Automatic Segmentation of the Secondary Austenite-phase Island Precipitates in a Superduplex Stainless Steel Weld Metal

Victor H. C. Albuquerque[1] & Rodrigo Y. M. Nakamura[2] & João P. Papa[2]
Cleiton C. Silva[3] & João Manuel R. S. Tavares[4]

[1]*Universidade de Fortaleza, Centro de Ciências Tecnológicas, Fortaleza, Brazil*
[2]*Departamento de Computação, UNESP - Univ Estadual Paulista, Bauru, Brazil*
[3]*Departamento de Engenharia Metalúrgica e Materiais, Universidade Federal do Ceará, Fortaleza, Brazil*
[4]*Universidade do Porto, Faculdade de Engenharia, Porto, Portugal*

Duplex and superduplex stainless steels are class of materials of a high importance for engineering purposes, since they have good mechanical properties combination and also are very resistant to corrosion. It is known as well that the chemical composition of such steels is very important to maintain some desired properties. In the past years, some works have reported that γ_2 precipitation improves the toughness of such steels, and its quantification may reveals some important information about steel quality. Thus, we propose in this work the automatic segmentation of γ_2 precipitation using two pattern recognition techniques: Optimum-Path Forest (OPF) and a Bayesian classifier. To the best of our knowledge, this if the first time that machine learning techniques are applied into this area. The experimental results showed that both techniques achieved similar and good recognition rates.

1 INTRODUCTION

Duplex and superduplex stainless steels are a important class of materials for engineering, which have an exceptional corrosion resistance and good mechanical properties combination (Nilsson 1992). The success of these alloys is associated to the microstructural balance of phases, in which ferrite and austenite have approximately the same proportions. All these characteristics have motivated the use of duplex and super-duplex stainless steels in a wide variety of industrial sectors, such as chemical ones, petrochemical and oil & gas (Tavares et al. 2010; Bastos et al. 2007).

The balance of phases is influenced by the chemical composition of the alloys, and also by the cooling rate experimented during its production (Hemmer and Grong 1999; Hemmer et al. 2000). However, depending on the manufacturing process, this proportion can be changed and then the properties degraded. One of the most important processes used in the manufacturing and repairing of pipes and equipments for industrial applications is the welding, in which the steel is subjected to a high cooling rate. The high temperature reached during the welding cycle causes the austenite dissolution, and consequently one may observe an increasing in the ferrite content, harming the toughness and ductility (Kotecki and Hilkes 1994; Hertzman et al. 1997). In multipass welding, the reheated zone by deposition of subsequent weld beads causes, as main microstructural changes, the dissolution of chromium nitrides and also the precipitation of secondary austenite (γ_2) (Ramirez et al. 2004; Ramirez et al. 2003).

Some works have reported that γ_2 precipitation improves the toughness of the duplex and super-duplex stainless steels (Lippold and Al-Rumaih 1997; Lee et al. 1999). On the other hand, the low chromium, molybdenium and nitrogencontents of the γ_2 are harmful to corrosion resistance (Nilsson and Wilson 1993; Nilsson et al. 1995). Based on these aspects, it is very important to quantify the amount of γ_2 in welded joints, especially in fusion zone, in order to improve the weld quality. However, this quantification is not straightforward, mainly because the secondary austenite formed is more evident when the precipitates are located inside the ferrite grain, with needles shape and also with the presence of γ_2 islands. Thus, the quantification of such islands is usually carried out by manual operations using all purpose image analysis softwares, demanding a long time and user experience.

In this paper, we propose the automatic segmentation of γ_2 islands using machine learning techniques, focusing on the Optimum-Path Forest (OPF) (Papa, Falcão, and Suzuki 2009) and a Bayesian classifier (Duda, Hart, and Stork 2000). As far as we know, this is the first time that OPF is applied into this domain, as well as any other computational technique, once that these precipitates have never been automatically segmented up to date.

The remainder of the paper is organized as follows. Section 2 revisits the classifiers, and Section 4 discuss the experimental results. Finally, Section 5 states the conclusions.

2 MACHINE LEARNING BACKGROUND

This section addresses a review about the pattern recognition techniques applied.

2.1 Optimum-path Forest classifier

The OPF classifier works by modeling the problem of pattern recognition as a graph partition in a given feature space. The nodes are represented by the feature vectors and the edges connect all pairs of them, defining a full connectedness graph. This kind of representation is straightforward, given that the graph does not need to be explicitly represented, allowing us to save memory. The partition of the graph is carried out by a competition process between some key samples (*prototypes*), which offer optimum paths to the remaining nodes of the graph. Each prototype sample defines its optimum-path tree (OPT), and the collection of all OPTs defines de optimum-path forest, which gives the name to the classifier (Papa, Falcão, and Suzuki 2009).

The OPF can be seen as a generalization of the well known Dijkstra's algorithm to compute optimum paths from a source node to the remaining ones (Dijkstra 1959). The main difference relies on the fact that OPF uses a set of source nodes (prototypes) with any path-cost function. In case of Dijkstra's algorithm, a function that summed the arc-weights along a path was applied. For OPF, we used a function that gives the maximum arc-weight along a path, as explained before.

Let $Z = Z_1 \cup Z_2$ be a dataset labeled with a function λ, in which Z_1 and Z_2 are, respectively, a training and test sets such that Z_1 is used to train a given classifier and Z_2 is used to assess its accuracy. Let $S \subseteq Z_1$ a set of prototype samples. Essentially, the OPF classifier creates a discrete optimal partition of the feature space such that any sample $s \in Z_2$ can be classified according to this partition. This partition is an optimum path forest (OPF) computed in $\Re^n$ by the image foresting transform (IFT) algorithm (Falcão, Stolfi, and Lotufo 2004).

The OPF algorithm may be used with any *smooth* path-cost function which can group samples with similar properties (Falcão, Stolfi, and Lotufo 2004). Particularly, we used the path-cost function f_{max}, which is computed as follows:

$$f_{max}(\langle s \rangle) = \begin{cases} 0 & \text{if } s \in S, \\ +\infty & \text{otherwise} \end{cases}$$

$$f_{max}(\pi \cdot \langle s, t \rangle) = \max\{f_{max}(\pi), d(s, t)\}, \quad (1)$$

in which $d(s, t)$ means the distance between samples s and t, and a path π is defined as a sequence of adjacent samples. As such, we have that $f_{max}(\pi)$ computes the maximum distance between adjacent samples in π, when π is not a trivial path.

The OPF algorithm assigns one optimum path $P^*(s)$ from S to every sample $s \in Z_1$, forming an optimum path forest P (a function with no cycles which assigns to each $s \in Z_1 \setminus S$ its predecessor $P(s)$ in $P^*(s)$ or a marker nil when $s \in S$. Let $R(s) \in S$ be the root of $P^*(s)$ which can be reached from $P(s)$. The OPF algorithm computes for each $s \in Z_1$, the cost $C(s)$ of $P^*(s)$, the label $L(s) = \lambda(R(s))$, and the predecessor $P(s)$.

The OPF classifier is composed of two distinct phases: (i) training and (ii) classification. The former step consists, essentially, into finding the prototypes and computing the optimum-path forest, which is the union of all OPTs rooted at each prototype. After that, we pick a sample from the test sample, connect it to all samples of the optimum-path forest generated in the training phase and we evaluate which node offered the optimum path to it. Notice that this test sample is not permanently added to the training set, i.e., it is used only once. The next sections describe in more detail this procedure.

2.1.1 Training

We say that S^* is an optimum set of prototypes when OPF algorithm minimizes the classification errors for every $s \in Z_1$. S^* can be found by exploiting the theoretical relation between minimum-spanning tree (MST) and optimum-path tree for f_{max} (Allène, Audibert, Couprie, Cousty, and Keriven 2007). The training essentially consists in finding S^* and an OPF classifier rooted at S^*.

By computing an MST in the complete graph (Z_1, A), we obtain a connected acyclic graph whose nodes are all samples of Z_1 and the arcs are undirected and weighted by the distances d between adjacent samples. The spanning tree is optimum in the sense that the sum of its arc weights is minimum as compared to any other spanning tree in the complete graph. In the MST, every pair of samples is connected by a single path which is optimum according

to f_{max}. That is, the minimum-spanning tree contains one optimum-path tree for any selected root node.

The optimum prototypes are the closest elements of the MST with different labels in Z_1 (i.e., elements that fall in the frontier of the classes). By removing the arcs between different classes, their adjacent samples become prototypes in S^* and OPF can compute an optimum-path forest with minimum classification errors in Z_1. Note that, a given class may be represented by multiple prototypes (i.e., optimum-path trees) and there must exist at least one prototype per class.

2.1.2 Classification

For any sample $t \in Z_2$, we consider all arcs connecting t with samples $s \in Z_1$, as though t were part of the training graph. Considering all possible paths from S^* to t, we find the optimum path $P^*(t)$ from S^* and label t with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t) \in S^*$. This path can be identified incrementally by evaluating the optimum cost $C(t)$ as:

$$C(t) = \min\{\max\{C(s), d(s,t)\}\}, \forall s \in Z_1. \quad (2)$$

Let the node $s^* \in Z_1$ be the one that satisfies Equation 2 (i.e., the predecessor $P(t)$ in the optimum path $P^*(t)$). Given that $L(s^*) = \lambda(R(t))$, the classification simply assigns $L(s^*)$ as the class of t. An error occurs when $L(s^*) \neq \lambda(t)$.

2.2 Bayesian Classifier

Let $p(\omega_i|x)$ be the probability of a given pattern $x \in \Re^n$ to belong to class ω_i, $i = 1, 2, \ldots, c$, which can be defined by the Bayes Theorem (Jaynes 2003):

$$p(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)}, \quad (3)$$

where $p(x|\omega_i)$ is the probability density function of the patterns that compose the class ω_i, and $P(\omega_i)$ corresponds to the probability of class the ω_i itself.

A Bayesian classifier decides whether a pattern x belongs to the class ω_i when:

$$p(\omega_i|x) > p(\omega_j|x), \ i,j = 1,2,\ldots,c, \ i \neq j, \quad (4)$$

which can be rewriten as follows by using Equation 3:

$$p(x|\omega_i)P(\omega_i) > p(x|\omega_j)P(\omega_j), \ i,j = 1,2,\ldots,x, \ i \neq j \quad (5)$$

As one can see, the Bayes classifier's decision function $d_i(x) = p(x|\omega_i)P(\omega_j)$ of a given class ω_i strongly depends on the previous knowledge of $p(x|\omega_i)$ and $P(\omega_i)$, $\forall i = 1, 2, \ldots, c$. The probability values of

$P(\omega_i)$ are straightforward and can be obtained by calculating the histogram of the classes, for instance.

However, the main problem is to find the probability density function $p(x|\omega_i)$, given that the only information we have is a set of patterns and its corresponding labels. A common practice is to assume that the probability density functions are Gaussian ones, and thus one can estimate their parameters using the dataset samples (Duda, Hart, and Stork 2000). In the n-dimensional case, a Gaussian density of the patterns from class ω_j can be calculated by:

$$p(x|\omega_i) = \gamma exp\left[-\frac{1}{2}(x-\mu_j)^T C_i^{-1}(x-\mu_j)\right], \quad (6)$$

in which

$$\gamma = \frac{1}{(2\pi)^{n/2} \mid C_i \mid^{1/2}}, \quad (7)$$

and μ_i and C_i stand for, respectively, to the mean and the covariance matrix of class ω_i. These parameters can be obtained by considering each pattern x that belongs to class ω_i using:

$$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x \quad (8)$$

and

$$C_i = \frac{1}{N_i} \sum_{x \in \omega_i} (xx^T - \mu_i\mu_i^T), \quad (9)$$

in which N_i means the number of samples from class ω_i.

3 MATERIALS AND METHODS

In order to evaluate the performance of the machine learning algorithms for automatic identification of the secondary austenite islands in microstructure of superduplex stainless steels, multipass welds were performed using gas metal arc welding process (GMAW). A testing bench with an industrial robot and an electronic welding power supply was used to produce the sample. The alloy used was the UNS S32750 (SAF 2507) superduplex stainless steel pipes with 19 mm thickness as base metal and as filler metal was AWS ER 2594.

Samples for metallographic evaluation were extracted from welded joints and conventionally prepared through mechanical grinding and polishing using silicon carbide sandpaper and diamond past, respectively. An electrochemical etching to reveal the microstructure was carried out using an aqueous solution with 40% vol. of nitric acid (HNO_3) and applying a potential of 2.0 V during 40 seconds.

We used optical microscopy images with $200\times$ and $1000\times$ of magnifications. These images were previously labeled by a technician into positive (γ_2 islands)

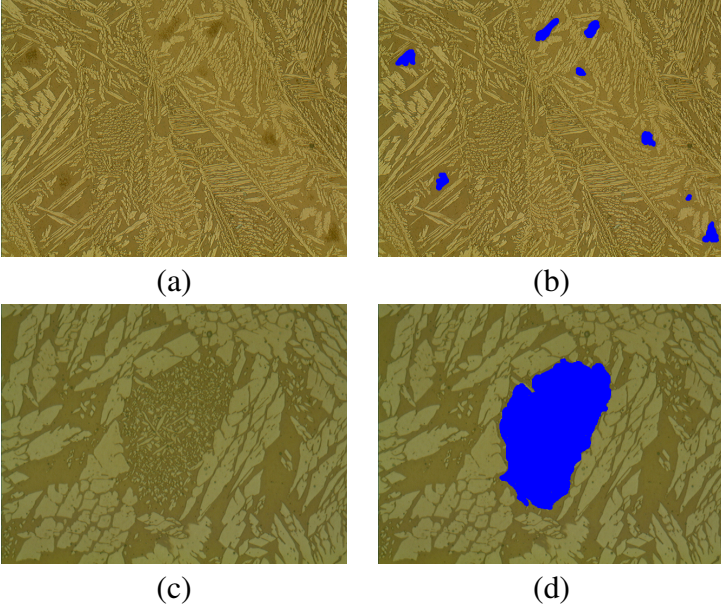and negative (background) samples. Figure 1 displays these images.

Figure 1: Microscopic images used in the experiments: original images with magnifications of (a) $200\times$ and (c) $1000\times$, and the respectively manual segmentations in (b) and (d).

Now, imagine an interactive classification tool in which the user can select same positive and negative samples in order to classify the remaining image. After that, the use may want to refine the classification process by marking another set of samples, and then to execute the process again. In most applications, one know that the effectiveness of classification is strongly related with the training set size, since we have more information to train the classifier. In this work, we would like to simulate this user behavior by randomly selecting some samples for training, and then to classify the remaining image. The percentages used for training were: 30% and 50%.

In this work, each pixel to be classified was described by a texture kernel around its neighborhood and also by its gray value. In order to extract texture information, we applied the Gabor filter (Feichtinger and Strohmer 1997), which can be mathematically formulated as follows:

$$G(x,y,\theta,\gamma,\sigma,\lambda,\psi) = e^{\frac{x'^2+y'^2\sigma^2}{2\sigma^2}} \cos\left(2\pi\frac{x'}{\lambda}+\psi\right), \quad (10)$$

where $x' = x\cos(\theta) + y\sin(\theta)$ and $y' = x\sin(\theta) + y\cos(\theta)$. In the above equation, λ means the sinusoidal factor, θ represents the orientation angle, ψ is the phase offset, σ is the Gaussian standard deviation and γ is the aspect spatial ratio.

The main idea of Gabor filter is to perform a convolution between the original image I and $G_{\theta,\gamma,\sigma,\lambda,\psi}$ in order to obtain a Gabor-filtered representation as:

$$\hat{I}_{\theta,\gamma,\sigma,\lambda,\psi} = I * G_{\theta,\gamma,\sigma,\lambda,\psi}, \quad (11)$$

in which $\hat{I}_{\theta,\gamma,\sigma,\lambda,\psi}$ denotes the filtered image. Thus, one can obtain a filter bank of Gabor filtered images by varying its parameters. We used a convolution filter of size 3×3 with the following Gabor parameters that were empirically chosen and based on our previous experience:

- 6 different orientations: $\theta = 0°$, $45°$, $90°$, $135°$, $225°$ and $315°$;

- 3 spatial resolutions: $\lambda = 2.5$, 3 and 3.5. Notice that, for each one of λ values, we applied different values for σ, say that $\sigma = 1.96$, 1.40 and 1.68;

- $\psi = 0$ and

- $\gamma = 1$.

Once we get the Gabor-filtered images (one can see that we have $6 \times 3 = 18$ images), we then compute the texture features at pixel p as the set of corresponding gray values among these images. Thus, each pixel is described by 19 features, being 18 of them related with texture and the remaining one is the original gray value. After classification process, we applied a 3×3 mode filter in order to post-processing the image.

In regard to the pattern recognition techniques, for OPF we used the LibOPF (Papa, Suzuki, and Falcão 2009), which is a free tool to the design of classifiers based on of optimum-path forest. For Bayesian classifier (BC) we used our own implementation.

4 EXPERIMENTAL RESULTS

We describe in this section the results obtained. Figures 2 and 3 display, respectively, the images classified with BC and OPF.

In order to emphasize the importance of mode filter, Figure 4 displays the image of Figure 1a classified by BC with and without mode filter. Notice that the image in Figure 4b is equal to the image of Figure 2a.

One can see that the results obtained using 30% of the whole image for training are better when we used the BC classifier in case of Figure 1a. Using 50% the results appear to be similar. In regard to Figure 1c, both classifiers achieved close results using 30% and 50% for training. It is important to shed light over that, for both techniques, the mode filter played an important role to filter the images after classification. Table 1 displays the recognition rates for Figures1a.

One can see that BC outperformed OPF using 30% of the whole image for training, while OPF outperformed BC the second case, i.e., using 50% of the samples to train the classifiers. Table 2 displays the recognition rates for Figure1c.
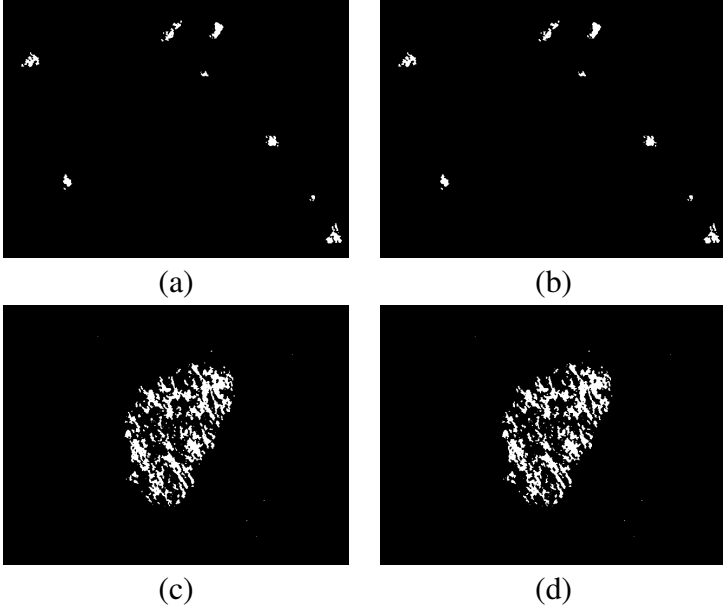
(a)   (b)

(c)   (d)

Figure 2: Classified images with BC using: (a) 30% and (b) 50% for training and (c) 30% and (d) 50% for training. The images (a)-(b) and (c)-(d) refer, respectively, to the original images in Figure 1a and Figure 1c.
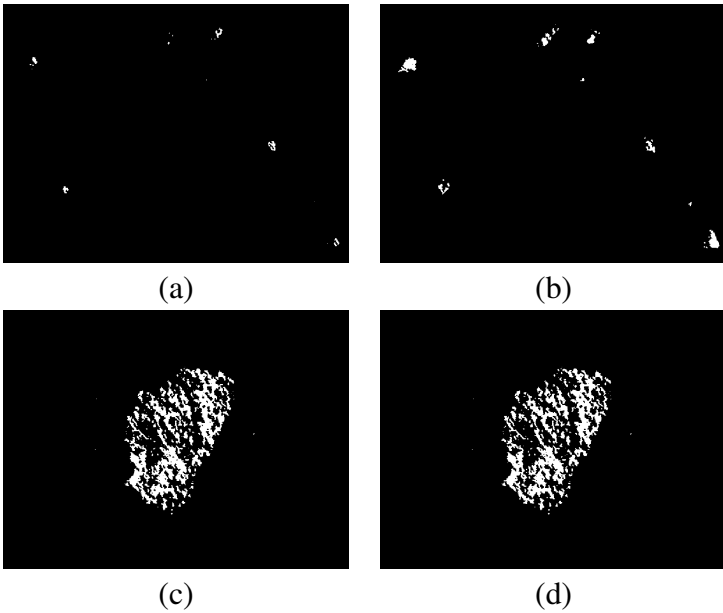

(a)   (b)

(c)   (d)

Figure 3: Classified images with OPF using: (a) 30% and (b) 50% for training and (c) 30% and (d) 50% for training. The images (a)-(b) and (c)-(d) refer, respectively, to the original images in Figure 1a and Figure 1c.

Classifier	Training %	Accuracy
OPF	30	68.51%
BC	**30**	**69.14%**
OPF	**50**	**77.31%**
BC	50	75.96%

Table 1: Recognition rates for the image in Figure 1a. The most accurate classifiers are bolded.
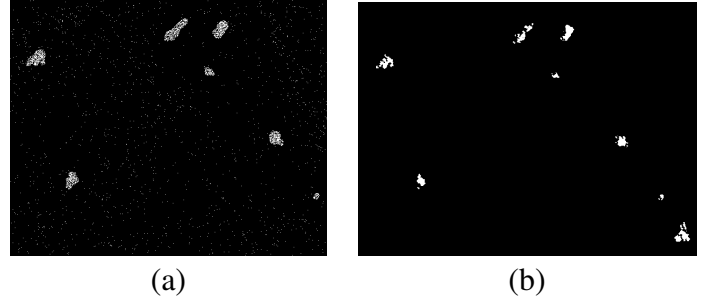
The OPF classifier achieved better results than BC

(a)   (b)

Figure 4: Figure 1a classified: (a) without and (b) with the mode filter.

Classifier	Training %	Accuracy
OPF	**30**	**70.62%**
BC	30	69.85%
OPF	50	79.13%
BC	**50**	**82.41%**

Table 2: Recognition rates for the image in Figure 1c. The most accurate classifiers are bolded.

using 30% for training, while the latter outperformed in case of 50%. However, one can see that the results are very similar for both classifiers using 30% and 50% for training in the employed images.

5 CONCLUSIONS

This paper was concerned on the problem of γ_2 island segmentation, which can provide important information about steel's quality and mechanical properties. In order to do that, we applied two supervised pattern recognition techniques, Optimum-Path Forest (OPF) and a Bayesian classifier (BC), on two labeled images with $200\times$ and $1000\times$ of magnifications, respectively.

Aiming to simulate an user behavior to select positive and negative samples, we conducted experiments with 30% and 50% of the whole images for training, to further classify the remaining pixels. The training samples were randomly chosen, and described by their gray values and texture features. In regard to recognition rates, both classifiers achieved similar results. A mode filter was applied to enhance the quality of images after classification.

Thus, we may conclude that the results were very promising, since this was the first work that addressed the problem of automatic segmentation of γ_2 islands in secondary austenite-phase precipitates.

REFERENCES

Allène, C., J. Y. Audibert, M. Couprie, J. Cousty, and R. Keriven (2007). Some links between min-cuts, optimal spanning forests and watersheds. In *Mathematical Morphology and its Applications to Image and Signal Processing (ISMM'07)*, pp. 253–264. MCT/INPE.

Bastos, I. N., S. S. M. Tavares, F. Dallard, and R. P. Nogueira (2007). Effect of microstructure on corrosion behavior of superduplex stainless steel enviroment conditions. *Scripta Materialia (Oxford) 57*, 913–916.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 269271.

Duda, R. O., P. E. Hart, and D. G. Stork (2000). *Pattern Classification*. Wiley-Interscience Publication.

Falcão, A. X., J. Stolfi, and R. A. Lotufo (2004). The image foresting transform theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*(1), 19–29.

Feichtinger, H. G. and T. Strohmer (1997). *Gabor Analysis and Algorithms: Theory and Applications* (1st ed.). Birkhauser Boston.

Hemmer, H. and O. Grong (1999). A process model for the heat-affected zone microstructure evolution in duplex stainless steel weldments: Part i. the model. *Metallurgical and Materials Transactions A 30*(11), 2915–2929.

Hemmer, H., O. Grong, and S. Klokkehaug (2000). A process model for the heat-affected zone microstructure evolution in duplex stainless steel weldments: Part ii. application to electron beam welding. *Metallurgical and Materials Transactions A 31*(13), 1035–1048.

Hertzman, S., B. Brolund, and P. Ferreira (1997). An experimental and theoretical study of heat-affected zone austenite reformation in three duplex stainless steels. *Metallurgical and Materials Transactions A 28*(2), 277–285.

Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press.

Kotecki, D. J. and J. L. P. Hilkes (1994). Welding processes for duplex stainless steels. In *Proceedings of the Fourth International Conference on Duplex Stainless Steels*, Volume 2, Glasgow, Scotland.

Lee, K. M., H. S. Cho, and D. C. Choi (1999). Effect of isothermal treatment of saf 2205 duplex stainless steel on migration of δ/γ interface boundary and growth of austenite. *Journal of Alloys and Compounds 285*(1-2), 156–161.

Lippold, J. C. and A. M. Al-Rumaih (1997). Toughness and pitting corrosion of duplex stainless steel weld heat affected zone microstructures containing secondary austenite. In *Proceedings of the Conference of Duplex Stainless Steels*, Maastrisht, The Netherlands, pp. 1005–1010.

Nilsson, J. O. (1992). Overview: Super duplex stainless steels. *Materials Science and Technology 8*, 685–695.

Nilsson, J. O., L. Karlsson, and J. O. Andersson (1995). Secondary austenite formation and its relation to pitting corrosion in duplex stainless steel weld metal. *Materials Science and Technology 11*(3), 276–283.

Nilsson, J. O. and A. Wilson (1993). A. influence of isothermal phase transformation on toughness and pitting corrosion of superduplex stainless steel saf 2507. *Materials Science and Technology 9*(6), 545–554.

Papa, J. P., A. X. Falcão, and C. T. N. Suzuki (2009). Supervised pattern classification based on optimum-path forest. *International Journal of Imaging System and Technology 19*(2), 120–131.

Papa, J. P., C. T. N. Suzuki, and A. X. Falcão (2009). *LibOPF: A library for the design of optimum-path forest classifiers*. Software version 2.0 available at: `http://www.ic.unicamp.br/~afalcao/LibOPF`.

Ramirez, A. J., S. D. Brandi, and J. C. Lippold (2004). Secondary austenite and chromium nitride precipitation in simulated heat affected zones of duplex stainless steels. *Science and Technology of Welding and Joining 9*(4), 301–313.

Ramirez, A. J., J. C. Lippold, and S. D. Brandi (2003). The relationship between chromium nitride and secondary austenite precipitation in duplex stainless steels. *Metallurgical and Materials Transactions A 34A*(8), 1575–1597.

Tavares, S. S. M., C. Scandian, J. M. Pardal, T. S. Luz, and F. J. da Silva (2010). Failure analysis of duplex stainless steel weld used in flexible pipes in off shore oil production. *Engineering Failure Analysis 17*, p. 1500–1506.