NPL Report MATC (A) 21

Application of Neural Networks to Predict Sulphide Stress Corrosion Cracking of Duplex Stainless Steels

Shengqi Zhou, David Coleman and Alan Turnbull

May 2001

ISSN 1473 - 2734

National Physical Laboratory
Teddington, Middlesex, UK, TW11 0LW

Approved on behalf of Managing Director, NPL, by Dr C Lea,
Head, Materials Centre

Application of Neural Networks to Predict Sulphide Stress

Corrosion Cracking of Duplex Stainless Steels

S Zhou, D Coleman and A Turnbull
Materials Centre
National Physical Laboratory
Teddington
Middlesex
TW11 0LW

Abstract

The development of the "Corrosion Database of Duplex Stainless Steels" was a significant advance in encouraging the safe use of these materials. However, as corrosion is affected by many variables such as temperature, pressure of H_2S, yield strength of material, there are many conditions for which specific data are not available and prediction of susceptibility using traditional statistical methods is not possible. Neural networks have been used to achieve such predictions.

Algorithms have been developed using multi-layer feed-forward neural networks to predict the susceptibility to sulphide stress corrosion cracking of both standard 22 Cr and high alloy duplex stainless steels. Kohonen neural networks have also been created to aid the development of these algorithms. The latter also serve as a search engine enabling the nearest neighbours that exist in the "Corrosion Database of Duplex Stainless Steels" to be accessed. When used together, these techniques have the potential to provide a fully functional software package capable of accessing existing information and prediction based on the database data.

Finally, the networks have been analysed using network weights. These indicate the relative importance of parameters such as partial pressure of H_2S, NaCl content and temperature. Neural-fuzzy algorithms have been developed that "mimic" the multi-layer feed-forward networks and allow some "rules" to be constructed.

Introduction

The safe use of duplex stainless steels (DSS) depends on knowing the bounds within which these materials will be resistant to corrosion. Although duplex stainless steels usually have good resistance to corrosion, they can be susceptible to stress corrosion cracking particularly in sour (H_2S-containing) environments. There is an abundance of published data regarding this phenomenon but locating the most relevant data to a specific application can be time consuming.

To remedy this situation considerable effort has been made to collate data on the performance of DSS from the open literature, materials suppliers and end users. These data have been integrated by NPL, with a significant contribution from TWI to form the "Corrosion Database of Duplex Stainless Steels". This database consists of over 6000 entries on general corrosion, stress corrosion cracking, pitting corrosion and crevice corrosion.

The development of the database was a significant advance in encouraging the safe application of duplex stainless steels and in avoiding unnecessary testing. However, susceptibility to corrosion is affected by many variables, such as temperature, pH, chloride level and partial pressure of hydrogen sulphide. There still exist many situations for which specific corrosion data are not available. Prediction of corrosion performance based on existing data can be very difficult, particularly in borderline conditions. However, tools which enable prediction based on multivariate data do exist. Following a preliminary evaluation of the tools available[1], neural networks are now being utilised to predict the susceptibility of duplex stainless steels to corrosion, using the data in the Duplex Database.

Sulphide stress corrosion cracking of duplex stainless steels

As the name suggests, duplex stainless steels contain a two-phase structure of ferrite and austenite. Consequently these steels combine good corrosion resistance and mechanical properties. As a result, duplex stainless steels have been increasingly used in demanding environments (such as oil wells). However, in the presence of significant quantities of hydrogen sulphide (i.e. sour environments), sulphide stress corrosion cracking may occur.

There are many factors that influence the susceptibility of duplex stainless steels to sulphide stress corrosion cracking. For example, material properties, such as composition and heat treatment, environmental variables, such as partial pressures of H_2S, and CO_2, and the presence of aggressive species such as chloride, may all influence susceptibility to cracking. Whilst much has been published regarding the effect of each of these variables[2, 3], there is not as yet a model which encompasses all of them in order to predict the probability of cracking. In the current work, neural networks have been applied to the Corrosion Database of Duplex Stainless Steels to produce such algorithms.

Basic Principles

A neural network is essentially a type of information processing technology inspired by studies of the structures in the brain and nervous system. It consists of many simple, highly interconnected processing elements (called neurons) that dynamically interact with each other to "learn" or "respond to" information rather than simply carry out programmed instructions[4]. A single neuron together with the connections from its input and output make up a simple perceptron (Figure 1).

In this single perceptron (i.e. in which there is only one neuron in the network), the value of each of the variables inputted into the neuron via a connection is used to produce an output (Figure 1). Every input to the neuron also has an associated "weight" attached. The neuron simply adds together all the weighted inputs connected to it and calculates the output to be transmitted. This output is then modified by a function, which in simple cases is a binary transfer function. If the value of the summation from the neuron is greater than or equal to a certain value, *y*, then the function's output is 1; if the neuron's summation is less than *y*, the output is 0.

As an example consider a simple perceptron, with only two inputs, that has learnt to reproduce Table I below:

INPUT 1	INPUT 2	OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

Table I. Example of an input/output pattern.

Figure 2 illustrates a perceptron that can reproduce this table and includes associated weights. If the case in which both inputs are zero is presented to the neuron, the total sum will be zero, and therefore the output will be zero. Likewise, if only one input is 1, the total sum calculated by the neuron will be less than 1 (i.e. [1 x 0.6]+[0 x 0.6]<1), and therefore the output will be zero. Only in the case where both inputs are 1 will the output be 1 (i.e. analogous to a logic "AND" gate).

Multi-Layer Feed Forward Network

Single perceptrons can only be used to output simple patterns such as the previous example. For more complex patterns, a multi-layer perceptron (MLP) is required. This is commonly known as a Multi-Layered Feed-Forward (MLF) neural network. In these networks (Figure 3), neurons are arranged in layers with the input data fed to the network at the input layer. These data then pass through the trained network to the output layer to provide the output, i.e. the prediction.

Before such networks can be used to generate outputs given a pattern of input variables, they must be trained so that the weights on each connection are correctly assigned. This is achieved using back-propagation, during which sets of training data are presented to the network along with their respective outputs (i.e. the outcomes). This process is known as supervised learning. Quite simply, an error from the network is calculated by subtracting the value of the calculated output, R, from the actual output (or target), T. The errors of the neurons in the previous layer are then calculated. These are calculated simply by multiplying the error in the output layer by the weight on the connection from the previous layer, and summing this for all the relevant connections (see Figure 3). The weights are then recalculated by adding to the original ones a function of the calculated error. This process is iterated until the error from the output is less than a pre-set level. The network is then said to be trained and (following validation and testing) can be used to make predictions given a set of data presented to it at the input layer.

MLFs have previously been used in many corrosion related problems [5-15], including:

- The development of an expert system to determine whether localised or general corrosion would occur based on data from polarisation scans [5];
- The modelling of atmospheric corrosion [6];
- The prediction of pitting corrosion [7];
- The prediction of stress corrosion cracking occurring in austenitic stainless steels in chloride containing waters [8].

The output of neural network predictions for sulphide stress corrosion cracking can be presented in two ways:

1. Generation of a continuous value between 0 and 1 representing the probability of a "pass" occurring;
2. Generation of a "binary" output classifying whether a "pass" or "fail" will occur.

As will be shown, the former is more meaningful since there is often scatter in test data. This is due to the random nature of SSCC and the variations in the laboratory test methods used to generate the data.

Kohonen Neural Network

This type of network is an *unsupervised* neural network and trains using the Kohonen algorithm, hence its name. The description that follows is rather simplified. Readers interested in the details regarding Kohonen networks can refer to Tarassenko [16], or Kohonen [17].

Suppose there are *lines* of data (henceforth known as datasets) of which each has n variables attributed to it. Each dataset can then be represented by a co-ordinate in n-dimensional space, the position of which is determined by the value of each variable. Before these datasets can be inputted into a Kohonen neural network, a set of K randomised vectors (neurons) in the same n-D space is established.

As each individual dataset is inputted into the network, its co-ordinate in n-D space is compared to those of the randomised neurons. The closest neuron "wins" that dataset and moves closer to the co-ordinate of that set's location (by an amount determined by a *learning rate*). The other neurons within a preset *neighbourhood* also move by an amount proportional to the *neighbourhood size*. This process is then repeated for every dataset. Once all the datasets have been inputted, the *neighbourhood size* and *learning rate* are reduced and the process repeated. This process continues until the neurons no longer move by any significant amount. At this stage what were once randomised neurons have now become ordered within the n-dimensional space. Each neuron therefore represents a collection of *nearest neighbours* or clusters of data.

A two dimensional co-ordinate (i,j) is also attributed to each of the neurons such that a 2-D array can be constructed in which the neurons or clusters of data which are close to each other in the original n-dimensional input space will be neighbours in the 2-D array [16].

Comparing the actual outcomes or results from each of the datasets for each cluster, it may be possible to identify certain clusters which generate a common behaviour and are hence

predictable. It is also possible to identify outliers, i.e. lines of test data that are quite different from the rest and do not fall within a large cluster of data.

Whilst the above uses could be applied to the corrosion database, it could not be used to generate algorithms which directly predict an outcome from the given dataset parameters, and consequently the ability to interpolate between clusters is limited. However, this technique has been used in this work for the pre-processing of data for the development of MLF neural networks as described later.

The technique is also useful as a search engine. The *real data* which are positioned closest in the n-dimensional space to that of the exposure condition which is being queried can be accessed. This can therefore be used by the user to back-up the MLF prediction with actual data and to identify references in which the relevant test results are reported. It also serves to identify regions in which test data may be sparse.

Neural Fuzzy Network

One of the limitations of neural networks is that they are often perceived to be "black boxes" in which a prediction is given with no indication of how this conclusion was drawn. However, advanced "neural-fuzzy" networks have been developed which allow a network's predictions to be described in a logical manner. Such networks are supervised and are trained in much the same way as MLFs, i.e. the input variables are presented to the network along with the results/outcomes during the training stage. The difference is that instead of attaching weights to neurons to map the input/output relationships, a series of rules are constructed and refined.

These rules are based on IF…AND…THEN logic. However, the rules constructed are fuzzy in nature. In other words they utilise classes of both the input variables and outputs, all of which are not precisely defined (see Figure 4). For example, one network could well suggest the following rules:

> IF NaCl concentration = High, AND temperature = High…THEN probability of failure = High (0.6).

The number in brackets at the end of the above rule represents the level of confidence in the given rule (ranging from 0 to 1 where the latter is the highest level of confidence).

The network defines all the boundaries of each class so that the analyst can determine how the network has come to its conclusions.

The analyst can also modify fuzzy rules. For example, if a rule is created by the neural network as a result of its training but is known to be incorrect (perhaps due to lack of data in a crucial area), this can be rectified.

Reducing the number of classes for each variable is also possible and this allows simplification of the algorithms, together with greater transparency.

Development Of Predictive Algorithms

Data Pre-Processing

In order to successfully develop predictive algorithms using MLF neural networks, it is necessary to pre-process the database data by: dealing with any missing parameter values; balancing the data such that the neural network can be trained, validated and tested using a full range of environmental conditions; and organising the data such that they can be split (i.e. partitioned) into separate training, validation and test sets. The methodology used in this work to pre-process the data is summarised below.

Missing parameter values

Although the database is quite comprehensive there remain some datasets for which a parameter value, e.g. yield strength, is missing. Where such parameters were required for successful neural network development, effort was made to insert values. This was achieved by deriving values from known relationships, or using mean values of parameters from other datasets for which the relevant conditions were the same. For example, where the yield strength was not quoted for a given grade of DSS in a known condition, a mean value of the yield strength quoted in all other datasets for the same material in the same condition was used. Where parameter values could not be derived for a particular dataset, the dataset was not used for neural network development.

Balancing of the data

Although it is important to create neural networks that are capable of prediction over a broad range of conditions, the inclusion of data that deviate significantly from the majority (i.e. those which lie far outside the range of say 95 % of the total data) can result in erroneous models. A rather crude method of analysing the spread of data for each of the input parameters was used. In each case, i.e. for yield strength, pH, temperature, partial pressure of H_2S ($p.p._{H2S}$), ranked percentile values (i.e. minimum value, 10^{th} percentile, 20^{th} percentile… 100^{th} percentile) were plotted together with best fit (linear regression) lines. The spread of data could then be evaluated by using the linear correlation coefficient, R^2 (i.e. square of Pearson product moment). Parameters for which the value of R^2 was close to 1 had an even spread of values whereas as the spread became less even, R^2 tended towards zero. A value of 0.7 was used as a threshold for R^2. If the coefficient fell below this, datasets containing parameter values in the extreme regions were gradually removed until the threshold value was reached. (These datasets played no further part in the neural network development). The threshold value was selected on the basis that too high a requirement would not yield enough data for successful development and would also limit the range of conditions within which predictions could be generated. Too low a value would result in erroneous algorithms that would not be able to interpolate effectively between the clusters of data.

Kohonen self-organisation and partitioning

The balanced data were all passed through a Kohonen network so that they could be arranged in clusters, and consequently could be presented in some order (within the n-dimension space). The data was then split into three equal partitions, i.e. for training, validation and testing, by the repositioning of every third dataset into these respective groups. In using this

technique to partition the data, each group covered as fully as possible the broad range of data that existed. This is important, particularly for training, in order that the neural network learns from the "big picture" rather than from sporadic clusters which may lead to ignorance of experience in important situations, i.e. boundary conditions.

Although there may be situations in which a greater proportion of training data is required, e.g. where data are sparse, equal partition sizes are preferable[16]. For the neural network algorithms reported here, this criterion has been adhered to.

MLF Development

Multi-layer feed-forward (MLF) neural networks were used to predict the susceptibility of both solution-annealed and cold-worked 22 Cr and high alloy DSS to sulphide stress corrosion cracking (SSCC). These networks were trained as follows:

1. The MLF was trained with the *training data* and validated (at the same time) with the *validation data*. In each case, discrete output (target) values of 1 were used for datasets that resulted in a pass, whilst output values of 0 were used for datasets that failed.
2. When the average error on the validation set either remained constant or started to rise, training was halted.
3. Steps 1-2 were repeated twice more to account for the random nature of training.
4. This training cycle was repeated several times, each time with a different network architecture, consisting of either more neurons, or, if necessary, more hidden layers.
5. The optimal network architecture was selected based on the performance (see below).
6. Training with the optimal network architecture was then repeated ten times. The three networks from these ten with the best overall performance (see below) were selected for the generation of predictions (i.e. an average prediction was used).

For the developed algorithms reported here, the inputs to the MLF neural networks were yield strength (MPa), temperature (°C), log $p.p._{H2S}$ (bar), log wt NaCl (%), and pH. A sigmoid transfer function was employed for both the hidden and output layers. This particular function has been used by many others in the field of corrosion [5-8] and is of the following form:

$$f(x) = \frac{1}{1 + e^{-x}} \qquad (1)$$

The logarithms of NaCl and $p.p._{H2S}$ were used since corrosion phenomena are often observed to respond to the logarithm of concentrations of chemicals involved [8]. Also, the data cover the range of these parameters more fully in logarithmic form.

As mentioned above, the output (target) values used in the training and validation stages were discrete values of 0 or 1. However, on querying the trained networks to generate predictions for a given set of input parameters, a continuous value between 0 and 1 is outputted where 0 represents 0% chance of a pass and 1 represents 100% chance of a pass. This transformation from a discrete to a continuous output arises primarily due to the random nature of SSCC and the variations in the laboratory test methods used to generate the data. However, even with "perfect" but incomplete data (i.e. data which does not fully cover the range of input parameters but in which there is no variability) there would be some degree of uncertainty in predictions for datasets not explicitly covered in the training data. This uncertainty would

increase as the input parameters lie closer to the decision boundary. Consequently, there is no clear boundary between regions in which a pass will occur and those in which a fail will occur, but rather there is a gradual transition.

Performance Determination Of Neural Networks

There are two factors that are important in determining the performance of the MLF neural networks constructed. Firstly, it is important to have a network that models the data accurately. Secondly, a model is required which will generalise for conditions other than those in which it was trained, as opposed to simply memorising the data. Since the training and validation data both influence the construction of the algorithms, the unseen (i.e. previously unused) test data has been used to assess the network performance, both in isolation and in conjunction with the training and validation data (the latter of which may result in a more statistically sound performance).

In both cases, the performance of each of the ten networks of optimal architecture were evaluated as follows:

- Predictions were generated by the neural network algorithms for each of the data.
- The data was then ranked and split into quintiles (i.e. five groups of equal size which classified the data according to the value of predictions).
- The average prediction in each group was then compared to the proportion of data that *actually* resulted in a pass.
- Both the spread of predictions and the maximum error were recorded. The value of the spread of predictions divided by the maximum error that occurred in any one class was used as the parameter for performance.

Results and Discussion

Table II illustrates the range of data used to train the MLF neural networks (and hence the boundaries of the predictive capability). The average performances (i.e. from the three optimal networks) for each DSS grade are shown in Figures 5-8. These figures illustrate the proportion of data that passed in each band of predictions (both for the test data and the overall set). The solid series (to the right) in the figures, i.e. the *average predicted %*, illustrates the proportion of passed data that are expected to occur in each category and is (as the name suggests) equal to the average prediction in each class.

It can be seen from Figures 5-8 that the proportion of test data that passed in each band of predictions follows quite closely the expected trend. Although there is some error in the network performances, these errors are at least transparent and can be easily transferred to users of the algorithms (e.g. an error value can be presented together with the predicted probability of resistance to stress corrosion cracking).

	Solution-annealed 22Cr DSS data		Cold-worked 22Cr DSS data		Solution-annealed High alloy DSS data		Cold-worked High alloy DSS data	
	min.	max.	min.	max.	min.	max.	min.	max.
Yield stress, MPa	418	650	593	1579	450	780	822	1210
Temperature, °C	20	300	20	300	20	300	10	300
$p.p._{H2S}$, bar	0	25	0	25	0	14.5	0	10
NaCl, wt. %	0.1	25	0.5	25	0.1	50	1	25
pH	2.5	5.5	2.5	4	1.35	5.1	2.8	4.8
Total number of data:	453		414		269		276	
Number of training data:	151		138		89		92	

Table II. Properties of data used for MLF neural network development.

In order to explore the predictive capability and the application of these preliminary algorithms together with any limitations, plots of $p.p._{H2S}$ versus temperature can be constructed from predictions generated by querying the networks with a range of input parameters that are of practical relevance. An example of such a plot is shown in Figure 9.

Considering the plot of $p.p._{H2S}$ versus temperature (Figure 9) constructed with the predictions for a range of input parameters, one can see that for the standard 22 Cr DSS, curves are generated which demarcate the transition from a high risk of cracking to a low risk. The maximum susceptibility to cracking at a given H_2S level occurs between 80 and 100 °C. At lower temperatures the susceptibility to cracking reduces as the solubility of hydrogen is lower and therefore less hydrogen is absorbed into the steel. At higher temperatures, a smaller proportion of the absorbed hydrogen atoms become trapped at microstructural sites leading to a reduced likelihood of embrittlement.

If an end-user is particularly interested in the actual laboratory test data which are closest to a specific set of conditions (as opposed to a predicted probability), the Kohonen neural network can be queried to find the neuron which represents the cluster in which the condition of interest and its nearest neighbours are present. Data from the DSS database that fall within this cluster can then be listed. For example, the nearest neighbours to the condition X, represented on Figure 9, are listed in Table III.

Further analysis of the MLF networks is achievable by viewing the weights attributed to each connection, although it is not possible to establish any equations as such using these weights. It is however possible to establish the relative importance of parameters for each of the networks created. The average importance of each input parameter is shown in Figure 10.

It can be seen from Figure 10 that in every case, as expected, the most important parameters are partial pressure of hydrogen sulphide followed by the NaCl content. The effect of temperature is also very significant for solution-annealed standard 22 Cr DSS (a), although this has much less influence in the other cases (b-d).

Test	Temperature, °C	p.p.$_{H2S}$, bar	NaCl, wt. %	Result	Reference
SSRT	225	0.1	2.5	Pass	18
SSRT	225	0.3	2.5	Pass	18
SSRT	225	1	2.5	Pass	18
SSRT	300	0	2.5	Pass	18
SSRT	300	0.1	2.5	Pass	18
SSRT	300	0.3	2.5	Pass	18
SSRT	300	1	2.5	Pass	18
SSRT	300	10.1	2.5	Fail	18
U-bend	250	0.36	5	Fail	19
U-bend	200	0.1	5	Fail	20
C-ring	300	1	5	Pass	21
C-ring	250	0.1	5	Pass	21
C-ring	300	0.1	5	Pass	21

Table III. Parameters of nearest neighbours to the condition of interest, X, as shown on Figure 9.

Further exploration of the neural network algorithms has been possible with the use of neural-fuzzy technology. Models have been generated which "mimic" the MLF algorithms by using the MLF network predictions as the output data in training. Rules extracted from these networks have then been modified, by cutting out rules that are known to be wrong, or changing the confidence attached to rules to that of the nearest rules. The rules extracted from the networks, together with the performance of the prediction based on these rules are shown in Figures 11–14.

For example, the network may have "decided" from the data that the effect of increasing the yield strength is to increase slightly the resistance to SSCC. However, for cold-worked materials, this is contrary to knowledge elicited from the literature[2] and therefore the relevant rules have been removed. In other cases, confidences in rules have been slightly adjusted in order to maintain logic. For example, for cold-worked high alloy DSS (Fig. 14), the effect of increasing H_2S appeared to increase the resistance to SSCC (rule 6). This cannot be the case and therefore the confidence attributed to that rule has been set at the same value as that attached to what is deemed to be the closest rule (i.e. rule 5).

Having modified the neural-fuzzy networks, the rules extracted from each of the neural-fuzzy networks (Figures 11-14) are summarised as follows:

Solution-annealed 22 Cr DSS (Figure 11)

- As partial pressure of hydrogen sulphide increases, the susceptibility to SSCC increases.
- As NaCl content increases, the susceptibility to SSCC increases.
- This steel is most susceptible to SSCC around 90 ºC.
- An increase in yield strength increases the susceptibility to SSCC.

Cold-worked 22 Cr DSS (Figure 12)

- As the partial pressure of hydrogen sulphide increases, the susceptibility to SSCC increases.
- As NaCl content increases, the susceptibility to SSCC increases. This effect is much more pronounced at higher temperatures.
- An increase in pH reduces the susceptibility to SSCC.

Solution-annealed high alloy DSS (Figure 13)

- In general, as partial pressure of hydrogen sulphide increases, so does the susceptibility to SSCC. However, the resistance to stress corrosion cracking decreases with increasing hydrogen sulphide when higher levels of NaCl are present.

Cold-worked high alloy DSS (Figure 14)

- As the partial pressure of hydrogen sulphide increases, the susceptibility to SSCC increases.
- As the NaCl content increases, the susceptibility to SSCC increases. However, the effect of both NaCl content and hydrogen sulphide pressure is much lower at higher pH values.
- Increasing the temperature reduces the susceptibility to SSCC.

All of the above extracted rules coincide with existing knowledge of duplex stainless steels and confirm that fuzzy neural networks can be used to extract "quantitative" rules on the effect of the material and environmental variables on sulphide stress corrosion cracking. Figures 11-14 also show that the fuzzy neural network constructed based on the above rules give reasonably accurate predictions of sulphide stress corrosion cracking of duplex stainless steels. However, the rules generated by the fuzzy neural networks are not based on a physical model and hence should be used with caution.

USER-FRIENDLY SOFTWARE

A description of the development of algorithms to predict general corrosion of duplex stainless steels have is given elsewhere.[22] A user-friendly software package that incorporates algorithms for sulphide stress corrosion cracking and general corrosion of duplex stainless steels has been developed, as shown in Figure 15. By querying the trained MLF neural networks with input parameters that are of practical relevance, the users can generate predictions in their selected environments. Where appropriate, confidence indicators are presented alongside these predictions. The user can also generate corrosion maps demarcating the transition from a high risk of SSCC to a low risk, as in Figure 9. Kohonen neural networks are also be available to recall data that are closest to the condition of interest, as in Table III. In this way, the user can back-up the MLF predictions with *actual* data and identify references in which the most relevant tests have been carried out. It also serves to identify regions in which test data may be sparse.

Conclusions

- Algorithms for predicting the susceptibility of duplex stainless steels to sulphide stress corrosion cracking have been developed. A multi-layer feed-forward neural network has been shown to provide predicted probabilities of "passing" a test in a given sour environment, whilst a Kohonen network has been shown to provide ready access to data closest to the conditions of interest. Neural-fuzzy networks allow extraction of the most important "rules" associated with each material's behaviour.

- By querying neural networks with a range of input parameters of practical relevance it is possible to demarcate the transition from regions of high to low risk.

References

1. Coleman, D., Zhou, S., and Turnbull, A., "Modern tools for the extraction of information from Duplex Stainless Steel Database", NPL Report CMMT(D)191, February 1999.
2. Gunn, R. N., *Duplex Stainless Steels: Microstructure, properties and applications*, Abington Publishing, Cambridge, 1997.
3. Cottis, R. A., and Newman, R. C., *Stress Corrosion Cracking Resistance of Duplex Stainless Steels*, HSE OTH 94440, 1995.
4. Liebowitz, J., (ed.), *The Handbook of Applied Expert Systems*, CRC, 1998, pp. 13-2.
5. Rosen, E. M., and Silverman, D. C., "Corrosion Prediction from Polarisation Scans Using an Artificial Neural Network Integrated with an Expert System", Corrosion, NACE, September 1992, pp. 734-745.
6. Cai J., Cottis R. A., Lyon S. B., "Phenomenological modelling of atmospheric corrosion using an artificial neural network", Corrosion Science, Vol. 41, 1999, pp. 2001-2030
7. Urquidi-Macdonald, M., and Macdonald, D., "Performance Comparison Between a Statistical Model, a Deterministic Model, and an Artificial Neural Network Model for Prediction Damage from Pitting Corrosion", J. Research of National Institute Standards and Technology, Vol. 99, No. 4, 1994, pp. 495-504.
8. Smets, H. M. G., and Bogaerts, W. F. L., "SCC Analysis of Austenitic Stainless Steels in Chloride-Bearing Water by Neural Network Techniques", Corrosion, NACE, August 1992, pp. 618-623.
9. Urquidi-Macdonald, M., and Egan, P., "Validation and Extrapolation of Electrochemical Impedance Spectroscopy Data", Corrosion Reviews, Vol. 15, No. 1-2, 1997, pp. 169-194.
10. Nesic, S., Postlethwaite, J., and Vrhovac, M., "CO_2 Corrosion of Carbon Steel - From Mechanistic to Empirical Modelling", Corrosion Reviews, Vol. 15, No. 1-2, 1997, pp. 211-240.
11. Trasatti, S. P., and Mazza, F., "Crevice Corrosion: A Neural Network Approach", British Corrosion Journal, Vol. 31, No. 2, 1996, pp. 105-112.
12. Silverman, D. C., "Artificial Neural Network Predictions of Degradation of Non-Metallic Lining Materials from Laboratory Tests", Corrosion, Vol. 50, No. 6, NACE, 1994, pp. 411-418.
13. Ben-Haim, M., and Macdonald, D., "Modelling Geological Brines in Salt-Dome Level Nuclear Waste Isolation Repositories by Artificial Neural Networks", Corrosion Science, Vol. 36, No. 2, 1994, pp. 385-393.
14. Lu, P-C., and Urquidi-Macdonald, M., "Prediction of IGSCC in Type 304 Stainless Steel Using an Artificial Neural Network", Corrosion '94 Conference, NACE International, Paper No. 151.
15. Barton TF, Tuck DL, Wells DB, *The Identification of Pitting and Crevice Corrosion Spectra in Electrochemical Noise Using an Artificial Neural Network*, 'Electrochemical Noise Measurement for Corrosion Applications', ASTM STP 1277, Kearns, Scully, Roberge, Reichart, and Dawson (Eds.), American Society for Testing and Materials, 1996, pp. 157-169
16. Tarassenko, L., *A Guide to Neural Computing Applications*, Arnold, UK, 1998.
17. Kohonen T, "Self-organized formation of topologically correct feature maps", Biological Cybernetics, Vol. 43, 1982, pp. 59-69
18. Onoyama M, Shitani K, Hayashi N, Murata M, "Resistances of some high alloys to CO_2-H_2S Corrosion at Elevated Temperatures", Corrosion '84 Conference, NACE Intl., Paper No. 291
19. Barteri M, Mancia F, Tamba A, Bruno R, "Microstructural study and corrosion performance of duplex and superaustenitic steels in sour well environment", Corrosion '86 Conference, NACE Intl., Paper No. 157
20. Barteri M, Rondelli G, Scoppio L, Tamba A, " Cold working and composition effect on the performance of duplex stainless steels for OCTG", Duplex Stainless Steels '91 Conference, 1203-1218
21. Fliethmann J, Schlerkmann H, Schwenk W, "Autoclave investigation of stress corrosion cracking behaviour of Fe-Cr-Ni alloys in NaCl/CO_2/H_2S-environment", Werkstoffe und Korrosion, Vol. 43, 1992, pp. 467-474
22. Zhou, S., Coleman, D., and Turnbull, A., "Development of neural networks to predict general corrosion of duplex stainless", NPL Report MATC (A) 22, May 2001.

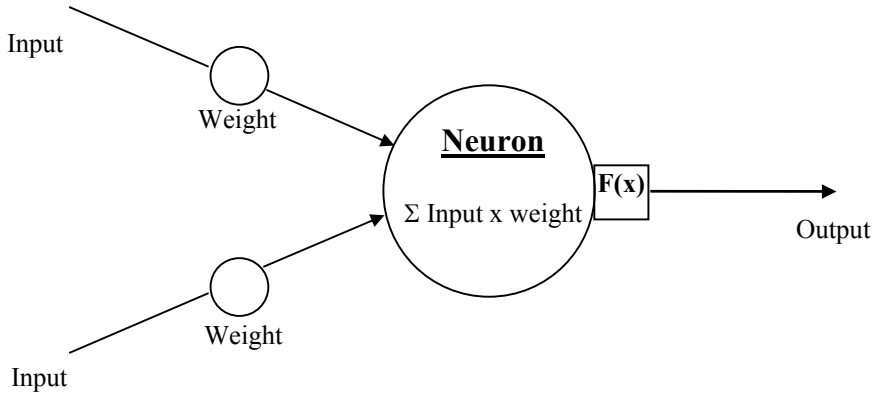


Figure 1. The overall structure and function of a simple perceptron.

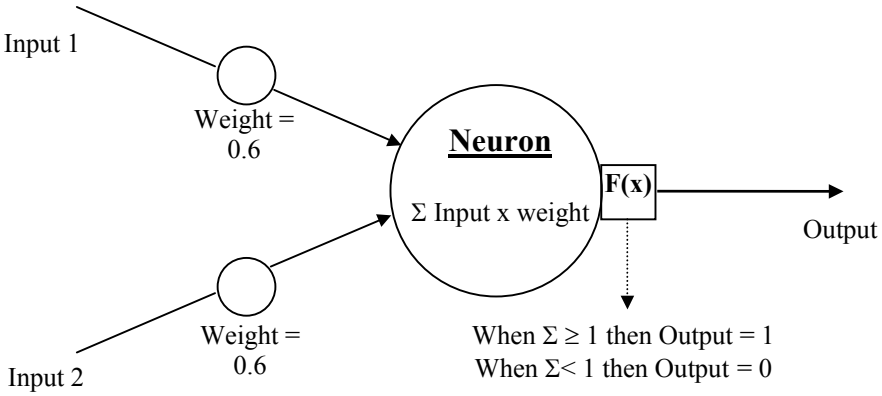
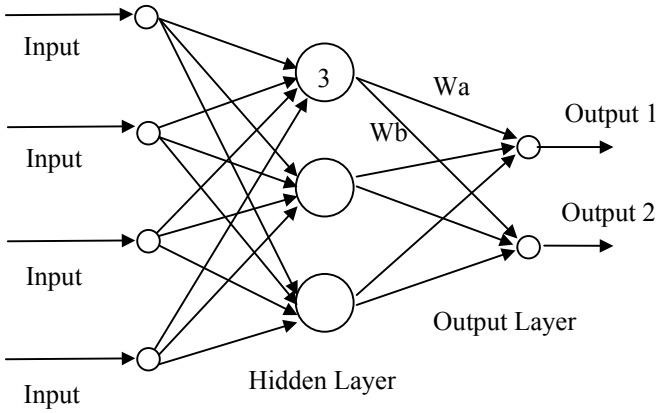


Figure 2. Example of a perceptron that can reproduce a simple pattern.



Wa and Wb are the respective weights on the connections between Neuron 3 and the outputs 1 and 2.

Error on Output 1 = Target value - Output Value = X_1
Error on Output 2 = Target value - Output Value = X_2
Error on Neuron 3 = (X_1 x Wa) + (X_2 x Wb) = X_3
New Wa = Wa + F(X_3)
New Wb = Wb + F(X_3)
And so on…

Figure 3. A multi-layer feed-forward neural network.

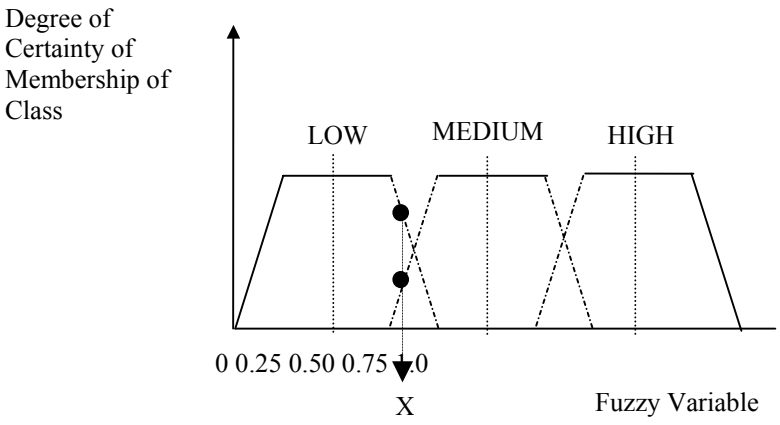


Figure 4. Examples of fuzzy classes that are not precisely defined. The fuzzy variable valued at X exhibits mainly properties associated with the LOW class. However, it also exhibits some properties of the next MEDIUM class.

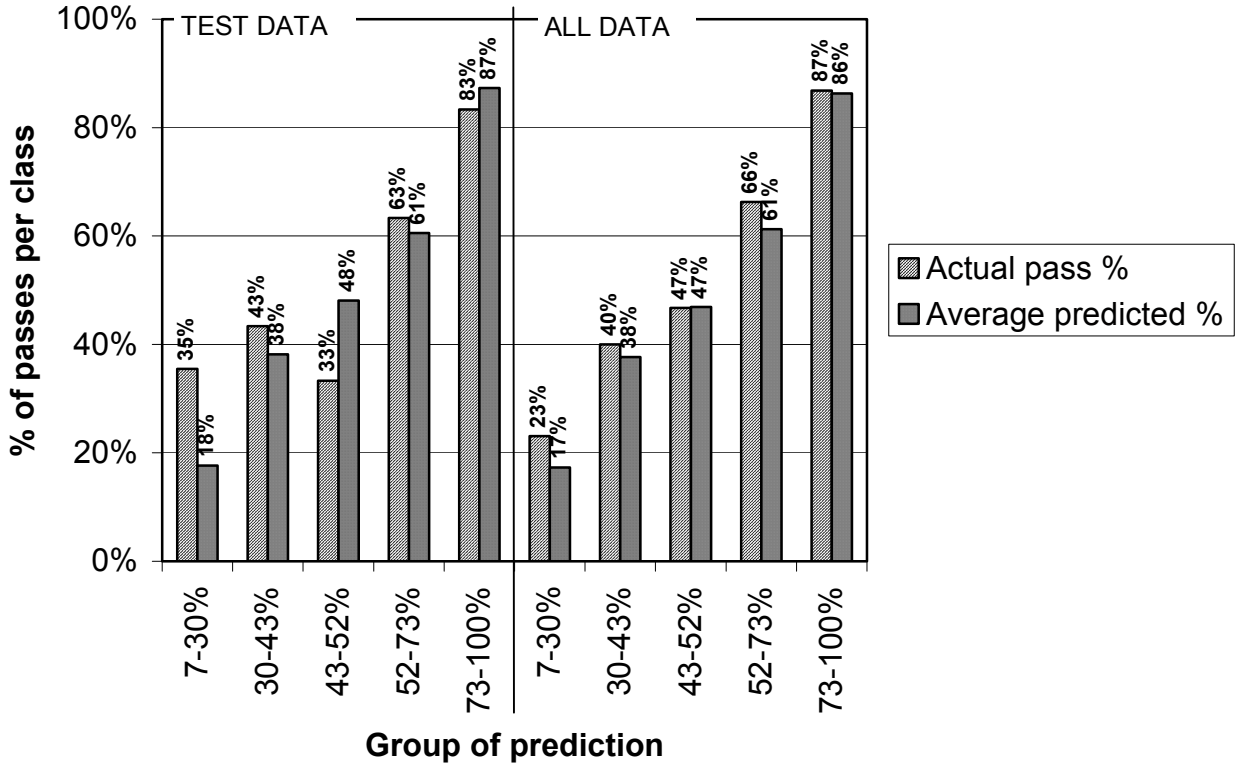


Figure 5. Performance of solution annealed standard 22 Cr DSS neural network.

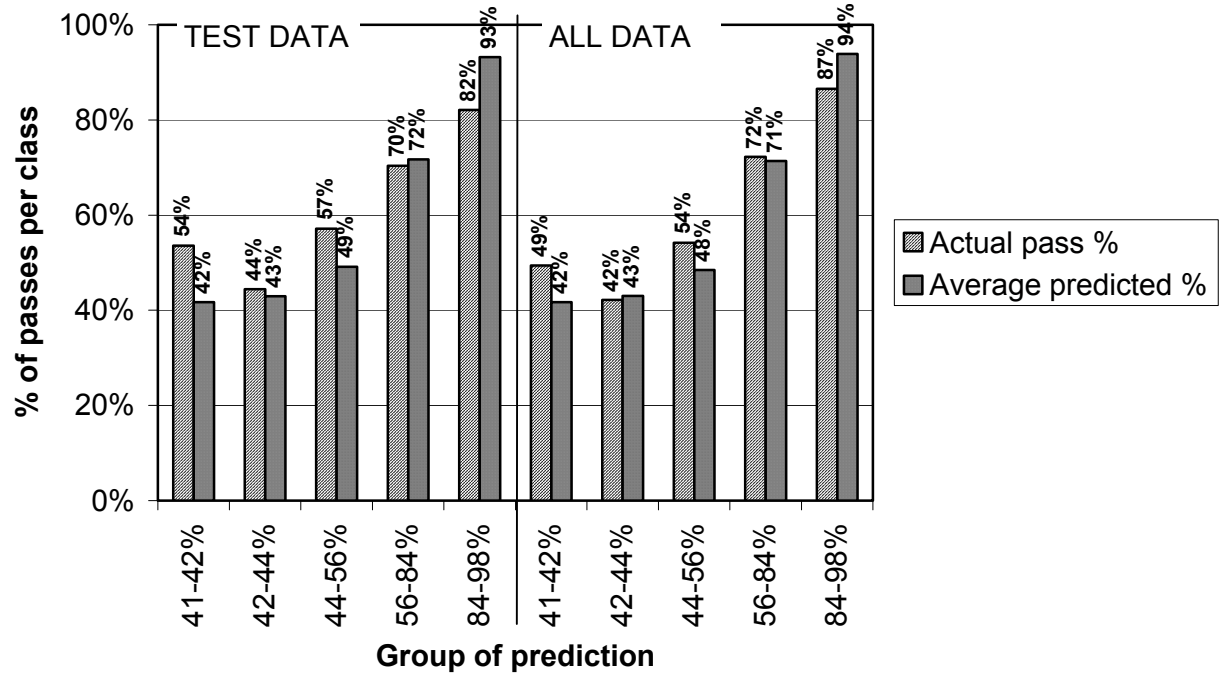


Figure 6. Performance of cold-worked standard 22 Cr DSS neural network.

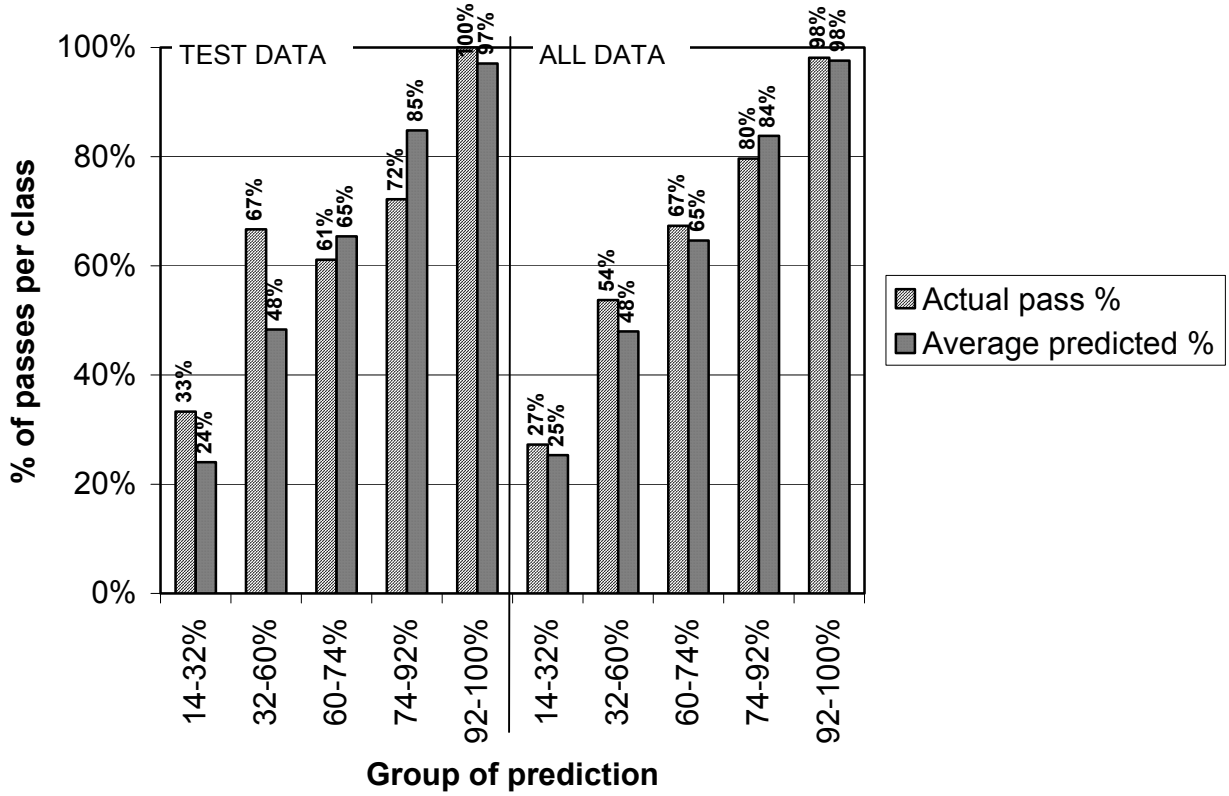


Figure 7. Performance of solution annealed high alloy DSS neural network.

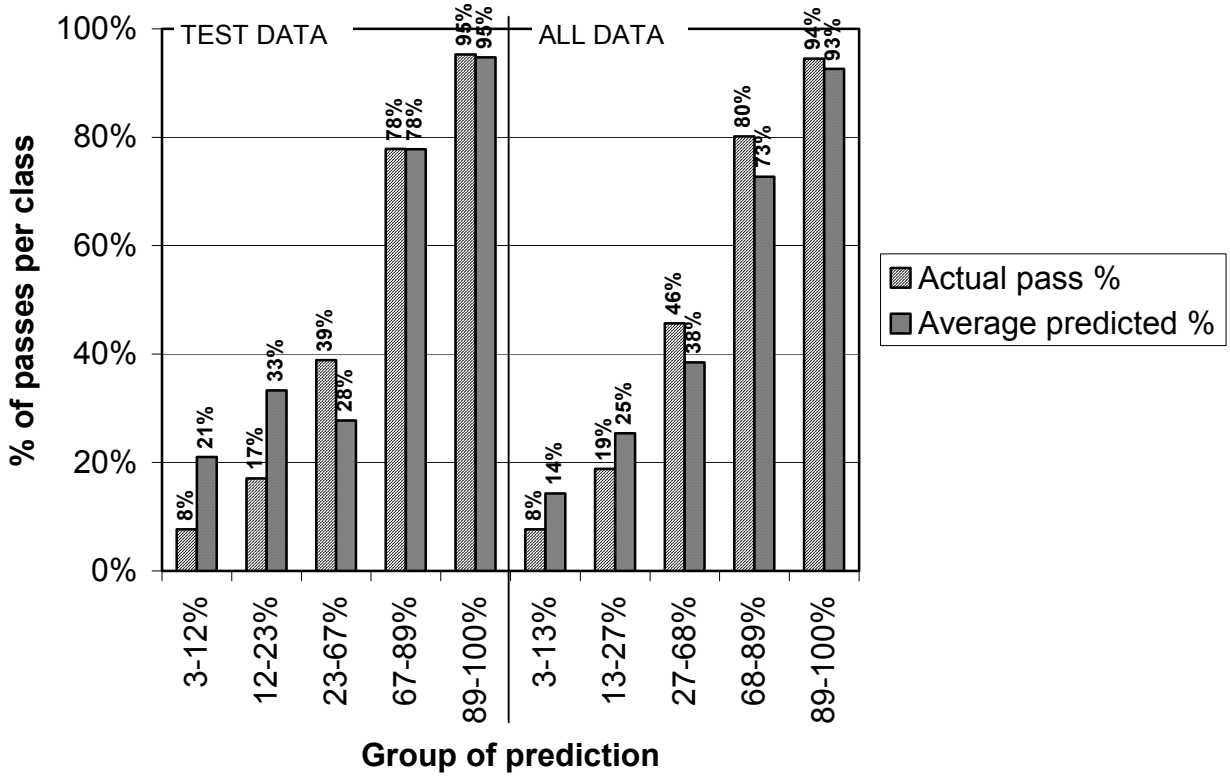


Figure 8. Performance of cold-worked high alloy DSS neural network.

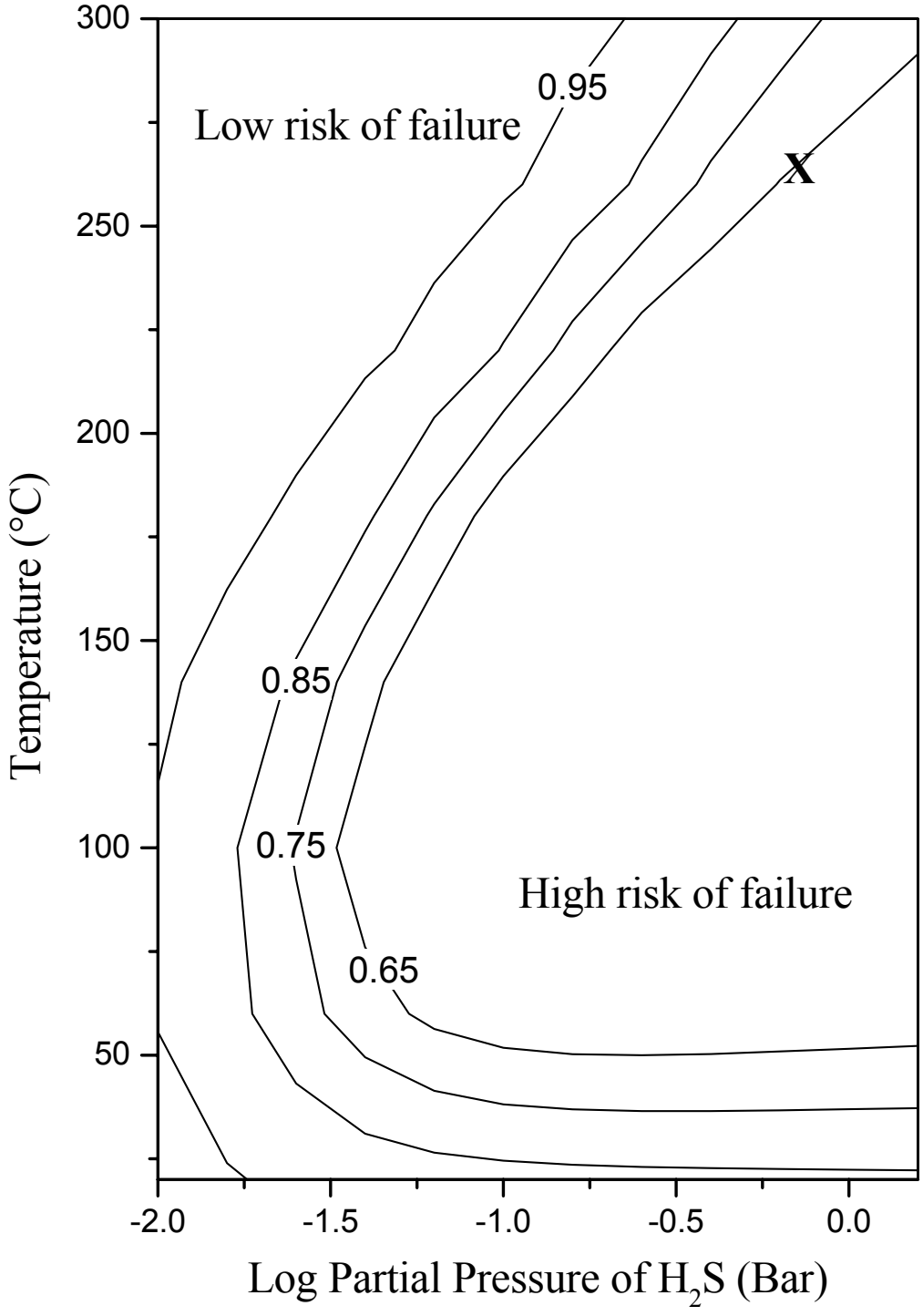


Figure 9. Change in Predicted probability of "pass" with temperature and H_2S. Solution annealed standard 22 Cr Duplex Stainless Steel; Yield Strength = 450 MPa; pH = 3.5; 4.5 wt. % NaCl.

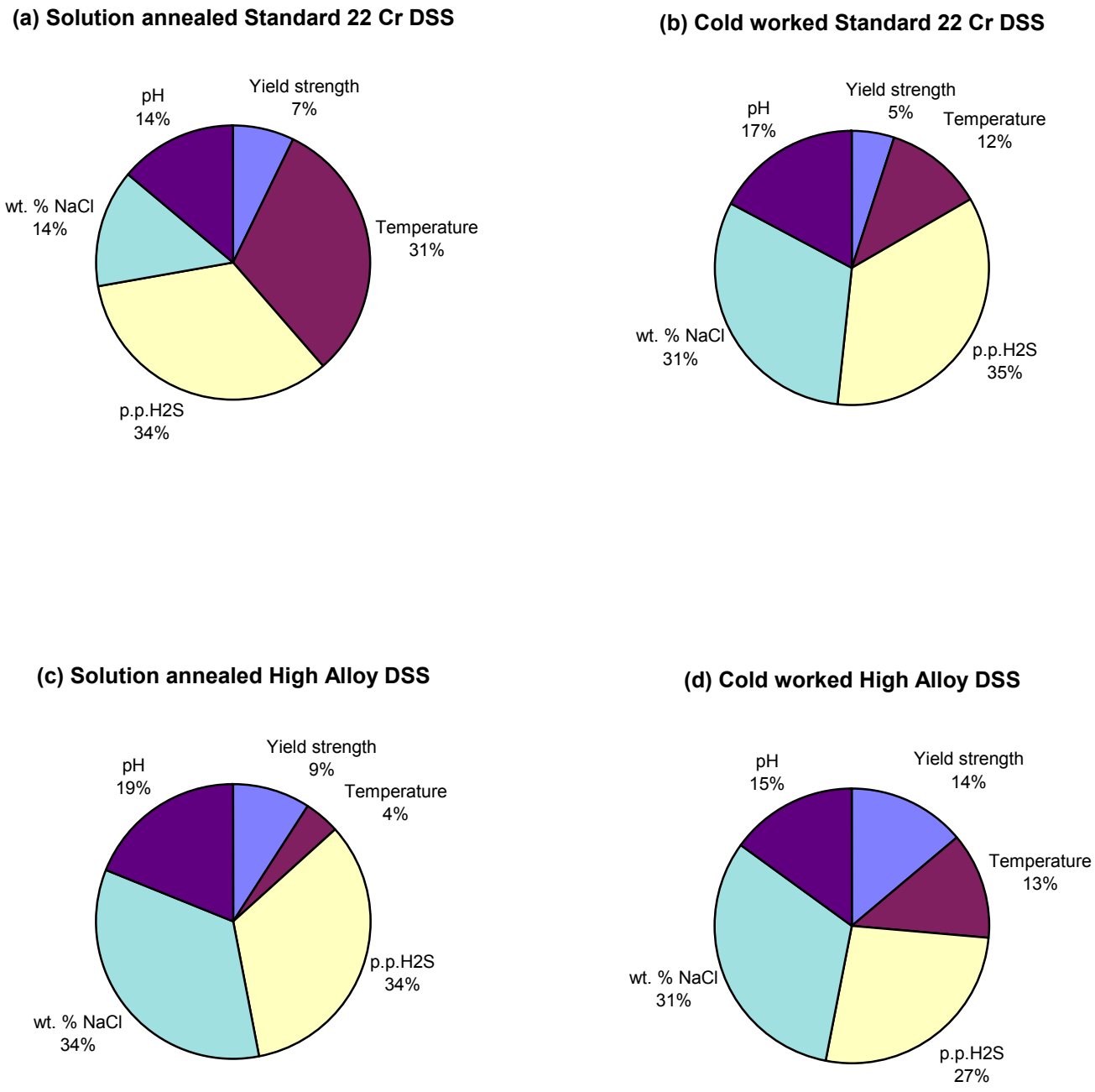
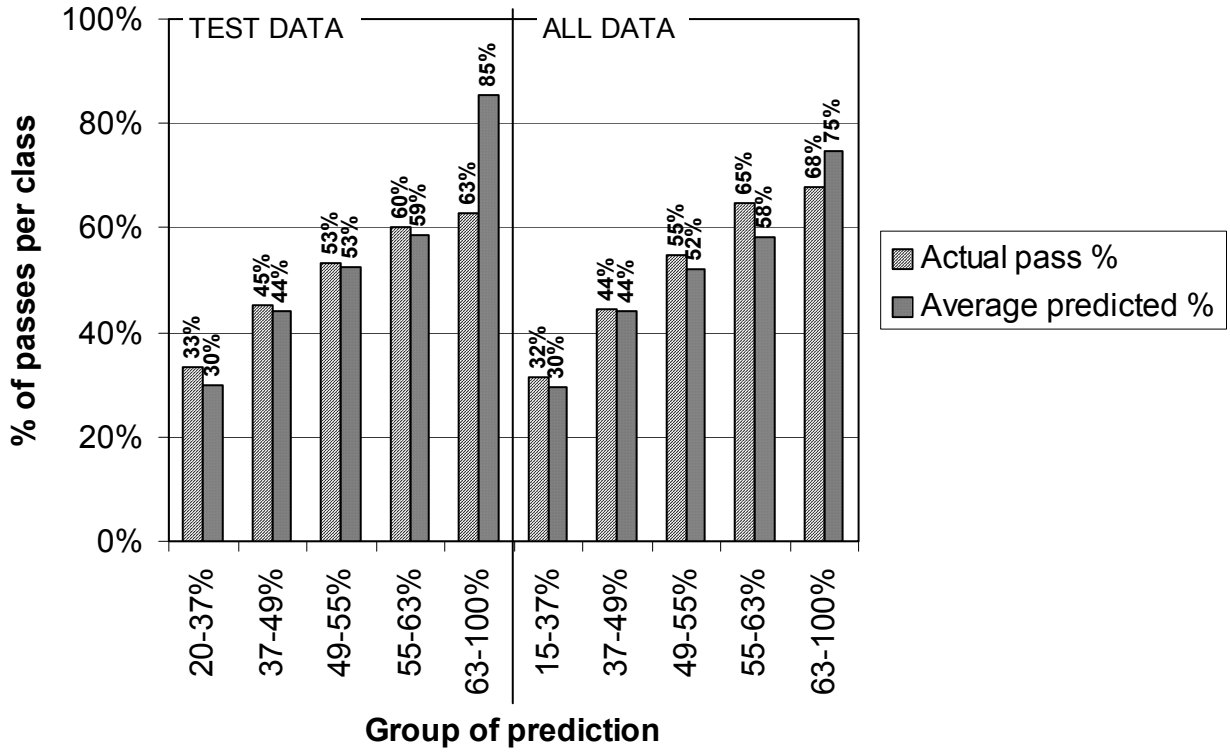


Figure 10. Relative importance of input parameters for each set of algorithms.



Rules:

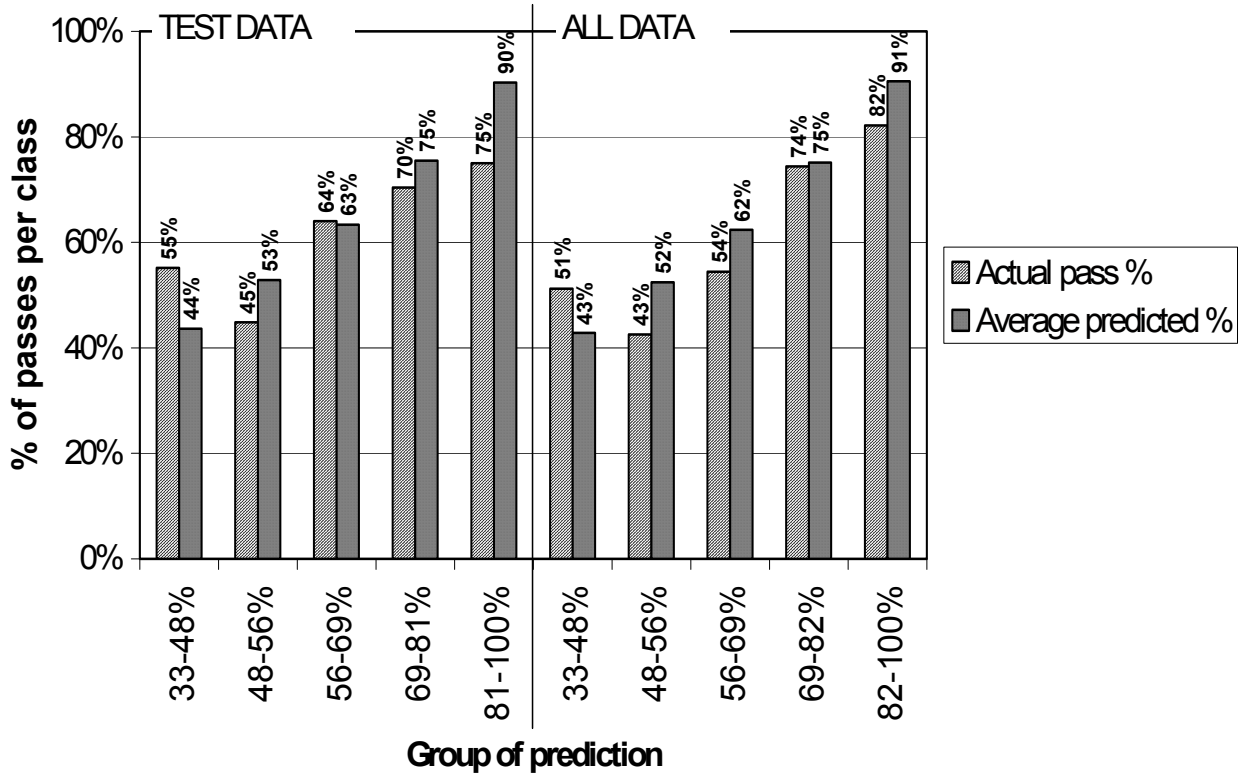
1: IF Log $p.p._{H2S}$ is low THEN Resistance is high (1.00)
2: IF Log $p.p._{H2S}$ is med THEN Resistance is low (0.68) OR Resistance is high (0.32)
3: IF Log $p.p._{H2S}$ is high THEN Resistance is low (0.87) OR Resistance is high (0.13)

4: IF Log NaCl is low THEN Resistance is low (0.53) OR Resistance is high (0.47)
5: IF Log NaCl is high THEN Resistance is low (0.70) OR Resistance is high (0.30)
6: IF Log NaCl is very high THEN Resistance is low (1.00)

7: IF Temperature is low THEN Resistance is low (0.66) OR Resistance is high (0.34)
8: IF Temperature is about 90 C THEN Resistance is low (0.86) OR Resistance is high (0.14)
9: IF Temperature is high THEN Resistance is low (0.35) OR Resistance is high (0.65)

10: IF Yield strength is low THEN Resistance is low (0.57) OR Resistance is high (0.43)
11: IF Yield strength is high THEN Resistance is low (0.85) OR Resistance is high (0.15)

Figure 11. Performance of solution annealed standard 22 Cr DSS neural-fuzzy network and "rules" generated



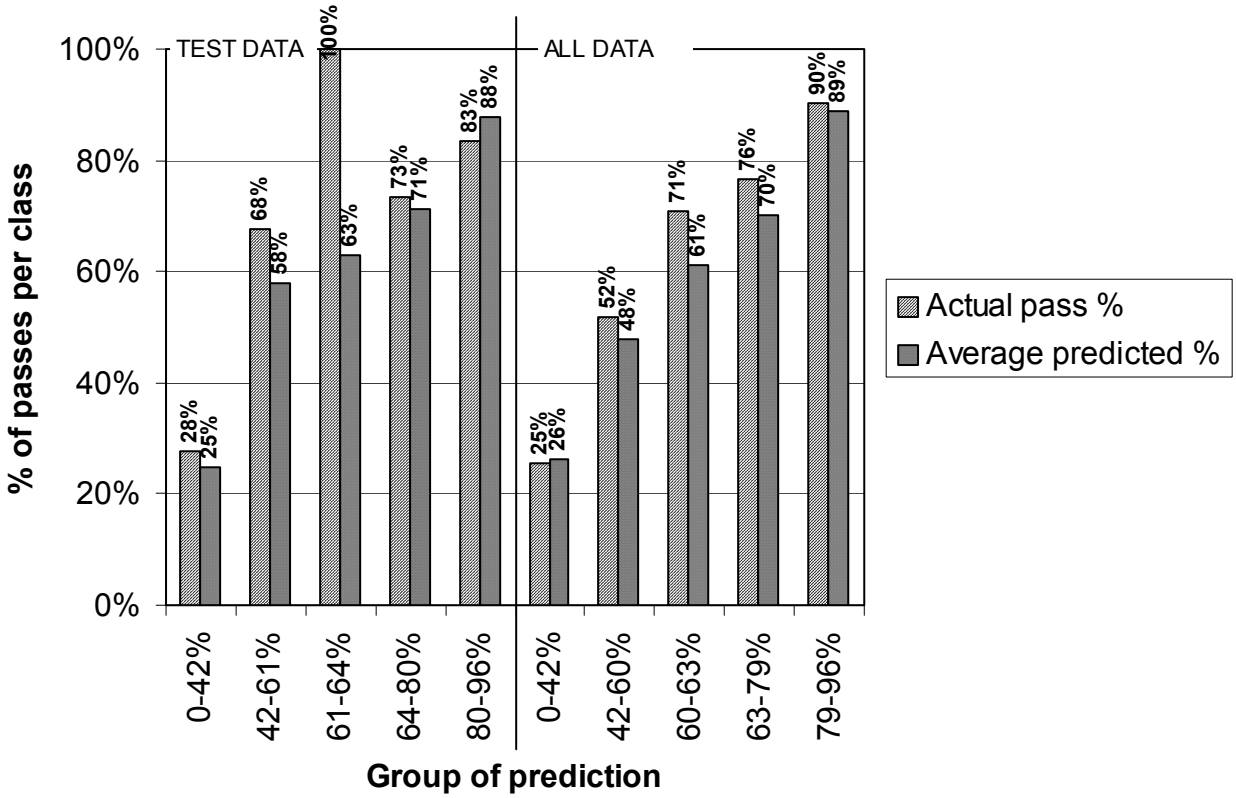
Rules:

1: IF Log $p.p._{H2S}$ is low THEN Resistance is low (0.53) OR Resistance is high (0.47)
2: IF Log $p.p._{H2S}$ is med THEN Resistance is low (0.78) OR Resistance is high (0.22)
3: IF Log $p.p._{H2S}$ is high THEN Resistance is low (1.00)

4: IF Temperature is low AND Log NaCl is low THEN Resistance is low (0.74) OR Resistance is high (0.26)
5: IF Temperature is high AND Log NaCl is low THEN Resistance is high (1.00)
6: IF Temperature is low AND Log NaCl is med THEN Resistance is low (0.78) OR Resistance is high (0.22)
7: IF Temperature is high AND Log NaCl is med THEN Resistance is low (0.59) OR Resistance is high (0.41)
8: IF Temperature is low AND Log NaCl is high THEN Resistance is low (0.89) OR Resistance is high (0.11)
9: IF Temperature is high AND Log NaCl is high THEN Resistance is low (0.81) OR Resistance is high (0.19)

10: IF pH is low THEN Resistance is low (0.87) OR Resistance is high (0.13)
11: IF pH is med THEN Resistance is low (0.79) OR Resistance is high (0.21)
12: IF pH is high THEN Resistance is low (0.69) OR Resistance is high (0.31)

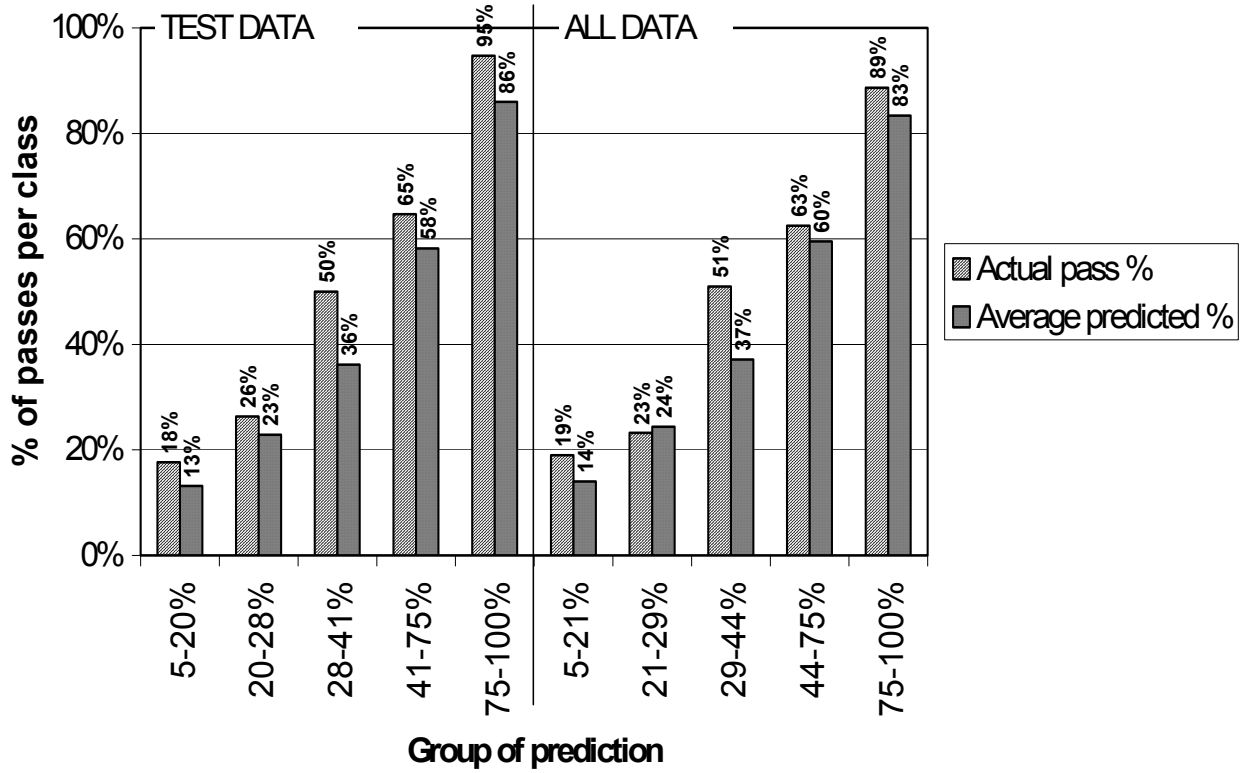
Figure 12. Performance of cold-worked standard 22 Cr DSS neural-fuzzy network and "rules" generated



Rules:

1: IF Log $p.p._{H2S}$ is low AND Log NaCl is low THEN Resistance is low (0.09) OR Resistance is high (0.91)
2: IF Log $p.p._{H2S}$ is med AND Log NaCl is low THEN Resistance is low (0.09) OR Resistance is high (0.91)
3: IF Log $p.p._{H2S}$ is high AND Log NaCl is low THEN Resistance is low (0.09) OR Resistance is high (0.91)
4: IF Log $p.p._{H2S}$ is low AND Log NaCl is med THEN Resistance is low (0.09) OR Resistance is high (0.91)
5: IF Log $p.p._{H2S}$ is med AND Log NaCl is med THEN Resistance is low (0.09) OR Resistance is high (0.91)
6: IF Log $p.p._{H2S}$ is high AND Log NaCl is med THEN Resistance is low (0.30) OR Resistance is high (0.70)
7: IF Log $p.p._{H2S}$ is low AND Log NaCl is high THEN Resistance is low (0.09) OR Resistance is high (0.91)
8: IF Log $p.p._{H2S}$ is med AND Log NaCl is high THEN Resistance is low (0.33) OR Resistance is high (0.67)
9: IF Log $p.p._{H2S}$ is high AND Log NaCl is high THEN Resistance is low (1.00)

Figure 13. Performance of solution annealed high alloy DSS neural-fuzzy network and "rules" generated



Rules:

1: IF Log p.p.$_{H2S}$ is low AND Log NaCl is low AND pH is low THEN Resistance is low (0.26) OR Resistance is high (0.74)
2: IF Log p.p.$_{H2S}$ is high AND Log NaCl is low AND pH is low THEN Resistance is low (0.42) OR Resistance is high (0.58)
3: IF Log p.p.$_{H2S}$ is low AND Log NaCl is high AND pH is low THEN Resistance is low (0.58) OR Resistance is high (0.42)
4: IF Log p.p.$_{H2S}$ is high AND Log NaCl is high AND pH is low THEN Resistance is low (1.00)
5: IF Log p.p.$_{H2S}$ is low AND Log NaCl is low AND pH is high THEN Resistance is low (0.26) OR Resistance is high (0.74)
6: IF Log p.p.$_{H2S}$ is high AND Log NaCl is low AND pH is high THEN Resistance is low (0.26) OR Resistance is high (0.74)
7: IF Log p.p.$_{H2S}$ is low AND Log NaCl is high AND pH is high THEN Resistance is low (0.26) OR Resistance is high (0.74)
8: IF Log p.p.$_{H2S}$ is high AND Log NaCl is high AND pH is high THEN Resistance is low (0.89) OR Resistance is high (0.11)

9: IF Temperature is low THEN Resistance is low (0.92) OR Resistance is high (0.08)
10: IF Temperature is high THEN Resistance is low (0.71) OR Resistance is high (0.29)

Figure 14. Performance of cold-worked high alloy DSS neural-fuzzy network and "rules" generated

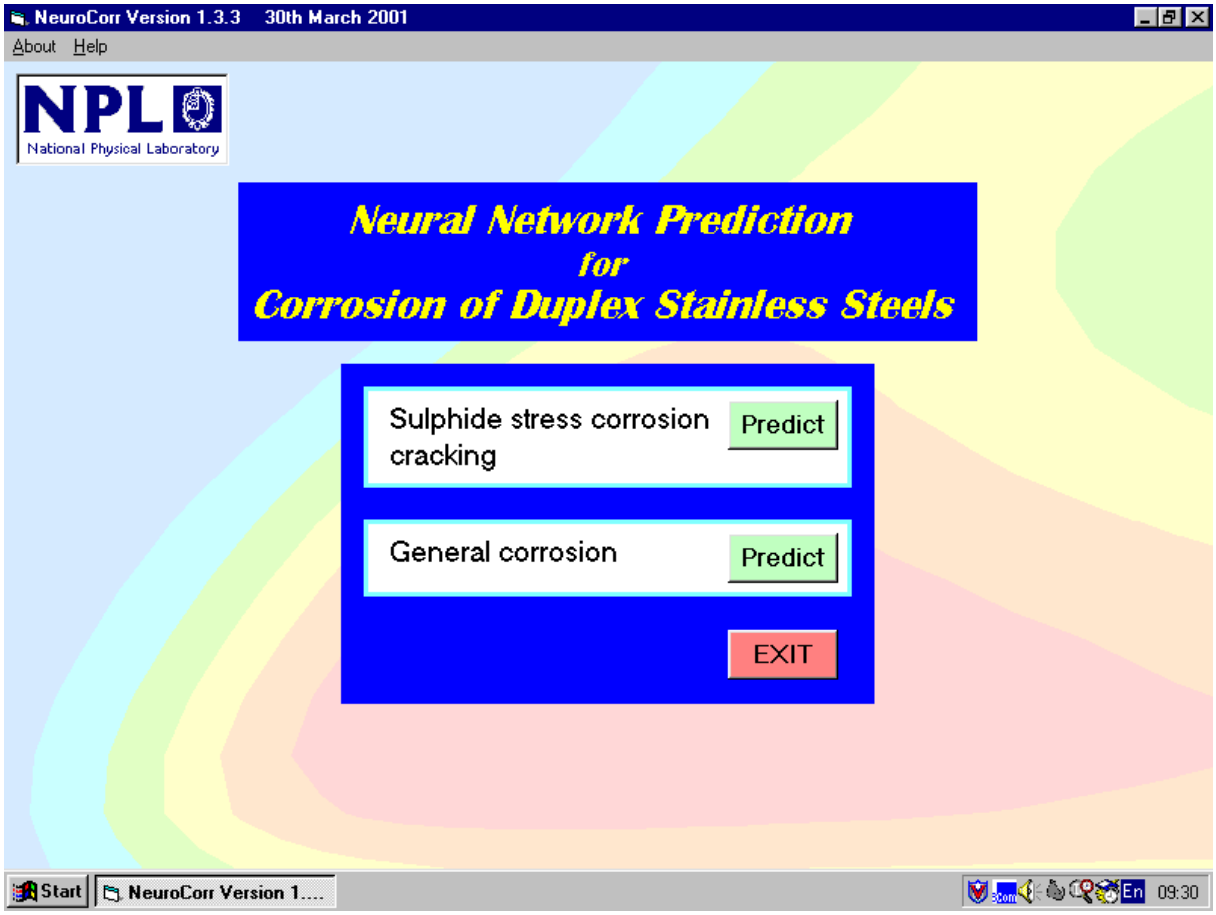


Figure 15. User-friendly interface of the neural networks software for predicting general corrosion and sulphide stress corrosion cracking of duplex stainless