A Constructive Neural Network to Predict Pitting Corrosion Status of Stainless Steel

Daniel Urda[1], Rafael Marcos Luque[1], Maria Jesus Jiménez[2], Ignacio Turias[3], Leonardo Franco[1], and José Manuel Jerez[1]

[1] Department of Computer Science, University of Málaga, Málaga, Spain
{durda,rmluque,lfranco,jja}@lcc.uma.es
[2] Department of Civil and Industrial Engineering, University of Cádiz, Cádiz, Spain
mariajesus.come@uca.es
[3] Department of Computer Science, University of Cádiz, Cádiz, Spain
ignacio.turias@uca.es

Abstract. The main consequences of corrosion are the costs derived from both the maintenance tasks as from the public safety protection. In this sense, artificial intelligence models are used to determine pitting corrosion behaviour of stainless steel. This work presents the C-MANTEC constructive neural network algorithm as an automatic system to determine the status pitting corrosion of that alloy. Several classification techniques are compared with our proposal: Linear Discriminant Analysis, k-Nearest Neighbor, Multilayer Perceptron, Support Vector Machines and Naive Bayes. The results obtained show the robustness and higher performance of the C-MANTEC algorithm in comparison to the other artificial intelligence models, corroborating the utility of the constructive neural networks paradigm in the modelling pitting corrosion problem.

Keywords: Constructive neural networks, Austenitic stainless steel, Pitting corrosion.

1 Introduction

Corrosion can be defined as the degradation of the material and its properties due to chemical interactions with the environment. The main consequences of corrosion are important maintenance costs in addition to endangering public safety. The annual cost of corrosion worldwide has been estimated over 3% of the gross world product [1]. Therefore, corrosion has become one of the most relevant engineering problems. This phenomenon occurs so often that it has been necessary to develop models in order to predict corrosion behaviour of materials under specific environmental conditions.

Many authors have applied neural networks models to study corrosion: Kamrunnahar and Urquidi-MacDonald [2] presented a supervised neural network method to study localized and general corrosion on nickel based alloys. Cavanaugh et al. [3] used these models to model pit growth as a function of different environmental factors. Lajevardi et al. [4] applied artificial neural networks to predict the time to failure as a result of stress corrosion cracking in austenitic stainless steel. While, Pidaparti et al. [5] developed computational model based on cellular automata approach to predict the multi-pit corrosion damage initiation and growth in aircraft aluminium.

I. Rojas, G. Joya, and J. Cabestany (Eds.): IWANN 2013, Part I, LNCS 7902, pp. 88–95, 2013.

In spite of the numerous researches in corrosion risk of materials, no reliable method to predict pitting corrosion status of grade 316L stainless steel has yet been developed by others authors. Based on our studies about pitting corrosion [6,7], constructive neural networks (CNNs) are proposed in this paper to develop an automatic system to determine pitting corrosion status of stainless steel, with no need to check pits occurrence on surface material by microscopic techniques. Particularly, C-MANTEC model [8] is compared with other different standard classification models such as Linear Discriminant Analysis (LDA), k-Nearest Neighbor (kNN), Multilayer Perceptron (NeuralNet), Support Vector Machines (SVM) and Naive Bayes, in order to check the robustness and reliability of this algorithm on industrial environments. The use of C-MANTEC is motivated in the good performance results previously obtained in other areas [9,10] and due to its relatively small and compact neural network architecture leading to possible hardware implementation on industrial environments.

The remainder of this paper is organized as follows: Section 2.1 and Section 2.2 provides respectively a description of the dataset utilized on the experiments and the use of several classifiers models to be compared with C-MANTEC, and Section 3 shows the experimental results over several classifying algorithms. Finally, Section 4 concludes the article.

2 Material and Methods

2.1 Material

In order to study corrosion behaviour of austenitic stainless steel a European project called “Avoiding catastrophic corrosion failure of stainless steel” CORINOX (RFSR-CT-2006-00022) was partially developed by ACERINOX. In this project, 73 different samples of grade 316L stainless steel were subjected to polarization tests in order to determine pitting potentials values in different environmental conditions: varying ion chloride concentration ($0.0025-0-1$M), pH values ($3.5-7$) and temperature ($2-75^oC$) using NaCl as precursor salt.

Pitting potential is one of the most relevant factors used to characterize pitting corrosion [11]. This parameter is defined as the potential at which current density suffers an abrupt increase. It can be determined based on polarization curves as the potential at which current density is $100\mu A/cm^2$ [12].

All the polarization tests were carried out using a Potentiostate PARSAT 273. For each of the 73 sample, the potential and current density values registered during the tests were plotted on semi-logarithmic scale to determine pitting potential values (see Figure 1). After polarization tests, all samples were checked microscopically for evidence of localized corrosion. In this way, all species were characterized by the environmental conditions tested (chloride ion concentration, pH and temperature) in addition to corrosion status: 1 for samples where pits appeared on the material surface and 0 otherwise.

2.2 Methods

In this work, we propose the use of constructive neural networks as classifiers models, in particular C-MANTEC, to predict corrosion behaviour of austenitic stainless steel.

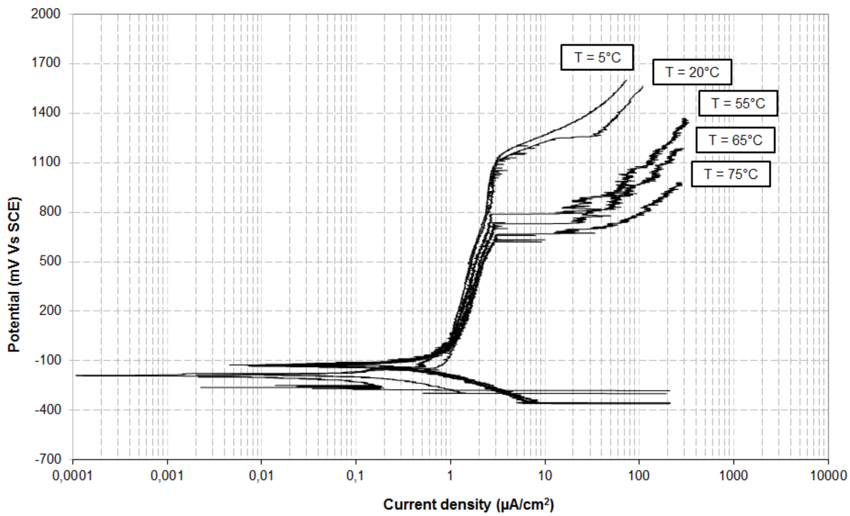

Fig. 1. Polarization curves measured for grade 316L stainless steel using NaCl as precursor salt. The conditions tested were: 0.0025 M (mol/L) chlorides ions, pH = 5.5 and temperature (5°C-75°C).

C-MANTEC (Competitive Majority Network Trained by Error Correction) is a novel neural network constructive algorithm that utilises competition between neurons and a modified perceptron learning rule to build compact architectures with good prediction capabilities. The novelty of C-MANTEC is that the neurons compete for learning the new incoming data, and this process permits the creation of very compact neural architectures. At the single neuronal level, the algorithm uses the thermal perceptron rule, introduced by Marcus Frean in 1992 [13], that improves the convergence of the standard perceptron for non-linearly separable problems. In the thermal perceptron rule, the modification of the synaptic weights, $\Delta\omega_i$, is done on-line (after the presentation of a single input pattern) according to the following equation:

$$\Delta\omega_i = (t - S)\psi_i T_{fac} \quad (1)$$

where t is the target value of the presented input, and ψ represents the value of input unit i connected to the output by weight ω_i. The difference to the standard perceptron learning rule is that the thermal perceptron incorporates the factor T_{fac}. This factor depends on the value of the synaptic potential and on an artificially introduced temperature (T) that is decreased as the learning process advances.

The topology of a C-MANTEC created network consists of a single hidden layer of thermal perceptrons that maps the information to an output neuron that uses a majority function. The choice of the output function as a majority gate is motivated by previous experiments in which very good computational capabilities have been observed for the majority function among the set of linearly separable functions [14]. The results so far

Table 1. Brief pseudo-code summary of the C-MANTEC learning algorithm

C-MANTEC learning algorithm

```
Initialise the parameters of the algorithm;

while (exists patterns to be learned) {
  input a random pattern;
  if (pattern target value == network output) {
    remove temporarily the pattern from the dataset;
  }
  else {
    the pattern has to be learned by the network;
    select the wrong neuron with highest temperature;
    if (Tfac >= Gfac) {
      the neuron will learn the pattern;
      update its synaptic weights according to the thermal perceptron rule;
    }
    else {
      a new neuron is added to the network;
      this new neuron learns the pattern;
      iteration counters are reset;
      noisy patterns are deleted from the training dataset;
      reset the set of patterns;
    }
  }
}
```

obtained with the algorithm [15,8,10] show that it generates very compact neural architectures with state-of-the-art generalization capabilities. It has to be noted that the algorithm incorporates a built-in filtering stage that prevent overfitting of noisy examples.

The C-MANTEC algorithm has 3 parameters to be set at the time of starting the learning procedure. Several experiments have shown that the algorithm is very robust against changes of the parameter values and thus C-MANTEC operates fairly well in a wide range of values. The three parameters of the algorithm to be set are: (i) I_{max} as maximum number of iterations allowed for each neuron present in the hidden layer per learning cycle, (ii) G_{fact} as growing factor that determines when to stop a learning cycle and include a new neuron in the hidden layer, and (iii) Fi_{temp} that determines in which case an input example is considered as noise and removed from the training dataset according to Eq. 2, where N represents the number of input patterns of the dataset, NTL is the number of times that the pattern X has been learned on the current learning cycle, and the pair $\{\mu,\sigma\}$ corresponds to the mean and variance of the normal distribution that represents the number of times that each pattern of the dataset has been learned during the learning cycle.

$$\forall X \in \{X_1, ..., X_N\}, \text{delete(X)} \mid \text{NTL} \geq (\mu + Fi_{temp}\sigma) \tag{2}$$

A summary of the C-MANTEC pseudo-code algorithm is described in Table 1. This learning procedure is essentially based on the idea that patterns are learned by those neurons, the thermal perceptrons in the hidden layer of the neural architecture, whose output differs from the target value (wrongly classified the input) and for which its internal temperature is higher than the set value of G_{fac}. In the case in which more than one thermal perceptron in the hidden layer satisfies these conditions at a given iteration, the perceptron that has the highest temperature is the selected candidate to

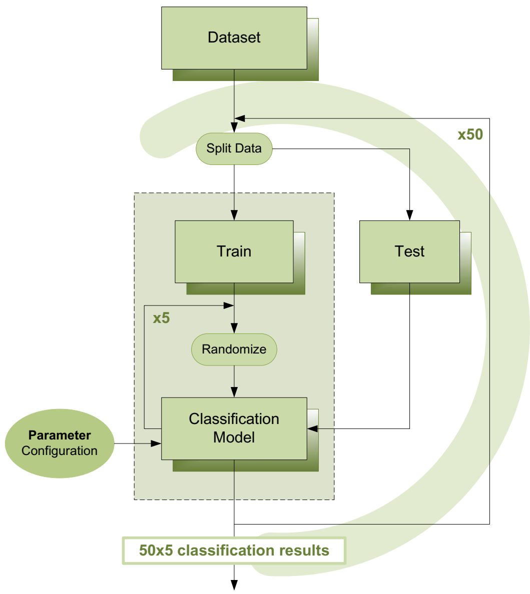

Fig. 2. Bootstrap resampling technique procedure used for each classification model, both for estimating the parameters configuration of each model and to predict pitting corrosion behaviour

learn the incoming pattern. A new single neuron is added to the network when there is no a thermal perceptron that complies with these conditions and a new learning cycle starts. The learning process ends when there are no more patterns to be learned, as all of them are classified correctly or are outside of the initial dataset because are considered noisy by an internal built-in filter.

Several classification models have been used to evaluate our proposal in this paper: LDA, kNN, NeuralNet, SVM and Naive Bayes. As Figure 2 shows, first a bootstrap resampling technique is applied $50x5$ times for each of these models varying the values of their required parameters, including C-MANTEC. Although it is not an honest parameter estimation procedure, it allows us to estimate a parameter configuration set in order to test the robustness of different classification models [16]. Afterwards, bootstrapping is reapplied $200x10$ for each model with the obtained parameters in order to predict pitting corrosion behaviour in terms of accuracy and standard deviation.

3 Experimental Results

It is not easy to determine in advance the appropriate parameters to get a generalization rate, which requires a tedious empirical assessment of the data to assign these values. In this way, multiple configurations of the comparative techniques are generated by combining the values of the parameters shown in Table 2 in every possible manner, which also includes the final quantitative results in the column "Accuracy". These results are obtained by setting the algorithms parameters as follows: $\{k = 1, d = \text{cosine-similarity}\}$ in kNN; $\{NHidden = 20, \alpha = 0.05, NCycles = 25\}$

Table 2. Parameter settings tested during evaluation of the classification algorithms. The combination of all the values of the parameters generate a set of configurations for each method. The third column shows the quantitative results for the best parameter setting of each algorithm.

Algorithm	Test Parameters	Accuracy
LDA	No parameters	72.560±0.49
kNN	Neighbours, k= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} Distance type, d= {euclidean, chi-squared, cosine-similarity}	79.867±0.44
NeuralNet	Hidden neurons, $NHidden$= {2, 4, 6, 8, 10, 15, 20} Alpha, α= {0.05, 0.1, 0.2, 0.3, 0.5} Number cycles, $NCycles$= {10, 25, 50}	**87.254±0.47**
SVM	Kernel type, t= {linear, polynomial, radial base function, sigmoid} Cost, C= {1, 3, 5, 7, 9, 10, 12, 15} Degree, d= {1, 2, 3, 4, 5} Gamma, g= {0.001, 0.005, 0.1, 0.15, 0.2, 0.4, 0.6, 0.8, 1, 2, 3, 5} Coef0, r= {0, 1, 2}	85.508±0.50
NaiveBayes	Kernel density, K= {0, 1}	66.882±0.55
C-MANTEC	Max. Iterations, I_{max}= {1000, 10000, 100000} GFac, g_{fac} = {0.01, 0.05, 0.1, 0.2, 0.25, 0.3} Phi, ϕ = {1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6}	**89.788±0.56**

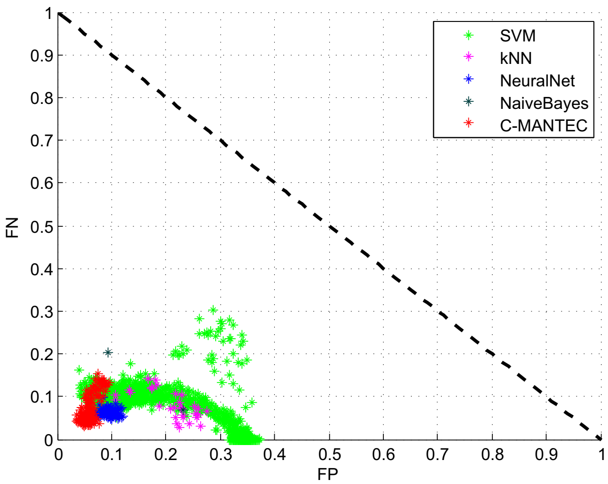

Fig. 3. False Positives (FP) and False Negatives (FN) ratios after applying each method to the dataset with all the parameter configurations. Each coloured point '*' is considered as a different configuration of that method. The closer the points are to the origin, the better the segmentation is. Additionally, the method is less sensible to a parameters' change if the points' cloud keeps compact and grouped.

in NeuralNet; $\{t = \text{polynomial}, C = 15, d = 2, g = 5, r = 0\}$ in SVM; $\{K = 0\}$ in NaiveBayes; and $\{I_{max} = 10000, g_{fac} = 0.3, \phi = 4.5\}$ in C-MANTEC. In concrete, C-MANTEC (89.78% in accuracy) clearly outperforms LDA (72.56%), kNN (79.86%) and NaiveBayes (66.88%) models, and it also improves the NeuralNet and SVM classification accuracies but only in 2 and 4 percentage points respectively.

A thorough analysis is presented in Figure 3, where the influence of the parameter setting for different algorithms is evaluated in the classification accuracy variability. The horizontal axis corresponds to the average percentage of the false positives (FP) on the data, while the vertical axis is associated with the false negatives values (FN). Each point of the plot represents the average FP and FN of a generated configuration when it is applied to the dataset. The closer the points are to the origin, the better the classification process. The optimum performance occurs if $FN = 0$ and $FP = 0$, which implies there is a perfect match between the output of the algorithm and the real output of the problem. The results are always below the diagonal of the plot because we always have $FN + FP <= 1$.

The variability for each classifier depends largely on the analysed dataset, but the robustness of the method has also an influence, i.e. more robust methods yield smaller values. If the configuration cloud is compact, it means that the results do not vary significantly after a change in its parameters. On the other hand, if several configurations are far from each other, it implies that the variation of a parameter causes abrupt changes in the results, which is a very undesirable property for a classification algorithm. As shown in Figure 3, the compactness for kNN and SVM methods is rather poor, while our C-MANTEC approach and NeuralNet model have their configurations very close together. In other words, the performance of the proposed method is not very sensitive to the parameter selection. Additionally, our approach is closer to the zero point than the remaining alternatives, which implies that C-MANTEC provides the best classification result. Since the NaiveBayes classifier do not require many values for its parameters, the cloud of points for this method (i.e. number of configurations) is not relevant.

4 Conclusions

This work presents a novel application of a constructive neural network (the C-MANTEC algorithm) to the prediction of pitting corrosion status of stainless steel as function of environmental conditions. The results demonstrate that C-MANTEC outperforms, in terms of classification accuracy, the other algorithms under study. In addition, the compact neural network architecture generated by the C-MANTEC algorithm makes it suitable to be implemented in a hardware architecture for industrial scope.

The high cost of the polarization tests, in addition to the complexity of its interpretation because of the multiples factor involved, justifies the development of a model to predict pitting corrosion behaviour of austenitic stainless steel without resorting polarization tests. In this sense and as further work, it could be interesting to provide a useful tool to determine the existence of pits on the material surface by automatic technique without need of microscope analysis reducing the cost of experimental tests. Moreover, due to the multiple variables affecting pitting corrosion behaviour of stainless steel, it would also be interesting to test classification models varying some of the environmental factors studied in this paper. A remarkable case of study would be to analyse the influence of precursor salts on classification performance.

Acknowledgements. The authors acknowledge support through grants TIN2010-16556 from MICINN-SPAIN and P08-TIC-04026 (Junta de Andalucía), all of which include FEDER funds.

References

1. Schmitt, G.: Global needs for knowledge dissemination, research, and development in materials deterioration and corrosion control. The World Corrosion Organization (2009)
2. Kamrunnahar, M., Urquidi-Macdonald, M.: Prediction of corrosion behaviour of alloy 22 using neural network as a data mining tool. Corrosion Science 53, 961–967 (2011)
3. Cavanaugh, M., Buchheit, R., Birbilis, N.: Modeling the environmental dependence of pit growth using neural network approaches. Corrosion Science 52, 3070–3077 (2010)
4. Lajevardi, S., Shahrabi, T., Baigi, V., Shafiei, M.A.: Prediction of time to failure in stress corrosion cracking of 304 stainless steel in aqueous chloride solution by artificial neural network. Protection of Metals and Physical Chemistry of Surfaces 45, 610–615 (2009)
5. Pidaparti, R.M., Fang, L., Palakal, M.J.: Computational simulation of multi-pit corrosion process in materials. Computational Materials Science 41, 255–265 (2008)
6. Jiménez-Come, M.J., Muñoz, E., García, R., Matres, V., Martín, M.L., Trujillo, F., Turias, I.: Austenitic stainless steel en 1.4404 corrosion detection using classification techniques. In: Corchado, E., Snášel, V., Sedano, J., Hassanien, A.E., Calvo, J.L., Ślęzak, D. (eds.) SOCO 2011. AISC, vol. 87, pp. 193–201. Springer, Heidelberg (2011)
7. Jiménez-Come, M.J., Muñoz, E., García, R., Matres, V., Martín, M.L., Trujillo, F., Turias, I.: Pitting corrosion behaviour of austenitic stainless steel using artificial intelligence techniques. J. Applied Logic 10, 291–297 (2012)
8. Subirats, J.L., Franco, L., Jerez, J.M.: C-mantec: A novel constructive neural network algorithm incorporating competition between neurons. Neural Networks 26, 130–140 (2012)
9. Urda, D., Cañete, E., Subirats, J., Franco, L., Llopis, L., Jerez, J.: Energy efficient reprogramming in WSN using Constructive Neural Networks. International Journal of Innovative Computing, Information and Control 8, 7561–7578 (2012)
10. Urda, D., Subirats, J.L., Franco, L., Jerez, J.M.: Constructive neural networks to predict breast cancer outcome by using gene expression profiles. In: García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., Ali, M. (eds.) IEA/AIE 2010, Part I. LNCS, vol. 6096, pp. 317–326. Springer, Heidelberg (2010)
11. Galvele, J.: Present state of understanding of the breakdown of passivity and repassivation. The Electrochemical Society, 285–326 (1979)
12. Merello, R., Botana, F., Botella, J., Matres, M., Marcos, M.: Influence of chemical composition on the pitting corrosion resistance of non-standard low-Ni high-Mn N duplex stainless steels. Corrosion Science 45, 909–921 (2003)
13. Frean, M.: A "thermal" perceptron learning rule. Neural Comput. 4, 946–957 (1992)
14. Subirats, J.L., Franco, L., Gómez, I., Jerez, J.M.: Computational capabilities of feedforward neural networks the role of the output function. In: Proceedings of the XII CAEPIA 2007, Salamanca, Spain, vol. 2, pp. 231–238 (2008)
15. Subirats, J.L., Jerez, J.M., Franco, L.: A new decomposition algorithm for threshold synthesis and generalization of boolean functions. IEEE Transactions on Circuits and Systems 1, 3188–3196 (2008)
16. Jiang, W., Simon, R.: A comparison of bootstrap methods and an adjusted bootstrap approach for estimating the prediction error in microarray classification. Statistics in Medicine 26, 5320–5334 (2007)