Kernel and Random Extreme Learning Machine applied to Submersible Motor Pump Fault Diagnosis

Thomas Walter Rauber*, Thiago Oliveira-Santos*,
Francisco de Assis Boldt*, Alexandre Rodrigues†, Flávio M. Varejão*
Universidade Federal do Espírito Santo, 29075-910, Vitória, Brazil
* Departamento de Informática - {thomas,todsantos,fboldt,fvarejao}@inf.ufes.br
† Departamento de Estatística - alexandre.rodrigues@ufes.br

Marcos Pellegrini Ribeiro
Petrobras, CENPES/PDP/TE
Av. Horácio Macedo 950 - Ilha do Fundão
21941-598, Rio de Janeiro, Brazil
mpellegrini@petrobras.com.br

***Abstract*—This paper presents an extension of a comparative study of classifier architectures for automatic fault diagnosis, with a special emphasis on the Extreme Learning Machine (ELM), with and without kernel mapping. Besides the explanation of the ELM model, an attempt is made to find theoretical hints of the excellent generalization capabilities of this model, based on the findings of Cover about dichotomies and the equivalence of Mean Squared Error minimization in the high-dimensional feature spaces induced by kernels, and spaces defined by a finite sample set. The field of application is a practical problem in the context of offshore petroleum exploration where sophisticated submersible motor pumps are extensively tested before being deployed. The work juxtaposes the performance of ELM to an existing statistically sound comparison of state of the art classifier methods for a hand-crafted feature model tailored specially to the spectra of the vibrational signals of the pump. The results suggest the remarkably good generalization capability of ELM, exhibiting the highest scores for the chosen F-measure performance criterion.**

***Index Terms*—Extreme Learning Machine; Fault Diagnosis; Submersible Motor Pump; Classification; Performance Criteria; Feature Models**

I. Introduction

In this work, the ELM is investigated in the context of fault diagnosis of submersible motor pumps. Its performance is compared to the state-of-the art and consolidated classifier architectures, considering the scenario of the artificial intelligence system (presented in [1]) to automatically diagnose faults of submersible motor pumps. To validate the proposed method, a comparative study was performed using real data acquired by measuring vibrational patterns of submersible motor pumps with accelerometer sensors. The dataset is composed of thousands of accelerometer sensor data examples that were labeled by a human expert into one of the five following categories: normal operation, faulty sensor, faulty pump with rubbing, misalignment or unbalance.

The rest of the paper is organized in the following manner. Section II elaborates the theoretical aspects of the ELM for both the conventional and kernel based architecture. Besides the conventional ELM graphical scheme, typically seen in works related to ELM, the architecture of the kernel-based ELM is also introduced schematically with the intent to explain its calculus. In section III the application domain of motor pump fault diagnosis is explained. Subsequently in section IV, the experimental evaluation procedure is described. To the best of our acknowledge, this is the first work that uses a large range of hyperparameters to be selected by grid-search for ELM and Kernel-ELM, with a bias aware and statistically sound performance evaluation. The experimental results of section V show that ELM outperforms all previously investigated classifiers. It also presents a chart that shows the evolution of the performance criterion for the ELM with and without regularization. Conclusions are drawn in section VI.

II. Extreme Learning Machine

The core of the ELM [2], [3] is a feature extraction process that maps a d-dimensional pattern $\mathbf{x}$ from an input domain $\mathcal{X}$ which is usually the Euclidean vector space $\mathbb{R}^d$ to the hidden-layer feature space (ELM feature space) $\mathcal{H}$. The final mapping from the hidden space to the output space $\mathcal{Y}$ is just a linear combination of the patterns of the hidden space which can easily be learned deterministically by virtue of a generalized inverse matrix of the mapped patterns of the hidden layer. However, the original ELM can be explored with other more elaborated feature mapping such as kernel-based mapping [4]. When no kernel is used, the ELM feature space $\mathcal{H}$ is an explicit nonlinear combination of the original feature space $\mathcal{X}$ with finite dimension L, where L is the number of hidden neurons. When kernel mapping is applied, the ELM feature space $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) of possibly infinite dimension, implicitly defined by $L = N$ kernel functions of an arbitrary pattern with each of the N training patterns.

A. Random Hidden Feature Map ELM

The basic ELM of fig. 1 relies on ideas of the Perceptron [5], [6] since both calculate random new features from the original pattern and then combine them linearly. The hidden layer of the ELM is composed of L neurons that represent the extracted features $h_i, i = 1, \ldots, L$. In order to calculate the i-th extracted feature h_i, a d-dimensional pattern $\mathbf{x} = [x_1 \cdots x_\ell \cdots x_d\ ]^\mathsf{T}$ is linearly combined with d random weights $\mathbf{w}_i = [w_{i,1} \cdots w_{i,\ell} \cdots w_{i,d}\ ]^\mathsf{T}$, followed by the addition of a random bias $b_i, i = 1, \ldots, L$, and finally the passage through a nonlinear activation function f, so

$$h_i(\mathbf{x}; \mathbf{w}_i, b_i) = f\left(\sum_{\ell=1}^{d} w_{i,\ell} x_j + b_i\right) = f\left(\mathbf{w}_i \cdot \mathbf{x} + b_i\right). \quad (1)$$

Consequently, there are $(d+1) \times L$ random weights w and biases b that are once initialized and then frozen. Mostly, the activation function $f : \mathbb{R} \to \mathbb{R}$ is the logistic sigmoid function $f(n) = 1/(1+\exp(-n))$. It must be nonlinear to allow the ELM to globally calculate a nonlinear approximation of the desired regression or classification problem. Alternative activation functions are the radial basis transfer function $f(n) = \exp(-n^2)$, the sine function, hard-limit, or triangular basis [7]. The choice of the activation function is an additional hyperparameter for this architecture. All output neurons form the L-dimensional hidden-layer feature vector $\mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) \cdots h_i(\mathbf{x}) \cdots h_L(\mathbf{x}) \end{bmatrix}^\mathsf{T}$.

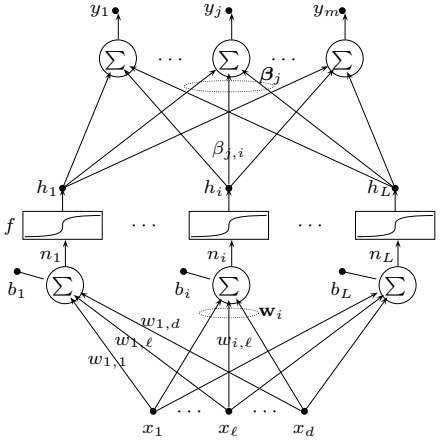

Fig. 1: Architecture of the basic Extreme Learning Machine. The new features h_i at the hidden layer are random linear combinations of the original features x_ℓ, followed by a nonlinear activation function f.

The final hidden-to-output mapping is purely linear. The j-th output component y_j is a linear combination of all hidden neurons as

$$y_j(\mathbf{h}(\mathbf{x}); \boldsymbol{\beta}_j) = \sum_{i=1}^{L} \beta_{j,i} h_i(\mathbf{x}) = \boldsymbol{\beta}_j \cdot \mathbf{h}(\mathbf{x}), \tag{2}$$

where the weight vector of the j-th output y_j is defined as $\boldsymbol{\beta}_j = \begin{bmatrix} \beta_{j,1} \cdots \beta_{j,i} \cdots \beta_{j,L} \end{bmatrix}^\mathsf{T}$. All $L \cdot m$ hidden-to-output weights are stored in the $L \times m$ matrix

$$\mathbf{B} = \begin{bmatrix} \boldsymbol{\beta}_1 & \cdots & \boldsymbol{\beta}_m \end{bmatrix} = \begin{bmatrix} \beta_{1,1} & \cdots & \beta_{m,1} \\ \vdots & \ddots & \vdots \\ \beta_{1,L} & \cdots & \beta_{m,L} \end{bmatrix}. \tag{3}$$

The m components y_j define the final m-dimensional output vector of the ELM functional mapping

$$\mathbf{y}(\mathbf{x}; \mathbf{B}) = \begin{bmatrix} y_1(\mathbf{h}(\mathbf{x}); \mathbf{B}) \cdots y_m(\mathbf{h}(\mathbf{x}); \mathbf{B}) \end{bmatrix}^\mathsf{T} = \mathbf{h}(\mathbf{x})^\mathsf{T} \mathbf{B}. \tag{4}$$

This is a generalized linear function which calculates a random variable $\mathbf{y}$ as output from another random variable $\mathbf{x}$ as input. In the context of a supervised learning problem, the error between the desired $\mathbf{y}$ and predicted output $\mathbf{y}(\mathbf{x}; \mathbf{B})$ is defined as the squared norm of the difference vector as

$$E(\mathbf{B}) := ||\mathbf{y} - \mathbf{y}(\mathbf{x}; \mathbf{B})||^2. \tag{5}$$

This is another random variable which depends on parameter $\mathbf{B}$. Its expected value $\mathbb{E}[E(\mathbf{B})]$ can be estimated as the Mean Squared Error (MSE) over all N samples as

$$\widehat{\mathbb{E}[E(\mathbf{B})]} = \frac{1}{N} \sum_{k=1}^{N} ||\mathbf{y}^{(k)} - \mathbf{y}(\mathbf{x}^{(k)}; \mathbf{B})||^2 \tag{6}$$

Since the input-to-hidden map $\mathbf{h}(\mathbf{x})$ is a one time calculus of all training patterns, all N original patterns $\mathbf{x}^{(k)}, k = 1, \ldots, N$ are mapped to N mapped hidden feature vectors $\mathbf{h}(\mathbf{x}^{(k)}), k = 1, \ldots, N$ that are stored transposed as the lines of the $N \times L$ matrix $\mathbf{H}$

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}^{(1)}) \\ \vdots \\ \mathbf{h}(\mathbf{x}^{(N)}) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}^{(1)}) & \cdots & h_L(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}^{(N)}) & \cdots & h_L(\mathbf{x}^{(N)}) \end{bmatrix}. \tag{7}$$

The final map of the intermediate data set $\mathbf{H}$ is linear and hence an ELM as a classifier belongs to the category of Generalized Linear Discriminant Functions [8]. The total set of N output patterns $\mathbf{y}^{(k)}, k = 1, \ldots, N$ is a $N \times m$ matrix $\hat{\mathbf{Y}} := \mathbf{Y}(\mathbf{H}; \mathbf{B})$ where the k-th line is the transposed output pattern $\mathbf{y}(\mathbf{x}^{(k)}; \mathbf{B})^\mathsf{T}$. The global predicted mapping of the whole training set can compactly be formulated as

$$\hat{\mathbf{Y}} = \mathbf{H}\mathbf{B}. \tag{8}$$

Assuming that the training set does not imply an underdetermined linear system (i.e. linear independent columns in $\mathbf{H}$ and more patterns N than hidden units L), the weights $\mathbf{B}$ can deterministically be obtained by means of the $L \times N$ Moore-Penrose pseudoinverse

$$\mathbf{H}^\dagger = (\mathbf{H}^\mathsf{T}\mathbf{H})^{-1}\mathbf{H}^\mathsf{T} \tag{9}$$

as

$$\mathbf{B} = \mathbf{H}^\dagger \mathbf{Y}, \tag{10}$$

where $\mathbf{Y}$ is the $N \times m$ matrix of the desired output, with one line as the transposed desired output of the k-th pattern. This is the solution to the MSE minimization of (5), hence

$$\mathbf{B} = \arg\min_{\tilde{\mathbf{B}}} ||\mathbf{y} - \mathbf{y}(\mathbf{x}; \tilde{\mathbf{B}})||^2 \tag{11}$$

and known to be the best linear approximation to the theoretical bound of the Bayes error rate. Other methods to obtain the hidden-to-output weights are not considered here, e.g. a memory saving incremental technique [9] or gradient descent based optimization.

B. Kernel ELM

The success of the Support Vector Machine [10] for regression and classification has drawn considerable attention to the use of kernels which perform a inner product of two patterns that have been implicitly mapped to the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. The original d-dimensional pattern $\mathbf{x}$ is mapped to a new feature vector $\boldsymbol{\phi}(\mathbf{x})$ in RKHS of potentially infinite dimension by a map $\boldsymbol{\phi}$

$$\begin{aligned} \boldsymbol{\phi} : \mathcal{X} &\to \mathcal{H} \\ \mathbf{x} &\mapsto \boldsymbol{\phi}(\mathbf{x}). \end{aligned} \tag{12}$$

This map is not defined explicitly in general, only the calculus of the inner product[1] of two mapped patterns is possible. Let $\mathbf{x}$ and $\mathbf{y}$ be two patterns from $\mathcal{X}$ and $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\phi}(\mathbf{y})$ their mapped counterparts from $\mathcal{H}$. The kernel function k is equivalent to the inner product in the mapped space defined by the map

$$\begin{aligned} k : \langle\cdot,\cdot\rangle : \mathcal{X}\times\mathcal{X} &\to \mathcal{H}\times\mathcal{H}\to\mathbb{R} \\ (\mathbf{x},\mathbf{y}) &\mapsto (\boldsymbol{\phi}(\mathbf{x}),\boldsymbol{\phi}(\mathbf{y})) \mapsto \langle\boldsymbol{\phi}(\mathbf{x}),\boldsymbol{\phi}(\mathbf{y})\rangle = k(\mathbf{x},\mathbf{y}). \end{aligned} \quad (13)$$

Hence the kernel function

$$k(\mathbf{x}^{(p)},\mathbf{x}^{(q)}) = \boldsymbol{\phi}(\mathbf{x}^{(p)})\cdot\boldsymbol{\phi}(\mathbf{x}^{(q)}) \quad (14)$$

of two arbitrary patterns $\mathbf{x}^{(p)}$ and $\mathbf{x}^{(q)}$ is equivalent to their inner product in the implicitly defined Reproducing Kernel Hilbert feature space $\mathcal{H}$. The Radial Basis Function kernel $k(\mathbf{x}_i,\mathbf{x}_j) = \exp(-||\mathbf{x}_i-\mathbf{x}_j||^2/(2\sigma^2))$ with spread parameter σ^2, the inhomogeneous polynomial kernel $k(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i\cdot\mathbf{x}_j+1)^p$ which maps to all possible monomials up to degree p and the linear kernel $k(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i\cdot\mathbf{x}_j$ are the most widely employed kernels [11].

The architecture of the kernel based ELM is illustrated in fig. 2. It possesses $L=N$ hidden nodes. Each node h_i calculates the kernel function $k(\mathbf{x},\mathbf{x}^{(i)})$ of some input pattern $\mathbf{x}$ with an existing pattern $\mathbf{x}^{(i)}$ from the training set. This is equivalent to the inner product $\boldsymbol{\phi}(\mathbf{x})\cdot\boldsymbol{\phi}(\mathbf{x}^{(i)})$ of the two patterns after they were mapped to the RKHS as $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\phi}(\mathbf{x}^{(i)})$ respectively. All N nodes together form a N-dimensional vector, the so called empirical kernel map [12], [13] as

$$\mathbf{h}^{\text{ker}}(\mathbf{x}) := [\ k(\mathbf{x},\mathbf{x}^{(1)}) \quad \cdots \quad k(\mathbf{x},\mathbf{x}^{(N)})\ ]^{\mathsf{T}}. \quad (15)$$

This map constitutes the hidden layer value of any pattern $\mathbf{x}$. Consequently the number of hidden layer features is fixed at $L=N$ with $h_i^{\text{ker}}(\mathbf{x}) = k(\mathbf{x},\mathbf{x}^{(i)}), i=1,\ldots,L=N$. Note that the calculus of the kernel based ELM is deterministic, the mapped feature vector of the hidden layer is not arbitrarily obtained, as it was in the case of the conventional ELM.

Analogously to the one time calculus of the mapping of the whole training set to the hidden layer, the matrix $\mathbf{H}$ of (7) becomes

$$\mathbf{K} = \begin{bmatrix} \mathbf{h}^{\text{ker}}(\mathbf{x}^{(1)}) \\ \vdots \\ \mathbf{h}^{\text{ker}}(\mathbf{x}^{(N)}) \end{bmatrix} = \begin{bmatrix} k(\mathbf{x}^{(1)},\mathbf{x}^{(1)}) & \cdots & k(\mathbf{x}^{(N)},\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}^{(1)},\mathbf{x}^{(N)}) & \cdots & k(\mathbf{x}^{(N)},\mathbf{x}^{(N)}) \end{bmatrix}, \quad (16)$$

which can be recognized as the $N\times N$ Gram matrix, aka kernel matrix [12] of the kernel k with respect to the training patterns $\{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)}\}$.

The deterministic calculus of the weight matrix $\mathbf{B}$ defined in (3) is additionally stabilized by a regularization parameter C, an idea similar to ridge regression [14]. Before the inversion of the matrix $\mathbf{K}^{\mathsf{T}}\mathbf{K}$ to obtain the weights $\mathbf{B}$ of (10), a constant $1/C$ is added to the diagonal elements to avoid an eventual singularity in the case that the rank of $\mathbf{K}^{\mathsf{T}}\mathbf{K}$ is less than N. Hence the modified generalized inverse $\mathbf{K}_{\text{C}}^{\dagger}$ with regularization parameter C, in analogy to $\mathbf{H}^{\dagger}$ of (9), can be defined as

$$\mathbf{K}_{\text{C}}^{\dagger} := (\mathbf{K}^{\mathsf{T}}\mathbf{K} + \frac{\mathbf{I}}{C})^{-1}\mathbf{K}^{\mathsf{T}}. \quad (17)$$

The regularization can also be applied to the Random Hidden Feature Map ELM of section II-A, substituting the term $(\mathbf{H}^{\mathsf{T}}\mathbf{H})$ on the right hand side of (9) by $(\mathbf{H}^{\mathsf{T}}\mathbf{H}+\frac{\mathbf{I}}{C})$. The kernel based hidden-to-output ELM weights in form of a $N\times m$ matrix can identically to (10) be obtained as

$$\mathbf{B}^{\text{ker}} = \mathbf{K}_{\text{C}}^{\dagger}\mathbf{Y}. \quad (18)$$

The calculus of the output of an arbitrary input pattern $\mathbf{x}$ is identical to that formulated for the conventional ELM (8) using the hidden-to-output weights (18) as

$$\mathbf{y}(\mathbf{x})^{\mathsf{T}} = \mathbf{h}(\mathbf{x})^{\mathsf{T}}\mathbf{B}^{\text{ker}}. \quad (19)$$

C. Generalized Linear Separation Capabilities of ELM

In practice, ELM has shown supreme performance in regression and classification problems, e.g. in handwritten character recognition [15]. The seminal work of Cover [16] analyzes the separation capabilities of linear classifiers. The number of dichotomies, i.e. arbitrary attribution of N d-dimensional patterns in a binary classification problem is 2^N. An important result is that the probability of the dimension d^* at which a random pattern set first becomes linearly separable has a binomial distribution

$$\Pr[d^*=d] = \left(\frac{1}{2}\right)^{N-1}\binom{N-1}{d-1}, \quad d=1,2,\ldots,N \quad (20)$$

and expected value

$$\mathbb{E}[d^*] = \frac{N+1}{2}. \quad (21)$$

This implies that for a fixed number N of training patterns, when there is the possibility to increase the dimension d to about half the number of patterns, the patterns become separable. Obviously for certain classification problem, the patterns are *not* randomly distributed. Nevertheless, the hidden layer of the ELM with a high dimension L suggests that the mapped patterns $\mathbf{h}(\mathbf{x})$ are much better separable than the original patterns $\mathbf{x}$ with a low dimension d. Moreover, the randomization of the hidden layer might favor the underlying conditions of the theory of Cover, since the new patterns are randomly relocated in the mapped hidden space.

Another interesting study was presented by Ruiz and López-de-Teruel [13] which relates the kernel mapping to the mapping realized by the Gram matrix (16) with a finite number of training patterns. They establish the equivalence that the mapping to the possibly ∞-dimensional Reproducing Kernel Hilbert Space by the map (12) is also done by the N-dimensional empirical kernel map (15). The kernel based ELM is explained as the kernel version of the well known

[1]The symbols $\langle\mathbf{x},\mathbf{y}\rangle$ and $\mathbf{x}\cdot\mathbf{y}$ for the inner product are equivalent.

Least Mean Square Error optimization problem already mentioned in (11), to find the optimal weights that minimize the cost

$$\arg\min_{\mathbf{B}_{\mathcal{H}}} ||\mathbf{Y} - \mathbf{\Phi}\mathbf{B}_{\mathcal{H}}||^2, \tag{22}$$

where $\mathbf{\Phi}$ is the $N \times L_{\mathcal{H}}$ matrix of all N patterns $\left\{\boldsymbol{\phi}(\mathbf{x}^{(1)}), \ldots, \boldsymbol{\phi}(\mathbf{x}^{(N)})\right\}$ that were mapped to the $L_{\mathcal{H}}$-dimensional RKHS. $\mathbf{B}_{\mathcal{H}}$ is the corresponding hidden-to-output weight matrix of dimension $L_{\mathcal{H}} \times m$. Formally, eq. (45) of [13], in the nomenclature of this work, states that for a general mapping of one single pattern the following equivalence holds

$$\mathbf{y}(\mathbf{x})^{\mathsf{T}} = \boldsymbol{\phi}(\mathbf{x})^{\mathsf{T}}\mathbf{B}_{\mathcal{H}} = \mathbf{h}(\mathbf{x})^{\mathsf{T}}\mathbf{B}^{\text{ker}}. \tag{23}$$

The left hand side is the transposed m-dimensional output of the kernel ELM. In the expression in the middle, $\boldsymbol{\phi}(\mathbf{x})$ is the RKHS map of pattern $\mathbf{x}$ which has dimension $L_{\mathcal{H}} \leq \infty$. For instance when the Radial Basic Kernel [17] is used, $L_{\mathcal{H}} = \infty$. Note that $\boldsymbol{\phi}(\mathbf{x})$ and $\mathbf{B}_{\mathcal{H}}$ are only defined analytically. They cannot be represented in computer memory, since their RKHS dimension in general is not explicitly available. For instance the RBF kernel would require an array of infinite size to store $\boldsymbol{\phi}(\mathbf{x})$ or one column of $\mathbf{B}_{\mathcal{H}}$. The so called kernel trick [12] circumvents the explicit representation of patterns in RKHS, because it uses only inner products of the same, which can be calculated as kernel functions of two patterns. The right hand side in (23) is the output of the kernel based ELM defined in (19). Hence the MSE minimization problem of (22) is equivalently solved by the MSE minimization with the finite set of N patterns

$$\arg\min_{\mathbf{B}^{\text{ker}}} ||\mathbf{Y} - \mathbf{K}\mathbf{B}^{\text{ker}}||^2, \tag{24}$$

Conclusively, (23) proves that linear regression of the kernel based ELM with a finite number of hidden nodes in fact does realize implicitly a linear regression in the ultra-high dimensional RKHS, since both minimize the same MSE problem.

III. Fault Diagnosis of Submersible Motor Pumps

Submersible motor pumps are used to support off-shore exploration of crude petroleum and gas. The equipment must operate reliably because suspending production might lead to multi million dollar losses. In addition to losses due to the lack of production, removal and replacement of defective systems under operation is extremely expensive due to the location of operation (i.e. under deep water), therefore it must be avoided. In addition, real time supervision under deep water is not feasible. Once deployed, only a few parameters can be monitored (e.g. current consumption). In order to avoid such interventions, the equipment must be carefully examined in special testing environment [18], [19] before the actual acquisition.

Defects usually change the vibrational pattern of systems that are operating properly. Accelerometer sensors are used to capture the vibrational pattern of the pump systems under various operational conditions within a experimental environment. These signals are the raw input of the fault diagnosis system. Pumps are tested under various operational conditions with sensors equally distributed along their main axis. The complete string from which the data in this work was acquired, is a compound of six parts, upper pump, upper protector, upper motor, lower pump, lower protector and lower motor, c.f. fig. 3. Sensors are placed in pairs and orthogonally to each other in order to cover the whole plane of vibration (i.e. a plane orthogonal to the main axis of the equipment). One pair of sensors times three different positions (top, middle and bottom) per part of equipment, results in a total of 36 vibrational signals. The sensor signals are collected in the time domain and transformed to the frequency domain (through conventional Fourier transform) to be analyzed. Typically, a human expert analyses this vibrational patterns to diagnose the operational condition of the equipment. In [20], the authors proposed a system to automatically diagnose faults in the aforementioned equipment. In order to perform the diagnosis, the system firstly converts the vibrational spectrum into a more meaningful set of features and finally performs a classification.

Three categories of motor pump faults are considered:

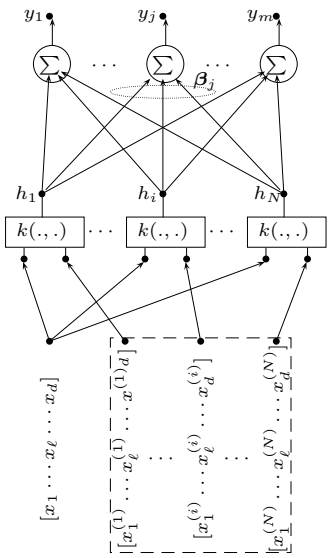

Fig. 2: Architecture of the kernel-based Extreme Learning Machine. The new features h_i at the hidden layer are the inner products of a certain pattern $\mathbf{x}$ with each of the training patterns $\mathbf{x}^{(i)}$. The inner product is calculated in the Reproducing Kernel Hilbert Space by virtue of a kernel function $k(\mathbf{x}, \mathbf{x}^{(i)})$. The dashed box represents all N samples of the training set.

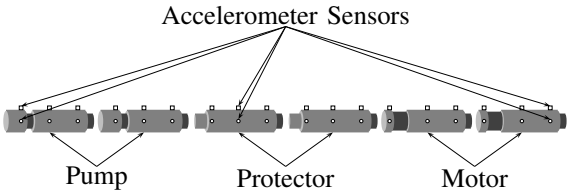

Fig. 3: Submersible pump layout.

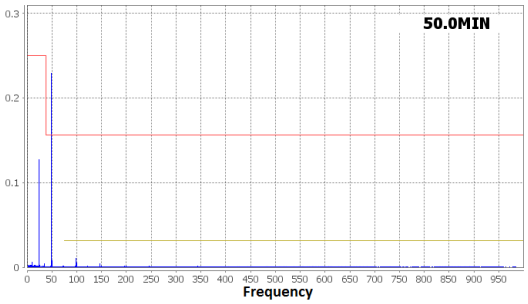

Fig. 4: A representative spectrum of a typical unbalance fault. The high peak in the first harmonic (50 Hz) is a strong indicator that at every multiple of the rotation frequency, the accelerometer measures a high amplitude.

operating with shaft misalignment, operating with pump blade unbalance, operating with mechanical rubbing. If the sensor is faulty, it is not possible to identify the motor pump fault. In order to deal with this fact, the accelerometer fault was also considered as a category. This fault categories together with the normal operation condition, gives a total of five mutually exclusive categories: misalignment, unbalance, rubbing, sensor, normal. Signals with evidence of two or more categories are not considered due to the lack of examples. An example of a faulty spectrum can be seen in fig. 4, in this case an unbalance. For a more detailed description of the spectra and their relation with the considered fault categories, c.f. [1].

IV. Experimental Evaluation

This section describes the dataset, the features and experimental framework used in [1] to measure the performance of the classifiers. The current study completes the previous work with the F-measure performance scores of the ELM with and without kernel and the K-Nearest-Neighbor classifier with kernel distances.

A. Data Acquisition

The dataset consists of 4570 labeled vibration spectra from nine motor pumps with distinct string configurations (18 pumps, protectors and motors) that were tested prior to underwater deployment. Five operation conditions were defined, c.f. table I for their a priori distribution. Please refer to [1] for more details on the data acquisition.

TABLE I: A-priori percentages of normal and fault classes.

Condition class	A priori class distribution [%]
Normal operating condition	81.10
Misalignment	1.09
Unbalance	10.61
Rubbing	0.77
Accelerometer fault	6.43

B. Feature Model

The feature set is composed of eight hand-crafted (described in table II) statistical parameters that focus on specific characteristics of the Fourier spectrum. The feature set is designed with the intent to replicate the human expert knowledge used to identify normal and fault classes. To obtain the spectrum of frequencies, the raw time domain signal of $37.27\ s$ is sampled at 4166.7 Hz, resulting in 155302 sampled values from which the magnitudes of the Fourier transform are calculated.

TABLE II: Statistical feature model of the vibration signal frequency spectrum

X_{rf}	Rotation frequency ν_1 (first harmonic) of the submersible motor pump during the test
X_{rfm}	Magnitude $F(\nu_1)$ in the rotation frequency (first harmonic)
X_{rfm2}	Magnitude $F(\nu_2)$ in the double of the rotation frequency (second harmonic ν_2)
X_{rfrms}	Root mean square of the magnitudes $F(\nu)$ around the rotation frequency, $\nu \in [X_{\mathrm{rf}} - 1, X_{\mathrm{rf}} + 1]$
X_{rfmm}	Median of the magnitudes $F(\nu)$ around the rotation frequency, $\nu \in [X_{\mathrm{rf}} - 1, X_{\mathrm{rf}} + 1]$
X_{m3to5}	Median of the magnitudes $F(\nu)$ of the low frequencies, interval $\nu \in [3\mathrm{Hz}, 5\mathrm{Hz}]$
X_{ilr}	Intercept (a) of the linear regression of logarithm of the frequency magnitudes $F(\nu)$ over the interval of frequencies $\nu \in [5\mathrm{Hz}, 19\mathrm{Hz}]$, c.f. (25)
X_{slr}	Slope (b) of the linear regression of logarithm of the frequency magnitudes $F(\nu)$ over the interval of frequencies $\nu \in [5\mathrm{Hz}, 19\mathrm{Hz}]$, c.f. (25)

The exponential regression model of the two features X_{ilr} and X_{slr} is defined as

$$\log \mathrm{F}(\nu) = a - b \cdot \nu. \tag{25}$$

The first five features $\{X_{\mathrm{rf}}, X_{\mathrm{rfm}}, X_{\mathrm{rfm2}}, X_{\mathrm{rfrms}}, X_{\mathrm{rfmm}}\}$ focus on peaks around the first two harmonics, the last three features $\{X_{\mathrm{m3to5}}, X_{\mathrm{ilr}}, X_{\mathrm{slr}}\}$ are the resulting parameters of an exponential regression analysis of the decay behaviour of low frequency magnitudes. Univariate standardization is applied to all eight features, i.e. centralizing to zero mean and scaling to unit variance.

C. Classifier Architectures

In addition to the ELM explained before, other classifiers were also used as base for comparison. This section briefly describes each of them.

1) K-Nearest Neighbor Classifier with and without Kernel Distance: The K-Nearest-Neighbor (KNN) classifier [21] in its most basic form should always be included, since it has no hyperparameters besides the number of neighbors and gives a qualitative idea of the theoretical performance limits for a given data set. Since the RKHS mapping via kernels is applied for ELM and the Support Vector Machine, it will also be applied to the KNN, extending the study of [1]. The idea of the kernel version [22], [23] of the conventional Nearest Neighbor Classifier [8] is to substitute the Euclidean distance in the original d-dimensional feature space by the Euclidean distance in the high-dimensional RKHS. Let again $\mathbf{x}$ and $\mathbf{y}$ be two patterns from $\mathcal{X}$ and $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\phi}(\mathbf{y})$ their mapped versions from the RKHS $\mathcal{H}$. The squared Euclidean distance between the original patterns $\mathbf{x}$ and $\mathbf{y}$ can be expanded as

$$||\mathbf{x} - \mathbf{y}||^2 = \mathbf{x} \cdot \mathbf{x} - 2\mathbf{x} \cdot \mathbf{y} + \mathbf{y} \cdot \mathbf{y}. \tag{26}$$

The squared Euclidean distance between the mapped patterns can be expanded analogously and expressed purely in terms of kernel function terms, considering (14), as

$$\begin{aligned} ||\boldsymbol{\phi}(\mathbf{x}) - \boldsymbol{\phi}(\mathbf{y})||^2 = \\ \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}) - 2\boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{y}) + \boldsymbol{\phi}(\mathbf{y}) \cdot \boldsymbol{\phi}(\mathbf{y}) = \\ k(\mathbf{x}, \mathbf{x}) - 2k(\mathbf{x}, \mathbf{y}) + k(\mathbf{y}, \mathbf{y}). \end{aligned} \tag{27}$$

For instance in the case of the linear kernel, (27) simplifies to (26), and for the RBF kernel, the Euclidean distance instantiates to $\left[2 - 2\exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))\right]^{\frac{1}{2}}$ where the argument of the square root should be lower bounded to zero due to eventual numerical instabilities. The hyperparameters of the KNN with kernel are the number of neighbors, the type of kernel, and the kernel parameters associated to each kernel.

2) Support Vector Machine, Decision Tree and Random Forests: Support Vector Machine (SVM), Decision Tree (DT) and Random Forests (RF) are important classifiers in the field of machine learning. SVM has already been used for fault diagnosis as a field of application [24], [25], [26]. DT [27], [28], [14] and RF [29], [30], [31] have also shown great performance in classification tasks. Therefore, they are used as base for comparison.

3) Hyperparameters: The performance evaluation of this work has a tuning stage which tries to obtain the best hyperparameters of each classifier architecture, c.f. [1] for details. The hyperparameter values used for the grid search during the tuning phase are compiled in table III. The chosen values are initial guesses around promising points of the grid search. As each classifier model has its own peculiarities, the respective number of hyperparameters is usually different. It is worth to highlight that all experiments in this work are bias aware. Thus, for each fold, the grid search algorithm chooses the hyperparameters using only the training set. Consequently, there is no guarantee that the set of hyperparameters are the same for the folds. The performed experiments in general have shown different sets of hyperparameters for each fold for all classifiers.

TABLE III: Hyperparameters used for each classifier architecture.

Method	Hyperparameters	Range
ELM	# hidden neurons	$\{1600, 1760, \ldots, 9600\}$
	Regularization parameter	$C \in \{\text{none}, 1.0\}$
	Activation function	logistic sigmoid, radial basis
Kernel ELM	RBF Kernel σ	$\{0.25, 0.5, 1, 2, 4, 8\}$
	Polynomial Kernel order	$\{2, 3, 4, 5\}$
	Regularization parameter	$C \in \{\text{none}, 1.0\}$
KNN	Number of neighbors	$\{1, 3, 5\}$
Kernel KNN	Number of neighbors	$\{1, 3, 5\}$
	RBF Kernel σ	$\{8, 2, 0.5, 0.125, 0.03125, 0.0078125\}$
	Polynomial Kernel order	$\{2, 3, 4, 5\}$
SVM	RBF Kernel C	$\{32, 128, 8192, 32768\}$
	γ	$\{2, 8\}$
DT	Algorithm	{Exact, PCA}
	Max number of splits	$\{10, 50, 100\}$
	Max number of categories	$\{10, 50, 100\}$
	Pruning	{yes,no}
RF	Number of trees	$\{100, 1000\}$
	Number of features	$\{1, 2, 3, 4, 5\}$

D. Performance Evaluation

The performance evaluation presented here reaches beyond the common cross validation methods observed in literature for the problem of fault diagnosis which usually divides the total data set into training and test sets. Here a $R \times K$ performance score matrix of F-measure values are produced for each of the classifier models, where $R = 10$ is the number of rounds (experiments) and $K = 10$ the parameter of the K-fold cross validation. This gives a total of $R \cdot K = 100$ performance scores. Theses scores are submitted to a statistical analysis which extracts location and dispersion descriptive measures in order to present the results. Moreover, a corrected Bonferroni-Holm t-test mutually compares two classifiers to judge if they are significantly different. The framework for obtaining the performance matrix is described in more detail in [1]. In each of the R rounds K stratified folds are generated from scratch. As the performance criterion, the F-measure with weighting parameter $\beta = 1$ is used. Under this assumption the F-measure is defined as

$$F = \frac{(1 + \beta^2)\text{tp}}{(1 + \beta^2)\text{tp} + \beta^2\text{fn} + \text{fp}}. \tag{28}$$

With $\beta = 1$, the F-measure is the harmonic mean of *precision* tp/(tp+fp) and *recall* tp/(tp+fn), where the true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn) are obtained from the confusion matrix of a binary classification problem. To account for the multi-class situation, the final F-measure score is the average of the F-measure obtained from the one-against-one binary classification task for the normal and fault classes.

E. Statistical Analysis

This stage attempts to answer the question if the performance scores of two different classifiers are significantly different. A pairwise comparison of all $R \cdot K$ differences of F-measures is done by the correlated t-test [32]. This technique introduces a correction of an otherwise biased statistical parameter since it considers the implicit overlap of training and test sets inevitably occurring by the proposed cross validation. The t-test statistic is defined as

$$t = \frac{\mu}{\sigma}\left[\frac{1}{R \times K} + \frac{n^{\text{TE}}}{n^{\text{TR}}}\right]^{-1/2} \tag{29}$$

where μ and σ are the mean and standard deviation of the $R{\cdot}K$ differences between two classifier F-measure performance scores. The size of the training and test sets are n^{TR} and n^{TE} respectively. The null hypothesis is that two classifiers are similar. Given a significance level α, the null hypothesis is rejected, if the probability of a standard Student's t-distribution with $R \times K - 1$ degrees of freedom (T) being greater than

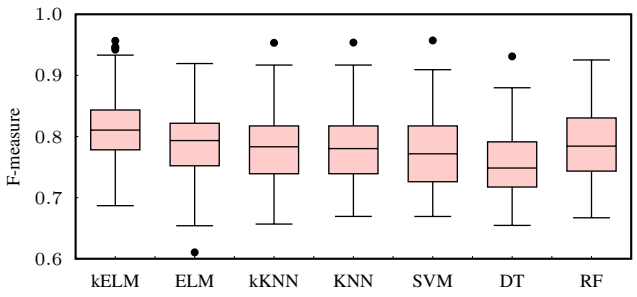

Fig. 5: Boxplot for F-Measure results of each method. Kernel ELM (kELM), Random Hidden Feature Map ELM (ELM), Kernel Nearest-Neighbor Classifier (kKNN), Nearest-Neighbor Classifier with Euclidean distance (KNN), Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF).

the observed t of (29) is less than α, in other words, if $P(T > |t|) < \alpha$. The probability $P(T > |t|)$ is the p-value of the correlated t-test. Holm's step-down procedure [33] is used to control the family-wise error, i.e. the probability of making at least one Type I error in any of the comparisons. All $q = m(m-1)/2$ p-values of the pairwise comparisons of all m different classifier methods are increasingly ordered, so $p_1 < p_2 < \ldots < p_q$. Then Holm's procedure rejects the null hypotheses of tests with p-values $p_1, \ldots, p_{i^*}$, where i^* is determined as the largest i such that $p_i < \alpha/(q+1-i)$.

V. Experimental Results and Discussion

Experiments were performed with the total amount of $n = 4570$ examples for $R = 10$ rounds, in each round using $K = 10$ fold cross validation procedure, hence $n(K-1)/K = 4113$ training samples and $n/K = 457$ test samples in each fold.

A. Basic performance score statistics

Fig. 5 illustrates the results of the F-measures for each method in a boxplot, table IV shows the corresponding numerical values.

TABLE IV: Parameters of the boxplot for each classification method. Q_1 is the lower Quartile, Q_2 the median, Q_3 the upper Quartile. k='Kernel'.

Classifier	min	max	Q_1	Q_2	Q_3	mean
kELM	**0.6869**	0.9567	**0.7783**	**0.8106**	**0.8443**	**0.8112**
ELM	0.6105	0.9195	0.7539	0.7935	0.8252	0.7908
kKNN	0.6568	0.9534	0.7433	0.7832	0.8216	0.7839
KNN	0.6693	0.9539	0.7410	0.7803	0.8198	0.7810
SVM	0.6692	**0.9573**	0.7297	0.7718	0.8176	0.7778
DT	0.6546	0.9312	0.7207	0.7485	0.7932	0.7576
RF	0.6671	0.9254	0.7444	0.7843	0.8332	0.7920

B. Difference Significance Test

With respect to hypotheses testing, it can be stated that no significant difference between two classifiers could be detected for $\alpha = 0.05$. This does not necessarily mean that all classifiers have equal performance. To achieve a $\alpha = 0.05$ significance level, a very large gap between the performance scores of two classifiers must exist. For instance in [1] only a leap from about 0.6 with a KNN classifier, using the non-standardized feature values, to about 0.8 with a KNN classifier, using the standardized feature values, caused the rejection of the null hypothesis that the two methods are significantly the same. The results presented here suggest that the kernel ELM classifier is reasonably superior than the remaining non-ELM approaches with an increase of average performance of 3.5%, 3.9%, 4.3%, 7.1% and 2.4% over the kKNN, KNN, SVM, Decision Tree and Random Forest, respectively.

TABLE V: Pairwise comparisons of classifier architectures. Upper triangle shows the p-values of the correlated t-test of each pair. Lower triangle shows the values of the t statistic of (29).

	Classifier p values						
Classifier t values	kELM	ELM	kKNN	KNN	SVM	DT	RF
kELM	—	0.4929	0.3577	0.3230	0.2614	0.0808	0.5350
ELM	0.6882	—	0.7641	0.6546	0.6468	0.2822	0.9688
kKNN	0.9240	0.3009	—	0.7311	0.8325	0.3572	0.7838
KNN	0.9932	0.4488	0.3446	—	0.9065	0.4169	0.7055
SVM	1.1295	0.4597	0.2120	0.1178	—	0.3908	0.5560
DT	1.7642	1.0812	0.9250	0.8153	0.8619	—	0.1222
RF	0.6226	-0.039	-0.275	-0.379	-0.591	-1.559	—

C. Learning Curve

Revealing observations about the learning behaviour of the random hidden feature map ELM, with and without regularization are illustrated in fig. 6. The x-axis shows the number of hidden nodes L. The y-axis is the F-measure performance score estimated as the mean of $R = 3$ rounds with $K = 10$ folds, i.e. 30 values (Less rounds were used here to speed up the performance estimation). The horizontal red dotted line is the theoretical limit of one, the vertical red dotted line shows the situation $L = N$ when the number N of training patterns matches the number L of hidden nodes. In [7], a theoretical analysis about the upper bound $L_{\max}$ for the number of hidden nodes proves that with at most N hidden neurons and with any bounded nonlinear activation function which has a limit at one infinity, ELM can learn these N distinct samples with zero error. This can be confirmed in fig. 6 for the ELM without regularization, since the maximum performance of 1.0 for the training patterns, i.e. resubstitution, is reached at $L = 2880$ (green line). For $N = 4113$ training patterns in each fold, $L_{\max} = 4113$, hence for this particular data set the theoretical results are experimentally corroborated. The price of an error-free resubstitution is overfitting. This can be seen for the very low performance for the 457 test samples (magenta line) at $L_{\max} = 4113$ (vertical red dotted line). The curve recovers again, probably due to the higher amount of randomness with more hidden nodes. For the case with regularization, i.e. modifying (9) like (17), the regularization parameter was set to $C = 1$. The training performance (blue line) enters saturation, but never reaches the theoretical limit. The test performance (orange line) is much better with regularization, oscillating around 0.79 after reaching saturation.

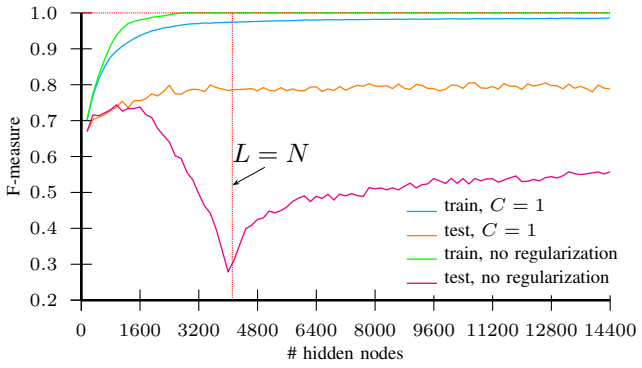

Fig. 6: Evolution of the F-measure performance criterion. Each training and test performance is the mean of the 30 scores of 3-rounds and 10-folds. Activation function is radial basis. Incremental step for the number of hidden nodes is 20.

VI. Conclusion

This work compared the classification performance of the Extreme Learning Machine to alternative classifier methods, considered state-of-the art. As a practical application, fault diagnosis of a complex motor pump system was chosen. The performance criterion was the F-measure obtained in a elaborated evaluation procedure with hyperparameter tuning. Posterior statistical analysis of the performance scores were done. The results experimentally confirm the excellent learning capabilities of the ELM.

Acknowledgments

This work was supported by CENPES-Petrobras, Grant *Termo de Cooperação 0050.00070332.11.9 Petrobras-UFES.*

References

[1] T. Oliveira-Santos, T. W. Rauber, F. M. Varejão, L. Martinuzzo, W. Oliveira, and M. P. Ribeiro, "Submersible motor pump fault diagnosis system: A comparative study of classification methods," in *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*. IEEE, 2016, pp. 1–8.

[2] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 2. IEEE, 2004, pp. 985–990.

[3] ——, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[4] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.

[5] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[6] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, Madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, Aug. 2002. [Online]. Available: http://dx.doi.org/10.1109/5.58323

[7] G.-B. Huang and H. A. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions," *IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 224–229, 1998.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: John Wiley and Sons, 2001.

[9] J. Tapson and A. van Schaik, "Learning the pseudoinverse solution to network weights," *Neural Networks*, vol. 45, pp. 94–100, 2013.

[10] V. Vapnik, *The Nature of Statistical Learning Theory*. N.Y.: Springer, 1995.

[11] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Annals of Statistics*, vol. 36, pp. 1171–1220, 2008.

[12] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT Press, 2002.

[13] A. Ruiz and P. López-de Teruel, "Nonlinear kernel-based statistical pattern analysis," *IEEE Trans. on Neural Networks*, vol. 12, no. 1, pp. 16–32, 2001.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer Series in Statistics. Springer New York, 2013.

[15] E. Cambria, G.-B. Huang, L. L. C. Kasun, H. Zhou, C. M. Vong, J. Lin, J. Yin, Z. Cai, Q. Liu, K. Li, V. C. Leung, L. Feng, Y.-S. Ong, M.-H. Lim, A. Akusok, A. Lendasse, F. Corona, R. Nian, Y. Miche, P. Gastaldo, R. Zunino, S. Decherchi, X. Yang, K. Mao, B.-S. Oh, J. Jeon, K.-A. Toh, A. B. J. Teoh, J. Kim, H. Yu, Y. Chen, and J. Liu, "Extreme learning machines [trends controversies]," *Intelligent Systems, IEEE*, vol. 28, no. 6, pp. 30–59, Nov 2013.

[16] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE transactions on electronic computers*, no. 3, pp. 326–334, 1965.

[17] S. K. Zhou and R. Chellappa, "From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 6, p. 917, 2006.

[18] H. Toliyat, S. Nandi, S. Choi, and H. Meshgin-Kelk, *Electric Machines: Modeling, Condition Monitoring, and Fault Diagnosis*. CRC Press, 2016.

[19] H. Pasman, *Risk Analysis and Control for Industrial Processes - Gas, Oil and Chemicals: A System Perspective for Assessing and Avoiding Low-Probability, High-Consequence Events*. Elsevier Science, 2015.

[20] T. W. Rauber, F. M. Varejao, F. Fabris, A. Rodrigues, and M. P. Ribeiro, "Automatic diagnosis of submersible motor pump conditions in offshore oil exploration," in *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*. IEEE, 2013, pp. 5537–5542.

[21] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, jan 1967.

[22] M. Aizerman, E. Braverman, and L. Rozonoer, "Extrapolative problems in automatic control and the method of potential functions," *Am. Math. Soc. Transl.*, vol. 87, pp. 281–303, 1970.

[23] K. Yu, L. Ji, and X. Zhang, "Kernel nearest-neighbor algorithm," *Neural Processing Letters*, vol. 15, pp. 147–156, 2002, 10.1023/A:1015244902967.

[24] A. Nourmohammadzadeh and S. Hartmann, "Fault classification of a centrifugal pump in normal and noisy environment with artificial neural network and support vector machine enhanced by a genetic algorithm," in *International Conference on Theory and Practice of Natural Computing*. Springer, 2015, pp. 58–70.

[25] Z. Yin and J. Hou, "Recent advances on svm based fault diagnosis and process monitoring in complicated industrial processes," *Neurocomputing*, vol. 174, pp. 643–650, 2016.

[26] Y. Liu, Z. Yu, M. Zeng, and Y. Zhang, "Lle for submersible plunger pump fault diagnosis via joint wavelet and svd approach," *Neurocomputing*, vol. 185, pp. 202–211, 2016.

[27] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*, ser. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984.

[28] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[29] T. K. Ho, "Random decision forests," in , *Proceedings of the Third International Conference on Document Analysis and Recognition, 1995*, vol. 1, Aug. 1995, pp. 278–282 vol.1.

[30] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[31] R. Genuer, J.-M. Poggi, and C. Tuleau, "Random Forests: some methodological insights," *arXiv:0811.3619*, Nov. 2008.

[32] C. Nadeau and Y. Bengio, "Inference for the generalization error," *Machine Learning*, vol. 52, no. 3, pp. 239–281, 2003.

[33] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics*, pp. 65–70, 1979.