ABSTRACT

Chemical process installations are exposed to aggressive chemicals and conditions leading to corrosion. The damage from corrosion can lead to an unexpected plant shutdown and to the exposure of people and the environment to chemicals. Due to changes within and on the surface of materials subjected to corrosion, energy is released in the form of acoustic waves. This acoustic activity can be captured and used for corrosion monitoring in chemical process installations. Wavelet packet coefficients extracted from the acoustic activity have been considered to determine whether corrosion occurs, and to identify the type of corrosion process, at least for the most important corrosion processes in the chemical process industry. A challenging problem, after the extraction of the wavelet coefficients from the Wavelet Packet Transform (WPT), is to capture as much information as possible using a few wavelet coefficients to predict the corrosion type. Due to the statistical dependencies that potentially exist between the wavelet coefficients, the latter should not be selected independently from each other. Local discriminant basis selection algorithms do not take the statistical dependencies between wavelet coefficients into account. In this paper, we have used several mutual information-based approaches, that take into account these dependencies, to improve the selection of a few informative basis functions, and compared them to the wavelet-specific local discriminant basis selection algorithm. We have compared the following three mutual
information filter approaches: a high-dimensional density-based method, a high-dimensional distance-based method, and a relevance-redundancy approach based on the normalized mutual information. Furthermore, a hybrid filter-wrapper genetic algorithm, which uses the relevance-redundancy approach as a local search procedure, was designed. All mutual information-based approaches require fewer coefficients than the local discriminant basis one and achieve a higher classification performance. The highest classification accuracies are most often obtained with the hybrid filter-wrapper genetic algorithm, for almost all classifiers used in this paper. A naïve Bayes classifier that uses the features selected by the hybrid filter-wrapper genetic algorithm was able to identify the absence of corrosion, uniform corrosion, pitting and stress corrosion cracking, with an accuracy of up to 86.8%.

**KEY WORDS:**

Acoustic emission; corrosion monitoring; feature subset selection; genetic algorithm; mutual information; Wavelet Packet Transform.
1. INTRODUCTION

1.1. The continued need for corrosion monitoring

Each year, corrosion destroys a large part of the world’s economy. The global cost of corrosion, consisting of direct and indirect costs, is estimated at 3.8% of the GWP (gross world product) [1]. This global cost equals $1930 billion in US dollars for the year 2004. For the United States, the total cost of corrosion is estimated at $504 billion ($304 billion direct costs + $200 billion indirect costs) per year in 2004 [1,2], which represents about 4.7% of its GDP (gross domestic product). The direct costs are the costs incurred by the owners or operators. These costs consist of the following [1,2]: the use of more expensive or additional materials to prevent corrosion, the labor and equipment for corrosion management, the loss of revenue due to disruption of the supply of the product, the loss of reliability, and the loss of capital due to corrosive deterioration. The indirect costs are incurred by the user of the products or the society. In the chemical, petrochemical and pharmaceutical sectors [2], which are the targeted sectors here, the direct costs are extrapolated to $1.9 billion for 2004 [1,2]. About 60% of the mechanical failures in the chemical process industry are due to corrosion [3]. A large part, 25 to 40%, of the direct and indirect costs can be saved by the use of corrosion monitoring and control systems [4]. Corrosion detection provides feedback to operators about the state of the plant so that they can participate in managing the high corrosion costs [4]. Direct costs that can be avoided by the use of monitoring systems are due to the increased reliability of the plant, avoidance of the disruption of the supply of products, decreased loss of capital and avoidance of lawsuits against companies (e.g., due to pollution caused by leaks of the installations), among other factors. Indirect costs can be equally important as these costs have an impact on the society and environment. In some sectors, damage due to corrosion can be tolerable, but in the chemical, petrochemical and nuclear
sectors, corrosion damage can be catastrophic, even resulting in the loss of lives and environmental damage.

Regular practice in the chemical process industry consists of periodic inspections of the plant, e.g., every 3 months, every 6 months or every year [5]. A recurring problem with such periodic inspections is that one can overlook the active damage that occurs in the plant; furthermore, immediately after inspection, the damage can continue to grow until the next periodic inspection is scheduled. Clearly, such situations should be avoided. A solution is offered by continuous monitoring using corrosion monitoring systems. Different techniques are available for corrosion detection and monitoring in the chemical process industry [5,6]. In this research, we identify the most important types of corrosion in the chemical process industry using the acoustic emission signals that are emitted during the corrosion process. Chemical reactions, as occurring during corrosion, emit acoustic activity [7,8] as well as the microscopic damage and fracture processes that occur during corrosion [9]. The acoustic emission technique has the advantage that it is low cost and allows for a continuous, on-line monitoring so that the damage can be detected as soon as it occurs [6].

1.2. Wavelet packet feature extraction and selection from acoustic emission

Although future successes in corrosion prevention still depend on selecting and developing more corrosion resistant materials, it is expected that the main progress in corrosion prevention will be achieved with better information-processing strategies and the development of more efficient monitoring tools that support corrosion control programs [10]. Feature extraction, feature subset selection, and classifier choice and design are all information-processing strategies that should be explored in the design of better corrosion monitoring systems.
Features to characterize the acoustic emission activity have often been obtained in the time-amplitude domain [5,8,11], the frequency domain [5,8,12], or the time-frequency domain using the Continuous Wavelet Transform (CWT) [13,14], the Discrete Wavelet Transform (DWT) [14] or the Wavelet Packet Transform (WPT) [15]. The process of constructing informative features that discern between different classes is often not trivial, but some generic approaches are available [16]. One generic approach is to consider basis functions that can be used to extract features. A library of basis functions can be obtained from the Wavelet Packet Transform [17-19]. The Wavelet Packet Transform can be preferred over other feature extraction and construction techniques such as PCA [20] (principal component analysis), LDA [20] (linear discriminant analysis) and MMI [21] (maximization of mutual information) due to the ease of interpretation of the wavelet packet coefficients in terms of acoustically relevant variables as frequencies and bandwidths [22]. Moreover, Wavelet Packet Decompositions are more flexible than the Discrete Wavelet Transform (DWT) and the Fourier Transform (FT) [19].

One of the challenges that arises after the use of the Wavelet Packet Transform is the selection of a basis that is optimal in some sense, or the selection of a few coefficients for signal compression or pattern recognition purposes [18,23-28]. The current paper contributes to the selection of the most informative basis functions, from a library of wavelet packets, to distinguish between different classes of corrosion, using information theory. We use mutual information [29] to guide the search for informative basis functions by taking into account the statistical dependencies between the wavelet coefficients. Mutual information is intensively used in chemometrics as a feature selection criterion [30-34]. It is a filter-based variable selection technique, meaning that it does not take the interaction with the final machine learning algorithm used for pattern prediction into account [16,35]. This may lead to an inferior performance compared to wrapper [35] approaches. However, the latter often come
with an increased computational cost. A wrapper-based approach may become computationally expensive when thousands of features are obtained, which is typically the case after wavelet packet coefficients have been extracted from acoustic emission signals. A solution exists in combining the mutual information-based approach with a wrapper search, leading to a so-called hybrid filter-wrapper approach [36,37]. In this article, we will follow this hybrid filter-wrapper strategy by performing the expensive local search in a genetic algorithm [38] with a simple, mutual information-based filter. Our approach is generically applicable to classification problems only requiring a set of training signals with corresponding class labels. The approach proposed here is called GIBFS (Genetic Informative Basis Function Selection). It is a genetic algorithm driven approach using mutual information as a local search procedure applied to a dictionary of basis functions for feature extraction.

2. MATERIALS AND METHODS

2.1. Discrimination between different corrosion types

The most frequently occurring corrosion processes in the chemical process industry are as follows: uniform corrosion (or general corrosion), pitting and stress corrosion cracking (SCC) [4,5]. It may also be possible that no corrosion process is active during the measurement. Therefore, this work considers in addition to the first three classes, also absence of corrosion to be discriminated from each other. A corrosion monitoring system in the chemical process industry should be able to detect and discriminate between these corrosion types for several reasons. Firstly, pitting and SCC are more harmful types of corrosion compared to uniform corrosion. Uniform corrosion reduces the thickness of the material relatively uniformly, which is usually taking a long time before the material properties are compromised. On the other hand, pitting punches through the material and SCC causes cracks which can grow much
faster due to the locally increased mechanical forces on the material. The latter two corrosion types will cause leaks to occur sooner in chemical plants. Therefore, the occurrence of pitting and SCC acoustic emission (AE) events should trigger a visual inspection of the installation sooner. Secondly, the discrimination between different corrosion types should be performed prior to the correlation of acoustic emission activity to the corrosion rate. In [39], a quantitative analysis has shown that the count rate (this is defined by the authors as the total number of threshold crossings of AE signals per unit area of the exposed part of the metal sample per unit time) is correlated with the rate of corrosion measured by means of the weight loss of the metal sample. A quantitative relation between the number of AE events in pitting and the number of pits as well with the pitted area and volume was established in [40]. In stress corrosion cracking, a relationship between AE parameters (counts change per unit time and energy change per unit time) and the speed of the corrosion (change of crack length per unit time) has been established [41]. This shows that in different corrosion processes one can estimate the corrosion speed from AE parameters, although one should first link an AE event to the type of corrosion process: erroneously relating AE events originating from pitting to SCC leads to a wrong estimate of the speed of the corrosion.

In the absence of corrosion, the signals are characterized by a continuous acoustic signal with low amplitude [5]. Examples of acoustic signals recorded from stainless steel and carbon steel are shown in Fig. 1.

FIGURE 1 HERE (ONLY COLOR ON THE WEB)

2.1.1. Uniform corrosion
Acoustic signals in the uniform corrosion experiments are characterized by a continuous-type acoustic emission signal [5,15]. The fact that, during uniform corrosion, only a very limited number of events (bursts) can be detected, as opposed to non-uniform corrosion, and intense localized corrosion, such as pitting and SCC, is also supported by [39]. Also in [42], it was observed that, in neutral aerated media, burst-type acoustic emission in uniform corrosion is very limited. Examples of continuous emission signals during uniform corrosion are shown in Fig. 2.

FIGURE 2 HERE (ONLY COLOR ON THE WEB)

2.1.2. Pitting
Localized forms of corrosion, such as pitting and stress corrosion cracking, lead to a burst-type acoustic activity. The following diverse sources have been suggested to generate acoustic events during pitting: scale cracking, oxide fracture, rupture of an oxide or salt cap covering the pits [43], hydrogen bubble formation and motion from the growing pits [44], and instantaneous stress changes on the metal surface [45]. Examples of pitting acoustic events are shown in Fig. 3.

FIGURE 3 HERE (ONLY COLOR ON THE WEB)

2.1.3. Stress corrosion cracking
Stress corrosion cracking is characterized by a burst-type acoustic activity. Examples for stainless steel and carbon steel are shown in Fig. 4.

FIGURE 4 HERE (ONLY COLOR ON THE WEB)
Among the causes that give rise to AE events in stress corrosion cracking are: volumetric expansion during rust growth in fine crevice or cracks on the free surface [46], crack growth [9], metal dissolution [9], hydrogen gas evolution [9], a breakdown of the thick oxide film [9], fracture or decohesion of precipitate and inclusions at the crack tip [9], plastic deformation by slip or twin at the crack tip [9,41], stress induced martensitic transformation [9], and micro- or macro-cracking [9].

Hence, different mechanisms can be responsible for acoustic activity within each corrosion class, which can lead to a statistical spread of the extracted features within each class.

2.2. Experimental set-up

This section briefly sketches the experimental set-up for recording the acoustic emission signals. The experimental set-up is shown on the left side of Fig. 5.

FIGURE 5 HERE

The probe is designed such that the corrosion process to which it is subjected is representative for that of the corrosion occurring in the chemical process plant [5]. This allows one to estimate the corrosion speed of the plant from the corrosion of the probe. This implies that the probe is exposed to the same environmental conditions as in the plant, i.e., the corrosive medium, temperature and pressure. This is represented in Fig. 5, where the probe is exposed in a by-pass of the process plant to the same conditions, due to the input flow that arrives from the plant and the output flow that is guided back to the plant.

The advantage of using a reference probe can be seen as follows. The probe is relatively small, approximately 30 cm in height. This means that any dampening of the waves, as they
propagate over such small distances, is small. On the other hand, when performing measurements on the large installation itself, AE waves may have dampened out before they reach an AE sensor. Moreover, due to the large difference in distances over which waves may have travelled, AE events can be deformed to different degrees, e.g., due to dispersion. This deformation will hamper the recognition of the type of corrosion from the AE waveforms. Additionally, installations are often exposed to external sources that can create AE events, e.g., mechanical vibrations, rain drops, etc. [5]. These sources may be confounded with AE events originating from corrosion events.

The elastic waves that are created by the damage that occurs to the probe can be captured by means of piezoelectric sensors attached to the corroding probe. In order to guarantee a good acoustical transfer from the probe to the sensor, a “high vacuum” grease (DOW Corning®) is applied between the sensor and the probe. The sensors used [5] here are broadband sensors (B1025, Digital Wave Corporation). These sensors have a guaranteed frequency bandwidth of 50 kHz to 2 MHz and can be used in a temperature range of -50 °C to 100 °C. Subsequently, the signals are amplified by an amplification factor of approximately 40 dB. The signals are then bandpass-filtered between 50 kHz - 2 MHz, because outside this range the sensor does not guarantee reliable information [5]. Signals were sampled at 20 MHz or 25 MHz, and both sampling rates are safely higher than the Nyquist sampling rate of 4 MHz for signals of up to 2 MHz. Before computing the Wavelet Packet Transform, the signals are resampled at 25 MHz if they were sampled at 20 MHz. Hence, after resampling, the same wavelet packet coefficients from different signals contain information about the same bandwidths and center frequencies.

2.3. Experimental conditions
Two types of steel, carbon steel and stainless steel, are considered. They are most often the ones involved in failures in petrochemical plants: 42.1% for stainless steel and in 20.9% of the cases carbon steels are involved [4]. The carbon steel used here is number 1.0038 (German Material Number), name S235JRG2 (DIN EN 10025) or RSt 37-2 (DIN 17100). The stainless steel used here is number 1.4541 (German Material Number), name X6CrNiTi18-10 (DIN EN 10088-2) and similar to AISI 321. The chemical composition of the 2 considered steels can be found in [47]. All materials and experimental conditions are summarized in Table 1, together with the number of different experiments for the material-environment combinations (the environment is the combination of a corrosive medium and a temperature). The total number of time series obtained from these experiments is indicated between brackets. The signals for each experiment were often collected over several days of measuring to obtain a representative set of signals. The acoustic emission data set contains 197 time series of “absence of corrosion” (indicating that no corrosion was active during these experiments), 194 time series of uniform corrosion, 214 time series of pitting and 205 time series of SCC. The time series have been assigned a corrosion class label by an expert [5] based on the visual inspection of the damage to the probe, the experimental conditions, and the inspection of the acoustic emission signals [5]. Each time series consists of “N” = 1024 samples.
Table 1

The steel types, the corrosive medium and the number of different experiments considered. The data was obtained from [5].

<table>
<thead>
<tr>
<th>Type of corrosion</th>
<th>Material</th>
<th>Corrosive medium + conditions</th>
<th>Number of experiments (number of time series)</th>
<th>Total number of experiments per class (number of time series)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absence of corrosion</td>
<td>1.0038</td>
<td>NaOH 20 weight% + NaCl 3 weight% 80°C</td>
<td>1 (99)</td>
<td>4 (197)</td>
</tr>
<tr>
<td></td>
<td>1.4541</td>
<td>CaCl₂ 40 weight% 85°C</td>
<td>3 (98)</td>
<td></td>
</tr>
<tr>
<td>Uniform corrosion</td>
<td>1.0038</td>
<td>H₃PO₄ 10 weight% T_environment</td>
<td>6 (194)</td>
<td>6 (194)</td>
</tr>
<tr>
<td>Pitting</td>
<td>1.4541</td>
<td>brackish water + FeCl₃ 1 weight% 45°C</td>
<td>9 (214)</td>
<td>9 (214)</td>
</tr>
<tr>
<td>Stress corrosion cracking</td>
<td>1.0038</td>
<td>Ca(NO₃)₂ 60 weight% 105°C</td>
<td>1 (147)</td>
<td>10 (205)</td>
</tr>
<tr>
<td></td>
<td>1.4541</td>
<td>CaCl₂ 40 weight% 85°C</td>
<td>9 (58)</td>
<td></td>
</tr>
</tbody>
</table>

2.4. Software

The local discriminant bases algorithm was kindly provided by Prof. Dr. N. Saito [25] in Matlab. All mutual information-based filtering algorithms (Parzen density, K-nearest neighbors and normalized mutual information) and the hybrid-filter wrapper genetic algorithm were written by the first author in Matlab. The classification algorithms were used from the following software packages: K-nearest neighbors (Netlab [48]), Gaussian mixture model (Netlab [48]), decision tree J48 (Weka 3.4 [49]) and naïve Bayes (Weka 3.4 [49]).

3. THEORY
3.1. Wavelet packet features

This section provides the basic definitions of wavelet packet coefficients used as features. The mathematical details about the construction of wavelet packet basis functions, their properties and fast computations of wavelet coefficients can be found in excellent textbooks [19,50]. Let us represent a single time series by means of a sequence of observations $x(t)$: $x(0)$, $x(1)$, … $x(N-1)$, where “t” refers to the time index and “N” is the number of samples. The observations can be considered as being sampled from variables $X(t)$: $X(0)$, $X(1)$, … $X(N-1)$. It is instructive to consider the wavelet packet coefficients as being computed from the inner product between the wavelet basis functions $\psi_{i,j,k}(t)$ and the time series as follows (using a continuous time notation):

$$\gamma_{i,j,k} = \langle x(t), \psi_{j}(t - 2^k) \rangle = \int_{-\infty}^{+\infty} x(t)\psi_{j}'(t - 2^k)dt. \quad (3.1)$$

A feature, in this case a wavelet coefficient, in the Wavelet Packet Decomposition needs to be specified by the scale index “i”, frequency index “j” and time index “k”. Hence, the coefficient $\gamma_{i,j,k}$ can be considered as quantifying the similarity, by means of the inner product, between the time series $x(t)$ and the wavelet function $\psi_{i,j,k}(t - 2^k)$ at position $2^k$ in time. The variables that can be associated with the coefficients $\gamma_{i,j,k}$ are further denoted by capitals $\Gamma_{i,j,k}$. The index “i” is the scale index and used to obtain dilation (commonly called a “stretching”) of the wavelet function $\psi_{i,j}(t)$ by a factor $2^i$:

$$\psi_{i,j}'(t) = \frac{1}{\sqrt{2^i}}\psi_{i,j}'\left(\frac{t}{2^i}\right). \quad (3.2)$$

This index will determine the bandwidth of a frequency band in time-frequency space.

The index “j” determines the shape of the wavelet function; in time-frequency space, it determines the position of the frequency band on the frequency axis. A full Wavelet Packet
Decomposition results in many wavelet coefficients. In cases where one can assume that the exact time location “k” of the basis function is of no importance, one can, e.g., consider the energy or the entropy of wavelet coefficients over time for each possible sub-band, i.e., a particular combination of the scale index “i” and the frequency index “j”. This will lead to fewer features to select from. Here, we will consider the full complexity of the problem, when the exact time location of the basis function may be important, and we will consider all coefficients from a full Wavelet Packet Decomposition as possible candidates to select from.

A full Wavelet Packet Decomposition leads to $N^*(\log_2 N + 1)$ features, as illustrated in Fig. 6. In Fig. 6, we show a graphical representation of the first different subspaces that are obtained in a Wavelet Packet Decomposition up to scale index “i” equal to 3, together with the coefficients that are extracted from within these subspaces.

![FIGURE 6 HERE](image)

A particular subspace at scale “i” and index “j” is denoted by $W^i_j$. The number of subspaces is determined by the scale index “i”: the number of subspaces at scale “i” is equal to $2^i$. The frequency index “j” at a certain scale “i” will be an integer from $[0, 2^i - 1]$, indicating the starting position of the subspace at scale “i”.

As can be seen from Equation 3.1, at scale “i” the inner products are computed at discrete time positions $2^i k$. At scale 0, we have “N” (length of the signals) coefficients: $\gamma_{0,0,0}, \ldots, \gamma_{0,0,N-1}$. At the next scale “i” = 1, we obtain $N/2$ coefficients in each subspace, i.e., $\gamma_{1,0,0}, \ldots, \gamma_{1,0,N/2-1}$ and $\gamma_{1,1,0}, \ldots, \gamma_{1,1,N/2-1}$. At the highest frequency resolution, “i” = $\log_2 N$, and we obtain the following coefficients: $\gamma_{\log_2 N,0,0}, \ldots, \gamma_{\log_2 N,N-1,0}$. Hence, at each scale there are “N” coefficients and in total there are $\log_2 N + 1$ different scale levels. This leads overall to $N^*(\log_2 N + 1)$ different coefficients to select from. For our acoustic emission time series “N” = 1024 leading
to 11264 features for each time series. In pattern recognition and machine learning this amount of features is considered to be a large scale feature selection problem.

3.2. Local discriminant basis (LDB)

The local discriminant basis algorithm can be considered as a benchmark for basis function selection, from the Wavelet Packet Decomposition, against which new algorithms should be compared. This is a filter-based feature selection method, because it does not take any feedback into account from the classification paradigm used. The local discriminant basis algorithm [23] was proposed as an extension of the entropy-based best basis selection algorithm [18], developed for signal compression, to select a best basis for pattern recognition. A method based on the normalized energy of signals has been developed [23,24], and there is another version based on a probability density function (pdf) estimate of the wavelet coefficients [24,25]. The pdf based method carries also the more subtle information such as the phase [24,25]. This pdf based algorithm is summarized here.

Assume that we are given a set of training signals \( x_j \) and, for each one of them, the associated target class \( c_j: \{(x_j, c_j)\}_j. \)

Step 0: Expand each training signal into a time-frequency dictionary \( D \), which involves the computation of all coefficients \( \gamma_{i,j,k} \) for each training signal and assumes that we choose a particular conjugate mirror filter [19] in advance that will define the basis functions.

Step 1: Estimate the class conditional probability density functions \( \hat{p}^y(\Gamma_{i,j,k}) \) (pdf’s) for each wavelet coefficient variable \( \Gamma_{i,j,k} \) in the dictionary. Superscript “y” refers to the class label, with \( y = 1, 2, \ldots, \#C \) and \( \#C \) is the total number of classes. These pdf’s were estimated by means of the averaged shifted histograms method (ASH) as in [25].
Step 2: For each wavelet coefficient variable $\Gamma_{i,j,k}$, compute the discriminant measure $\delta_{i,j,k}$.

The computational cost of this procedure is $O((N+1)\log_2 N)$. Many discriminant measures can be used in practice. We use the symmetric relative entropy, Equation 3.5, as in [25]. The relative entropy for $\Gamma_{i,j,k}$ between two classes, $y = 1$ and 2, is computed as [29]:

$$D(\hat{p}_1(\Gamma_{i,j,k}), \hat{p}_2(\Gamma_{i,j,k})) = \int \hat{p}_1(\gamma_{i,j,k}) \log_2 \frac{\hat{p}_1(\gamma_{i,j,k})}{\hat{p}_2(\gamma_{i,j,k})} d\gamma_{i,j,k}.$$  \hspace{1cm} (3.3)

Because this discriminant measure is, in general, not symmetric, a symmetric version is obtained as:

$$\delta_{i,j,k} = D^S\left(\hat{p}_1(\Gamma_{i,j,k}), \hat{p}_2(\Gamma_{i,j,k})\right) = D\left(\hat{p}_1(\Gamma_{i,j,k}), \hat{p}_2(\Gamma_{i,j,k})\right) + D\left(\hat{p}_2(\Gamma_{i,j,k}), \hat{p}_1(\Gamma_{i,j,k})\right).$$  \hspace{1cm} (3.4)

When more than two classes are considered, $\delta_{i,j,k}$ is defined as the sum over all $\#C \times (\#C - 1)/2$ pairs of different classes as:

$$D^S_{\text{pair}}\left(\hat{p}_1(\Gamma_{i,j,k}), \hat{p}_2(\Gamma_{i,j,k}), \ldots \hat{p}_C(\Gamma_{i,j,k})\right) = \sum_{m=1}^{\#C-1} \sum_{n=m+1}^{\#C} D^S\left(\hat{p}_m(\Gamma_{i,j,k}), \hat{p}_n(\Gamma_{i,j,k})\right).$$  \hspace{1cm} (3.5)

Step 3: Evaluate the discriminant power of each basis $B \in D$ (the dictionary) and obtain the best basis $\Psi$ for which the discriminant power is maximal:

$$\Psi = \arg \max_{B \in D} \sum_{i,j,k: B \in B} \delta_{i,j,k}.$$  \hspace{1cm} (3.6)

Step 4: Select “m” basis functions, $\psi_i(t - 2^j k)$, from $\Psi$ corresponding to the “m” largest $\delta_{i,j,k}$. The number of basis functions “m” to be retained is not defined in [25]. Therefore, we perform experiments for “m” ranging from 1 to 50 basis functions.

Step 5: Construct classifiers with features derived from the “m” basis functions. We will use here the “m” selected coefficient variables $\Gamma_{i,j,k}$ as the features.
The total discriminant power in step 3 is computed as the sum of the discriminant measures of each of the coefficients in a basis $B$:  
\[
\sum_{(i,j,k) \in B} \delta_{i,j,k}
\]. The reason is that searching for the basis with the highest discriminant power can then be achieved very rapidly. It can be shown that an optimal basis can be found in $O(N)$ comparisons, with “$N$” the length of the signal (see, e.g., proposition 9.5 in [19] for the details of such algorithms). However, it was proven in [15] that the joint discriminant power, in the case of the relative entropy, can be written as a sum of marginal discriminant measures of the coefficients, if the coefficients are class conditional independent. Class conditional independence means that the coefficients are independent, when conditioned on each class label. When the joint discriminant power is estimated as the sum of the marginal discriminant powers as in Equation 3.6, the statistical dependencies between the wavelet coefficients are not taken into account, and the selection of basis functions, for pattern recognition purposes, may therefore be suboptimal, as will be seen in section 4. Analytical results on the correlations between wavelet coefficients for a class of stochastic signals (fractal Brownian motion) have been obtained previously in [51,52]. The correlations between wavelet coefficients for fractal Brownian motion decrease with increasing distance between scales and increasing distance between time locations, but neighboring coefficients in scale and time may still show a significant correlation.

Dependencies are also present in the corrosion data set. We computed the wavelet coefficients for all corrosion signals up to level 3 in Fig. 7.

FIGURE 7 HERE (ONLY COLOR ON THE WEB)

A second restriction of the LDB algorithm is present in step 4. Once a basis has been obtained, the basis functions are ordered according to the descending order of the individual discriminant measures $\delta_{i,j,k}$ of the basis functions. Hence, the information that is present in the
previous coefficient variables $\Gamma_{i,j,k}$ is not taken into account when selecting the next coefficient variables.

### 3.3. Mutual information filters

As was true with relative entropy, mutual information can be shown to satisfy an additive property under some conditions: the high-dimensional mutual information can be written as a sum of marginal mutual information contributions. This additive property will be satisfied after independence and conditional independence assumptions between the features [37,53].

The reason to use mutual information here is that it is a well-established criterion for taking dependencies between variables into account. In the last two decades, a lot of research has been directed towards taking these dependencies into account by means of mutual information, although the criterion has in fact been used for the first time as a feature selection criterion almost half a century ago by Lewis [54]. At that time, Lewis did not call his criterion “mutual information”, but a “measure of goodness”.

The high dimensional mutual information between a feature vector $F$ and class variable $C$ can be defined as:

$$MI(F;C) = \sum_{i=1}^{\log_2(N)} \int_F p(f,c) \log_2 \left( \frac{p(f,c)}{p(f)p(c)} \right) df.$$  \hspace{1cm} (3.7)

We use both filter methods and a hybrid filter-wrapper method (section 3.4), all of which are based on mutual information. In the filter methods, we perform a sequential forward search (SFS) over all wavelet coefficients using a mutual information criterion. In the SFS, we start with the empty feature set $S = \{\emptyset\}$ as the selected coefficients so far and the whole dictionary $D = \{\Gamma_{i,j,k}\}$, with $0 \leq i \leq \log_2 N$, $0 \leq j \leq 2^i - 1$ and $0 \leq k \leq N/(2^i) - 1$, as the available feature set. In each iteration of the SFS, the variable $\Gamma'_{i,j,k}$ which achieves the highest value of the mutual information criterion, taking into account the previously selected features, is selected.
S is updated in each iteration as: $S = S \cup \Gamma_{i,j,k}$ and the dictionary is updated as $D = D \setminus \Gamma_{i,j,k}$. Three different mutual information criteria were compared in the SFS filter: a density-based method (section 3.3.1), a distance-based method (section 3.3.2) and a relevance-redundancy method (section 3.3.3).

### 3.3.1. Parzen window density (MI Parzen)

The estimation of the mutual information by means of a Parzen window density estimator was proposed in [55]. If a Gaussian window function is used, the mutual information is estimated as (a hat is used to indicate an estimator):

$$\hat{\text{MI}}(F; C) = \frac{1}{n} \left( -\sum_{j=1}^{n} \sum_{c=1}^{#C} \hat{p}(c | f_j) \log_2 \hat{p}(c | f_j) \right), \tag{3.8}$$

$$\hat{p}(c | f_j) = \frac{\sum_{j=1}^{n} \exp \left( -\frac{(f - f_j)^T \Sigma^{-1} (f - f_j)}{2h^2} \right)}{\sum_{l=1}^{#C} \sum_{j=1}^{n} \exp \left( -\frac{(f - f_l)^T \Sigma^{-1} (f - f_l)}{2h^2} \right)}. \tag{3.9}$$

The functional $H(.)$ is the entropy [29]. Further, $I_k$ is the set of indices of data points which belong to class “k”, $f_j$ is the feature vector of the j’th training data point and $#C$ is the total number of classes. The covariance matrix $\Sigma$ is estimated as the full sample covariance matrix. The parameter “h” is set to a default value as suggested in the experiments in [55]: $h = 1/\log_2(n)$, where “n” is the sample size of the training set. This estimator is referred to as “MI Parzen”.

### 3.3.2. K-nearest neighbors (MI knn)

Instead of estimating the probability density functions, the mutual information between a discrete class variables and a feature vector $F$ can be estimated based on the pairwise distances between data points [37,53,56]. These approaches rely on the Kozachenko-Leonenko entropy estimator [57] of the differential entropy:
\[ \bar{H}(F | c) = -\psi(k) + \psi(n_c) + \ln(c_d) + \frac{d}{n_c} \sum_{i \in I} \ln(\varepsilon_c(i, k)), \]  
(3.10)

which is plugged into:

\[ \text{MI}(F; C) = \bar{H}(F) - \sum_{c=1}^{\text{C}} \bar{H}(F | c) \hat{p}(c). \]  
(3.11)

In Equation (3.10), \( \psi(.) \) is the psi-function, “\( n_c \)” the number of training data points in class “\( c \)”, \( \varepsilon_c(i, k) \) is twice the distance from the \( i \)'th data point in class “\( c \)” to its \( k \)'th neighbor in class “\( c \)” in the training set, “\( d \)” the dimensionality of the data points and “\( c_d \)” the volume of the \( d \)-dimensional unit ball. We used the Euclidean distance between data points, in this case “\( c_d \)" = \( \pi^{d/2}/\Gamma(1 + d/2) \), with \( \Gamma(.) \) the gamma-function.

The unconditional entropy \( \bar{H}(F) \) in Equation (3.11) can be estimated similarly as the conditional entropy in Equation (3.10), but with “\( n_c \)” replaced with the total number of training points “\( n \)” and \( \varepsilon_c(i, k) \) replaced by \( \varepsilon(i, k) \), i.e., twice the distance from data point “\( i \)” to its “\( k \)” nearest neighbor when all training data points from all classes are merged in one set.

The prior probabilities \( \hat{p}(c) \) are estimated as the number of training points in class “\( c \)” divided by the total number of training points as follows: \( n_c/n \). In the experiments, the number “\( k \)” of nearest neighbors was set equal to 6. This estimator is referred to as “\( \text{MI knn} \)”.

3.3.3. Relevance-redundancy approaches

Relevance-redundancy approaches select variables that are highly relevant with respect to the class variable, but penalize a variable if it is redundant with respect to previously selected variables. These approaches often use mutual information to estimate both the relevance and the redundancy. Suppose that \( F_i \) is a candidate variable to be selected and that \( S \) is the set of already selected variables; relevance-redundancy criteria are then generically represented as:

\[ \text{Crit}_S(F_i, C) = \text{MI}(F_i; C) - \beta \sum_{F_j \in S} \alpha(F_i, F_j) \text{MI}(F_i; F_j). \]  
(3.12)
Note that, as opposed to Equations (3.8) and (3.11), here only the lower dimensional MI(F_i;C) and MI(F_i;F_s) are required. The first relevance-redundancy criterion was proposed by Battiti [58], which is obtained from Equation (3.12) by setting: \( \alpha(F_i,F_s) = 1 \) and \( 0.5 \leq \beta \leq 1 \). The criterion in [59] is obtained by setting \( \alpha(F_i,F_s) = 1 \) and \( \beta \) is adaptively chosen as \( 1/|S| \). A normalized mutual information criterion was proposed in [60], this criterion can be obtained from Equation (3.12) with \( \alpha(F_i,F_s) = 1/\min\{H(F_i),H(F_s)\} \) and with \( \beta \) adaptively chosen as \( 1/|S| \). Among all the relevance-redundancy approaches NMIFS resulted for many cases in the highest accuracies [60]. In this paper, we used the normalized mutual information as the relevance-redundancy approach in the genetic algorithm. In the computation of the normalized mutual information, the features were first discretized into 3 states [61]: values of \( F_i < \mu(F_i) - (\sigma(F_i))/2 \) were set to state 0, \( \mu(F_i) - (\sigma(F_i))/2 \leq F_i \leq \mu(F_i) + (\sigma(F_i))/2 \) were set to state 1 and values of \( F_i > \mu(F_i) + (\sigma(F_i))/2 \) were set to state 2. Note that \( \mu(F_i) \) and \( \sigma(F_i) \) are respectively the mean and standard deviation of \( F_i \). The mutual information was then computed from the contingency tables of the discretized variables.

### 3.4. Hybrid filter-wrapper genetic algorithm

Both the local discriminant basis algorithm and the mutual information based approaches select the variables independently of the final classifier. Wrapper approaches [35] include the classifier by providing feedback on the classification performance of selected subsets using the classifier. A comparison between feature selection algorithms has shown that genetic algorithms are more suitable than other feature selection algorithms for very large scale feature subset selection problems (more than 100 features) [62]. They were introduced as a feature selection strategy in chemometrics by Leardi et al. [63]. The strength of genetic algorithms exists in the application of two strategies that are important in optimization: exploitation and exploration [64-66]. Exploration allows one to explore a large part of the search space, so that different subsets are selected and tested at random. In an exploitation
strategy, on the other hand, one searches in the neighborhood of a starting point in order to reach a local optimum in the neighborhood of that starting point.

We have designed a genetic algorithm (GA) for the wrapper search which uses a relevance-redundancy filter as the local improvement operator.

Given the high dimensionality of the problem (about $10^4$ features) and the large number of samples (about $10^3$), we have used a steady-state genetic search procedure [38]. In a steady-state approach, one chromosome in the population is updated in each generation, whereas in generational genetic search procedures, the whole of the population or most it is updated in each generation. A steady-state GA requires less model evaluations compared to a generational GA given a maximum number of generations and a fixed population size. The proposed genetic algorithm also allows one to specify the number of features that need to be selected. Overall, our GA variable selection algorithm can be characterized as a hybrid filter-wrapper genetic algorithm that uses the steady-state procedure and has a preset number of features to be selected. The GA is also hybrid in the sense that it includes a local improvement operator. The genetic algorithm is outlined in Fig. 8.

FIGURE 8 HERE

Each chromosome in the population represents a feature subset. These chromosomes are encoded as a string of bits, where a 0 and a 1 bit indicate whether a feature is removed or included, respectively. The length of the chromosomes is equal to the total number of features to select from. Consider, e.g., the following chromosome Cr: 0110000101. This chromosome represents that there are 10 features in total (10 bits) from which 4 features are selected: the 2nd, the 3rd, the 8th and the 10th; the other features are not included in the subset.
3.4.1. Initialization of the population

A first choice to be made in the initialization of the population is the population size $n_{pop}$. A larger population allows a larger difference among the chromosomes to co-exist. Increasing the population size increases exploration and decreases exploitation.

The $n_{pop}$ chromosomes can be initialized in different ways. In [38], the authors initialized the whole population with random feature subsets, whereas in [60] the authors initialized a part of the chromosomes in the initial population randomly and the other part used subsets that included some good features. Our experience is that if some good features are already included in the initial population, this can speed-up the convergence of the genetic algorithms [36] and may lead to a superior classification performances as well [67].

A balance between random initialization (exploration) and the exploitation of already good feature subsets can be obtained by initializing a fraction of the population $f_{pop}$ (with $0 \leq f_{pop} \leq 1$) with random feature subsets. In the other fraction $(1-f_{pop})$ of the population, a number of features $d_{chrom}$ ($1 \leq d_{chrom} \leq d_{tar}$, i.e., larger than or equal to 1, but smaller than or equal to the target number of features) in each chromosome can be selected according to a feature selection criterion (3.12) [60]. The random initialization is obtained in pseudo-code (we don’t use “end” to close loops and conditions to save space) as follows:

% fraction $f_{pop}$ initialized randomly:

```matlab
for i = 1 to $f_{pop} \cdot n_{pop}$ % repeat $f_{pop} \cdot n_{pop}$ times.
    for (each gene g in the i’th chromosome) % iterate over all genes in each chromosome.
        if (rand() < $d_{tar} / \#$genes) % generate a uniform random number.
            g = 1; % set the current gene to 1.
        else
```
\[ g = 0; \quad \text{% set the current gene to 0.} \]

The result will be that in each chromosome about \( d_{\text{tar}} \) genes will be set to 1, and hence about \( d_{\text{tar}} \) features will be selected randomly.

The other fraction is initialized as follows:

\[
\begin{align*}
% \text{fraction } (1 - f_{\text{pop}}) \text{ of the population initialized with some } (d_{\text{chrom}}) \text{ good features and some randomly selected features.} \\
% \text{Apply (3.12) to obtain a set of } S \text{ of } d_{\text{chrom}} \text{ features.} \\
\text{for } i = f_{\text{pop}} \cdot n_{\text{pop}} + 1 \text{ to } n_{\text{pop}} \text{ % repeat } (1 - f_{\text{pop}}) \cdot n_{\text{pop}} \text{ times.} \\
\text{for all } F_j \in S \text{ set the } j'\text{th gene in the } i'\text{th chromosome to 1;} \\
\text{for } (\text{all other genes } g \text{ in the } i'\text{th chromosome}) \text{ % iterate over all other } (\#\text{genes-}d_{\text{chrom}}) \text{ genes not selected from } S. \\
\text{if } (\text{rand()} < (d_{\text{tar}} - d_{\text{chrom}}) / (\#\text{genes-}d_{\text{chrom}})) \text{ % generate a uniform random number.} \\
g = 1; \quad \text{% set the current gene to 1.} \\
\text{else} \\
g = 0; \quad \text{% set the current gene to 0.}
\end{align*}
\]

Hence, in each chromosome \( d_{\text{chrom}} \) genes are set to 1, according to the features selected by criterion (3.12), and an additional \( d_{\text{tar}} - d_{\text{chrom}} \) genes are expected to be set to 1 randomly. Increasing the fraction of randomly initialized subsets \( f_{\text{pop}} \) will increase exploration and decrease exploitation; increasing \( d_{\text{chrom}} \) for a fixed \( f_{\text{pop}} \) will decrease exploration and increase exploitation.

3.4.2. Select two parents from the population
Suppose that $X_{C_{ri}}$ is the set of selected features by chromosome $C_{ri}$. Then, the fitness is computed as: $\text{fitness}(C_{ri}) = \text{acc}(X_{C_{ri}}) - \text{penalty}(X_{C_{ri}})$, where, acc($X_{C_{ri}}$) is the 10 fold cross-validation performance, in percent, using the training set. The penalty ($X_{C_{ri}}$) is computed as: $w.||X_{C_{ri}}|-|\text{tar}|$, with $|X_{C_{ri}}|$ the size of the feature subset (the number of 1 genes). The penalty coefficient, $w$, is set to 0.5, as in [38]. Subsequently, a rank-based roulette wheel selection procedure is applied [64]. This procedure favors fitter chromosomes to pass their genes on to the next generation. The chromosomes in a generation are first sorted in non-increasing order, according to their fitness. Using this sorting, the chromosomes can be assigned a rank “j”. The fittest chromosome is assigned rank 1, the next fittest rank 2, and so on. Next, a non-linear function of the rank assigns a probability to each chromosome to pass on to the next generation: $P(j) = q(1-q)^{j-1}$. The parameter “q” controls the selection pressure. When “q” is small, there is only a small influence of the rank “j”. A larger “q” increases the selection pressure. The parameter “q” is set equal to 0.25. Disposing of the probabilities $P(j)$, the selection of the chromosomes can be implemented as follows. Firstly, compute the cumulative probabilities as $p_r = \sum_{j=1}^{r} P(j)$ for $r = 1, 2, ..., n_{pop}$ and $p_0 = 0$. Next, two random numbers rand1() and rand2() with uniform distribution on the interval $[0, \frac{1}{n_{pop}}]$ are drawn. Finally, the chromosomes that correspond with rank “r1” such that $p_{r1-1} < \text{rand1()} < p_{r1}$, and with rank “r2” such that $p_{r2-1} < \text{rand2()} < p_{r2}$ are selected. To avoid that the same parent is selected twice, a new rand2() is drawn as long as $r2 = r1$.

3.4.3. Crossover and mutation

We used the standard m-point crossover operator [64,65]. Suppose that we have two parent chromosomes parent1 0110000101 and parent2 1010101010. As an example, we define three crossover points in parent1 01|1000|01|01 and parent2 10|1010|10|10. After the mutation, these two parents generate two children, child1 01|1010|01|10 and child2 10|1000|10|01. Each
child received copied segments alternately from the two parents. The number of crossover points, “m”, is set to five in the experiments. These crossover points were selected randomly. Increasing the number of crossover points favors the exploration [64,65] principle in optimization due to the fact that it tends to break schema.

After the crossover operation, the children are mutated. Simply applying a mutation operator to the large feature set size considered here (11264 features) would drastically change the number of features selected (the number of “1” bits). Consider a fixed probability of mutation $p_m$ equal to 0.1, and suppose that we have before mutation 20 “1” bits and $(11264-20)$ “0” bits. After mutation, this would lead to approximately $20 \times 0.9 + (11264-20) \times 0.1 = 1142$ “1” bits. Therefore, a bit dependent mutation operator is applied [38]. Bits which are equal to “1” in a child are set to “0” with a probability $p_m$. The bits which are equal to “0” are set to “1” with a probability $(n1/n0).p_m$, where $n1$ is the number of “1” bits and $n0$ the number of “0” bits in the child. The probability $(n1/n0).p_m$ guarantees that the expected number of “1” bits is not changed after the mutation operator has been applied: $E(n_1) = (1-p_m).n1 + (n1/n0).p_m.n0 = n1$, with $E(.)$ being the expectation operator.

3.4.4. Local improvement

Genetic algorithms are able to explore a large part of the search space and can escape from regions in the neighborhood of local optima. On the other hand, most classical optimization algorithms focus on a local search and succeed in finding an optimum in the neighborhood of the starting point. A simple GA lacks the ability to fine-tune its solutions to a local optimum. This may render the genetic search ineffective. Combining both a local search and a genetic search, leading to a so called hybrid genetic algorithm, leads to improved results [38,65,66]. In order to fine-tune the feature subsets after crossover and mutation, we apply a sequential forward search (SFS) or a sequential backward search (SBS). The local improvement operator
calls an SFS or SBS on its own in each generation and is, therefore, computationally expensive compared to crossover and mutation operator. In practice, either a filter or a wrapper approach could be used in the local improvement. However, in our case, we dealt with a very large scale feature set problem and opted for a filter approach. We used the criterion in (3.12) based on the normalized mutual information; in section 4, it will be observed that this was the best filter method for our data. The local improvements are applied to both children. Let us denote the selected subset of a child by $X$, the number of features in it by $|X|$ and the subset of available features $Y$. One iteration of the SFS on child $X$, denoted as $\text{SFS}(X)$, using (3.12), consists of: $F = \arg \max_{F \in Y} \text{Crit}_X(F, C)$, $Y = Y \setminus \{F\}$ and $X = X \cup \{F\}$. First, the feature $F$ from the set of available features $Y$ is searched, with $\text{Crit}_X(F, C)$ being maximal, then this feature is removed from the set of available features $Y$ and added to the child $X$. One iteration of the SBS on child $X$, denoted as $\text{SBS}(X)$, using (3.12) consists of: $F = \arg \min_{F \in X} \text{Crit}_{X \setminus \{F\}}(F, C)$, $Y = Y \cup \{F\}$ and $X = X \setminus \{F\}$. First, the feature $F$ from $X$ is searched that is least relevant and most redundant with respect to the other features in $X$, i.e. $X \setminus \{F\}$. Next, this feature is added to the set of available features $Y$ and removed from $X$. The SFS and/or SBS procedures are iterated multiple times, depending on whether the child contained equal, less or more features than the target number of features, $d_{\text{tar}}$, such that the target number of features is reached after the local improvement:

switch

\begin{align*}
\text{case } & d_{\text{tar}} = |X| \\
& \text{SBS}(X) \text{ followed by } \text{SFS}(X); \quad \% \text{ remove 1 feature, next add 1 feature.} \\
\text{case } & d_{\text{tar}} < |X| \\
& \text{for } (|X| - d_{\text{tar}} \text{ times}) \text{ SBS}(X); \quad \% \text{ remove } |X| - d_{\text{tar}} \text{ times 1 feature.} \\
\text{case } & d_{\text{tar}} > |X| \\
& \text{for } (d_{\text{tar}} - |X| \text{ times}) \text{ SFS}(X); \quad \% \text{ add } d_{\text{tar}} - |X| \text{ times 1 feature.}
\end{align*}
3.4.5. Replacement

After the local improvement, two children are obtained. In the steady-state approach only one child replaces a chromosome from the population in each generation, and we select the child that has the highest cross-validation within the training set. The replacement scheme from [38] and [68] is adopted. If the child is superior to both parents, the most similar parent is replaced. The most similar parent is defined as the one for which the Hamming distance to the child is the smallest. If the child’s performance is in between the two parents’ performances, the inferior parent is replaced. Otherwise, the most inferior chromosome from the population is replaced.

3.4.6. Summary of GA parameters

Several parameters in GA’s need to be set, the settings used in the experiments are summarized:

- the fraction of the population initialized randomly $f_{\text{rpop}} = 0.5$: increasing $f_{\text{rpop}}$ increases exploration and decreases exploitation;
- the number of features initialized with a relevance-redundancy criterion in each chromosome in a fraction $1-f_{\text{rpop}}$ of the population $d_{\text{chrom}} = 10$: increasing $d_{\text{chrom}}$ decreases exploration and increases exploitation;
- the population size $n_{\text{pop}} = 20$: increasing $n_{\text{pop}}$ increases exploration and decreases exploitation;
- the probability of mutation $p_{\text{m}} = 0.1$: increasing the probability of mutation increases exploration and decreases exploitation;
- the selection pressure coefficient $q = 0.25$: increasing $q$ favors the selection of the more fit chromosomes;
• the number of crossover points \( = 5 \): increasing the number of crossover points increases exploration and decreases exploitation;

• the number of generations \( n_{\text{gen}} = 500 \): this determines largely the computational cost;

• the penalty coefficient \( w = 0.5 \): increasing “w” favors the selection of chromosomes for which the number of selected features is closer to the target number of features;

• the target number of features \( d_{\text{tar}} = 20 \): increasing \( d_{\text{tar}} \) increases the computational cost and the risk of over-fitting.

4. RESULTS AND DISCUSSION

In the experiments, we have chosen the 12-tap Coiflet filter [69], which is available in the local discriminant basis software and in Matlab. The 12-tap Coiflet filter was chosen for all mutual information feature selection methods and for GIBFS. We used four different classifiers that are regularly used in chemometrics [20]:

- k-nearest neighbor (knn): the Euclidean distance is used with “k” set to 3;

- decision tree J48 (WEKA’s implementation of C4.5) from WEKA package 3.4.1, we used the default values from the WEKA package, i.e., the minimum number of instances per leaf (-M) equal to 2 and the confidence factor for pruning (-C) is equal to 0.25;

- Gaussian Mixture Model (GMM): the number of Gaussians per class is taken equal to 1 in the experiments and, hence, each class is modeled as a multivariate Gaussian distribution (see, e.g., McLachlan end Peel [70] for a reference on Gaussian mixture modeling);
• Naïve Bayes classifier (NB) from WEKA package 3.4.1 with, as an option, kernel estimation (-K) for modeling numeric attributes.

4.1. The evolution of the feature subset size in GIBFS

Initially, a larger part of the feature space is explored using the initialization with different feature subset sizes in the genetic algorithm [38]. For each of the four classifiers, the number of features selected in the chromosomes in the population converges to the target number of features, as a function of the number of generations, with the target number of features \( d_{\text{tar}} \) equal to 20. The evolutions of the “average upper bound”, the “average number of features” and the “average lower bound” for the knn classifier, the decision tree J48, the GMM and the NB classifier are shown in Fig. 9. The “average number of features” was computed as the average of the number of features selected in the 20 chromosomes in the population, and this was considered over all 10 training folds in a 10 fold cross-validation. The average upper (lower) bound was computed as the maximum (minimum) number of features selected in the 20 chromosomes, for each generation, and this was averaged over all 10 training folds in a 10 fold cross-validation.

FIGURE 9 HERE (ONLY COLOR ON THE WEB)

From Fig. 9, it can be observed that the average number of features is close to 20 over all generations, but the average upper bound and the average lower bound in the first generations are clearly higher and lower than 20, respectively. This is due to the initialization procedure in section 3.4.1: the expected number of features selected in each chromosome in the first generation is equal to 20, but the actual number of features selected may differ from chromosome to chromosome. Before the 500th generation is reached, all chromosomes in the
population will have selected 20 features. The generation in which the average upper bound and the average lower bound are both equal to 20 for the first time is indicated by a circle in Fig. 9. From this moment on, the average upper bound, the average lower bound and the average number of features stay all exactly equal to 20. The decreasing behavior of the upper bound down to 20 features and the increasing behavior of the lower bound up to 20 features are due to the local improvement operator in section 3.4.4 and the replacement operator in section 3.4.5. Two parents, which are selected according to the procedure of section 3.4.2, are transformed to two offspring, and the local improvement operator guarantees that both offspring have selected 20 features. The replacement operator will always replace one chromosome in the population by an offspring. Hence, it can never be that an offspring is introduced in the population that will increase the upper bound (given that the upper bound is larger than or equal to 20) or decrease the lower bound (given that the lower bound is smaller than or equal to 20).

4.2. Classification accuracies

Next, we compare the classification performances of the different filter approaches and the hybrid filter-wrapper GA approach. In the case of the filter approaches (LDB, MI Parzen, MI knn, relevance-redundancy), we have performed experiments using anything from 1 to 50 features with the different classifiers. We use a 10 fold cross-validation in which the testing folds have not been used to select features neither to train classifiers. It is observed in Fig. 10 to Fig. 13 that the classification testing accuracies as a function of the number of wavelet coefficients selected, increase initially much faster for the mutual information based approaches compared to the LDB algorithm.
The slower increase in accuracy for the LDB algorithm is related to the fact that the LDB algorithm ignores dependencies between the wavelet coefficients. In fact, the selected wavelet coefficients are largely redundant. In each of the training folds of the 10 fold cross-validation, the local discriminant basis selection algorithm selected subspace $W_0^0$ as the most discriminative basis. Although the coefficients in this subspace provide discriminative information between SCC (largest values), pitting (intermediate values) and uniform corrosion + absence of corrosion (these two classes have the smallest values), the LDB algorithm was mislead by the high dependencies that are present in subspace $W_0^0$. Indeed, in the scatter plot of Fig. 14 it can be seen that the first three features that occurred most often as a triplet in the 10 trainings sets of the 10 fold cross-validation are highly dependent. Each of the three coefficients provide about the same discriminative power, so adding up their discriminative powers to obtain the joint discriminative power is misleading.

The accuracy in the case of the MI knn and the relevance-redundancy approach increases faster than that of the MI Parzen approach. The highest classification accuracies among the
filters are achieved with the relevance-redundancy (normalized mutual information) criterion and this observation is consistent over all the classifiers used. A scatter plot of the triplet of features that was selected most often with the normalized mutual information filter is shown in Fig. 15.

FIGURE 15 HERE (ONLY COLOR ON THE WEB)

The selected coefficients shown are $\gamma_{10,4,0}$, $\gamma_{5,1,6}$ and $\gamma_{9,6,0}$. These coefficients can be related to their corresponding frequency interval

$$
\left[ g \frac{f_s}{2} 2^{-i}, (g + 1) \frac{f_s}{2} 2^{-i} \right],
$$

with, as center frequency,

$$
f_c = \left( g + \frac{1}{2} \right) \frac{f_s}{2} 2^{-i}.
$$

(4.1)

(4.2)

Here, “g” is the Gray order [19] of the wavelet packet and $f_s$ the sampling rate.

This leads for $\gamma_{10,4,0}$ to interval [85.4, 97.6] kHz with center frequency approximately 92 kHz, for $\gamma_{5,1,6}$ to interval [390.6, 781.3] kHz with center frequency approximately 586 kHz and for $\gamma_{9,6,0}$ to interval [97.7, 122.0] kHz with center frequency approximately 110 kHz.

The highest accuracy of the normalized mutual information justifies the use of this criterion in the proposed GIBFS genetic algorithm. We ran GIBFS with the number of generations set to 500. The results shown in Table 2 for GIBFS are the average test performances, from 10 fold cross-validation, averaged over all chromosomes in the last generation.

In Table 2, the highest accuracies were obtained with the naïve Bayes classifier: 86.81%. The Gaussian mixture model is a good second model because its accuracy was 85.25%.
It can be observed in Fig. 10 to Fig. 13 that GIBFS almost always results in higher accuracies than each of the filter approaches, no matter how many coefficients were used in the filters. Only in the case of the decision tree in Fig. 11, a few subsets were found with the relevance-redundancy approach that resulted in a slightly higher performance than GIBFS. However, this was only the case when the number of coefficients in the filter was higher than 20, which is the number of coefficients targeted in GIBFS. For an equal subset size of 20 coefficients, the performance of the filters is always lower than that for GIBFS.

To reveal the structure of the errors that were made, we computed the confusion matrices for the GIBFS algorithm for the subsets found at the 500th generation. The results for the k-nearest neighbor classifier, decision tree, Gaussian mixture model and naïve Bayes classifier are shown in Table 3 to Table 6, respectively.

<table>
<thead>
<tr>
<th></th>
<th>k-nearest neighbor</th>
<th>Decision tree J48</th>
<th>Gaussian mixture model</th>
<th>Naïve Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIBFS</td>
<td>76.80%</td>
<td>76.54%</td>
<td>85.25%</td>
<td>86.81%</td>
</tr>
</tbody>
</table>
Table 3
Confusion matrix for the subsets obtained with GIBFS using the k-nearest neighbors classifier. Accuracies are average test performances over the 20 chromosomes obtained in the last generation with a 10 fold cross-validation. Columns are normalized so that rows add up to 100% in each column.

<table>
<thead>
<tr>
<th>Predicted\True</th>
<th>Absence of corrosion</th>
<th>Uniform corrosion</th>
<th>Pitting</th>
<th>SCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absence of corrosion</td>
<td><strong>56.12 %</strong></td>
<td>20.77 %</td>
<td>6.92 %</td>
<td>9.66 %</td>
</tr>
<tr>
<td>Uniform corrosion</td>
<td>43.85 %</td>
<td><strong>79.23 %</strong></td>
<td>5.02 %</td>
<td>2.95 %</td>
</tr>
<tr>
<td>Pitting</td>
<td>0.03 %</td>
<td>0 %</td>
<td><strong>87.57 %</strong></td>
<td>4.24 %</td>
</tr>
<tr>
<td>SCC</td>
<td>0 %</td>
<td>0 %</td>
<td>0.49 %</td>
<td><strong>83.15 %</strong></td>
</tr>
</tbody>
</table>

Table 4
Confusion matrix for the subsets obtained with GIBFS using the decision tree J48 classifier. Accuracies are average test performances over the 20 chromosomes obtained in the last generation with a 10 fold cross-validation. Columns are normalized so that rows add up to 100% in each column.

<table>
<thead>
<tr>
<th>Predicted\True</th>
<th>Absence of corrosion</th>
<th>Uniform corrosion</th>
<th>Pitting</th>
<th>SCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absence of corrosion</td>
<td><strong>55.51 %</strong></td>
<td>23.68 %</td>
<td>6.85 %</td>
<td>5.24 %</td>
</tr>
<tr>
<td>Uniform corrosion</td>
<td>35.60 %</td>
<td><strong>74.28 %</strong></td>
<td>1.40 %</td>
<td>1.44 %</td>
</tr>
<tr>
<td>Pitting</td>
<td>5.74 %</td>
<td>1.60 %</td>
<td><strong>88.20 %</strong></td>
<td>6.61 %</td>
</tr>
<tr>
<td>SCC</td>
<td>3.15 %</td>
<td>0.44 %</td>
<td>3.55 %</td>
<td><strong>86.71 %</strong></td>
</tr>
</tbody>
</table>
Table 5
Confusion matrix for the subsets obtained with GIBFS using the **Gaussian mixture model**. Accuracies are average test performances over the 20 chromosomes obtained in the last generation with a 10 fold cross-validation. Columns are normalized so that rows add up to 100% in each column.

<table>
<thead>
<tr>
<th>Predicted\True</th>
<th>Absence of corrosion</th>
<th>Uniform corrosion</th>
<th>Pitting</th>
<th>SCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absence of corrosion</td>
<td><strong>64.95 %</strong></td>
<td>14.59 %</td>
<td>1.17 %</td>
<td>2.90 %</td>
</tr>
<tr>
<td>Uniform corrosion</td>
<td>32.97 %</td>
<td><strong>85.41 %</strong></td>
<td>1.37 %</td>
<td>0 %</td>
</tr>
<tr>
<td>Pitting</td>
<td>1.90 %</td>
<td>0 %</td>
<td><strong>93.11 %</strong></td>
<td>0.71 %</td>
</tr>
<tr>
<td>SCC</td>
<td>0.18 %</td>
<td>0 %</td>
<td>4.35 %</td>
<td><strong>96.39 %</strong></td>
</tr>
</tbody>
</table>

Table 6
Confusion matrix for the subsets obtained with GIBFS using the **naïve Bayes** classifier. Accuracies are average test performances over the 20 chromosomes obtained in the last generation with a 10 fold cross-validation. Columns are normalized so that rows add up to 100% in each column.

<table>
<thead>
<tr>
<th>Predicted\True</th>
<th>Absence of corrosion</th>
<th>Uniform corrosion</th>
<th>Pitting</th>
<th>SCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absence of corrosion</td>
<td><strong>64.21 %</strong></td>
<td>8.89 %</td>
<td>2.10 %</td>
<td>2.61 %</td>
</tr>
<tr>
<td>Uniform corrosion</td>
<td>32.76 %</td>
<td><strong>91.11 %</strong></td>
<td>1.31 %</td>
<td>0.02 %</td>
</tr>
<tr>
<td>Pitting</td>
<td>1.68 %</td>
<td>0 %</td>
<td><strong>94.23 %</strong></td>
<td>0.66 %</td>
</tr>
<tr>
<td>SCC</td>
<td>1.35 %</td>
<td>0 %</td>
<td>2.36 %</td>
<td><strong>96.71 %</strong></td>
</tr>
</tbody>
</table>
The confusion matrices show that the identification of each of the different corrosion types and the absence of corrosion is most accurate with the Gaussian mixture model and the naïve Bayes classifier compared to the knn classifier and the decision tree classifier. Comparing Table 6 with Table 2 to Table 5 reveals that the identification of SCC (96.71%), pitting (94.23%) and uniform corrosion (91.11%) is performed most accurately with the naïve Bayes classifier. The confusion of the absence of corrosion with uniform corrosion is consistently larger over all classifiers. Both the absence of corrosion and uniform corrosion were characterized in the experiments by a continuous-type emission. In fact, the accurate identification of SCC and pitting are the most important, because these localized forms of corrosion will lead to possible leaks sooner.

In our previous research [53], we have used electrochemical noise measurements (a non-destructive testing technique which monitors charge transfers between the surface of the metal and the electrolyte, see [71]) to identify uniform corrosion versus the absence of corrosion. Using this technique, uniform corrosion and the absence of corrosion could be distinguished with an accuracy of 91.5%. This suggests that monitoring systems could identify the localized forms of corrosion (pitting and SCC) with the acoustic emission technique, using GIBFS as the feature selection algorithm to achieve the highest accuracies, and uniform corrosion from the absence of corrosion with electrochemical noise measurements.

5. CONCLUSIONS

Wavelet basis functions are selected either based on filters or using a hybrid filter-wrapper genetic algorithm (GA). Wavelet packet coefficients that are selected with the local discriminant basis (LDB) algorithm lead to a slower increase in classification accuracy compared to mutual information-based approaches that take the redundancies between
wavelet coefficients into account. We compared the following three information theoretic filters: MI Parzen, MI knn and a relevance-redundancy criterion based on normalized mutual information. MI knn leads to higher accuracies than MI Parzen, but the highest classification accuracies are obtained for the normalized mutual information. We developed a steady-state genetic algorithm called GIBFS that uses a local improvement operator. An aspect of our genetic algorithm is that it enables us to combine filters and wrappers in a modular way. A filter of choice can be used as a local improvement operator and combined with a classification algorithm of choice. In this research, we have chosen the best filter as the local improvement operator in GIBFS and experimented with the following machine learning algorithms: k-nearest neighbors (knn), a decision tree (J48), the Gaussian mixture model (GMM) and the naïve Bayes classifier. The hybrid filter-wrapper genetic algorithm is able to select feature subsets that lead to higher classification accuracies compared to the filter-based algorithms.

The most important corrosion types occurring in the chemical process industry (uniform corrosion, pitting and stress corrosion cracking) can be identified accurately with the acoustic emission technique using wavelet packet coefficients as the extracted features. Among the four classification models tested here, the best results were obtained with the naïve Bayes classifier. This classifier, in combination with the feature subsets found with the GIBFS algorithm, was able to identify stress corrosion cracking with 96.71% accuracy, pitting with 94.23% accuracy, uniform corrosion with 91.11% accuracy, and the absence of corrosion with 64.21% accuracy. The largest confusion under the considered experimental conditions was observed between uniform corrosion and the absence of corrosion. The confusion between the absence of corrosion and uniform corrosion is less problematic because uniform corrosion is less likely to lead to a sudden plant shutdown than the localized forms of corrosion (pitting and stress corrosion cracking).
ACKNOWLEDGEMENTS

The authors are grateful to Prof. Dr. N. Saito, University of California, Davis, for providing the local discriminant basis selection algorithm and to Dr. M. Winkelmans for providing data from corrosion experiments. We are also grateful to Prof. Dr. M. Wevers for offering the opportunity to work on the problem of corrosion identification using the acoustic emission technique. GVD is supported by the CREA Financing (CREA/07/027) program of the K.U.Leuven. MMVH is supported by research grants received from the Excellence Financing program (EF 2005), the Belgian Fund for Scientific Research – Flanders (G.0588.09), the Interuniversity Attraction Poles Programme – Belgian Science Policy (IUAP P6/054), the Flemish Regional Ministry of Education (Belgium) (GOA 10/019), and the European Commission (IST-2007-217077). This work used the HPC (high-performance computing infrastructure) of the K.U.Leuven.

REFERENCES


**Figure Captions:**

Fig. 1. Absence of corrosion. The example of the absence of corrosion on the left was captured from stainless steel in CaCl$_2$ 40 weight% at 85°C environment. The example of the absence of corrosion on the right was captured from carbon steel NaOH 20 weight% + NaCl 3 weight% at 80°C environment.

Fig. 2. Uniform corrosion. The examples on the left and right are from continuous emissions during uniform corrosion of carbon steel in H$_3$PO$_4$ 10 weight% at environment temperature.

Fig. 3. Pitting. The left and right graphs are burst emission pitting signals captured from stainless steel in brackish water + FeCl$_3$ 1 weight% at 45°C environment.
Fig. 4. Stress corrosion cracking. The left SCC burst emission signal was captured from stainless steel in CaCl$_2$ 40 weight% at 85°C environment; the right SCC burst emission signal was captured from carbon steel Ca(NO$_3$)$_2$ 60 weight% at 105°C environment.

Fig. 5. Experimental set-up. 1) A steel probe is inserted in a by-pass of a chemical process installation. 2) The steel probe is exposed to the same environmental conditions (chemicals, temperature and pressure) as the process installation in the by-pass. 3) Input flow arriving from the installation. 4) Output flow guided back to the installation. 5) A broadband sensor captures the acoustic activity caused by corrosion processes occurring on the probe. 6) Signal conditioning by means of amplification and band-pass filtering.

In the machine learning part, the signals are divided into training and test sets. 7) Wavelet packet coefficients are extracted as the features from the training set signals. 8) The selection of the most informative basis functions based on the training set only, after which the test set signals are projected onto the selected basis functions. 9) The selected wavelet packet coefficients from the training set are used to train a classifier, the wavelet packet coefficients from the test set are used to test the classifier. The scheme applies to the filter based selection. The performance is estimated from the test set.

Fig. 6. Different subspaces in a wavelet packet decomposition up to scale index “i” equal to 3. The coefficients extracted in subspace $W^i_j$ are denoted by $\gamma_{i,j,k}$.

Fig. 7. Dependencies between the wavelet coefficients in the corrosion data set. The normalized mutual information (NMI), a value between 0 and 1, is used to compute the dependencies. A value equal to 0 means independent, a value equal to 1 means maximally dependent. Wavelet coefficients for the first 4 scales, i.e., for “i” = 0 to 3, are computed. The
coefficients from the different subspaces $W^j_i$ are concatenated as follows: $[W^0_0, W^0_1, W^1_0, W^2_0, \ldots W^3_0, W^3_1]$. Hence, coefficients with indices from 1 to 1024 are $\gamma_{0,0,0} \ldots \gamma_{0,0,1023}$. Coefficients with indices from 1025 to 2048 are $\gamma_{1,0,0} \ldots \gamma_{1,0,511}$, and so on. Dependencies are clearly present between the coefficients within subspace $W^0_0$ (NMI values in the block of the first 1024 rows and the first 1024 columns) and within the subspaces obtained from low-pass filtering, i.e., $W^1_0$ (a block consisting of rows from 1025 to 1536 and columns from 1025 to 1536), $W^2_0$ (a block consisting of rows from 2049 to 2304 and columns from 2049 to 2304), $W^3_0$ (a block consisting of rows from 3073 to 3200 and columns from 3073 to 3200). Dependencies are also obvious between coefficients of subspace $W^0_0$ and the low-pass filtered subspaces $W^1_0, W^2_0, W^3_0$. Consider, e.g., the dependencies between coefficients of $W^0_0$ and coefficients of $W^1_0$. These are in the block consisting of rows from 1 to 1024 and columns from 1025 to 1536.

Fig. 8. The proposed hybrid filter-wrapper steady-state genetic algorithm (GA). Two parents are selected in each generation of the steady-state approach. After mutation and crossover, a local improvement operator is applied to both children. This local improvement operator takes the statistical dependencies between wavelet coefficients into account by means of Equation (3.12).

Fig. 9. Evolution of the average upper bound, the average number of features, and the average lower bound for the GIBFS algorithm. The circles indicate the generation, where the average upper bound and the average lower bound both achieved 20 features.

Fig. 10. Evolution of the accuracy of the k-nearest neighbor classifier (k = 3) as a function of the number of wavelet coefficients selected with the LDB algorithm and the mutual
information filter algorithms. The horizontal line is the accuracy of the GIBFS algorithm at the 500th generation. The GIBFS algorithm achieves the highest accuracy compared to all filters and this over all number of wavelet coefficients considered in the filters.

Fig. 11. Evolution of the accuracy of the decision tree J48 classifier as a function of the number of wavelet coefficients selected with the LDB algorithm and the mutual information filter algorithms. The horizontal line is the accuracy of the GIBFS algorithm at the 500th generation. Only in the case of the relevance-redundancy filter was a slightly higher accuracy achieved compared to GIBFS, but that was only when the number of wavelet coefficients was larger than 20: it happened with 26, 39 and 40 selected wavelet coefficients.

Fig. 12. Evolution of the accuracy of the Gaussian mixture model classifier as a function of the number of wavelet coefficients selected with the LDB algorithm and the mutual information filter algorithms. The GIBFS algorithm achieves the highest accuracy compared to all filters and this over all number of wavelet coefficients considered in the filters.

Fig. 13. Evolution of the accuracy of the naïve Bayes classifier as a function of the number of wavelet coefficients selected with the LDB algorithm and the mutual information filter algorithms. The GIBFS algorithm achieves the highest accuracy compared to all filters and this over all number of wavelet coefficients considered in the filters.

Fig. 14. A scatter plot of the first 3 coefficients that was selected most often as a triplet in the 10 trainings sets of the 10 fold cross-validation. These are the coefficients $\gamma_{0,0,77}$, $\gamma_{0,0,78}$ and $\gamma_{0,0,79}$ in subspace $W_0^0$. 
Fig. 15. A scatter plot of the first 3 coefficients that are selected most often as a triplet with the normalized mutual information filter. These are the coefficients $\gamma_{10,4,0}$ (92 kHz center frequency), $\gamma_{5,1,6}$ (586 kHz center frequency) and $\gamma_{9,6,0}$ (110 kHz center frequency).