal{C}_i = \mathbf{X}^{target}$. This hard partition is obtained through the k-means.

The decision region of each cluster $\mathcal{C}_i$ of diameter $B(\mathcal{C}_i) = \delta(\mathcal{C}_i) + \varepsilon$ is constructed around the medoid c_i, bounded by the average radius $\delta(\mathcal{C}_i)$ plus a threshold ε, considered together with the dissimilarity weights $\overline{\boldsymbol{w}} = diag(\overline{\boldsymbol{W}})$ as free parameters. Given a test pattern ζ_j^{test} the decision rule consists in evaluating if it falls inside or outside the overall faults decision region, by checking if it falls inside the closest cluster. The learning procedure consists in clustering the training set composed by LF (target) patterns, adopting a standard Genetic Algorithm (GA), in charge of evolving a family of cluster-based classifiers considering the weights $\overline{\boldsymbol{w}}$ and the thresholds of the decision regions as search space, guided by a proper objective function. The last one is evaluated on a validation set composed by LFs and normal functioning states, taking into account a linear combination of the accuracy of the classification, that we seek to maximize, and the extension of the thresholds, that should be minimized. Moreover, in order to outperform the well-known limitations of the initialization of the standard k-means algorithm, the OCC_System initializes more than one instance of the clustering algorithm with random starting representatives. At test stage (or during validation) a voting procedure for each cluster model is performed. This technique allows building a more robust model of the power grid faults. More details can be found in (De Santis et al., 2015).

In the current version of the classification algorithm – as a further improvement compared to last versions – the soft decision value is computed by a Gaussian membership function, that is:

$$s(\zeta|\mathcal{C}_i;\hat{\sigma}) = \mu_{\mathcal{C}_i}(\zeta) = e^{\frac{d(\zeta,c_i)}{2\hat{\sigma}_i^2}}, \tag{7}$$

where $d(\zeta,c_i)$ is the medoid-data pattern distance, $\hat{\sigma}$ is a parameter defining the standard deviation of the Gaussian curve related to the cluster $\mathcal{C}_i$ geometry. The parameter $\hat{\sigma}_i$ for the i-th cluster is obtained as: $\hat{\sigma}_i = \frac{B(\mathcal{C}_i)}{\sqrt{2\log(2)}}$, where $B(\mathcal{C}_i)$ is the diameter of the decision region of cluster $\mathcal{C}_i$. The above expression is based on the fact that the rationale behind the soft decision is to assign $s = 0.5$ for a data pattern lying on the decision region boundary and that the relation between the width of the Gaussian curve at half height is $2\sqrt{2\log(2)} \cdot \overline{\sigma} = 2B(\mathcal{C}_i)$, with $\overline{\sigma}$ the standard deviation.

Hence, given a generic test pattern and given a learned model, it is possible to associate a score value (soft-decision) s that can be interpreted as *uncalibrated* probability, performing a non-linear mapping between power grid states and uncalibrated probability values.

4 EXPERIMENT SETTINGS AND RESULTS

The total number of power grid states within the ACEA data set considered for the following experiments is 2561, divided into 1162 LFs (target) and 1489 SFSs (non-target). In the following section, a comparison between two different approaches to the classification of faults will be offered, namely the OCC_System and GMM, reviewed in Sec. 3.2 and Sec. 3.1, respectively.

As concerns the OCC_System, the adopted training algorithm, in charge of computing a suitable partition, is the well-known k-means with random initialization of representatives. In this study, the k_{occ} parameter of the k-means algorithm is a meta-parameter fixed in advance, which is an index of the model structural complexity. Thereby, simulations are conducted through a linear search on k_{occ} and the model with the highest performance in terms of accuracy on the validation set is chosen. Performances are provided for various model structural complexity values. The adopted GA used to tune the classification model performs stochastic uniform selection, Gaussian mutation and scattered crossover (with crossover fraction of 0.80). It implements a form of elitism that imports the two fittest individuals in the next generation; the population size is kept constant throughout the generations and equal to 50 individuals. The stop criterion is defined by considering a maximum number of iterations (250) and checking the variations of the best individual fitness.

The GMM approach is declined in two variants. Grounding on what is reported in Sec. 3.1, the first approach relies on the adoption of a validation set for establishing the best number of Gaussian components through a linear search in a predefined interval of integers $k \in [1,20]$, selecting the model with the best accuracy. The second approach consists in an unsupervised search of the best model for each class using the BIC criterion searching for $k \in [1,20]$. The main difference is that in the case of model selection with the validation set the number of components per class is the same, while in the unsupervised approach each class has its own number of components. For the sake of investigating the behaviour of both approaches along with the embedding techniques discussed in Sec. 3, performance results are collected for each integer $k_{emb} \in [1,50]$. Moreover, a comparison with the case where components share or not share the covariance matrix $\mathbf{\Sigma}$ is provided. In the last case, the regularization parameter is set to $\lambda = 0.01$ for numerical stability.

For robustness purposes, during the training phase, in both approaches 10 random initializations of the EM algorithm are adopted and, in the first approach (model tuned with validation set), the average accuracy results are collected.

The reported evaluation metrics of the classifier are the accuracy (A), the true positive rate (TPR), the false positive rate (FPR), the specificity, the precision, the F-measure (F_Score), the area under the Receiving Operating Characteristic curve (AUC) elaborated from the confusion matrix and the informedness (IFM). As concerns the calibration of output probabilities (score values) the Brier score and the Log-Loss are provided (see. Sec. 3).

In Tab. 1 are reported the performances of the various experiments. It is noted that in the table Σ_{not-sh} represents the case where the covariance matrix is not shared, while Σ_{sh} means thai it is. Furthermore, GMM $\mathcal{S}_{vs}$ means that the model is trained through the adoption of a validation set, while GMM (BIC) means that it is trained with the BIC as objective function. For both the GMM and the OCC_System, average performances (mean and standard deviation in brackets) are computed on five runs. For the GMM also the

Table 1: Performance evaluation for several experiments (averaged on five runs) conducted with various versions of the GMM and the OCC_System. In brackets are reported the standard deviations.

Class. type	k	k_{emb}	Accuracy	TPR	FPR	Specificity	Precision	F_Score	AUC	IFM	Brier	Log loss
OCC_Syst. ($k_{occ} = 15$)	/	/	0.9718 (0.0097)	0.9552 (0.0149)	0.0153 (0.0151)	0.9847 (0.0151)	0.9803 (0.0190)	0.9674 (0.0028)	0.9884 (0.0050)	0.9700 (0.0095)	0.0410 (0.0131)	0.2059 (0.0427)
OCC_Syst. ($k_{occ} = 30$)	/	/	**0.9838** (0.0058)	0.9747 (0.0155)	0.0090 (0.0063)	0.9910 (0.0063)	0.9885 (0.0080)	0.9814 (0.0068)	0.9947 (0.0102)	0.9829 (0.0067)	0.0238 (0.0059)	0.1779 (0.2037)
OCC_Syst. ($k_{occ} = 100$)	/	/	0.9824 (0.0033)	0.0094 (0.0043)	0.9772 (0.0035)	0.9905 (0.0043)	0.9938 (0.0028)	0.9855 (0.0028)	0.9958 (0.0014)	0.9839 (0.0034)	0.0302 (0.0047)	0.1892 (0.0453)
GMM $\mathcal{S}_{vs}$ Σ_{sh}	5.8 (1.3)	42.8 (11.4)	**0.9453** (0.01984)	0.9758 (0.02165)	0.0852 (0.06016)	0.9147 (0.06016)	0.9038 (0.06207)	0.9369 (0.0238)	0.6370 (0.0865)	0.9453 (0.0198)	0.3130 (0.0510)	19.4013 (0)
GMM $\mathcal{S}_{vs}$ Σ_{sh}-best	7	23	**0.9668**	0.9425	0.0090	0.9910	0.9880	0.9647	0.6375	0.9668	0.3249	19.4013
GMM $\mathcal{S}_{vs}$ Σ_{not-sh}	15.2 (1.3)	9.4 (6.9)	0.7401 (0.0134)	0.8103 (0.0578)	0.3300 (0.0696)	0.6700 (0.0696)	0.6601 (0.0345)	0.7255 (0.0125)	0.5258 (0.0766)	0.7401 (0.0134)	0.4477 (0.0114)	19.4013 (0)
GMM $\mathcal{S}_{vs}$ Σ_{not-sh}-best	16	5	0.7603	0.8391	0.3184	0.6816	0.6728	0.7468	0.4281	0.7603	0.4536	19.4013
GMM (BIC) Σ_{sh}	19.4, 20 (0.8), (0)	34.8 (91.2)	0.8645 (0.0029)	0.8379 (0.0084)	0.1148 (0.0050)	0.8852 (0.0050)	0.8554 (0.0063)	0.8436 (0.0044)	0.5166 (0.0506)	0.8616 (0.0031)	0.4551 (0.0012)	19.4013 (0)
GMM (BIC) Σ_{sh}-best	20, 20	40	0.9093	0.8793	0.0673	0.9327	0.9107	0.8947	0.7750	0.9060	0.3980	19.4013
GMM (BIC) Σ_{not-sh})	10.2, 7.2 (2.7), (0.7)	4.6 (0.8)	0.6851 (0.0022)	0.4908 (0.0224)	0.1632 (0.0052)	0.8368 (0.0052)	0.7085 (0.0032)	0.5669 (0.0120)	0.5354 (0.0116)	0.6638 (0.0031)	0.4002 (0.0001)	19.4013 (0)
GMM (BIC) Σ_{not-sh})-best	8, 8	5	0.7632	0.6897	0.1794	0.8206	0.7500	0.7186	0.4359	0.7551	0.4062	19.4013

best ones, within the five runs, are provided. As concerns the OCC_System $k_{occ} = \{15, 30, 100\}$ are experimented. In terms of accuracy and informedness, best performances (accuracy=98%) are reached by OCC_System for a structural complexity of the model obtained with $k_{occ} = 30$ clusters. Experiments with $k_{occ} = 100$ clusters do not show remarkable improvements. As concerns GMM models, the configuration that reached comparable average performances (accuracy=94%) is the one with shared covariance, where the hyper-parameters are obtained adopting a validation set $\mathcal{S}_{vs}$, with a low average number of components $k = 5.8$ and an embedding dimension of $k_{emb} = 42.8$. The worst experimented results, with an average accuracy of 68%, are achieved with a GMM model with no shared covariance matrix, where the model complexity (i.e. the number of components k) is optimized through the BIC criterion. In summary, in terms of classification performances, results are slightly different in favor of the OCC_System, even if the GMM does its best with a lower model complexity, measured as the number of components k. In Fig. 1 is reported the average accuracy and informedness as function of the embedding dimension k_{emb}, measured on the test set $\mathcal{S}_{ts}$. Performances varies widely with k_{emb}. As expected, a great variability is found even in single experiments due to the well-known strong dependence to initial conditions of the EM algorithm. As concerns the calibration status of the output classifiers, the Brier score and Log-Loss – see Tab. 1 – show that both techniques needs calibration. For example, in Fig. 2 is depicted the Reliability diagram for the output probabilities obtained through the OCC_System. It confirms that the Reliability curve is very far from the bisector line, indicating that output score values are uncalibrated. The same is confirmed by the calibration performances reported in Tab. 1. Specifically, the OCC_System is found more reliable than the GMM in terms of Brier score.

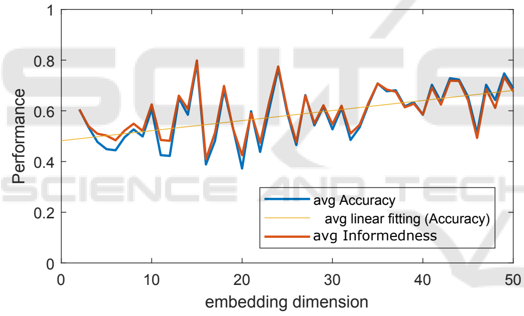

Figure 1: Average accuracy and informedness for the GMM (shared covariance) as function of the embedding dimension k_{emb}, measured on the test set $\mathcal{S}_{ts}$.

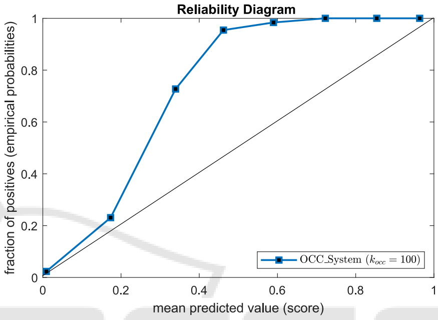

Figure 2: Reliability diagram for the score values s obtained from the OCC_System on the test set $\mathcal{S}_{ts}$.

5 CONCLUSIONS

A comparison between some versions of the GMM classification algorithm, able to operate in real valued vector domains, and the OCC_System, that works with a custom-based weighted dissimilarity measure, shows that remarkable performances can be obtained choosing the right embedding for the first one. It is well known that the perfect ML model does not exist because each one possesses its own peculiar characteristics. In our case the EM algorithm is fast and the best GMM model obtained on the current ACEA data set has a low computational complexity in terms of the number of components, working with a shared covariance matrix. As concerns the OCC_System, it reaches very good results in terms of accuracy, yielding also classification models characterized by a low number of clusters, even if the evolutionary procedure slows down the training process. It is the price for obtaining a robust model where the weights of the custom based dissimilarity measures can be also interpreted as the importance of each feature in the classification task. This interesting feature, together with clusters content analysis, allows knowledge discovery applications. Moreover, some applications require calibrated probabilities as output scores and both the compared techniques show a weak calibration degree. Future works will be grounded on the study and on the application of several classical and newly proposed calibration techniques for OCC_System output scores, as requested by the objectives of the main project.

ACKNOWLEDGEMENTS

The authors wish to thank ACEA Distribuzione S.p.A. for providing the data and for their continuous support during the design and test phases. Special thanks to Ing. Stefano Liotta, Chief Network Operation Division, to Ing. Silvio Alessandroni, Chief Electric Power Distribution, and to Ing. Maurizio Paschero, Chief Remote Control Division.

REFERENCES

ACEA (2014). The acea smart grid pilot project (in italian).

Akaike, H. (1974). A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer.

Bella, A., Ferri, C., Hernández-Orallo, J., and Ramírez-Quintana, M. J. (2010). Calibration of machine learning models. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 128–146. IGI Global.

Bellet, A., Habrard, A., and Sebban, M. (2013). A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709.

Bianchi, F., De Santis, E., Rizzi, A., and Sadeghian, A. (2015). Short-term electric load forecasting using echo state networks and pca decomposition. *Access, IEEE*, 3:1931–1943.

Brier, G. W. (1950). Verification of forecast expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3.

Cai, Y. and Chow, M.-Y. (2009). Exploratory analysis of massive data for distribution fault diagnosis in smart grids. In *2009 IEEE Power & Energy Society General Meeting*, pages 1–6. IEEE.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

De Santis, E., Livi, L., Sadeghian, A., and Rizzi, A. (2015). Modeling and recognition of smart grid faults by a combined approach of dissimilarity learning and one-class classification. *Neurocomputing*, 170:368 – 383.

De Santis, E., Martino, A., Rizzi, A., and Mascioli, F. M. F. (2018a). Dissimilarity space representations and automatic feature selection for protein function prediction. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

De Santis, E., Paschero, M., Rizzi, A., and Mascioli, F. M. F. (2018b). Evolutionary optimization of an affine model for vulnerability characterization in smart grids. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

De Santis, E., Rizzi, A., and Sadeghian, A. (2017a). A learning intelligent system for classification and characterization of localized faults in smart grids. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 2669–2676.

De Santis, E., Rizzi, A., and Sadeghian, A. (2018c). A cluster-based dissimilarity learning approach for localized fault classification in smart grids. *Swarm and evolutionary computation*, 39:267–278.

De Santis, E., Rizzi, A., Sadeghian, A., and Mascioli, F. (2013). Genetic optimization of a fuzzy control system for energy flow management in micro-grids. In *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, pages 418–423.

De Santis, E., Sadeghian, A., and Rizzi, A. (2017b). A smoothing technique for the multifractal analysis of a medium voltage feeders electric current. *International Journal of Bifurcation and Chaos*, 27(14):1750211.

DeGroot, M. H. and Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 32(1/2):12–22.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Duin, R. P., Pękalska, E., and Loog, M. (2013). Non-euclidean dissimilarities: causes, embedding and informativeness. In *Similarity-Based Pattern Analysis and Recognition*, pages 13–44. Springer.

Guikema, S. D., Davidson, R. A., and Liu, H. (2006). Statistical models of the effects of tree trimming on power system outages. *IEEE Transactions on Power Delivery*, 21(3):1549–1557.

Khan, S. S. and Madden, M. G. (2010). A survey of recent trends in one class classification. In Coyle, L. and

Freyne, J., editors, *Artificial Intelligence and Cognitive Science*, volume 6206 of *Lecture Notes in Computer Science*, pages 188–197. Springer Berlin Heidelberg.

Kuleshov, V., Fenner, N., and Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. *arXiv preprint arXiv:1807.00263*.

Liu, H., Davidson, R. A., Rosowsky, D. V., and Stedinger, J. R. (2005). Negative binomial regression of electric power outages in hurricanes. *Journal of infrastructure systems*, 11(4):258–267.

Martino, A., De Santis, E., Baldini, L., and Rizzi, A. (2019). Calibration techniques for binary classification problems: A comparative analysis. In *Proceedings of the 11th International Joint Conference on Computational Intelligence*, volume 1 of *IJCCI2019*.

Müller, M. (2007). Dynamic time warping. *Information retrieval for music and motion*, pages 69–84.

Murphy, A. H. and Winkler, R. L. (1977). Reliability of subjective probability forecasts of precipitation and temperature. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 26(1):41–47.

Pękalska, E. and Duin, R. (2005). *The dissimilarity representation for pattern recognition: foundations and applications*. Series in machine perception and artificial intelligence. World Scientific.

Possemato, F., Paschero, M., Livi, L., Rizzi, A., and Sadeghian, A. (2016). On the impact of topological properties of smart grids in power losses optimization problems. *International Journal of Electrical Power & Energy Systems*, 78:755–764.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Storti, G. L., Paschero, M., Rizzi, A., and Mascioli, F. M. F. (2015). Comparison between time-constrained and time-unconstrained optimization for power losses minimization in smart grids using genetic algorithms. *Neurocomputing*, 170:353–367.

Wang, Z. and Zhao, P. (2009). Fault location recognition in transmission lines based on support vector machines. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pages 401–404. IEEE.

Zhang, Y., Huang, T., and Bompard, E. F. (2018). Big data analytics in smart grids: a review. *Energy informatics*, 1(1):8.