Mechanical Systems and Signal Processing 70-71 (2016) 1026–1037

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Classification of acoustic emission signals using wavelets and Random Forests : Application to localized corrosion

N. Morizet, N. Godin *, J. Tang, E. Maillet, M. Fregonese, B. Normand

INSA de Lyon, MATEIS Laboratory - UMR CNRS 5510. 7, Avenue Jean-Capelle 69621 Villeurbanne Cedex, France

ARTICLE INFO

Article history:
Received 2 July 2014
Received in revised form
17 September 2015
Accepted 19 September 2015
Available online 23 October 2015

Keywords:
Acoustic emission
Corrosion monitoring
Wavelets
Random Forests
Supervised classification
Machine learning

ABSTRACT

This paper aims to propose a novel approach to classify acoustic emission (AE) signals deriving from corrosion experiments, even if embedded into a noisy environment. To validate this new methodology, synthetic data are first used throughout an in-depth analysis, comparing Random Forests (RF) to the k-Nearest Neighbor (k-NN) algorithm. Moreover, a new evaluation tool called the alter-class matrix (ACM) is introduced to simulate different degrees of uncertainty on labeled data for supervised classification. Then, tests on real cases involving noise and crevice corrosion are conducted, by pre-processing the waveforms including wavelet denoising and extracting a rich set of features as input of the RF algorithm. To this end, a software called *RF-CAM* has been developed. Results show that this approach is very efficient on ground truth data and is also very promising on real data, especially for its reliability, performance and speed, which are serious criteria for the chemical industry.

1. Introduction

Acoustic Emission (AE) is the transient elastic sound waves produced when a material undergoes stress, caused by the release of localized stress energy. AE can typically be detected in frequency ranges within 50 kHz to 1 MHz, and one major application is health monitoring of structural materials (bridges, pressure containers, pipe lines, etc.). Different kind of evolving damages can be detected by AE technique, but special attention was paid to localized corrosion processes during the last two decades [1–5]. In these works, specific AE sources associated to corrosion damage were identified to be gas evolution (mainly H_2 produced by cathodic reactions), corrosion products formation and rupture, and stress corrosion cracking initiation and propagation. Localized corrosion phenomena, such as crevice corrosion, mainly affect passive metals and alloys in chemical and oil industries. Corrosion degrades the useful properties of materials and structures including strength, appearance and permeability to liquids and gases. Thus, real time detection and understanding the electrochemical processes involved in these phenomena are fundamental when it comes to implement a forward-looking strategy of operational maintenance of facilities. Over the past eighteen years, several studies have been conducted regarding the identification and the classification of different types of corrosion. In 1996, Barton et al. [6] developed an artificial neural network (ANN) to identify the onset and classify the type of localized corrosion from electrochemical noise (ECN) spectra. In 2004, Van Dijck et al. [7] presented a pattern recognition system to classify corrosion processes from ECN time series using the continuous wavelet transform (CWT), a Bayesian classifier and a genetic algorithm. In 2009, Piotrkowski et al. [8] applied

* Corresponding author.
E-mail address: nathalie.godin@insa-lyon.fr (N. Godin).

http://dx.doi.org/10.1016/j.ymssp.2015.09.025
0888-3270/

wavelet analysis (WA) and bispectrum analysis (BA) to AE signals for damage identification and evaluation of corroded galvanized steel whereas Griffin et al. [9] performed both Short-Time Fourier Transform (STFT) and Wavelet-Packet Transform (WPT) on AE signals extracted during burn and chatter anomalies, using genetic programming as a classifier algorithm. In 2010, Zhao et al. [10] classified AE signals in composite laminates using wavelet packet analysis (WPA) and support vector machine (SVM). In 2011, Van Dijck and Van Hulle [11] used a hybrid filter-wrapper genetic algorithm and a naïve Bayes classifier to identify the absence of corrosion, uniform corrosion, pitting and stress corrosion cracking. In 2012, Yu and Zhou [12] proposed a classification method of AE signals deriving from oil storage tank damage, combining SVM and an optimized grid search algorithm whereas Li et al. [13] studied the classification of AE signals of 304 stainless steel during stress corrosion process based on K-means clustering. Except the latter work, all of these researches are based on supervised learning algorithms but, to our knowledge, no attempt using decision trees has been made so far.

Considering the crevice corrosion process, emitted bubbles coming from chemical reactions generate AE activity, which can be recorded by sensors located on the surface of the specimen. Since AE signals associated to crevice corrosion are characterized by low energy content, it is very difficult to separate those signals from the environmental noise [14]. Thus, an in-depth work has been realized to preprocess the corresponding waveforms and a major motivation was to find the most relevant set of features. Chosen classification algorithm must be fast, reliable and not very sensitive to a mislabeled learning database (due to real-time and reliability industrial constraints). Moreover, it is preferable to provide a confidence level for the final decision.

This paper is organized as follows: waveform preprocessing and some details about the extracted features are given in Section 2. In Section 3, the RF algorithm is explained before an in-depth analysis is performed on ground truth data in Section 4. Classification results on real cases involving noise and crevice corrosion are shown in Section 5. Finally, some conclusions are drawn and improvements will be proposed.

2. Feature extraction from preprocessed waveforms

2.1. Waveform preprocessing

This important preliminary step is performed on waveforms directly acquired from sensors. The motivation here is to normalize those AE signals for consistent comparison. It is possible to discard useless information, numerically store the waveforms for further analysis and denoise them. The waveform preprocessing consists in the four following steps.

2.1.1. Pre-trigger removing

Pre-trigger removing simply deletes samples from the waveform corresponding to the very first points of the acquisition process. The pre-trigger value to be removed is totally customizable, depending on experiment conditions. This step is useful to remove digital noise from acquisition.

2.1.2. Tail cutting

Tail cutting resides in dynamically cutting the end of the waveform according to an energy criterion. For each point in the waveform, the cumulative energy computed from the beginning is compared to the energy contained in a 10 μs length window following that point. If this energy is less than a certain threshold T (in %) of the cumulative energy, then the corresponding point represents the end of the signal (Fig. 1).

This step is especially useful when it comes to remove the “zero-padding” which may have been applied at the end on some waveforms during the acquisition process (for length normalization purpose), thus removing useless information.

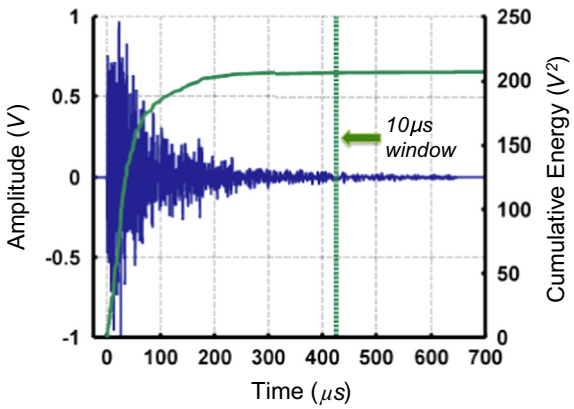

Fig. 1. Illustration of the tail cutting process on a waveform.

2.1.3. Shape preserving interpolation (SPI) resampling

SPI resampling is based on the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) technique [15,16] and the Weierstrass Approximation Theorem [17]. The motivation behind the use of this specific resampling technique is that, after tail cutting is applied, each waveform has a different number of points, thus it is not possible to numerically store them all in a matrix for further processing. Thanks to the SPI resampling, the original sampling frequency can be recovered and the full preprocessed waveforms can be stored.

2.1.4. Wavelet denoising

Wavelet denoising [18] can also be performed and uses the `wden` function from the *Matlab Wavelet Toolbox*. Specific parameters have been set: the universal threshold of Donoho [19] is used to select the wavelet coefficients in combination with a *soft thresholding* being rescaled using level dependent estimation of level noise. Decomposition is made at level 3 with the *symmlet8* as the mother wavelet.

An example of a full waveform preprocessing is depicted in Fig. 2.

2.2. Extracting features

The final decision process must be as fast and reliable as possible. Thus, each waveform is turned into a compact representation through a set of 30 features, in time, frequency and wavelet domains (Table 1). Besides common features such as *amplitude*, *duration*, *energy*, *rise time*, *partial powers* or *peak frequency*, other features derive from speech recognition and sound description studies [20]. Wavelet features are actually a specific set of features using wavelet packet energy [21,22]. The energy percentage of the terminal nodes of the wavelet packet tree is computed, leading to $2^3 = 8$ wavelet packet energy features.

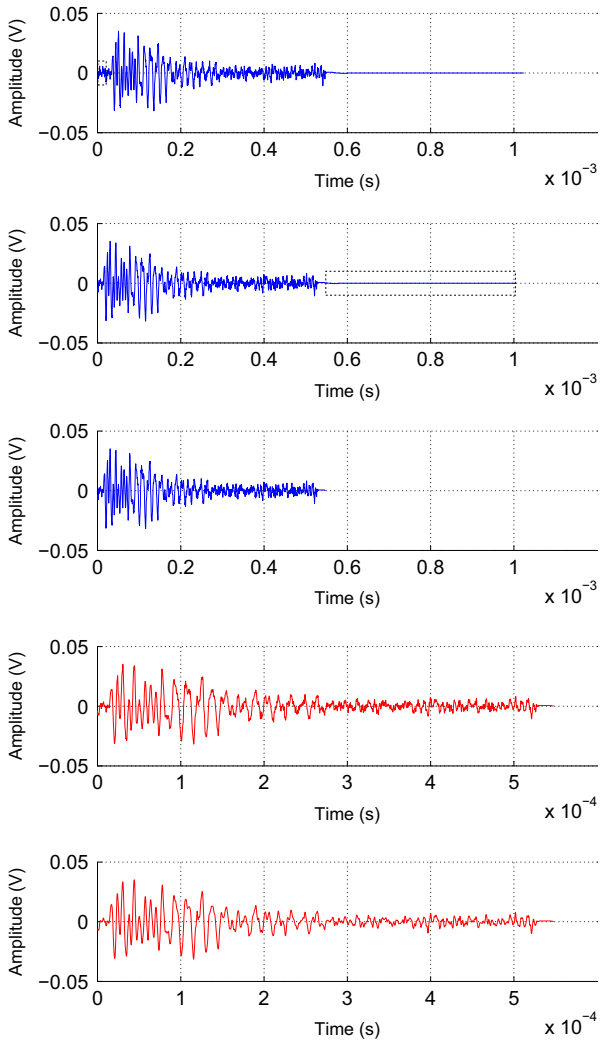

Fig. 2. From top to bottom: original, pre-trigger removed, tail cut, SPI resampled and final wavelet denoised waveforms.

Table 1
The set of the 30 features. Those features are recalculated from waveforms. "L" = Low-pass filter, "H" = High-pass filter. Thus, "LLH" consists in cascading two low-pass filters and one high-pass filter.

	ID	Feature	Unit
Time Features			
	R1	Amplitude	V
	R2	Duration	*s*
	R3	Energy	V^2
	R4	Zero-crossing rate	%
	R5	Rise time	s
	R6	Temporal centroid	s
	R7	Temporal decrease	Vs^{-1}
Frequency Features			
	R8	Partial Power 1 ([50; 100] kHz)	%
	R9	Partial Power 2 ([100; 200] kHz)	%
	R10	Partial Power 3 ([200; 250] kHz)	%
	R11	Partial Power 4 ([250; 400] kHz)	%
	R12	Frequency centroid	Hz
	R13	Peak frequency	Hz
	R14	Spectral spread	Hz
	R15	Spectral skewness	–
	R16	Spectral kurtosis	–
	R17	Spectral slope	Hz^{-1}
	R18	Roll-off frequency	Hz
	R19	Spectral spread to peak	Hz
	R20	Spectral skewness to peak	–
	R21	Spectral kurtosis to peak	–
	R22	Roll-on frequency	Hz
Wavelet Features			
	R23	Wavelet Packet Energy 1 (LLL)	%
	R24	Wavelet Packet Energy 2 (LLH)	%
	R25	Wavelet Packet Energy 3 (LHL)	%
	R26	Wavelet Packet Energy 4 (LHH)	%
	R27	Wavelet Packet Energy 5 (HLL)	%
	R28	Wavelet Packet Energy 6 (HLH)	%
	R29	Wavelet Packet Energy 7 (HHL)	%
	R30	Wavelet Packet Energy 8 (HHH)	%

3. Supervised classification using Random Forests

Random Forests are an ensemble learning method for supervised classification (and regression) that operate by constructing a multitude of decision trees during training, each capable of producing a response (*vote*) when presented with a new set of features during testing. The algorithm was originally developed by Leo Breiman and Adele Cutler in 2001 [23]. The term *Random Forests* is their trademark and comes from random decision forests that was first proposed by Tin Kam Ho of *Bell Labs* in 1995 [24]. The method merges two important ideas : Breiman's "bagging" [25] and the random split selection of features, introduced independently by Ho, Amit and Geman [26] and Dietterich [27] in order to construct a collection of decision trees with controlled variation. Bagging (bootstrap aggregation) consists in building multiple training subsets by sampling with replacement (also known as *bootstrapping* [28]) from the original training set. It reduces the variance of the ensemble and smoothes decision boundaries. One tree is constructed per bootstrap but only 2/3 of a bootstrap (the *In-Bag* data) are actually used to build the tree, the remaining 1/3 (the *Out-Of-Bag* (OOB) data) are used to get an unbiased estimate of the classification error of the corresponding tree (Fig. 3). The second randomness phenomenon is introduced through the random split selection idea : at each node of the tree, only a random subset of all features is considered. It makes tree construction less greedy and gives seemingly *weak* features a chance to get into the tree and to become helpful in conjunction with other features. During the testing phase, each AE signal is ran down each tree of the Forest, leading to *T* votes. The final decision can be obtained two different ways. The first one is simply the usual majority voting (*MV*) rule. In this work, another decision rule is introduced called security voting (*SV*) rule. In this special rule, one given AE signal is assigned to a specific class if more than 70% of the total number of trees voted for that class.

The whole approach combining the waveform preprocessing and the RF supervised classification has been implemented into the software *RF-CAM* ("Random Forests Classification for Acoustic emission Monitoring").

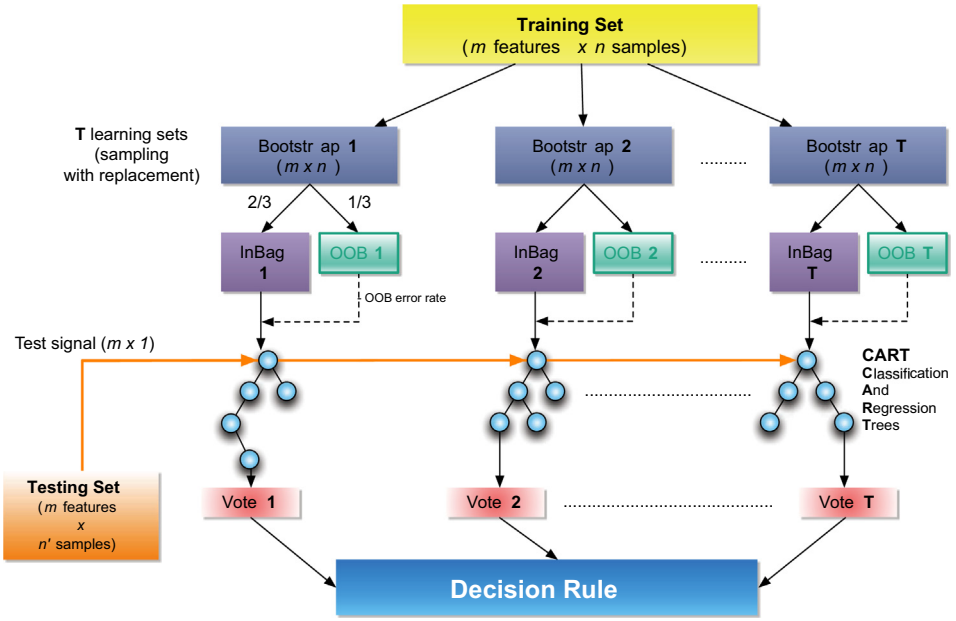

Fig. 3. Random Forests algorithm workflow.

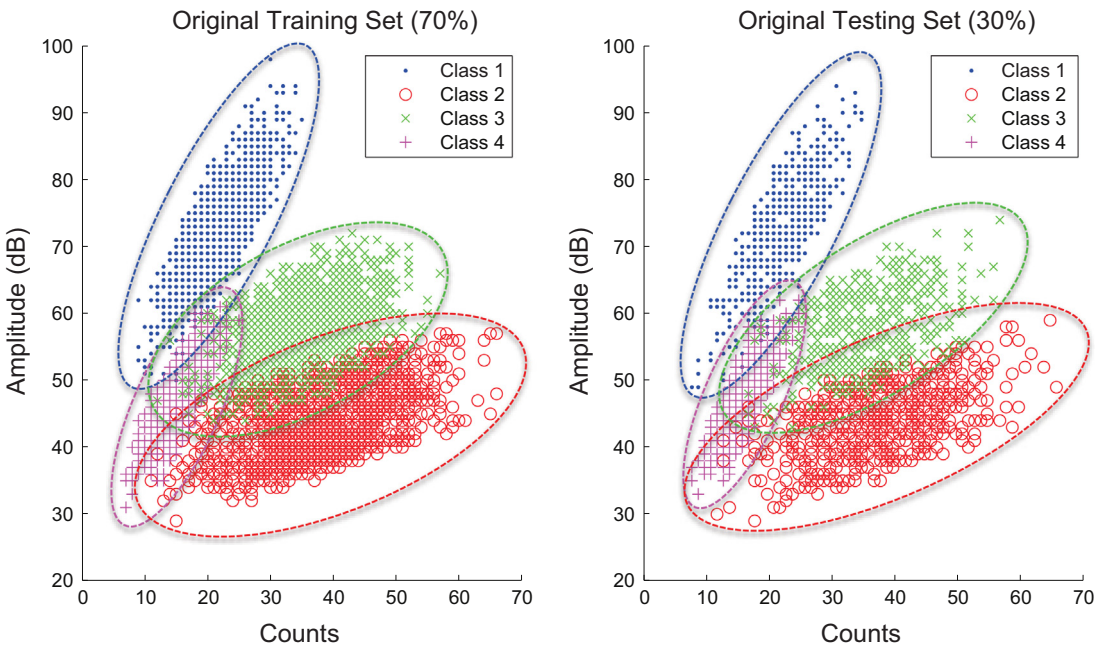

Fig. 4. Original Training and Testing Sets. Illustration of the data in the two-dimensional plane: threshold crossings (Counts) versus Amplitude (dB).

4. Ground truth data tests

4.1. The data

Ground truth data come from the synthetic dataset collected in [29]. Those data represent four clearly identified classes (2000 signals per class) and are described with a set of $M=9$ features. A Training Set is built, comprised of 70% of those data (5600 signals taken at random), the remaining 30% (2400 signals) constitute the Testing Set (Fig. 4).

4.2. The alter-class matrix (ACM)

In order to test the robustness of both algorithms regarding mislabeled data and the introduction of noise and outliers, a new evaluation tool called the *alter-class matrix* (ACM) is presented. The ACM is a particular n-square matrix designed to alter the original training set in order to simulate uncertainty on labeled data for supervised classification. It is based on a doubly stochastic matrix [30] and uses the 1-*norm scaling algorithm* from [31]. The ACM contains integer entries and has its sums along columns and rows equal to $S=100$ (see the following equation):

$$ACM_{(n,n)} = \begin{pmatrix} \mathbf{a_{1,1}} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & \mathbf{a_{2,2}} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & \mathbf{a_{n,n}} \end{pmatrix} \quad \sum_{i=1}^{n} a_{i,j} = \sum_{j=1}^{n} a_{i,j} = 100, \quad 1 < i \leq n, 1 < j \leq n \tag{1}$$

Each entry ($a_{i,j}$) of the ACM denotes a percentage of waveforms from the original class C_j. The diagonal values of the ACM are set according to a *trust factor* which reflect the confidence level given to each original class (a trust factor equals to 100 means the corresponding class is not altered). Also, the mean μ_{diag} and the standard deviation σ_{diag} of the diagonal values are imposed such that $\mu_{diag} = \pm 5\%$ of the trust factor and $\sigma_{diag} < 3$ respectively.

Eq. (2) shows an example of an ACM for a 4-class problem, given for a trust factor equals to 80 ($\mu_{diag} = 79, \sigma_{diag} = 2.94$):

$$ACM_{(4,4)} = \begin{pmatrix} \mathbf{79} & 2 & 15 & 4 \\ 7 & \mathbf{83} & 2 & 8 \\ 5 & 9 & \mathbf{76} & 10 \\ 9 & 6 & 7 & \mathbf{78} \end{pmatrix} \tag{2}$$

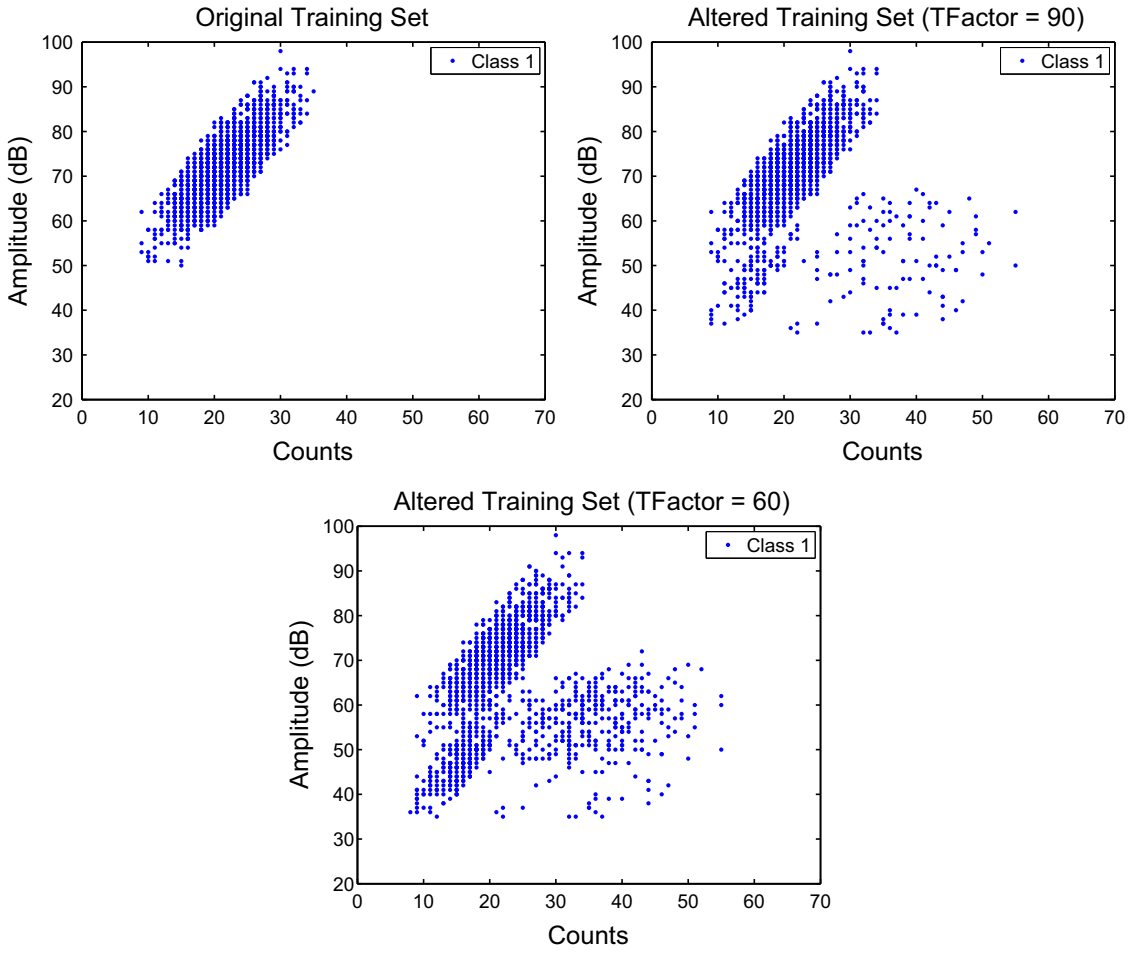

Fig. 5. Alteration of the class C1 from the Training Set, for different values of the *trust factor* (Counts vs. Amplitude (dB)).

The first row of $ACM_{(4,4)}$ must be read as follows: *C1* becomes $C1_{alter}$, composed of 79% of waveforms from *C1*, 2% from *C2*, 15% from *C3* and 4% from *C4*. This technique allows the simulation of different degrees of uncertainty regarding the labeled data of the original training set from one single matrix. Forcing the data to be mislabeled using the ACM can also be seen as a random and progressive introduction of noise and outliers in the original data (Fig. 5).

4.3. The tests

In all these tests, RF is compared to the widely used and efficient k-Nearest Neighbor (k-NN) algorithm [32]. The following parameters have been set for the algorithms:

- *Random Forests*: The number of trees $T=200$ has been set using the point where the *OOB error rate* stabilizes [23], the number of randomly selected features has been set to the recommended value for classification, *i.e.* $m=\lfloor\sqrt{M}\rfloor$, where M is the total number of features [23].
- *k-NN*: The optimal value of $k=15$ derives from the leave-one-out cross-validation (LOOCV) method [33], where the usual euclidean distance is used.

4.3.1. Cross-validation recognition rate

Recognition rates are computed for both RF and k-NN algorithms, for different values of the trust factor (Fig. 6). They correspond to the ratio of the number of predicted labels to the number of true labels. Globally, RF outperforms k-NN up to 10% and is less sensitive to a slightly mislabeled library (Table 2).

4.3.2. CPU time tests

To perform speed tests, we need a lot of data, especially features, which can only be obtained artificially. Focus must be made on the speed processing of the algorithms regarding the amount of data, not on the recognition rates. Thus, data sets are built from random matrices using the `randi(iMAX, N, M)` *Matlab* function. Entries are integers in [1, 100], the number of rows N (signals) and columns M (features) is variable. Training and testing sets are $(4000\times M)$ and $(N\times M)$ matrices respectively. There are four different classes, randomly assigned for each signal. The parameters for both RF and k-NN algorithms are set according to the beginning of Section 4.3. The computer used is a *Dell Vostro 640–Intel Core i7-2600* @3.40 GHz – *8Go RAM*, running *Windows 7 Pro (64 bits) SP1*. Results on those randomly generated data, for the testing phase, are depicted in Figs. 7 and 8.

k-NN is globally linearly sensitive to the increase of the number of waveforms and the number of features whereas RF is almost not influenced. During the testing phase, k-NN needs to compute many euclidean distances for each test sample

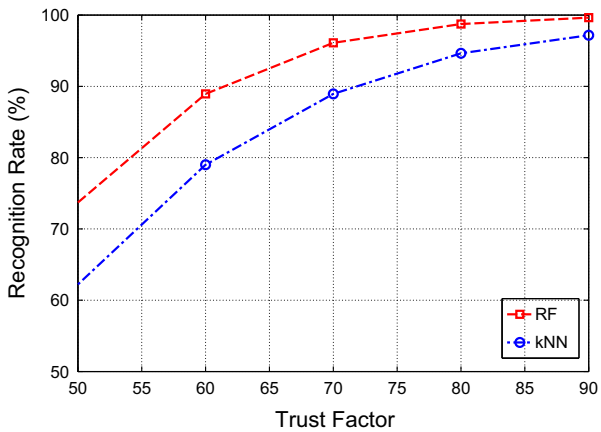

Fig. 6. Recognition rates (%) on synthetic data for different Trust Factor values.

Table 2
Recognition rates (%) on synthetic data for both RF and k-NN algorithms.

Algorithm	Trust Factor				
	60	70	80	90	100
RF	88.9	96.1	98.8	99.7	100.0
k-NN	79.0	88.9	94.6	97.2	99.2

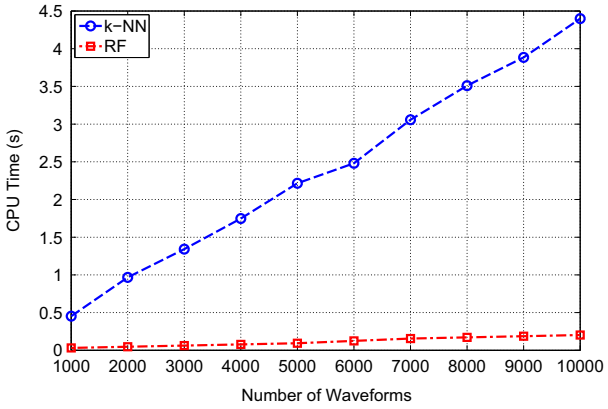

Fig. 7. CPU time for different numbers of waveforms (N varies from 1000 to 10 000, $M=9$ features).

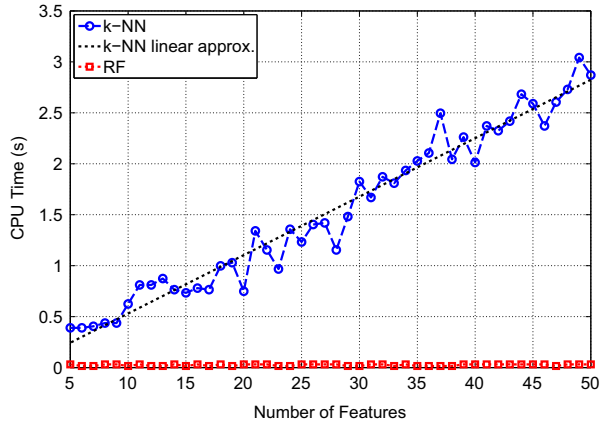

Fig. 8. CPU time for different numbers of features ($N=1000$ signals, M varies from 5 to 50).

before assigning the predicted label whereas RF does not need to compute anything, each test sample is simply ran through the Forest, hence the extremely fast decision process.

5. Application to real corrosion data

5.1. Corrosion experiments and monitoring

Experimental protocol to obtain crevice corrosion is described in detail in [14]. In the following, only the key points are reminded. 17 experiments are conducted on 304L stainless steel. Samples are immersed in a corrosive solution (with different values of NaCl (2 g/L or 35 g/L), pH (6.7, 8.3 or 10.5) and temperature 25 °C or 50 °C). In average, each corrosion experiment has been conducted twice under the same conditions. The pre-treatment (before immersion) of samples is performed step by step as follows: Grind to 400, rinse, chemical passivation (20 vol.% HNO_3 during 1 h, at room temperature), dry in air. Stainless steel sheet is then assembled by two formers made by polymethyl methacrylate (PMMA). This latter device allows two confined areas on both sides of the specimens in order to enhance crevice corrosion (Fig. 9). The open circuit corrosion potential (OCP) is continuously recorded owing to a saturated calomel electrode (SCE) as a reference (Fig. 10).

The AE acquisition system is a *Mistras AEDSP* embedded computer board. The sensors (R15) are applied on the surface of the specimen, outside the corrosive solution (distance sensors/sample = 40 mm) (Fig. 10). R15 sensors have been chosen for their good frequency sensitivity around 150 kHz. Sensor coupling is performed using vacuum grease. To ensure the repeatability of the results, assembly torque is controlled using a dynamometric key and set to 3 Nm. Acquisition parameters were set as follows: peak definition time (PDT) = 200 μs, hit definition time (HDT) = 400 μs and hit lockout time (HLT) = 200 μs. Acquisition threshold depends on environmental noise, thus varying from 19 dB to 28 dB. An optimal sampling frequency equals to 4 MSPS (and 4 K points per waveform) has been set as a very good compromise between the

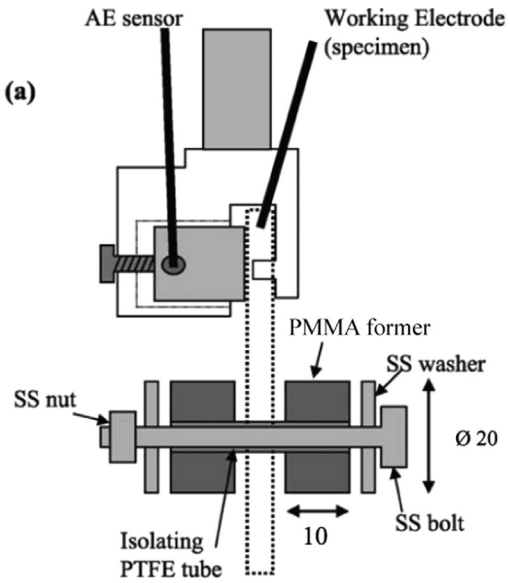

Fig. 9. Crevice assembly and AE sensor mounting (dimensions in mm, SS: stainless steel).

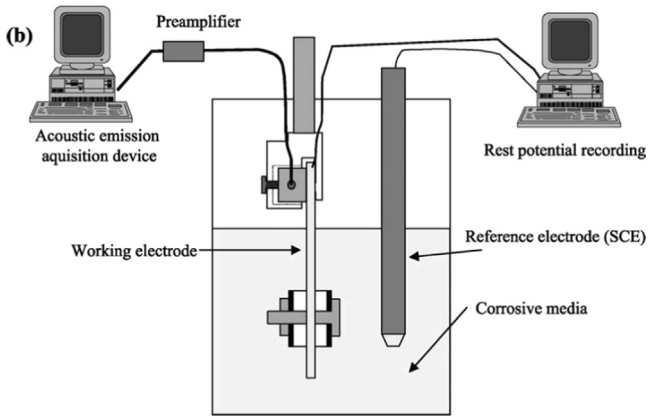

Fig. 10. Experimental device.

size of the data to process (real-time constraint) and the robustness of the extracted features (reliability constraint). Then, a set of 30 features are extracted from recorded waveforms (Table 1).

The open circuit corrosion potential (OCP) is continuously recorded owing to a saturated calomel electrode (SCE) as a reference (Fig. 10). This electrode is composed of metallic mercury (Hg) in contact with calomel ($Hg_2Cl_2(s)$), itself in equilibrium with a KCl saturated solution. When saturated, its potential is fixed and can serve as reference; it constitutes a half cell which allows measuring the OCP of the specimen which constitutes the second half cell.

Some experiments are performed with the addition of H_2O_2 in order to accelerate corrosion. The OCP drop shows that crevice initiates as soon as H_2O_2 is added (Figs. 11 and 12). For the sake of illustration, an experiment presenting no corrosion and performed at a higher temperature is depicted in Fig. 13.

5.2. Training and testing sets

In order to build various data sets, pH and temperature values, NaCl concentration and H_2O_2 addition are controlled to obtain crevice corrosion for some experiments and no corrosion for the others. 13 out of 17 experiments (almost 75%) are used for the training set. In absence of corrosion, OCP does not decrease (Fig. 13). However, even if there is no corrosion, some AE activity is observed all along the experiment. These signals are attributed to noise (i.e. dilatation phenomena and bubbles evolution within the liquid due to the temperature of the test (50 °C)). A first class (denoted as NC) is built from 1200 signals. Regarding the experiments involving crevice corrosion (Fig. 11), AE activity starts prior to the addition of H_2O_2, and is mainly assigned to noise. After H_2O_2 addition, gathered signals are mainly assigned to corrosion and constitute a second class (denoted as CC), composed of 1167 signals. It has to be noticed that this CC class is not *pure* and also contains

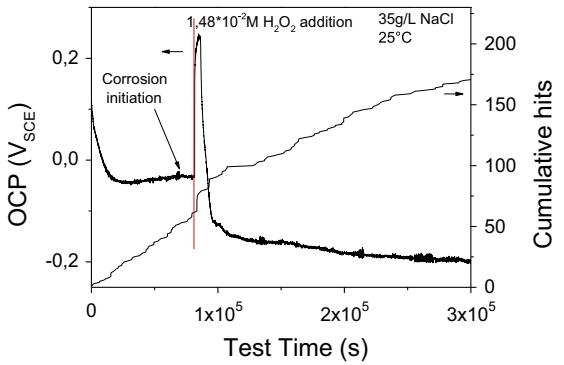

Fig. 11. OCP and AE monitoring of crevice corrosion in 35 g/L NaCl, accelerated by 1.48×10^{-2} MH_2O_2, pH6.7, 25 °C.

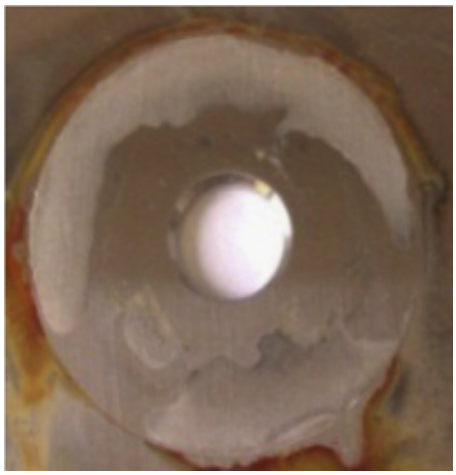

Fig. 12. Illustration of the corresponding crevice corrosion phenomenon on the sample.

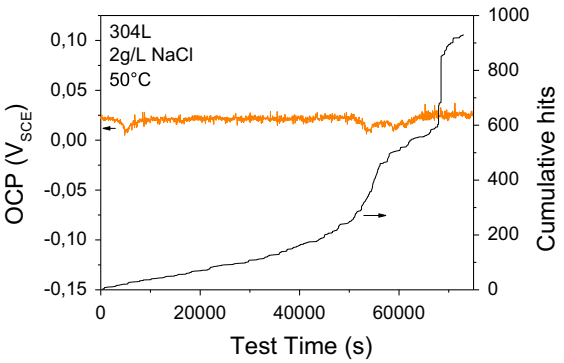

Fig. 13. OCP and AE monitoring of a specimen without corrosion in 2 g/L NaCl, 50 °C.

noise signals. The previous study on the use of the ACM for altered data shows that recognition results are satisfactory even if classes are altered up to 20%. Besides, since there is *no class-privilege* in real conditions, a special attention has been paid to build a well-balanced training set, for a total of 2367 waveforms. The remaining four experiments are used to construct the different testing sets. Two of them present no corrosion (namely NCa and NCb), the other two show crevice corrosion (namely CCa and CCb), for a total of 1311 waveforms. Preprocessing and features extraction have been applied to all waveforms, according to Section 2.

Table 3
RF classification results of the proposed method for NC and CC experiments.

Experiment	Number of votes (%) **NC** class	Number of votes (%) **CC** class	Total number of signals (%)	Decision rule
NCa	179 (76.8%)	54 (23.2%)	233 (100%)	MV
	103 (92.8%)	8 (7.2%)	111 (100%)	**SV**
NCb	459 (95.6%)	21 (4.4%)	480 (100%)	MV
	410 (98.3%)	7 (1.7%)	417 (100%)	**SV**
CCa	152 (34.9%)	283 (65.1%)	435 (100%)	MV
	89 (29.3%)	**215 (70.7%)**	304 (100%)	**SV**
CCb	69 (42.3%)	94 (57.7%)	163 (100%)	MV
	40 (37%)	**72 (64.3%)**	112 (100%)	**SV**

5.3. Classification results

The percentage of signals for the Majority Voting (MV) rule is given regarding the original total number of signals of the test set. The percentage of signals for the Security Voting (SV) rule is given regarding the number of remaining signals, after the security threshold of 70% is applied.

For each test, the proper majority class has been recognized and results show that using SV leads to a reinforcement of the usual MV decision, when it comes to assign signals to a specific class. Moreover, for each test case, most of the signals corresponding to the minority class is discarded, thus strengthening the trend of the majority class. For instance, considering the first no corrosion test (NCa), the percentage of rejected signals for the NC class is $(179-103)/179 = 42.5\%$ whereas the percentage of rejected signals for the CC class is $(54-8)/54 = 85.2\%$ (Table 3).

6. Conclusions

A precise waveform preprocessing, including wavelet denoising, has been performed on acquired waveforms, leading to normalized and *cleaned* signals. A new evaluation tool called the *alter-class matrix* (ACM) has been introduced to test the robustness to mislabeled classes and speed tests have been conducted on both RF and k-NN algorithms. Then, application to real corrosion data have been executed and classification results have been obtained with the software *RF-CAM*, developed for this work. RF is pretty well suited with regard to the facing issues, it is very fast and less sensitive to a learning library which may have been built incorrectly.

The results associated to the usual majority voting (MV) rule are satisfactory in terms of finding the proper majority class. In order to take into account the industrial reliability constraint, another decision rule called *security voting* (SV) has been implemented as a confidence level (set to 70%), and reinforces the final decision process taken by the MV. In average, signals deriving from the minority class are twice as discarded as signals coming from the majority class.

Future prospects to be considered are the enlargement of the learning library in order to identify other corrosion mechanisms such as pitting corrosion. Finally, this methodology which has been developed and validated at the laboratory scale, can be applied at the industrial scale, provided that a consistent new learning library is constructed.

Acknowledgments

The authors wish to thank the French Ministry of Economy, Finance and Industry alongside the *AXELERA* competitiveness cluster for their financial support, within the FUI *IREINE* project (“Innovation for the REliability of INdustrial Equipments”). The authors are also grateful to *Solvay*, *IFPEN*, *Arkema* and *Mistras* Group industrial partners for their technical collaboration. The authors are also thankful to R. Di Folco for his technical support regarding the corrosion experiments.

References

[1] H. Mazille, R. Rothea, C. Tronel, An acoustic emission technique for monitoring pitting corrosion of austenitic stainless steels, Corros. Sci. 37 (1995) 1365–1375.
[2] M. Fregonese, H. Idrissi, H. Mazille, L. Renaud, Y. Cetre, Monitoring pitting corrosion of aisi 316l austenitic stainless steel by acoustic emission technique: choice of representative acoustic parameters, J. Mater. Sci. 36 (2001) 557–563.
[3] Y. Kim, M. Fregonese, H. Mazille, D. Feron, G. Santarini, Study of oxygen reduction on stainless steel surfaces and its contribution to acoustic emission recorded during corrosion processes, Corros. Sci. 48 (2006) 3945–3959.
[4] C. Jomdecha, A. Prateepasen, P. Kaewtrakulpong, Study on source location using an acoustic emission system for various corrosion types, NDT & E Int. 40 (2007) 584–593.
[5] C. Jirarungsatian, A. Prateepasen, Pitting and uniform corrosion source recognition using acoustic emission parameters, Corros. Sci. 52 (2010) 187–197.
[6] T.F. Barton, D.L. Tuck, D.B. Wells, The identification of pitting and crevice corrosion spectra in electrochemical noise using an artificial neural network, in: ASTM Special Technical Publication, vol. 1277, 1996, pp. 157–169.

[7] G. Van Dijck, M. Wevers, M.M. Van Hulle, Corrosion time series classification using the continuous wavelet transform and mml density estimation, in: International Conference on Computational Intelligence, 2004, pp. 39–43.
[8] R. Piotrkowski, E. Castro, A. Gallego, Wavelet power, entropy and bispectrum applied to ae signals for damage identification and evaluation of corroded galvanized steel, Mech. Syst. Signal Process. 23 (2009) 432–445.
[9] J. Griffin, X. Chen, Multiple classification of the acoustic emission signals extracted during burn and chatter anomalies using genetic programming, Int. J. Adv. Manuf. Technol. 45 (2009) 1152–1168.
[10] J. Zhao, K. Wang, G. Xin, Acoustic emission signals classification based on support vector machine, in: 2010 2nd International Conference on Computer Engineering and Technology (ICCET), vol. 6, 2010, pp. 300–304.
[11] G. Van Dijck, M.M. Van Hulle, Genetic algorithm for informative basis function selection from the wavelet packet decomposition with application to corrosion identification using acoustic emission, Chemom. Intell. Lab. Syst. 107 (2011) 318–332.
[12] Y. Yu, L. Zhou, Acoustic emission signal classification based on support vector machine, TELKOMNIKA : Indones. J. Electr. Eng. 10 (2012) 1027–1032.
[13] J. Li, G. Du, C. Jiang, S. Jin, The classification of acoustic emission signals of 304 stainless steel during stress corrosion process based on k-means clustering, Anti-Corros. Methods Mater. 59 (2012) 76–80.
[14] Y. Kim, M. Fregonese, H. Mazille, D. Feron, G. Santarini, Ability of acoustic emission technique for detection and monitoring of crevice corrosion on 3041 austenitic stainless steel, NDT & E Int. 36 (2003) 553–562.
[15] Mathworks, Matlab pchip function documentation, 2013.
[16] F.N. Fritsch, R.E. Carlson, Monotone piecewise cubic interpolation, SIAM J. Numer. Anal. 17 (1980) 238–246.
[17] M. Stone, The generalized weierstrass approximation theorem, Math. Mag. 21 (1948) 167–184.
[18] D. Donoho, I. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika 81 (1994) 425–455.
[19] D. Donoho, De-noising by soft-thresholding, IEEE Trans. Inf. Theory 41 (1995) 613–627.
[20] G. Peeters, A large set of audio features for sound description (similarity and classification) in the CUIDADO project, Technical Report, IRCAM, 2004.
[21] R. Coifman, Y. Meyer, S. Quake, V. Wickerhauser, Signal processing and compression with wavelet packets, in: J.S.B. et al. (Ed.), Wavelets and Their Applications, vol. 442, Springer, 1994, pp. 363–379.
[22] Mathworks, Wavelet packets analysis documentation, 2013.
[23] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[24] T.K. Ho, Random decision forests, in: Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 1, IEEE, 1995, pp. 278–282.
[25] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[26] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 832–844.
[27] T. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization, Mach. Learn. 40 (2000) 139–157.
[28] B. Efron, Bootstrap methods: another look at the jackknife, Ann. Stat. 7 (1979) 1–26.
[29] A. Sibil, N. Godin, M. R'Mili, E. Maillet, G. Fantozzi, Optimization of acoustic emission data clustering by a genetic algorithm method, J. Nondestruct. Eval. 31 (2012) 169–180.
[30] M. Marcus, Some properties and applications of doubly stochastic matrices, Am. Math. Mon. 67 (1960) 215–221.
[31] P.A. Knight, D. Ruiz, B. Ucar, A symmetry preserving algorithm for matrix scaling, Technical Report, INRIA, 2012.
[32] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1967) 21–27.
[33] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Stat. Surv. 4 (2010) 40–79.