WELD DISCONTINUITIES CLASSIFICATION USING PRINCIPAL COMPONENT ANALYSIS AND SUPPORT VECTOR MACHINE

Giovanni Alfredo Guarneri*, Flávio Neves Junior*, Lúcia Valéria Ramos de Arruda*

*UTFPR - Av. Sete de Setembro, 3165
Curitiba, PR, Brasil

Emails: giovanni@utfpr.edu.br, neves@utfpr.edu.br, lvrarruda@utfpr.edu.br

Abstract— The present work proposes an automatic ultrasonic signal classification system to classify weld discontinuities in weld beads attaching metal plates of carbon steel. This system will be embedded into an autonomous robot and it must operates in real time. The classification system has three components: a preprocessing algorithm ultrasonic raw data conditioning; a Principal Component Analysis algorithm to extract features from preprocessed ultrasonic signals; and a Support Vector Machine classifier. This system is trained and tested with a signal database from A-scan ultrasonic simulator. This database contains signals with lack of fusion welds and longitudinal cracks discontinuities besides defect-free welds. A performance comparison with a Multilayer Perceptron based weld classification is carried out. As a result, the SVM based classification system performs out MLP results around 6% for discontinuities welds.

Keywords— Pattern recognition, support vector machine, nondestructive tests and ultrasonic technique.

Resumo— Este trabalho propõe um sistema automático de classificação de sinais de ultrassom para identificar descontinuidades em cordões de solda que unem placas de aço-carbono. Este sistema será embarcado em um robô autômono e deve operar em tempo real. O sistema de classificação tem três componentes: um algoritmo de pré-processamento para adequar os sinais de ultrassom para os demais componentes; um algoritmo de Análise de Componentes Principais para a extração de características dos sinais de ultrassom pré-processados; e uma Máquina de Vetores de Suporte como classificador. O sistema é treinado e testado com uma base de dados composta por sinais de A-scan para as descontinuidades de falta de fusão e trincas longitudinais, além de sinais para soldas sem defeito. Estes sinais de A-scan foram gerados por um programa de simulação de testes de ultrassom. É feita uma comparação no desempenho da classificação deste sistema com outro sistema que utiliza um classificador baseado em redes neurais artificiais to tipo *Multilayer Perceptron*. Tal comparação demonstra que o desempenho da classificação das descontinuidades melhorou em torno de 6%.

Palavras-chave— Reconhecimento de padrões, máquinas de vetores de suporte, ensaios não-destrutivos e técnicas de ultrassom.

1 Introduction

The pulse-echo is the most commonly technique used to nondestructive testing (NDT) of welded structures, mainly for its simplicity and efficiency (Veiga et al., 2005). In this technique, a piezoelectric ultrasonic transducer is used to generate ultrasonic waves which propagate through metal plate. Thus, they are reflected by defect and return back to transducer surface. The same ultrasonic transducer receives the reflected waves and converts them to electrical signals. These signals, called A-scan signals, contain information about type, size and orientation of weld defects (Sambath et al., 2011).

The identification of weld defects on an A-scan signal is a pattern recognition problem. It is usually called *ultrasonic pattern recognition* (Song et al., 2002). This problem is not a trivial task and it must be done by experts. The results depend on the experience and knowledge of those experts (Seyedtabaii, 2012). Nevertheless, experts are subject to mistake the identification. Thus, an automatic ultrasonic signal classification system (AUTSCS) to assist the inspector in that analysis may improve the reliability of the inspection process (Veiga et al., 2005).

Some researchers (Polikar et al., 1998; Song et al., 2002; Sambath et al., 2011; Seyedtabaii, 2012) define three major tasks to such systems: acquisition and preprocessing A-scan raw data; feature extraction; and classification. The preprocessing task normalizes the raw data signals values to a suitable range for other tasks. The feature extraction task is responsible to obtain attributes from normalized A-scan signals. These attributes must be able to characterize the flaws. Finally, the classification task aims to analyze the attributes from an A-scan signal and indicates one from a known set of flaws. The algorithm that implements a classification task is called *classifier*. There are several approaches to build a classifier, such as traditional statistical classifiers, rule-base classifiers and learning-base classifiers, which a typical example are artificial neural networks (Song et al., 2002; Moura et al., 2004). Recently, new approaches to build classifiers has been used (Vieira et al., 2008; Papa et al., 2012).

This work proposes an AUTSCS to classify defects in weld beads attaching metal plates of carbon steel. This AUTSCS is capable to identify two classes of defects: lack of fusion (LF) and longitudinal cracks (LC). Moreover, it also identifies defect-free welds (ND). This AUTSCS will be embedded into an autonomous robot which will inspect weld beads of storage tanks of oil and derivates. For that, the AUTSCS must be able to classify defects in real time. The AUTSCS has the following components: a preprocessing algorithm that normalizes the A-scan raw data; a Principal Component Analysis algorithm to feature extraction; and a learning-base classifier, using Support Vector Machine (SVM) algorithm. The knowledge database used to train and test the AUTSCS is obtained with a simulation software. This software is a ultrasonic testing called *simSUNDT*, developed at the Dept. of Mechanics at Chalmers University of Technology (Bovik and Bostrom, 1997; Wirdelius, 2004; Persson and Wirdelius, 2010). To evaluate the AUTSCS, its performance is compared to another AUTSCS, which uses the same algorithms for preprocessing and feature extraction, but uses a MultiLayer Perceptron (MLP) classifier.

This paper is organized as follows. Section 2 discusses how the knowledge database was assembled by using *simSUNDT* software. Section 3 describes the methods for building the AUTSCS architectures. Results obtained are presented in Section 4 and concluding remarks are presented in Section 5.

2 Knowledge database

2.1 *Weld discontinuities*

The American Welding Society (AWS) defines a weld discontinuity as an interruption of the typical structure of a weldment (AWS, 2000). It is considered to be a flaw when it does not meet any requirement of a particular specification. Depending on the nature of their appearance, the weld discontinuities can be divided into three groups (AWS, 2000): (1) procedures/process discontinuities, (2) metallurgical discontinuities and (3) base metal discontinuities. This work was limited to the study of only two types of discontinuities, lack of fusion and longitudinal cracks, both belonging to the class (1).

During the welding procedure, a lack of fusion discontinuity appears when the weld metal does not melt properly with the base metal. It can also be called incomplete fusion. According AWS (2000), normally they occur at interface between weld metal and base metal (sidewall) or adjoining weld beads (Figure 1). The main causes of its appearance are insufficient welding current, lack of access to all faces of the weld joint and insufficient preweld cleaning.

The cracks discontinuities are define as a result of localized stresses that exceeds the ultimate strength of the material (AWS, 2000). They can be divided into *hot cracks*, which occur during solidification, or *cold cracks* which occur in ambient temperature when the weldment has been placed in service. They can also be divided by its location, which may be in the weld metal and on the base metal.

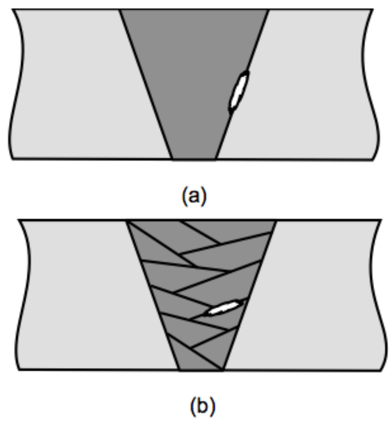

Figure 1: Lack of fusion examples (a) at sidewall and (b) adjoining weld beads

This research is limited to the study of cold cracks which occur both in weld metal and base metal along the weld bead (longitudinal cracks). These cracks are mainly caused by material fatigue.

2.2 *Software* simSUNDT

The software *simSUNDT* is a program which simulates the whole ultrasonic NDT procedure, generating the A-scan signals for defects insert into a weld bead. It is described in Wirdelius (2004). The simulator's mathematical model is three dimensional and the ultrasonic probe scans the object on a rectangular mesh. It can simulates tree types of planar defects: rectangular cracks; circular cracks and strip-like cracks. The lack of fusion defects were parameterized as *circular cracks* and longitudinal cracks were parameterized as *strip-like cracks*. The ultrasonic probe was parameterized with a circular shape (diameter of 5mm). The software was configured for shear waves of 60° sonic beam incidence in the metal. The frequency spectrum of electrical exciting pulse has *cossine square* shape, with central frequency set to 1.5 MHz and bandwidth set to 100%. The software can simulate the weld bead and the corresponding back scattered grain noise that is superposed to the defect signal that correlates better to a real inspection situation (Persson and Wirdelius, 2010). The weld metal and base metal were set to steel. The geometry of weld used in this research are showed in Fig. 2.

2.3 *A-scan signal database*

The A-scan signal database was built with the largest gain signal of a *simSUNDT* simulation section for LF and LC flaw classes. For this, we performed 100 simulation sessions for each type of defect. For defect-free welds, we taken all A-scan signals with gain greater than -30 dB (0 dB

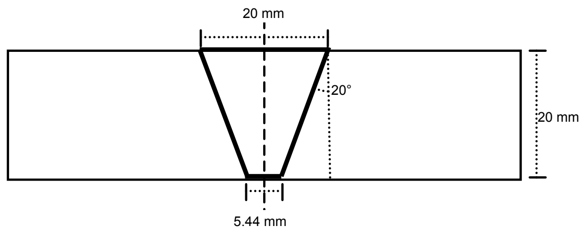

Figure 2: Weld geometry

is the gain of A-scan signal generated by simulator calibration process). This procedure gives us an A-scan signal database with 100 A-scan signals for lack of fusion defects, 100 A-scan signals for longitudinal cracks defects and 73 A-scan signals for defect-free welds. The raw A-scan signals produced in the simulator sessions are sampled at 50 MHz, from 0 to 50 μs. Therefore, all these signals have 2501 samples. Figure 3 shows an A-scan example for each defect.

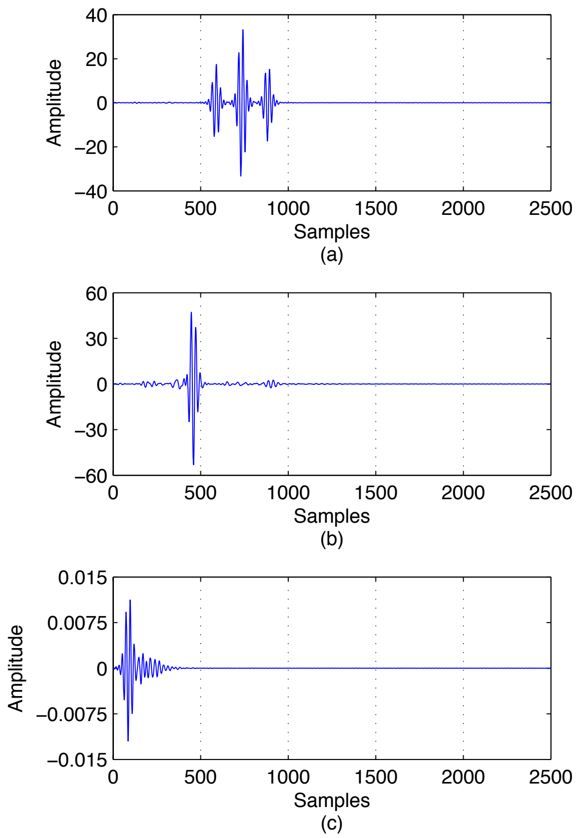

Figure 3: A-scan signals examples: (a) LF, (b) LC and (c) defect-free weld

3 Method

As previously mentioned in Sect. 1, an AUTSCS executes three main tasks: acquisition and preprocessing of raw data, feature extraction and classification. This section describes the methods applied in these three tasks to build the AUTSCS used in this research.

3.1 Preprocessing raw data

The knowledge database described in Sect. 2 is formed by raw A-scan signals. These are high-frequency modulated signals whose envelopes have information about defects. A method for envelope extract is based on Hilbert Transform. For this, it was used the `hilbert` function of MATLAB®. Figure 4a shows the envelope signal of the A-scan raw signal of Fig. 3a.

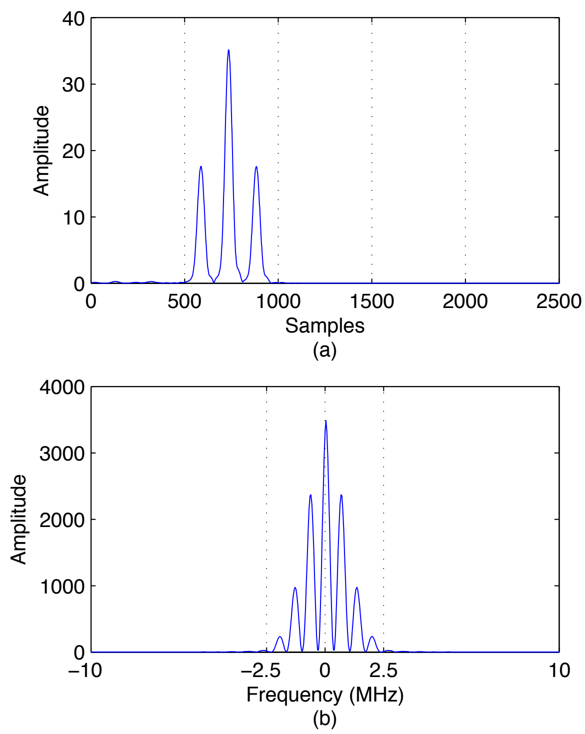

Figure 4: (a) A-scan envelope signal and (b) the envelope signal frequency spectrum

As showed in Fig. 4b (A-scan envelope signal frequency spectrum), the bandwidth of envelope signals are approximately 5 MHz. Thus, these signals can be downsampling from 50 MHz to 5 MHz without loss information and its size decreases from 2501 to 251 samples. Finally, the envelope A-scan signals amplitude are normalized for maximum and minimum amplitude value by taking the highest amplitude value to 1 and the lowest value to 0.

3.2 Feature extraction

The feature extraction is a processing which an *input space* is transformed to a *feature space* with reduced dimensionality (Haykin, 1998). The transformation is designed such that the feature data set keeps the information content of input data set. From an statistical point of view, the PCA is an effective technique for reducing dimensionality. In fact, PCA is an invertible linear transformation which maximizes variance decreasing rate. The resulting vector components are orthogonal ones.

In this work, the classifiers were trained and tested with two input data sets. The first input data set is the normalized A-scan envelope signals database, processed as described in Sect. 3.1. This data set will be called from now as Raw Data Set (RDS). In this set, each input data vector has 251 dimensions. The second input data set is the result of PCA processing on the RDS and will be called from now as PCA Data Set (PCADS). This PCA processing was performed by `processpca` function of MATLAB®. We used `maxfrac=0.01` as argument to `processpca` function. With this value, `processpca` function eliminates those principal components that contribute less than 1% to the total variation in the Raw Data Set. As a result, the dimension of input data vectors are reduced from 251 to 19.

3.3 SVM classifiers

The Support Vector Machines is a category of universal feedforward network which can be used as classifiers, like Multilayer Perceptrons and Radial Basis Functions networks (Haykin, 1998). The SVM learning process is based on statistical learning theory proposed by Vapnik (1995). From this theory, SVM classifiers can provide a good generalization performance.

The idea of a SVM is find an optimal hyperplane which separates positive and negative examples of a training data set with a maximum margin of separation (Haykin, 1998). Thus, given a finite training set of data-label pairs $(\mathbf{x}_i, y_i), i = 1, ..., N$ with size N, where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}^N$, the SVM solves the following optimization problem (Haykin, 1998):

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ \text{and} \quad & \xi_i \geq 0. \end{aligned} \tag{1}$$

In the Eq. 1, $\phi(\bullet)$ is a function which maps input data vector into a higher dimension space. The vectors $\phi(\mathbf{x}_i)$ in this space may be linearly separable according Cover's theorem. The separation hyperplane is defined by $\mathbf{w}^T\phi(\mathbf{x}) + b = 0$, where $\mathbf{w}$ is the vector (into the higher dimension space) normal to the hyperplane and b is a bias. The ξ_i are called *slack variables* which measure the derivation of the data points from their ideal condition of pattern separability (Haykin, 1998) and C is a regularization parameter determined experimentally by the user.

From optimization theory, Eq. 1 defines the *primal problem* and deals with linear constraints and a convex cost function. However, this problem can be rewrite using Lagrange multipliers (Haykin, 1998), in pattern recognition systems corresponding to the *dual problem*:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N}\alpha_i \\ & - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j(\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)) \\ \text{subject to} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \\ \text{and} \quad & 0 \leq \alpha_i \leq C, \forall i = 1, ..., N. \end{aligned} \tag{2}$$

It is not necessary to know $\phi(\bullet)$ function to solve Eq. 2. The term $\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)$ is a inner-product and its calculation results to a scalar value. Thus, only it is necessary to know how to calculate $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)$. The $K(\mathbf{x}_i, \mathbf{x}_j)$ is called *inner-product kernel.* The kernel function must follow the Mercer's theorem (Haykin, 1998).

In the present work, the SVM classifiers were implemented using the LIBSVM library and its interface to MATLAB®. The quadratic problem of SVM is solve in LIBSVM using an Sequential Minimal Optimization (SMO) like decomposition method. This algorithm is described in Chang and Lin (2011). The kernel function used was RBF one ($K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2), \gamma > 0$). To obtain the best values to C and γ was used the grid-search procedure with multifold cross-validation method, described by Haykin (1998, p.239). The grid-search procedure has used the LIBSVM cross-validation internal mode configured for 3-folds.

3.4 Multilayer Perceptron classifiers

The MLP are the most widely studied and used ANN classifier (Haykin, 1998; Zhang, 2000). The MLP classifiers used in this work were implemented using the Neural Network Toolbox™ of MATLAB®. To obtain a suitable number of neurons from the hidden layer the 3-fold multifold cross-validation method was used for model selection (Haykin, 1998). The databases used to training and test the MLP classifiers were the same used to SVM classifiers. The MLP classifiers had used these parameters: log-sigmoid transfer function, Scaled Conjugate Gradient backpropagation training algorithm (`trainscg`) and performance goal was set to 0.01. Was choose the MLP classifier with the best test success classification rate (TSCR).

4 Results

The results presented in this section were obtained by the same procedure. All classifiers were tested 20 times with the same data set. For each training/testing cycle the data set (RDS or PCADS) was partitioned into training set (182 input vectors) and testing set (91 input vectors). All values presented are the mean values. The test success classification rate (TSCR) is the relation between the number of correctly predict data and the total testing data.

After the grid-search procedure to obtain the best values to C and γ, the TSCR mean values obtained for SVM classifiers are shown in Tab. 1.

Table 1: Best SVM classifiers models chosen by cross-validation method.

Classifier Model	TSCR mean (%)			
	LF	ND	LC	Over All
SVM (RDS)	85.23	100.0	89.02	90.58
SVM (PCADS)	95.19	100.0	85.77	93.01

After cross-validation method for model selection of MLP classifiers, the best MLP model selected for RDS has 251 neurons at input layer, 9 neurons at first hidden layer, 8 neurons at second hidden layer and 3 neurons at output layer (251-9-8-3). The best MLP model selected for PCADS has 19 neurons at input layer, 6 neurons at first hidden layer, 4 neurons at second hidden layer and 3 neurons at output layer (19-6-4-3). The TSCR mean values obtained for MLP classifiers are presented in Tab. 2.

Table 2: Best MLP classifiers models chosen by cross-validation method.

Classifier Model	TSCR mean (%)			
	LF	ND	LC	Over All
251-9-8-3 (RDS)	86.93	100.0	79.61	87.72
19-6-4-3 (PCADS)	87.23	99.80	87.34	90.64

By observing the values of Tab. 1 and Tab. 2, we can assert:

- the SVM classifiers have improved classification performance comparing with MLP classifiers. It confirms that a SVM classifier provide a better generalization performance than a MLP classifier for this problem;
- the PCA feature extraction has improved classification performance too. The increase in classification performance using PCA feature extraction was around 3%;
- the SVM classifier trained with RDS had the same classification performance that the MLP classifier trained with PCADS;
- the combination PCA feature extraction plus SVM classifier have improved classification performance around 6% comparing to MLP classifier without feature extraction.

The Tab. 3 and Tab. 4 show the confusion tables to SVM and MLP classifiers. Both classifiers did not have showed false negative results, but the

Table 3: Confusion table of SVM classifier (RDS).

Classifier output			
LF	ND	LC	Target
85.23	1.5	13.28	LF
0	100	0	ND
10.98	0	89.02	LC

Table 4: Confusion table of MLP classifier (RDS).

Classifier output			
LF	ND	LC	Target
86.93	2.71	10.37	LF
0	100	0	ND
14.7	5.7	79.61	LC

MLP classifier has showed 2.71% and 5.7% false positive results to LF and LC defects, respectively. The SVM classifier has showed better results, only 1.5% false positive result to LF defect.

Table 5: Elapsed times for tasks execution.

	PCA algorithm	Training algorithm	Testing process
MLP (RDS)	0s	2.821s	0.042s
MLP (PCADS)	0.092s	1.047s	0.042s
SVM (RDS)	0s	0.015s	0.009s
SVM (PCADS)	0.092s	0.006s	0.0006s

The Tab. 5 shows the elapsed times to execute the PCA feature extraction algorithm, the training algorithm and the test process for all AUTSCS tested in this work, excluding the time to execute the grid-search procedure of SVM parameters. These tests were performed using a PC with Intel® Core 2 Duo (2.4 GHz) processor and 8 GB RAM. As a result, the SVM classifiers have performed training and testing processes much faster comparing to the MLP classifiers. The elapsed time to execute PCA feature extraction algorithm was equal for both classifiers. Its value is very small compared to the elapsed time of training process for the MLP classifier, but it is relatively high when compared to the elapsed time of training process for SVM classifier. This fact could be impeditive to use PCA feature extraction algorithm in a real time system, but as the absolute value of elapsed time is small (90ms), other factors need be analyzed to validate the use of the algorithm PCA.

5 Conclusions

In present study, we were built an AUTSCS to identify lack of fusion, longitudinal cracks and defect-free welds through pulse-echo ultrasonic signals. The proposed AUTSCS will be embedded into an autonomous robot to perform real time flaw detection. The AUTSCS has used the following preprocessing techniques: envelope signal extraction from A-scan signal, downsampling from 50MHz to 5 MHz and normalization of amplitude values between 0 and 1. The feature extraction method used in AUTSCS was the PCA and the classifier was a SVM. The knowledge database used to train and test the AUTSCS was built using a simulation software for ultrasonic testing (*simSUNDT*). This AUTSCS got a success classification rate of 93%. When this AUTSCS was compared with a classic AUTSCS which no feature extraction and with a MLP classifier, the classification performance was improved around 6%. Regarding the elapsed time to execute some AUTSCS processes, the SVM classifier was much faster than MLP classifier, which reinforces the choice of the SVM classifier for this system. The PCA feature extraction algorithm contributed positively in reducing the training time of the MLP classifier. However, it did not occur in the SVM classifier because the decrease in training process time was shorter than the PCA algorithm processing time. Nevertheless, only the PCA feature extraction contributed to improve the classification performance on both classifiers around 3%. Thus, this study demonstrates that the use of PCA feature extraction method in combination with a SVM classifier leads to fair results of classification performance and significant improvement in the elapsed time to execute the tasks for training and use of the classifier.

Acknowledgment

We would like to thank Professor Håkan Wirdelius, Ph. D, director of Scientific Centre of Non-Destructive Testing (SCENDT), for providing us the software *simSUNDT*, plus information about it which allowed to read the results of the simulations in MATLAB®.

References

AWS (2000). *Welding Inspection Handbook (3rd Edition)*, American Welding Society (AWS).

Bovik, P. and Bostrom, A. (1997). A model of ultrasonic nondestructive testing for internal and subsurface cracks, *The Journal of the Acoustical Society of America* **102**(5): 2723–2733.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology* **2**: 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*, 2nd edn, Prentice Hall PTR, Upper Saddle River, NJ, USA.

Moura, E. P., Silva, R. R., Siqueira, M. H. S. and Rebello, J. M. A. (2004). Pattern recognition of weld defects in preprocessed tofd signals using linear classifiers, *Journal of Nondestructive Evaluation* **23**: 163–172. 10.1007/s10921-004-0822-4.

Papa, J. P., Falcao, A. X., De Albuquerque, V. H. C. and Tavares, J. M. R. (2012). Efficient supervised optimum-path forest classification for large datasets, *Pattern Recognition* **45**(1): 512–520.

Persson, G. and Wirdelius, H. (2010). Recent survey and application of the *simSUNDT* software, *AIP Conference Proceedings* **1211**(1): 2125–2132.

Polikar, R., Udpa, L., Udpa, S. and Taylor, T. (1998). Frequency invariant classification of ultrasonic weld inspection signals, *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on* **45**(3): 614 –625.

Sambath, S., Nagaraj, P. and Selvakumar, N. (2011). Automatic defect classification in ultrasonic NDT using artificial intelligence, *Journal of Nondestructive Evaluation* **30**: 20–28.

Seyedtabaii, S. (2012). Performance evaluation of neural network based pulse-echo weld defect classifiers, *Measurement Science Review* **12**(5): 168–174.

Song, S.-J., Kim, H.-J. and Cho, H. (2002). Development of an intelligent system for ultrasonic flaw classification in weldments, *Nuclear Engineering and Design* **212**(1-3): 307 – 320.

Vapnik, V. N. (1995). *The nature of statistical learning theory*, Springer-Verlag, New York.

Veiga, J. L. B. C., Carvalho, A. A. d., Silva, I. C. d. and Rebello, J. M. A. (2005). The use of artificial neural network in the classification of pulse-echo and TOFD ultra-sonic signals, *Journal of the Brazilian Society of Mechanical Sciences and Engineering* **27**: 394 – 398.

Vieira, A., de Moura, E., Gonçalves, L. and Rebello, J. (2008). Characterization of welding defects by fractal analysis of ultrasonic signals, *Chaos, Solitons & Fractals* **38**(3): 748 – 754.

Wirdelius, H. (2004). *User Guide to simSUNDT*, CHALMERS LINDHOLMEN HÖGSKOLA AB.

Zhang, G. (2000). Neural networks for classification: a survey, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **30**(4): 451–462.