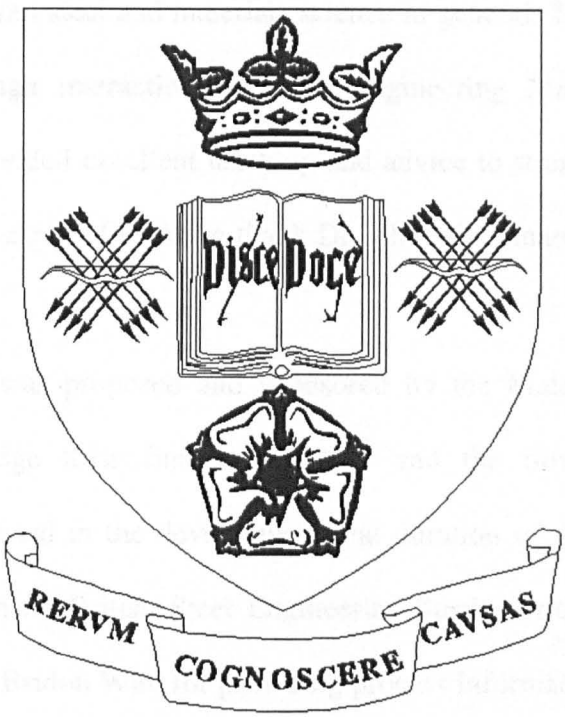
Optimisation of the Heat Treatment of Steel using Neural Networks

Jonathan Tenner



UNIVERSITY OF SHEFFIELD

Thesis submitted to the Department of Automatic Control & Systems Engineering in partial fulfilment for the degree of Doctor of Philosophy

October 1999

Acknowledgements

I would like to thank my supervisor, Professor Derek Linkens, for the right balance in support and allowance for creative freedom he has provided throughout this project.

Additionally, I would like to thank the members of the Intelligent Systems Laboratory for their friendship and help throughout the past three years. The research involved in this project has an interdisciplinary subject matter, part of its challenge was to learn more about steel and materials science in general. This has largely been made possible through interaction with the Engineering Materials Department, whose staff have provided excellent teaching and advice to strengthen my materials background. I would especially like to thank Dr John Whiteman for his experienced words.

This project was proposed and sponsored by the Materials Forum, and I gratefully acknowledge their financial support and the time the participating companies have invested in the development and duration of this project. I would particularly like to thank British Steel Engineering Steels, British Steel Technical, Aurora Forgings and Bridon Wire for providing process information and site visits at the beginning of the project.

Extensive interaction has continued with British Steel Engineering Steels and British Steel Technical throughout the project, with regular project review meetings involving Dr Peter Morgan, Dr Peter Morris, Mr Tevor Bailey and Mr J Pease, whose comments and guidance have been much appreciated. I especially acknowledge the help of Trevor for his work in the data cleaning stage of this project, and Peter Morris for his metallurgical modelling guidance.

Financial support from the Sheffield Metallurgical Engineering Association and the Worshipful Guild of Scientific Instrumentation Makers is also gratefully acknowledged.

To Rachel

Summary

Optimisation of the Heat Treatment of Steel using Neural Networks

Jonathan Tenner

Heat treatments are used to develop the required mechanical properties in a range of alloy steels. The typical process involves a hardening stage (including a quench) and a tempering stage. The variation in mechanical properties achieved is influenced by a large number of parameters including tempering temperature, alloying elements added to the cast, quench media and product geometry, along with measurement and process errors. The project aim was to predict the mechanical properties, such as Ultimate Tensile Strength, Proof Stress, Impact Energy, Reduction of Area and Elongation, that would be obtained from the treatment for a wide range of steel types.

The project initially investigated a number of data modelling techniques, however, the neural network technique was found to provide the best modelling accuracy, particularly when the data set of heat treatment examples was expanded to include an increased variety of examples.

The total data collected through the project comprised over 6000 heat treatment examples, drawn from 6 sites. Having defined a target modelling accuracy, a variety of modelling and data decomposition techniques were employed to try and cope with an uneven data distribution between variables, which encompassed non-linearity and complex interactions. Having not reached the target accuracy required the quality of the data set was brought into question and a structured procedure for improving data quality was developed using a combination of existing and novel techniques.

The stability of model predictions was then further improved through the use of an ensemble approach, where multiple networks contribute to each predicted data point. This technique also had the advantage of enabling the reliability of a given prediction to be indicated.

Methods of extracting information from the model were then investigated, and a graphical user interface was developed to enable industrial evaluation of the modelling technique. This led to further improvements enabling a user to be provided with an indication of prediction reliability, which is particularly important in an industrial situation.

Application areas of the models developed were then demonstrated together with a genetic algorithm optimisation technique, which demonstrates that automatic alloy design under optimal constraints can now be performed.

Contents

Chapter 1

Introduction

1.1 Aims

Specialist heat treatments are used to develop the required mechanical properties in a range of alloy steels. The heat treatment process consists of two main stages; hardening and tempering. The hardening stage involves the steel being heated to a temperature of typically 850°C for a period of time allowing transformation to austenite, following which it is quenched in an oil or water medium or is occasionally allowed to air cool. The tempering stage is required to improve ductility in the hardened material. Tempering involves heating the product to a lower temperature, for example 630°C, for a further period of time and then cooling again.

The mechanical properties of the material are dependent on many factors, however the tempering temperature is believed to be a major one. The process does involve many other variables, these include quench type used, composition of the steel, geometry of the bar, test sample location on the bar, batch distribution in the furnace, measurement tolerances and variations in process equipment and operators. A heat treatment metallurgist would balance these parameters through application of metallurgical knowledge and experience, to obtain the required mechanical properties.

It is not possible to accurately describe the process behaviour using mathematical models due to the complexity of the underlying physical mechanisms. With modern industrial processes being associated with large amounts of data, empirical models are increasingly being used to describe the industrial process. Through application of data modelling techniques, the aim was to utilise process data to build empirical models capable of predicting mechanical test results for steels covered by the range of the data available.

Once constructed, these models would allow optimisation of the heat treatment process by predicting the mechanical test results that would be achieved for a given treatment pattern on a particular steel product. It was envisaged that this would lead to improved product reliability and reduce process costs.

The types of mechanical test result of interest in this project include Ultimate Tensile Strength, Proof Stress, Reduction of Area, Elongation and Impact Energy. The precise nature of these tests will be described in chapter 3.

There are a number of methods that can be used to perform regression on data such as that obtained from an industrial process. One technique of particular interest was that of Artificial Neural Networks (ANN), however other relevant techniques of regression were evaluated, taking into account their suitability to this particular problem.

1.2 Project Foundation

IMMPETUS (Institute for Microstructural and Mechanical Process Engineering at The University of Sheffield) is an interdisciplinary research institute involving the Departments of Engineering Materials, Mechanical and Process Engineering, and Automatic Control and Systems Engineering. The project is part of this initiative and is supported by a Materials Forum Scholarship and has been proposed by British Steel Engineering Steels, Bridon Wire, Special Melted Products, Aurora Forgings and British Steel Technical. The Materials Forum is a group of steel manufacturers local to Sheffield, who support research with the aim of improving process techniques and knowledge.

The project was based in the Department of Automatic Control and Systems Engineering within the intelligent systems laboratory. There was however extensive

collaboration between University departments as well as with the industrial sponsors involved, with the aim of solving a practical problem to benefit British industry.

1.3 Brief Background of the Companies Involved

The companies involved in the project all perform heat treatment in some form on their steel products, however the production techniques of each company vary significantly depending on their business. To maximise the use of the industrial links associated with the project, preliminary visits were carried out to the main companies involved in the project. The purpose of these visits was to establish the part each company could play in the project's development.

British Steel Engineering Steels produce high-grade steels to order for a large variety of markets. The steels encompass a wide range of alloying elements. The specification required varies depending on the application, and the country ordering the product. It would usually, however, consist of Ultimate Tensile Strength and Impact requirements (the details of which will discussed in chapter 4), which are to be met for a specified range of temperatures. The mechanical properties are developed through a hardening and tempering process as mentioned briefly above. Product is usually in the form of rolled bars or slabs.

Bridon Wire manufacture wire for a range of applications which include wire rope for safety fences, rigging for boats, wires for mining and mineral extraction, suspension wires for bridges and strength members for fibre optic cables. The process is still essentially a heat treatment one, however unlike the other companies Bridon use a continuous heat treatment process as opposed to a batch one. Essentially this means that the product to be heat-treated passes though the hardening and tempering stages continuously.

Aurora Forgings produce precision forgings for a range of applications including Rolls Royce engines and earth moving machinery. The forgings are often manufactured from a range of alloy steels supplied by British Steel as rolled product or slabs that have not undergone the heat treatment process. The forging process causes mechanical work to be done on the steel and so heat treatment needs to follow the forging process in order to control the mechanical properties of the manufactured parts.

British Steel Technical is a research facility of British Steel. There are two sites, of which the Swinden Technology Centre in Rotherham was our main contact. The Technology Centre researches into many aspects of steel production and contains a wealth of expertise in the field of steel heat treatment. They aim to cater for short, medium and long-term research needs of the business. The purpose of their research is to improve product quality and process efficiency. Whilst the other companies involved in the project represented a good source of data and plant-based information, the Technology Centre offered interpretation of results and support on metallurgical issues.

1.4 Structure of the Thesis

This chapter provides a statement of the problem, together with a description of the companies involved in the project. What follows is a description of the key points covered in the following chapters.

Chapter 2 covers the main data modelling techniques that may be useful to the project. These include Linear, Polynomial, Neural Network and Partial Least Squares techniques. A comparison of accuracy and implementation requirements is made, together with details on the theoretical background behind each technique. Citations are made demonstrating where, within the related fields, each technique has

been used in heat treatment, together with their development and limitations. The citations provide an insight into the work that has already been done in this field, thereby providing clues as to the suitability of each technique to this particular application, whilst also highlighting where future work should concentrate.

Chapter 3 provides a detailed description of the heat treatment process from both a metallurgical and industrial perspective. Details of the specific processes from which data have been collected are described, together with their main sources of variability. This chapter also describes how data were collected from the processes, which are the important variables and what part they play in the resulting mechanical properties. Finally, the measurement error and incompatibility between mechanical test types are investigated along with the use of mechanical test facilities when supplying steels to companies such as Aurora.

Chapter 4 concentrates on the pre-processing aspects of data for empirical modelling. The data handling system used during the project together with methods to overcome data incompatibilities are described. Basic correlation analysis and initial modelling investigation are performed, aiming to break into subgroups the wide range of steels contained within the data, in order to establish the initial information within the data and evaluate the initial accuracy of the various data modelling techniques.

Chapter 5 investigates methods of data cleaning techniques, together with existing work in the area. The aim of data cleaning is to improve the predictive accuracy of the resulting models. A methodology for performing data cleaning on industrial process data is proposed and applied to the data set in the project. The improvement in neural modelling accuracy provided by the technique is then demonstrated experimentally.

Chapter 6 looks at possible ways in which the neural modelling technique can be improved. Having experimented with a variety of techniques, an ensemble approach, where a group of neural networks are used to make each prediction is employed, and careful consideration is given to the architecture of neural model used. The data set is then extended to include additional mechanical test results, and the data cleaning technique is applied to this additional data to develop further mechanical test result models. A comparison is made between the uncleaned and cleaned data sets for each new model developed.

Chapter 7 is concerned with enabling information extraction from the models developed. A sensitivity analysis method is investigated for finding the significance of each input variable to the model outputs, along with another technique, which allows the effects of each variable to be visualised. These techniques are used to help validate the functioning of the model. A Graphical User Interface (GUI) is then developed, which allows information extraction and validation of the model within industry. The feedback from these evaluation results shows that the model prediction accuracy is very dependent on the type of data used for prediction. Experimentation into possible techniques for assessing the reliability of the model's predictions is therefore conducted and a suitable solution is chosen and evaluated.

Chapter 8 considers using the developed models for process optimisation. A number of possible applications within industry are highlighted, followed by experimentation to demonstrate the ability of a model to generalise to a new process, and how measurement inaccuracies affect the models predicted values. Automatic and optimal design of alloys is then considered using the genetic algorithm optimisation approach, which demonstrates that the optimal design of alloys is now feasible.

Finally, chapter 9 details conclusions resulting from the work within this project, together with further research work required in this subject area.

Chapter 2

Data modelling techniques

2.1 Introduction

In the last chapter it was mentioned that there are a variety of regression techniques available. The principle of regression is the formulation of a model, relating one or more independent variables to one or more dependant variables. This project involved estimating a mechanical test result (the dependant parameter) from a set of process variables. For classical regression techniques this requires one to postulate a model form, for example linear, polynomial or some other non-linear function. Once a form of model is established, one then needs to estimate the values of the model parameters, this being done from a set of experimental data.

Once a model is established, interpolation can be performed to estimate the value of the dependant variable given the value of the independent variables.

Whilst a brief introduction to other techniques is made, the main content of the theory section of this chapter will be concerned with the neural network technique which was ultimately used in the project.

A literature review exploring the use of predictive models for mechanical property and other process parameters is then provided. This aims to summarise the current work that has been undertaken together with its limitations.

2.2 Least squares linear regression

The simplest form of regression is that of bivariate linear regression (regression of y on x). Such a linear model might take the form:

$$y = X\beta + \varepsilon \qquad (2.1)$$

In equation 2.1, y is an n by 1 vector of observations, X is an n by p matrix of regressors, β is a p by 1 vector of parameters and ε is an n by 1 vector of random disturbances. We therefore need to estimate the unknown vector of parameters β. This can be done using the least-squares solution of:

$$\hat{\beta} = (X'X)^{-1}X'y \tag{2.2}$$

Equation 2.2 is the basis of the least squares estimate, however it is important to note that there should not be dependancy between the regressors and hence methods such as orthoganal triangular decompositon are used to prevent the inversion of an ill-conditioned matrix.

We can then use the vector of estimated parameters to find the predicted y values as:

$$\hat{y} = X\hat{\beta} \tag{2.3}$$

2.3 Interaction effects and higher order regression

It has been assumed so far that the variable effects in a multiple linear regression case are additive. Interaction effects exist when the impact of one independent variable depends on the value of another independent variable. Such a model might take the form:

$$Y = \alpha_0 + \beta_1 x_1 + \beta_2 (x_1 x_2) + e \tag{2.4}$$

In a similar manner a variable effect might involve a quadratic or higher order term such as:

$$Y = \alpha_0 + \beta_1 x_1^{\,2} + \beta_2 x_2^{\,2} + e \tag{2.5}$$

Linear regression is a popular technique, firstly, because very many relationships have been found empirically to be linear. Secondly, it is sometimes argued that a

linear relationship is the most parsimonious. Thirdly, there is often little knowledge of what form the non-linear specification would be.

2.4 Principal component methods

Principal Component Analysis (PCA) is essentially a dimensionality-reduction technique, where, given a high dimensional data set which may have no obvious features, it is possible to describe the data via new variables which are linear combinations of the original variables. PCA is not a regression technique, however its function has been readily applied to two forms of regression, namely Principal Component Regression (PCR) and Partial Least Squares (PLS).

As well as regression, PCA can also be used for multi-dimensional data reduction and process fault detection; this is considered in chapter 5.

2.4.1. Principal Component Analysis (PCA)

When using the PCA technique, it is usual to scale data such that it is mean centred. If the data is represented by a set of p variables X_1, ..., X_p, PCA transforms this set of variables into a (preferably) much smaller set X'_1, ..., X'_k of linear combinations of the original variables $\{X_i\}$ which accounts for most of the variance of the original set. The new variables $\{X'_j\}$ are referred to as 'principal components' and are usually presented in the order of decreasing contribution to the total variance. The value of the j^{th} variable on the i^{th} object can be written as X_{ij}. All measurements for all variables can therefore be written in the form of an n x p matrix, which is known as the 'property matrix' :

$$X = \begin{bmatrix} X_{11} & \cdots & X_{1p} \\ X_{21} & \cdots & X_{2p} \\ \vdots & & \vdots \\ X_{n1} & \cdots & X_{np} \end{bmatrix} \tag{2.6}$$

The first principal component, X'_1, is defined as the linear combination of the original variables $\{X_i\}$:

$$X'_1 = \sum_{i=1}^{p} w_{1i} X_i \tag{2.7}$$

This combination is such that the variance of X'_1 is maximised, subject to the constraint:

$$\sum_{i=1}^{p} w_{1i}^2 = 1 \tag{2.8}$$

The coefficients w_{ij} can be thought of as components of a column vector. Subsequent principal components are defined analogously. The j^{th} principal component X'_j is the linear combination:

$$X'_j = \sum_{i=1}^{p} w_{ji} X_i \tag{2.9}$$

Whose variance is maximal under the constraints:

$$\sum_{i=1}^{p} w_{ji}^2 = 1$$

$$\sum_{i=1}^{p} w_{jk} w_{ki} = 0 \quad \text{for all } k<j \tag{2.10}$$

The variance-covariance matrix of the original values $\{X_i\}$ provides the solution of this constrained maximum problem. The variance-covariance matrix is defined in equation 2.11, where X_m is the mean corrected property matrix.

$$S = \frac{1}{n} X_m^T X_m \qquad (2.11)$$

The coefficients $\{w_{1i}\}$ of the linear combination defining the first principal component are given as the components of the eigenvector w_1 corresponding to the largest eigenvalue λ_1 of S. The coefficients of w may also be referred to as the 'scores'. The variances of the principal components are equal to their corresponding eigenvalues; therefore the percentage variance in the original data explained by one or more principal components can be calculated. Mardia et al provide a more detailed description of this technique[1].

PCA has also been combined with neural networks to improve learning problems in both supervised and unsupervised systems[2,3,4].

2.4.2. Principal Component Regression (PCR)

The PCR technique involves using the principal component transformation of the independent variables X in equation 2.2. The collinearity problem of multiple linear regression is now solved, since the component scores are mutually orthogonal. This means that an invertible matrix is guaranteed in equation 2.2. Only the principal components that are most intuitive need be included in the transformed independent variable matrix, such that the noise penalty of less significant variables can be reduced. There is a problem, however, that PCR is a two step method and thereby useful predictive information may end up in the discarded principal components.

2.4.3. Partial Least Squares regression (PLS)

This technique is similar to PCR except that the dependent variables are transformed as well as the independent variables. The components are now chosen such that the correlation between dependent and independent variables is maximised

as well as the retention of variance between the independent variables. The PLS model can therefore be considered as consisting of outer relations (input and output blocks individually) and an inner relation (linking both blocks).

The algorithm and full explanation of the PLS technique can be found in a tutorial by Geladi and Kowalski [5]. The PLS algorithm is regarded as one of the most flexible multivariate extensions of the linear regression problem. This flexibility allows it to be used in situations where the use of traditional multivariate methods would be severely limited, such as when there are fewer observations than predictor variables.

2.4.4. Extension of PCA and PLS to non-linear data sets.

As it stands, the PCA and PLS techniques described are designed to work on linear data sets (in that they cannot explain variance attributed to non-linear variables). There is however, a large amount of literature where attempts have been made to extend these linear techniques to cover non-linear data.

Hiden et al [6] investigated using a genetic programming method for this purpose. Other examples of non-linear PLS modelling involve the use of Neural Networks [7,8,9], however this is a rapidly developing field.

2.5 Neural networks: an introduction.

The neural network technique differs from the classical regression techniques in that a precise model structure does not have to be previously defined. Whilst the neural network technique is no more than a form of statistical non-linear function approximation, its foundation stems from modelling of biological nervous systems in the 1940's[10].

2.5.1 A Simple Artificial Neuron (SAN)

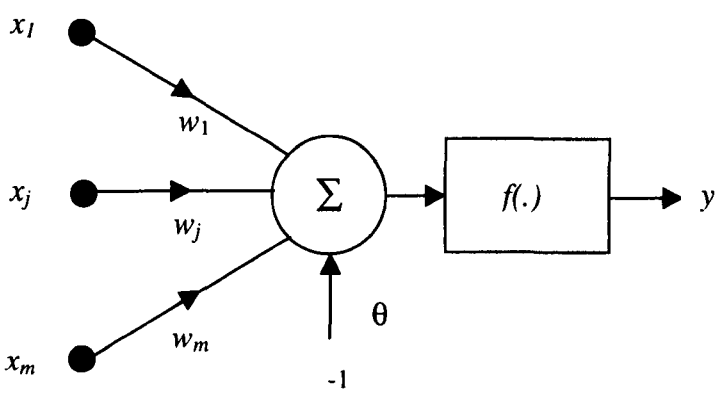


Fig.2.1. A simple artificial neuron

Neural networks were first used for classification problems before being used for regression purposes, although it is later shown that the technique's extension to regression is straightforward. A classification problem involves assigning input patterns to a class, for example given the binary inputs of an AND logic gate, a neuron can compute the binary output as class 1 or 0. MuCulloch and Pitts[10] first introduced the idea of an artificial neuron for processing data. Following this, neurons were arranged into layers to perform more complex pattern recognition tasks - this work was performed by Rosen in the late 1950s. A diagram of a Simplified

Artificial Neuron (SAN) is shown in Figure 2.1. The SAN represents the abstraction of several properties from the study of natural neural dynamics, in a vastly simplified form. The weights w, operate on the inputs to the neuron and are variable. The neuron can be used for simple pattern classification tasks in the following manner. The net input, which is the weighted sum of its n inputs, can be calculated as:

$$net_i = \sum_{j=1}^{m} w_{ij} x_j \tag{2.12}$$

where w_{ij} is used to denote the weights acting between the j th input and the i th neuron. The term, θ, shown in Figure 2.1 acts as a threshold, such that the net input must equal or exceed it for neuron activity. A function of the net input is then used to denote the state of the neuron, this being termed an activation function, a simple version of which is the step function which can be defined as:

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.13}$$

If f(net-θ) is used where f(.) is the step function shown in (2.13), then the output of the neuron, y, for a discrete time interval t is:

$$y(t) = \begin{cases} 1 & net \geq \theta \\ 0 & net < \theta \end{cases} \tag{2.14}$$

The weights of the SAN can therefore be adjusted such that, given two binary inputs, the output of the neuron represent that of the AND logic gate. It has therefore been seen that a SAN can be used to represent a two-class binary classification problem. The drawback with this approach, however, is that the user needs to set the weights manually, which implies that the problem has to be solved beforehand. An automatic method of finding the weights was therefore required and was invented in the form of the simple Perceptron.

2.5.2. The simple Perceptron

The simple Perceptron allows the process of weight adjustment to be automated, the correct weights can be found using previous experience (past data). Rosenblatt's Perceptron was originally designed as a pattern classifier, and is capable of classifying data which, in a two-dimensional sense is separable by a straight line or in an N-dimensional sense is separable by a (N-1) dimensional hyperplane. The simple Perceptron is shown in Figure 2.2. The function of the Perceptron is similar to that of the simple artificial neuron, except that the bias θ is replaced by w_0 with an input of –1, and that there is now error feedback and a learning algorithm.

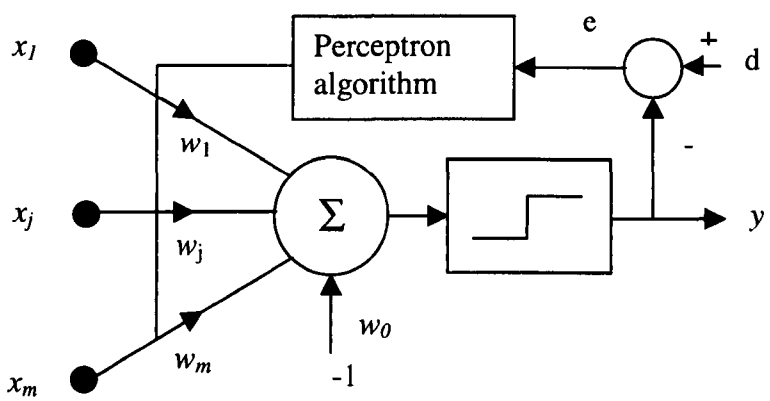


Fig.2.2. A simple Perceptron.

Equation (2.13) is still used as the node activation function and the net output is therefore given by:

$$net_i = \sum_{i=1}^{m} w_i x_i - w_0 = \sum_{i=0}^{m} w_i x_i \qquad (2.15)$$

where $x_0 = -1$, is the bias input. The output y can therefore be written as:

$$y(\mathbf{x}) = y(net) = \begin{cases} 1 & if\ \mathbf{w}^t\mathbf{x} \geq 0 \\ 0 & if\ \mathbf{w}^t\mathbf{x} < 0 \end{cases} \qquad (2.16)$$

where equation (2.15) is represented in vector form.

The Perceptron learns from data which comprises two finite sets, such that:

set X_1 of vectors **x**, belong to class 1 (desired output d=1) and,

set X_2 of vectors **x**, belong to class 0 (desired output d=0),

In order to achieve automatic learning, the following algorithm can then be followed:

1. Initialise the weights to small random values (including the threshold bias w_0).
2. The input vector is then presented, $\boldsymbol{x}(t) = (-1, x_1(t),\ldots,x_m(t))^t$, together with the desired response $d(t)$ corresponding to that particular pattern.
3. The Perceptron's response is then calculated for the input pattern from equation (2.15).
4. The weights are then adapted using the learning rule:

$$w_i(t+1) = w_i(t) + \eta e(t) x_i(t) \qquad (2.17)$$

where $e(t)$ is the error between the desired values and the Perceptron's output such that:

$$e(t) = (d(t)-y(t)) \in \{-1,0,1\} \qquad (2.18)$$

and η is a learning constant, where $0 < \eta \leq 1$.

5. The process is repeated from step two until there are no errors.

The important feature about the Perceptron is that it will only converge to a solution if the two pattern classes are linearly separable. Also, if a solution does exist then it will not always be a unique one. If a Perceptron does converge, then it can be used as a classifier. The constraint of linear separability, limits the usefulness of the Perceptron, however, it will be seen in the next subsection that this problem may be partially solved by using a different learning technique and error feedback loop, this new arrangement being termed the adaline.

2.5.3. The Adaline

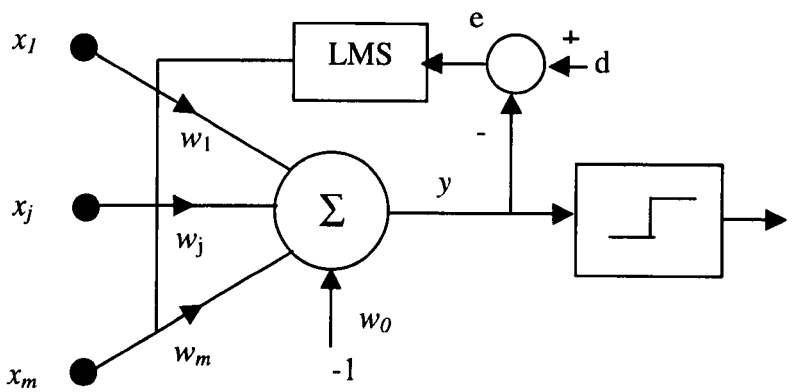


Fig.2.3. The adaline

A diagram of the adaline unit with error feedback is shown in Figure 2.3. This is quite similar to Figure 2.2 of the Perceptron, however, if the two diagrams are compared it is evident that the point where the error feedback is taken from is now different. The Perceptron requires that the actual unit output if fed back to update the weights, this means that the error will take only three discrete values {-1,0,1}. Therefore the error of the Perceptron only reflects the sign of the error, rather than its magnitude. The adaline, however, does take into account the magnitude of the error since the net input is fed back before it is passed through the node activation function. This gives the same weight update rule as in equation (2.17) except that the error can now be written as:

$$e(t) = (d(t)-y(t)) \tag{2.19}$$

The weight update rule is known as the Widrow-Hoff learning rule now updates the weight values in proportion to the size of the error. The term Least-Mean-Squares (LMS) is also used to describe the rule because when used within a training algorithm it attempts to minimise the mean square error.

The LMS rule can be explained by the fact that a cost function of the error can be defined as:

$$E = \frac{1}{2}\sum_{p=1}^{N} e^2(p) \tag{2.20}$$

where $e(p)=(d(p)-y(p))$ denotes the error for the p th pattern and N is the number of input patterns in the training set data. The cost function is with respect to the weights because $y(t) = \mathbf{w}^t\mathbf{x}$. The cost function in equation 2.20 can be thought of as an error surface where low error energy, which represents a low classification error, is the target. Finding the minimum error energy means finding a set of weights, which signify the optimum set of weights for the system, which may not necessarily produce the lowest rate of misclassification. The error surface can be thought of as a parabolic curve, where the minimum error energy can be found by following the steepest negative gradient of that curve. This leads to the gradient descent learning rule:

$$w_i(t+1) = w_i(t) - \eta\frac{\partial E}{\partial w_i} \tag{2.21}$$

Given the cost function in equation (2.20) it is possible to write that the error for one pattern is given by:

$$E_p = \frac{1}{2}e^2(p) = \frac{1}{2}(d(p) - y(p))^2 \tag{2.22}$$

which means that the total error energy can be written as the sum of the error energies contributed by each individual pattern, which gives:

$$E = \sum_{p=1}^{N} E_p \tag{2.23}$$

The total error E can then differentiated with respect to the weights, which, using the chain rule of partial differentiation gives:

$$\frac{\partial E}{\partial w_i} = \sum_{p=1}^{N}\frac{\partial E_p}{\partial w_i} = -(d(p) - y(p))x_i(p) \tag{2.24}$$

Combining equation (2.24) with equation (2.21) will then result in the Widrow-Hoff or LMS error learning rule.

The adaline is attempting to minimise the size of the error rather than the misclassification rate and will therefore always converge to a solution, which partly solves the problems of the Perceptron. This may mean, however, that even when attempting to solve a problem which is linearly separable (which the Perceptron could solve), the adaline may misclassify patterns even though the minimum error energy is still reached.

Development of neural networks slowed down after the late 1960s when it was shown by Minsky and Papert[11] that a single layer of Perceptrons could not compute the Exclusive-OR problem (Table 2.1)

X1	X2	Y
1	0	1
0	1	1
1	1	0
0	0	0

Table 2.1. The Exclusive-OR Logic Problem

It was realised that multiple layers of Perceptrons could be used to solve this problem however there were problems in finding a suitable learning algorithm since as will be seen in the next section it requires the differentiation of the activation function. Multiple layers of adaline units, without the step function included (since this only acts to quantise the outputs) can be formed, but are not useful, since their outputs are linear and have therefore been found to be equivalent to a single layer network. It was therefore realised that what was needed was a differentiable non-linear activation function to overcome the single layer equivalence problem. Research into

this area led to the development of the Multi-Layer-Perceptron containing a non-linear activation function, which is commonly used today.

2.5.4. The Multilayer Perceptron (MLP)

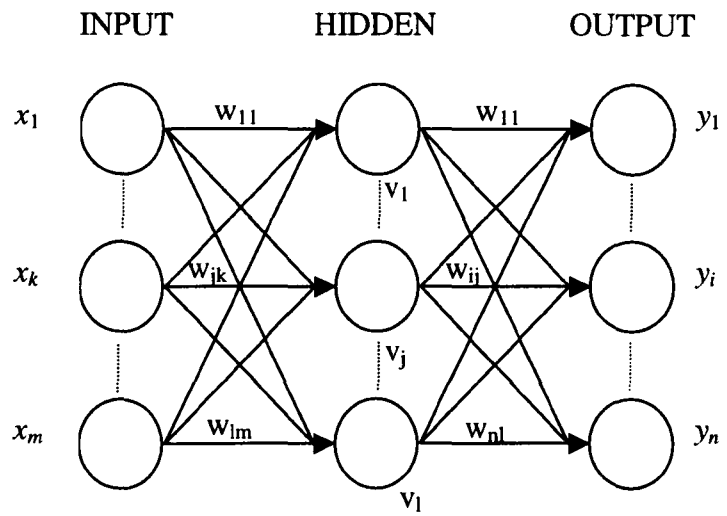


Fig.2.4. Multiple layers of neurons used to make a network of arbitrary size.

An example of the MLP with one hidden layer is shown in Figure 2.4. Each unit in the network is similar to that of the SAN described in section 2.5.1. (the bias weights are not shown for clarity) except that there is now a non-linear activation function as opposed to a step one. A sigmoid function provides a smooth differentiable function, which tends asymptotically to corresponding threshold function levels, an example of which is the logistic function which has the form:

$$f(x) = \frac{1}{1+\exp(-x)} \tag{2.25}$$

A diagram of this function is shown in Figure 2.5. It is argued in Bishop[12], that an alternative sigmoid function which may provide faster convergence times is the hyperbolic tangent function:

$$g(x)=\tanh(x) \tag{2.26}$$

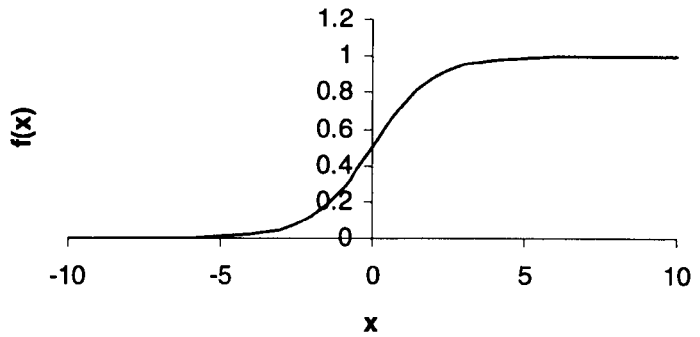


Fig.2.5. A sigmoid activation function

If the outputs of the hidden layer, v_j are treated as the inputs to the output layer, then a learning law for the output layer neurons can be derived as:

$$E=\frac{1}{2}\sum_{p=1}^{N}\sum_{i=1}^{n}(d_i(p)-y_i(p))^2 \tag{2.27}$$

where N denotes the number of patterns in the training set, n is the number of output units, and $y_i(p)=f(net^o_{pi})$ is the output activation of the output layer node, which is given by:

$$net^o_{pi}=\sum_{j=0}^{l} w^o_{ij} v_j(p) \tag{2.28}$$

The gradient descent update rule (equation 2.21) can now be written as:

$$w^o_{ij}(t+1)=w^0_{ij}(t)-\eta\frac{\partial E_p}{\delta w^0_{ij}} \tag{2.29}$$

for each pattern on a pattern-by-pattern basis, where E_p is the error for an individual training pattern, which is now given by:

$$E_p=\frac{1}{2}\sum_{i=1}^{n}(d_i(p)-y_i(p))^2 \tag{2.30}$$

The chain rule can now be used to differentiate (2.30) with respect to the weights such that:

$$\frac{\partial E_p}{\partial w_{ij}^o} = \frac{\partial E_p}{\partial y_i(p)} \frac{\partial y_i(p)}{\partial net_{pi}^o} \frac{\partial net_{pi}^o}{\partial w_{ij}} = -(d_i(p) - y_i(p)) f'(net_{pi}^o) v_j(p) \qquad (2.31)$$

therefore the weight update rule for units in the output layer is:

$$w_{ij}^o(t+1) = w_{ij}^o(t) + \eta \delta_{pi}^o v_j(p) \qquad (2.32)$$

where the output layer term $\delta^o_{pi} = -(d_i(p)-y_i(p))f'(net^o_{pi})$, and $net^o_{pi} = (\mathbf{w}^o_i)^t\mathbf{v}$ with $\mathbf{w}^o_j=(w^o_{i0},w^o_{i1},\ldots,w^o_{il})^t$ and $\mathbf{v}=(-1,v_1,\ldots,v_l)^t$.

The differential of the activation function in equation (2.25) using the quotient rule is:

$$f'(x) = \frac{\partial f(x)}{dx} = f(x)(1 - f(x)) \qquad (2.33)$$

Therefore the output layer error term δ becomes:

$$\delta_{pi}^o = (d_i(p) - y_i(p)) y_i(p)(1 - y_i(p)) \qquad (2.34)$$

The hidden layer learning rule can be found by apply the gradient descent rule again, where:

$$w_{jk}^h(t+1) = w_{jk}^h(t) - \eta \frac{\partial E_p}{\partial w_{jk}^h} \qquad (2.35)$$

The chain rule can once again be applied with the result that:

$$\frac{\partial E_p}{\partial w_{jk}^h} = \frac{\partial E_p}{\partial v_j(p)} f'(net_{pj}^h) x_k(p) \qquad (2.36)$$

E_p is not an explicit function of v_j, however through further application of the chain rule it can be found that:

$$\frac{\partial E_p}{\partial v_j} = -\sum_{i=1}^{n} (d_i(p) - y_i(p)) f'(net_{pi}^o) w_{ij}^o \tag{2.37}$$

If this is then substituted into (2.36) then,

$$\frac{\partial E_p}{\partial w_{jk}} = \left(-\sum_{i=1}^{n} (d_i(p) - y_i(p)) f'(net_{pi}^o) w_{ij}^o \right) f'(net_{pj}^h) x_k(p) \tag{2.38}$$

Using equation (2.34), the hidden layer δ term can be defined as,

$$\delta_{pi}^h = f'(net_{pj}^h) \sum_{i=1}^{n} w_{ij}^o \delta_{pi}^o \tag{2.39}$$

Therefore:

$$\frac{\partial E_p}{\partial w_{jk}^h} = -\delta_{pj}^o x_k(p) \tag{2.40}$$

which when substituted into the gradient descent rule of equation (2.35), gives the hidden layer weight update rule:

$$w_{jk}^h(t+1) = w_{jk}^h(t) + \eta \delta_{pj}^h x_k(p) \tag{2.41}$$

using the derivative of the sigmoid activation function, equation (2.33), the hidden layer error term becomes:

$$\delta_{pj}^h = v_j(1 - v_j) \sum_{i=1}^{n} w_{ij}^o \delta_{pi}^o \tag{2.42}$$

The output and hidden layer weight update rules (equations (2.32) and (2.41) respectively) together with their error terms (equations (2.34) and (2.42)) form the basis of the back error propagation algorithm used in the MLP.

2.5.5 Application of the MLP to continuous, real-world data

The outputs of the MLP are continuous, due to the use of logistic or tanh activation functions. For classification purposes a threshold can be used to determine if a given input pattern falls into class 1 or 0 (with a single output neuron). The continuous output of the network can also be used for function approximation, which was the interest of this project in the technique. Results by Cybenko[13] and Hornik[14] in 1989, independently showed that any continuous function of N real variables may be approximated by an MLP with a single hidden layer. These theorems do not, however, give practical advice as to the network size and configuration for a given mapping task. Some techniques for choosing these parameters will be investigated, however there are some further complications of the standard MLP back-propagation algorithm that have to be noted first. As with a single layer network, the gradient descent method does not necessarily mean that the minimum error energy will be reached. Figure 2.6 shows the problem of local minima, which can exist.

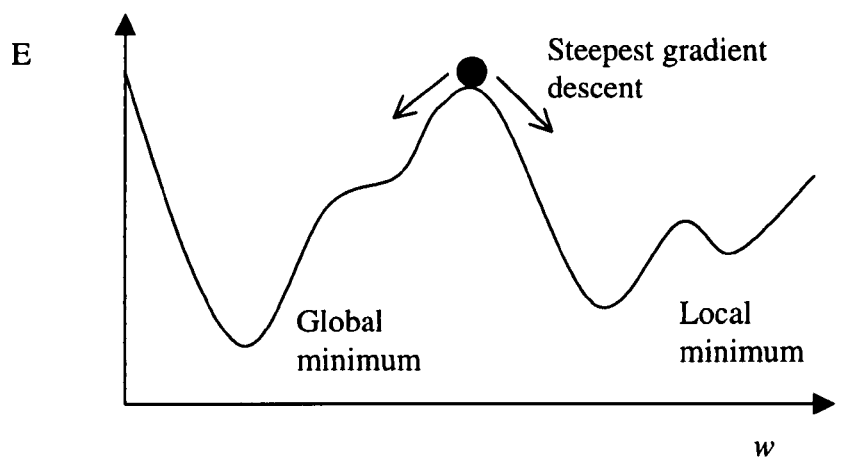


Fig.2.6. The local minima problem

The problem of local minima is related to the dimension of the weight space and the network architecture. There is still, however, a large amount of research into these areas and heuristic methods are often adopted. The local minima problem may to a certain extent be avoided by modifying the MLP with a momentum term; this consists of a proportion of the previous weight changes, which are combined with the weight update equation. Equation 2.43 shows the momentum term in the weight update equation, where $\Delta w(t)$ denotes the weight changes at time t for an arbitrary weight.

$$w(t+1) = w(t) + \Delta w(t+1) + \alpha \Delta w(t) \tag{2.43}$$

The momentum term helps 'push' a trajectory out of a local minima in a similar way to a ball rolling at a fast enough speed having enough momentum to overcome a barrier.

A second problem of the standard back error propagation algorithum is that of 'paralysis', this is due to the sigmoidal activation function. It has been shown that the weight changes are proportional to an error, which in turn, depends on the derivative of the sigmoidal activation function. If the hidden layer network input forces the activation function out of the linear region around zero, then the output is approximately constant. This causes the corresponding derivative to be close to zero, making the weight changes very small, hence further training achieves little. This problem can be overcome by adding a decay term to the update equation. Thus equation 2.43 can be written as in 2.44.

$$w(t+1) = \gamma w(t) + \Delta w(t+1) + \alpha \Delta w(t) \tag{2.44}$$

With equation 2.44, even if the weight changes, $\Delta w(t)$, are negligable, the weights will gradually decay, bringing the input to the activation funcion within the linear region.

Finally, a further problem of the MLP is that of overtraining. This problem is related to the data which are modelled, the network architecture and the number of itertions for which the network is trained. Before considering overtraining, the concept of generalization should be discussed. When a network has been trained on a given set of data, it will usually be used to predict the outcome for previously unseen inputs. It is usual to evaluate the prediction performance of a neural network for these new cases by using a test set. The ability of a given network to predict new input patterns is known as 'generalisation'. When training a neural network the aim is to obtain the best possible generalisation performance with the training data available.

Figure 2.7 illustrates the probem of overtraining on the prediction of some noisy data with a linear relationship. If a given network architecture is trained for too many iterations then the network will tend to learn the noise on the data rather than the true underlying function in the data. To help prevent this it is usual [15,12] to use a validaton

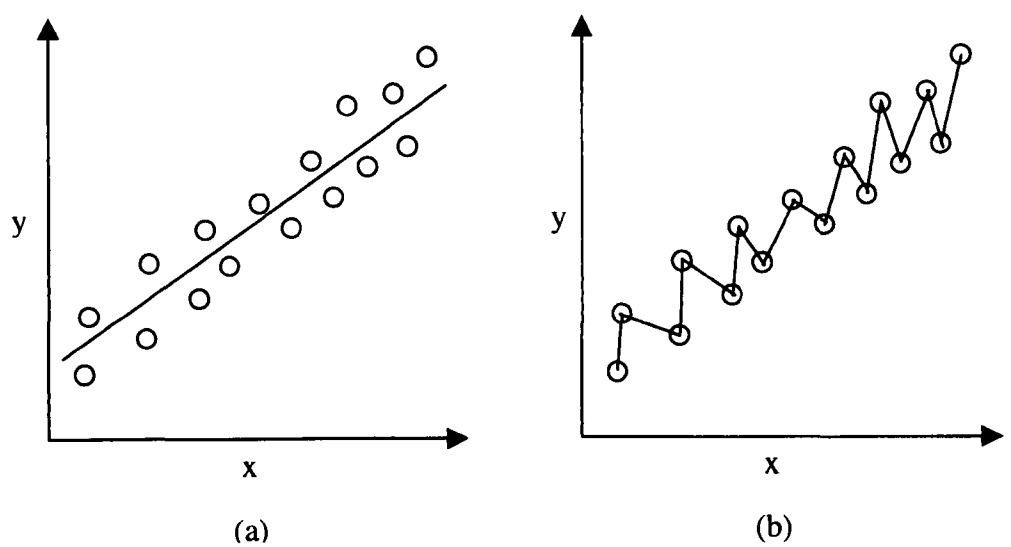


Fig.2.7 The problem of overtraining, (a) the true approximation, (b) an overtrained approximation.

set.

Thus for a given data set for which one would like to construct a neural model, the data is randomly partitioned into training, validation and test sets. The training data is that used to provide the inputs and target values that directly alter the weights in the network. The validation set is used to monitor the generalisation performance of the network as it trains. After each complete presentation of all the data points in the training data set (termed an eon), the ability of the network to generalise on the validation set is evaluated. This evaluation is made by using the mean squared error functon (MSE) shown in equation 2.45, for which t_n and y_n are the target and predicted outputs for a given input pattern. When the MSE of the validaton set starts to rise beyond its minima, training is halted. The generalisation performance of the network to a completely unseen data set, the test set, can then be investigated for performance evaluation.

$$E_{MSE} = \frac{1}{N}\sum_{n=1}^{N}(t_n - y_n)^2 \qquad (2.45)$$

The selecton of the number of hidden layers and the number or hidden layer neurons is also known to affect the generlisation ability of the MLP, however these factors are usually determined experimentally for a given data set. The effect of network architecture will be investigated in chapter 6.

It should be noted that there is a degree of disagreement in the definitons of the Validation and Test sets in the literature, as some authors use a Test set to validate the performance of the network as it is trained and do not use a third data set to analyse the true generalisation of the network. Tarassenko and Bishop define a test set as unseen data (not used in any part of the networks training) and use a validation set for deciding network architecture and when to stop training.

2.5.6 The Radial Basis Function network

The Radial Basis Function network (RBF) is another neural network that can be used for function approximation. It still uses the basic idea of an artificial neuron calculating the weighted sum of some inputs and then operating on an activation function, however the architecture and training used are different. RBF networks

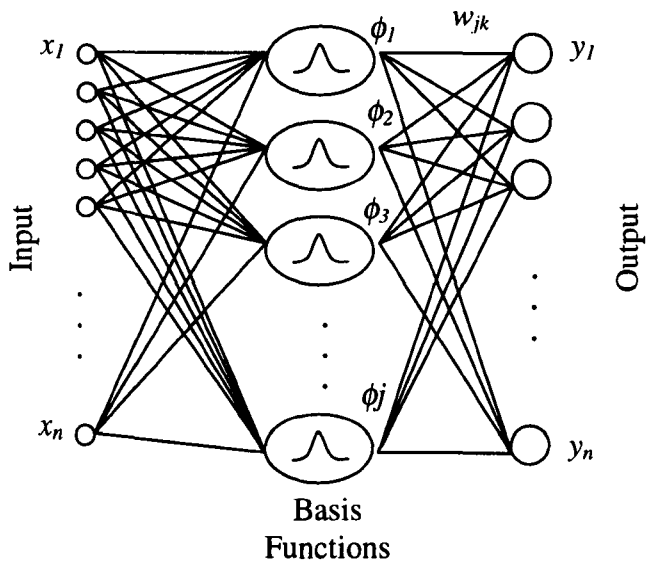


Fig.2.8. A Radial Basis Function Network

were not utilised in this work for several reasons that will be outlined at the end of this section, however a brief introduction to their architecture as found in Tarassenko[15] will be made so as to explain the main differences they have when compared to the MLP. Further information on the RBF technique can also be gathered from Bishop[12]. A simplified diagram of an RBF network is shown in Figure 2.8, the bias weights are not shown.

Before explaining the functionality of an RBF it is important to be aware that clustering algorithms exist functioning as Tarrassenko states:

Given P patterns in n-dimensional space, find a partition of the patterns into K groups, or clusters, such that the patterns in a cluster are more similar to each other than to patterns in different clusters.

Euclidean distance can be used as a measure of closeness between two vectors, and this is the case for a clustering algorithm known as the K-means clustering algorithm, which can iteratively partition data into K numbers of clusters. The outcome of such a clustering algorithm is a set of centres (with respect to a cluster of data in n-dimensional input space), which are also known as prototype patterns.

As shown in Figure 2.8, an RBF network has one hidden layer, an input layer and an output layer. The Euclidean distance between x and a set of prototype vectors determine the hidden unit activation function output in the RBF. Equation 2.46 shows the output of the j^{th} hidden layer unit, where μ_j is the centre of the basis function and σ_j is the width of the basis function.

$$\phi_j(x) = \exp\left(-\frac{\|x-\mu_j\|^2}{2\sigma_j^2}\right) \tag{2.46}$$

The output of the RBF (equation 2.47) is a weighted linear combination of non-linear basis functions.

$$y_k = \sum_{j=1}^{J} w_{jk}\phi_j + w_{0k} \tag{2.47}$$

Training an RBF is a two stage process, firstly clustering techniques are used to determine the centres, this is an unsupervised technique (no target values are required for a given set of input vectors). Once the clustering is complete the basis function widths are determined such that the spread of the data associated with each center is represented sufficiently. The second stage is a linear optimisation of the output layer weights using a supervised optimisation technique (taking into account the target values).

The initial unsupervised stage is advantageous in some problems as target values are not always known for a given data set, thus information can still be gained from this 'unlabelled' data.

Bishop[12] and Tarassenko[15] make a comparison of the RBF with the MLP, and the following points can be summarised.

- RBF and MLP networks provide similar techniques for approximating arbitrary-non-linear functional mappings between an input vector x and one or more output(s) y.
- An MLP network uses weighted summation of inputs, transformed by a continuously differentiable function. An RBF network uses a distance to a prototype vector and then a transformation with a localised function.
- The MLP forms a distributed representation in the space of activation values for the hidden units. For a given input, many of the hidden layer weights will usually contribute to the output value. These hidden layer units are then combined by the linear activation functions in the output later with weights such that the desired output is achieved over a range of output values. Slow convergence in MLPs arises because of the cross coupling and interference between the hidden units. RBF networks form a localised representation of the input space, for a given input pattern it is usual that only a few significant hidden layer activations occur.
- There is a global training strategy in an MLP, with all parameters being determined at the same time. A radial basis function has a two stage training procedure. The first stage involves determining the basis functions via an unsupervised technique and the second stage requires linear supervised techniques for determining the weights of the network.

The RBF and was not found to be suited to the data obtained in this project which had an uneven distribution, causing problems in cluster centre location (many centres may be assigned to dense areas of data). Additionally, the RBF's formulation of

localised models is not beneficial for interpolation between clusters of data within the input space in contrast to the MLP's distributed representation of the input space. The RBF's ability to benefit from unlabelled data was also not an advantage with the data in this project because all input data tended to have target values labelled.

2.6 Literature review

The project brief encompassed a multitude of disciplines, which made generating a concise literature review on all aspects of the project challenging. The aim of the literature review was to provide a base understanding of the heat treatment and modelling technique sides of the project, whilst also identifying where work relating to the project brief had already been carried out. A further aspect of the review, which could not be carried out in the review's initial stage was to keep pace with developments in the subject areas whist the project was undertaken, and investigate the feasibility of new ideas as the project developed. It was therefore decided to perform the review in a number of stages specific to the subject areas of interest as the project developed. An important starting point, however, was to concentrate initially on areas of the literature where modelling techniques, particularly non-linear versions such as neural networks, have been used in processes relating to manufacturing, especially in the steel industry.

2.6.1 Modelling of steel and other industrial processes

The literature search showed numerous hits for neural network or other techniques alone. It was therefore necessary to combine these hits such that literature specific to industrial process modelling was emphasised. This combination still yielded an unmanageable number of hits. When steel processes were also considered, this number was dramatically reduced, even when other intelligent systems

techniques were considered. The literature indicates that modelling of steel processes is performed for a number of reasons, these being as follows:

- Prediction of process or product parameters
- Process or product optimisation
- Model-based process control
- Fault Detection / Quality inspection

It is acknowledged that there is some overlap between these fields, for example one might predict some process parameters and then perform optimisation of the process using this model. In a similar respect, optimisation might be in the form of open-loop or closed-loop control. The techniques behind all of the above categories were therefore relevant to this project, which aimed to predict parameters, thus optimising the process.

The literature pertaining to steel processes also indicates that each of these categories may apply to either batch or continuous heat treatment or rolling processes.

2.6.2. Prediction of process or product parameters

There is not a large amount of literature on predicting the mechanical properties of steels, however there are several groups of authors who have performed research in these areas. Using neural techniques for mechanical property prediction seems to be a recent idea as most of the literature dates between 1986 and 1999.

One group of papers, which was relevant to the project aims was written by researchers at the Laboratory for Materials Science at Delft University. All of the papers in this group concentrated on the application of neural networks to heat

treatment processes. The first of these papers by van der Wolk et al.[16] investigated the prediction of the finishing temperature in a hot strip mill after the last finishing stand from a number of process parameters. Rolling processes can be categorised into hot or cold types. The mills tend to use a series of rolling stations (known as stands), to reduce the thickness of steel strip, such that the mechanical properties and surface finish of the strip are as required. The hot rolling process, similar to the batch heat treatment process this project was concerned with, is a very complex one. In order to produce strip metal with a uniform mechanical properties (and therefore microstructure), it is necessary to know the strip temperature at key stages of the process. The model in this paper was trained to predict the temperature at the last finishing stand, from data for 10 thickness groups of C-Mn steel. A feed- forward MLP neural network architecture with back-error propagation training similar to that illustrated in this chapter was utilised in all this research group's papers. This paper in particular investigated the performance of the neural network model in comparison to that of linear and PLS modelling techniques. It was felt that the neural model should be capable of encompassing the entire range of strip thickness (which affects the output parameter in a complex manner). When performing the linear and PLS techniques, the authors took two approaches, a group of models, one trained on each thickness, and a single linear and PLS model trained on the entire data set. The work concludes that the neural technique outperforms the linear model significantly. The optimised linear model (PLS technique), was as accurate as the neural model for some thickness groups, however overall the performance of the neural technique was superior in that it offered better generalisation to a range of thicknesses. An important feature of this paper is the recognition that for the model to be applicable to a new industrial process (not the one the model is trained on), an adaptation

procedure may be required to account for systematic differences between the model and the new process.

A further paper by this group looked at the use of a neural model for the prediction of martensite start temperature[17]. Martensite start temperature is a product parameter relating to the phase transformation kinetics of steels, and is the temperature at which super-cooled austenite transforms into martensite. The martensite start temperature is affected by the steel's composition, amongst other parameters. This paper is unusual in that the data points were collected from literature, rather than from an industrial process. The neural models were compared to the PLS and linear model techniques, and also to a statistical model found from the literature. There were only 164 data points used for model construction. This is an important area of neural modelling because it is often found that very large training data sets are required, depending upon the complexity of the problem and the number of input parameters. As this paper demonstrates, however, the location of the data points within the input space is important, and if the training and testing data are selected from specific areas, a good model can still be formed.

It is important to mention that not all of the literature reviewed used a separate test set in accordance with definitions set out earlier of this chapter. Sometimes authors accommodated for validation data within the training data, or simply used the validation performance as an assessment of generalisation ability. One technique in the martensite start temperature paper, which was seen across the range of parameter prediction papers, is that the effects of certain parameters can be investigated on the model's output parameter by keeping all other model inputs constant whilst varying only one input parameter. This does not show the effect of parameter interactions and so a progression is to vary two input parameters at once, such that the interaction effects between the two parameters could be seen.

The next paper by this group is concerned with the prediction of Continuous Cooling Transformation (CCT) diagrams for vanadium containing steels[18]. CCT diagrams show microstructural transformation start and finish temperatures at various, approximately constant, cooling rates, along with the resulting microstructure and hardness values at each cooling rate. The motivation behind the neural model, as with most of the projects found in the literature search, was that accurate physical models were not yet available and so an empirical model was required. The authors found that it is possible to predict CCT diagrams on the basis of chemical composition, however, fine detail of the diagrams was hindered by limited data.

Vermeulen et al. investigated the prediction of Jominy hardness profiles of steels [19]. The Jominy test is a simple test for determining the hardenabilty, that is the hardness as a function of depth into the bar. The key feature of this paper, apart from the actual model aim, is that the data for the model were pre-processed in a way such that abstract data points in the training set were excluded. For example, a steel grade out of the normal range of the rest of the data was excluded such that it would not hinder the other predictions of the model.

The final paper by this group of authors is that by van der Wolk et al [20], who looked at the problems associated with modelling hardenabilty of steels when the data is obtained from multiple sources.

The other main group of authors in this area is Cambridge University. The first paper, by Bhadeshia[21], concentrated on using neural networks for the prediction of C-Mn steel arc weld toughness. There is emphasis in this paper on the aspect of using neural networks to analyse the process, such that the models generated are used to learn something about the process, for example the effects of key variables. The neural network technique used by authors associated with Cambridge University

differs in some respects to the standard back-error propagation network explained at the beginning of this chapter. This difference is that they used a Bayesian statistical framework by MacKay[22,23], the techniques of which are the subject of a review paper[24]. The Bayesian framework helps control the complexity of the models generated, thus helping to prevent over-fitting and promote parsimonious models. Another feature claimed by the Bayesian neural network technique is that it provides error bars on model predictions, which give a guide to the confidence in the model's predictions. When choosing the input variables for the model, it is evident that metallurgical knowledge was utilised, such that only variables that give significant effect were included. The main reason for not simply adding all process variables available is that the variables which do not give a significant effect on the output variable carry a noise penalty with them which hinders the accuracy of the resulting model. It is therefore important that the variables selected for a model are relevant to the output variable concerned. There was, however, a problem with the input variables used in the arc weld toughness paper, which seemed to highlight the two approaches that there are for modelling the mechanical properties of steels. The variables used comprise, amongst other things, alloying additions to the steel and also microstructure. The mechanical properties are believed to be a function of microstructure, however the microstructure is also a function of alloy addition. Therefore the input variables were over-described as the author noted in his analysis. Another important aspect of this paper is that the author attempted to extrapolate beyond the input range of the model. It is widely felt that this is not a good idea, as network outputs tend to saturate[25]. The results are, as one would expect, unpredictable and hence this notion is confirmed.

A further paper by the Cambridge group, connected with weld properties is by Cool[26], who looked at predicting the yield and ultimate tensile strengths of steel

welds. The inputs to the UTS model are interesting here since the yield strength was used as an input parameter. Although not stated, this may have meant that there would be overlap between this and the other parameters used as model inputs such as alloy additions and tempering temperature, in a similar way to that found by Bhadeshia. Evaluation of the effect of the major alloying addition variables on the output was used to confirm the final model's representation of metallurgical knowledge.

Modelling of microstructure formation has also been performed by Bhadeshia et al[27]. This work involved predicting the start and finish temperatures of austenite formation. Like the martensite model of the Delft group, the data were drawn from literature. Various experiments were performed to confirm certain metallurgical phenomena and once again the effect of data quality was seen when it was noted that the prediction accuracy of start temperature was poorer than the finish temperature. The authors believe this to relate to problems in the procedure for measuring the start temperature.

Rolling processes have also been modelled using the Bayesian technique. A model using steel composition and screw gate settings for the mill over a range of passes has been built to predict the yield and ultimate tensile strengths of the resulting material[28]. This model had the largest number of inputs noted in the literature, which totalled 108 variables. From personal communication with experts it is a matter of opinion whether all were justified.

Finally, Jones[29] looked at predicting the yield and tensile strength of nickel-base super-alloys. The data set, comprised approximately 200 data points, 100 of which were used for validation of generalisation performance, i.e. testing. Some consideration was given to pre-processing of the data prior to the model training; this was done using max/min scaling of input variables such that all variables had the

same range. Sensitivity analysis of the network's output, to each of the inputs, was used to show the significance of certain variables in the model to the output value. This is another way of confirming metallurgical knowledge and learning from the model constructed.

Alloy design was the aim of a paper by Bulsari and Hocksell[30], who used a standard back-error propagation technique to predict yield and tensile strength of steels after a hardening process. A brief review of some of the areas where neural networks have been used in the steel industry is provided, followed by details of the modelling work undertaken. The ranges of the alloy additions in the steels covered by the model were unclear, however, only 236 observations were available. The training algorithm used is an optimised form of the gradient descent algorithm called the Levenberg-Marquardt method, which the authors claimed to provide faster training times with the overhead of more computer memory usage. No metallurgical analysis was made of the model, as in previous papers, however the prediction accuracy of linear and non-linear models was compared, using a range of network architectures. Non-linear versions of even the simplest network architectures were shown to be more accurate than the linear techniques. The authors viewed the prediction accuracy of their model to be slightly worse than the measurement error on the output values in the training data and attributed the rest of the variance in predicted results to errors in the measurement of the input data. They are therefore claiming that the variables used in the model fully describe the process.

Using neural networks for mechanical property prediction on a batch annealing process for thin steel strip was the subject of a paper by Myllyoski et al[31]. A data acquisition system was used in this work enabling data to be retrieved directly from a rolling mill giving ~3500 data points. Parameters describing the reduction in strip thickness during the rolling process, the temperatures in the annealing stage, the

composition of the steel strip and the sample locations in the strip were used to predict yield and tensile strength results. 70% of the data was generated from two steel grades, however the mechanical properties of some rarer grades were also modelled. During the model's design, hidden layer experimentation was used to determine the optimum network configuration, as seen before. Moreover, investigation into the optimum learning rate was also considered, such that the network could be trained in an optimum time, whilst not making the learning rate too high, such that the process becomes unstable. For one particular steel grade in the model, the prediction accuracy was poor. The authors considered that their neural model, covering a range of steel grades, might have been trying to form sub-models within the input space to accommodate for the different grades. It was postulated that there might not have been enough freedom in the descriptive power of the neural network concerned to accurately describe the results pertaining to one particular grade giving poor results when the network was also describing the functions behind the other grades. The authors therefore tried a 'localised' model, which was only trained on the problematic steel grade. It was seen that on this occasion the prediction accuracy did not improve beyond that of the model covering the entire range of data, and it was later found that the quality of this data was to blame. This is a poignant reminder that the neural network can only be used to describe an industrial process if the features of that process are accurately described within the training data set.

Kudav and Dengke[32] illustrate one example of a slightly misleading approach to modelling hardness characteristics in carbon steels. In a similar way to previous literature they utilised literature as a data source. These data consisted of a series of hardness versus tempering temperature curves for steels ranging in carbon, chromium and molybdenum (three alloying additions believed to affect the hardness of a steel). The network was basically trained on the data from these curves and then,

when 'tested', the network was used to predict the same compositions of steel, which was done with excellent accuracy. Whist this might have resulted in a network that was capable of predicting for the training set compositions, it provided no insight into the generalisation of the network to new compositions within the original input range and was essentially using the neural network as a look-up table.

There are two papers relating to the hot metal stage of steel production. The first paper, by Datta et al[33], is concerned with the desulphurization of hot metal, and specifically investigated the prediction of sulphur content in the desulphurization process. This process involves injecting powdered calcium diamide CaD through a submerged lance into a 'torpedo ladle' used to hold the molten material. The network was only trained on 40 examples, and yet the authors reported satisfactory generalisation. They did however note the need to select an optimum network architecture and demonstrated that if a particular data example is uncommon in the data set, it will not be modelled as accurately as a frequently occurring one.

The second hot metal paper, by Singh[34], attempted to predict the silicon content of hot metal in a blast furnace. This is a desirable thing to do because silicon has a number of effects on the basic oxygen steel-making process, which follows the blast furnace stage. It appears that like other steel processes covered in the literature, the blast furnace is a very complex process, with a number of important variables believed to have complex non-linear relationships. Some physical equations were designed by previous authors in an attempt to describe the level of silicon, however, as is so often the case, these were not adequate to describe such a complex process. The author has used some interesting approaches in the modelling stage. Linear regression was attempted but was unsuccessful, and the standard back-error propagation algorithm was also utilised. However, the interesting features include a number of modifications on the standard back error propagation algorithm being

utilised including adaptive learning rate algorithms and variable momentum terms. Although the performance of these techniques overall was good, training times were said to have been increased. One useful adaptation to the neural technique involved using a neuro-fuzzy technique, which did improve the models predictive performance under certain circumstances. The neuro-fuzzy technique consists of using fuzzy logic to encode the normally continuous inputs or outputs of the neural model into membership values of fuzzy sets. More information on the fuzzy logic technique can be found in Zadeh[35]. Briefly it can be explained that fuzzy logic enables imprecision in measurements to be expressed, for example one might say the weather is *hot*, where *hot* is a linguistic expression of temperature. To fuzzify a measured temperature one might have two sets in which temperature could be a member, a cold set, and a hot set. Depending on the shape and configuration of these sets, a temperature of say, 20°C might be 5% a member of the cold set and 80% a member of the hot set. Therefore, the temperature of 20% could be translated into two inputs of 0.2 and 0.8. The authors found that encoding the inputs to the network using the fuzzy logic technique produced the best results for their particular modelling problem. A fuzzy logic technique has also been used by Schooling et al[36] to model the fatigue threshold behaviour in Ni-Base Super-alloys. The authors here claim that the technique enables a priori knowledge to be readily combined into a neural model, and that when the technique is integrated throughout the architecture of the neural network, fuzzy rules are generated, which can provide 'transparency' to the so called 'black-box' modelling technique.

When considering the combination of neural networks with other intelligent techniques, Wiklund[37] used neural networks very differently for prediction. The temper rolling process was modelled using a Finite Element Modelling (FEM) technique, which is a way in which physical equations can be used to describe a

process by breaking the problem down into elements. This technique is effective, however, it requires a large amount of computer time to simulate the results. The neural network technique was used on this occasion to learn a variety of results for a number of parameter configurations of the FE model. The interpolation ability of the neural network allowed the results of the FE model to be accessed quickly. If enough parameter variation exists in the simulation of FE model then the neural network can save simulation time by predicting for a new set of parameters.

The final paper in this section is by Dumortier et al[38], who looked at modelling the yield and ultimate tensile strength of microalloyed steels in a rolling process. The author was motivated to investigate the neural technique when considering the poor prediction accuracy of regression formulae in the literature, which was demonstrated in a review paper on microalloy steels[39]. Dumortimer, unlike many others, considered the data integrity and utilised a multivariate data analysis technique to find data points that were outliers in a statistical sense. The model building was performed by starting with a number of input variables and then removing those variables that appeared not to have a significant effect on the output variable predicted. Binary or dummy variables were also used to describe aspects of the process which are non-continuous, such as the test type used. The authors once again conclude that the performance of the neural technique is superior to that of the linear techniques, but also recognise that data integrity and training set size are important factors in prediction accuracy.

It has been seen from the literature that there are examples of mechanical property prediction, as well as a number of other applications where other product parameters have been predicted. All of the literature covered in this part of the review used neural networks capable of non-linear function approximation to make the necessary predictions. The processes using these techniques were all complex

ones, containing non-linear effects and interactions between variables. It was frequently stated that traditional linear techniques were unable to describe these aspects of the processes and therefore produced less favourable models. Moreover it was felt that knowledge of the physical mechanisms behind these processes was insufficient to describe the processes adequately and therefore empirical techniques were required. Where, in the case of the FE model, knowledge did exist, the neural network technique was still useful in making a range of results accessible by interpolating between FE model results, enabling a variety of results to be produced in real-time. Generally, the inputs to the modelling process comprise process parameters such as roll gap or heat treatment temperatures and product parameters such as dimensions and alloy composition. In several of the papers, as well as producing a single predicted value, sensitivity analysis or procedures to investigate the effect of one or more variables were employed to confirm metallurgical knowledge and learn more about the processes concerned. The effect of the quality of the data was not widely investigated, and many of the models were trained on only a few hundred examples. Despite the use of Bayesian statistical techniques to aid network architecture selection, there are still few guidelines as to the appropriate architecture or training regime one should use for a given problem. It is indicated, however, that the network parameters are problem dependant and that experimentation seems to be the key to finding the optimum parameters for a given problem.

It is noted that the alloy ranges, which the models covered, tend to be quite narrow and are often restricted to a particular grade of interest over a range of manufacturing conditions. Moreover it is also noted that a neural model should only be used to predict data in the same range as that of the training data.

2.6.3. Product or process optimisation

There is a close link between product or process prediction and optimisation. The term optimisation can be quite vague, as simply prediction of a parameter can be classed as optimisation, since one is allowing experimentation with input parameters to the model in order to obtain a required output value which corresponds to the real process. In this section, however, examples of literature were considered where authors have utilised models on steel processes to enable an optimisation routine that finds a set of optimal parameters for process or product design.

The design of an accurate model is clearly still important and has therefore been noted. Apart from the research based at Cambridge University, one other author who used a Bayesian neural network for this type of model is Grylls[40]. His paper investigated using neural networks for modelling the mechanical properties of Marinel, a high strength marine alloy. The motivation of this paper is clear in that the author aimed to use the model generated to experiment with the composition and post-forging heat treatment in such a way as to produce a new alloy with a higher strength, but with maintained impact properties. The next chapter in this thesis details impact and tensile strength tests, however for now, the reader needs to accept that as a general rule there is a trade-off between strength and Impact energy (representing toughness of the material). A data set comprising 398 points was used to train the neural network to predict Proof Stress, UTS, Elongation and Impact Energy (four mechanical test results of a material that will be discussed in chapter3). Good model prediction accuracy was obtained, however it should be noted that the range of alloy additions was quite narrow. An iterative optimisation technique was then performed on the model and allowed a new alloy to be found which gave 50% more toughness without reducing the Proof Stress of the material.

Nakamura and Sagara[41] used an optimisation procedure for calculating the most cost-effective combination of reagents to use in a hot metal de-sulphurization process. This work aimed to find a replacement for an existing semi-automated optimisation technique based on a series of linear models. The model needed to predict the sulphur content at the end of a de-sulphurisation process for a range of operational conditions. The aim of the optimisation in this work was to obtain the required final sulphur content with the minimum financial costs. The authors looked for a unified model of the process to enable this optimisation to be successful, since piecewise linear models (one model for each region of the data), are difficult to maintain and often have gaps between the boundaries of each the models.

A unified model was constructed using the neural network technique with standard back error propagation. Although the authors said that the model was not very accurate due to the noise associated with the process data, optimisation was still possible. The authors mentioned that if the 'hidden' parameters pertaining to this noise could be found then the model could be improved. Input variables to the model were classed as controllable or uncontrollable. The controllable variables were examined using sensitivity analysis and two parameters, corresponding to two reagents used in the process were found to be significant to the optimisation procedure. The optimisation procedure employed used equal cost lines of the two reagents to analyse the operating range of the data collected from previous treatments. This enabled a series of guidance charts to be generated to optimise the use of the reagents for a given situation. This procedure was estimated to save 10% off the total annual reagent costs.

2.6.4. Model-Based process control

The motivation for process control may be to achieve better product quality or lower process costs. The control procedure may even be part of an optimisation routine. This area of the literature search concentrated on areas of steel processing where models were used as part of the control strategy. Kim et al[42] described the application of a model-based control strategy to a continuous annealing process. Steel strip passed through a number of stages in the process, each with a different temperature profile. One stage in particular, the rapid cooling section, required control dependent on the temperature, types of strip and other process conditions. The control actions available changed the amount of cooling applied to strips. Traditionally a look-up type table had been used to calculate the control action required, however it was felt that a more complex and continuous model may be more effective. A neural model was trained on 8,000 examples of process operation and was then used to provide the information previously gained by the look-up table. The results show that the neural network provided superior performance overall compared with the lookup table, however when the control action was very large, the look-up table gave a more stable result.

Rolling mills appear to be suited to control techniques incorporating neural networks, with their varying time delays, gains and complex interactions between stands. Most of the applications used the neural network for supervisory control architectures, where the neural network predicts the values of the controller set-up parameters. Examples of this type of work were a prediction model supplied set-up parameters in a supervisory manner include roll force prediction for a cold rolling mill[43] and parameter pre-setting for a temper mill[44]. Sbarbaro-Hofer, Neumerkel and Hunt[45] have used a neural model for direct control, and used a variety of control

schemes including model predictive control. In this work a radial basis network was used to construct a plant model. The authors used this technique as it encompassed a simple learning algorithm, and commented that it was easier to obtain a uniform distribution of the modelling error over the complete space of the training patterns. The noise sensitivity of the RBF technique was not a problem since training data was generated by solving an analytical model of the plant equations. Stability of the neural network control scheme was of primary concern and the authors showed that for certain kinds of control systems the models were input-output stable. The control performance of all neural schemes increased the control precision over a conventional PI controller.

In a hot strip mill, there is an area where strip is cooled, this is called the run-out table. The control of temperature in the hot strip mill is important because it greatly affects the mechanical properties of the product (in a similar way to batch heat treatment). Work by Loney, Roberts and Watson[46] investigated the cooling section of the run-out table as it was questioned whether the existing model used for feed-forward control was accurate enough. This work was based at the Port Talbot steelworks in Wales, and so a perspective of the industrialist's concerns was provided. The authors mentioned that a neural technique holds the advantage of providing complex non-linear mappings to describe process data, however they were aware that the technique is a black box one, and that it may yield unstable predictions under certain circumstances. It was also argued, however, that although concerns have been aired about empirical models over mathematical models based on process knowledge, mathematical models are often fitted to a particular plant using some process data, and therefore in some senses are data based. Various architectures of MLP network were trained, however no significant difference was seen between them. When compared to predictions of the existing control system model, the neural

network was shown to be superior, however for reasons of comparison, only the first stage of the cooling section was modelled. The neural model needed to be extended to cover the whole run-out table before it could be incorporated into the control scheme. In order for this to happen more data had to be available to train a reliable model. The authors placed emphasis on the importance of selecting appropriate examples from a larger database in order to build a suitable model.

It has therefore been shown that neural networks can be utilised for process control in two ways, either directly for model-based control or indirectly in a supervisory manner. The earlier case is rarer as the stability of the model is obviously very important and this work seems to be not as far advanced. Even with the supervised control schemes there are still concerns about stability[43], and a great deal of research still concentrates on creating reliable prediction models.

Although the work in this project did not encompass generating a control scheme for a heat treatment process, the considerations for making the prediction model still apply.

2.6.5. Fault detection/quality inspection

Fault detection and quality inspection are applications that span a range of industrial processes[47,48]. The problems of providing checks for product quality and consistency can often produce bottlenecks in processes, and fault detection on a process often requires the interpretation of multidimensional data.

As with the other categories in this review, one might argue that fault detection and quality inspection could be considered as an optimisation technique, since the result should be an improvement in product consistency and plant reliability. However special consideration is given to literature concentrating on fault detection and

quality inspection since the authors motivations were quite different to that of optimisation control and property prediction.

This section can therefore be divided into two categories, off-line fault detection and on-line (real-time) fault detection. Model-based offline fault detection has been applied to a rolling mill[49] to predict the occurrence of a defect known as "R5 Hook". The fault occurs at the roughing stage of the rolling mill, which is used to reduce the dimensions of a steel slab to a predetermined intermediate thickness or width. When the fault occurs the steel has to be removed from the mill and cannot be processed any further. A mathematical model was used to calculate the required set points to the mill regulators, however it could not account for the occurrence of the fault, which is related to the prior furnace conditions for a range of mill conditions. A variety of neural models were evaluated to predict the occurrence of this fault with the aim that the particular settings that cause the faults could be avoided. The authors comment MLP technique was found to provide better performance in the application than the RBF technique in this particular case. The model was used to evaluate which variables were most important in the generation of the fault.

Another off-line application is that of a ball steel process, where an adaptive logic network (ALN) was used to predict the percentage of faulty bars in a cast[50]. The ALN is a simplified, special case of the Multilayer Perceptron network. The difference is that its operation is strictly Boolean, with a fixed threshold so that each neuron is able to calculate the AND, OR, LEFT or RIGHT of its inputs. Because of its Boolean implementation, each output value contributes to one bit of a binary output value, and a series of nets is required to calculate the full output value. For continuous values to be evaluated all input and output data need to be encoded for a number of quantisation levels. In this paper, the bars were produced from scrap materials, which were melted down using an electric arc furnace. The molten metal

was then cast into blooms (bars with a square cross section), which were then rolled into rounds. After the rolling process, ultrasonic testing was used to detect voids. If the bar passed this test then it was reheated and made into balls. As with many of these types of modelling process, prior knowledge was used to select which variables were to be used in the model. The networks target values were the results of the ultrasonic testing. There were problems, however, that not all of the examples in the data set contained a full input vector, so there was missing information. This reduced the training set to 102 points. Due to the low number of examples in the data set, a technique known as "leave one out" cross validation was used to evaluate the model's generalisation performance. As the name suggests this technique entails using all but one example for training the network and then testing on that example. The example is then returned to the training set and the procedure is repeated until the entire data set has been tested. The cumulative test result can then be calculated, however as one might expect many iterations are required. One of the important features of this new type of model is the level of quantisation used in the encoding. When there are few data points it appears to be advantageous to use fewer quantisation levels and preserve the accuracy of the model (trading precision for the number of mistakes). The authors reported some success in the ability of the ALN to model the percentage of defect bars, considering it to be a quick way to obtain an off-line model for analysis of the fault, however they comment that the limited data reduced the success of the models.

When considering real-time fault detection the speed of the computation becomes more important. Neural networks are usually time consuming when the training phase is considered, however, they are able to provide a fast modelling technique for predicting. On-line models have been used in a variety of process fault detection applications, largely due to the need for interpretation of multidimensional

data. Although the cause of a fault may not be known, through supervised learning with appropriate process variables, detection may be possible.

Continuous casting has become an increasingly popular technique in recent years, enabling large quantities of molten steel to be cast into a continuous bar. There is however a serious fault that can occur in this process known as 'breakout'. This fault occurs because although the wall of the newly formed bar solidifies at an early stage, the centre of the bar is still liquid, and under certain conditions this liquid centre can literally break out of the bar's wall. This fault causes extended plant shutdown, and so model-based techniques have been employed to predict its occurrence. Nippon Steel in Japan used the neural network technique to forecast the occurrence of such a fault and claim "nearly 100% accuracy"[51], significantly better than previously existing methods. The models were tested off-line before implementing on-line. One problem, however is that, even with on-line implementation, by the time the fault is predicted, the breakout cannot be avoided. Production can, however, be halted to minimise the damage and down time associated with the fault. The models could also be used off-line to study the mechanisms behind the fault's occurrence.

Domingues, Campoy and Aracil[52] consider a very high-speed product fault detection application on a steel strip mill. This application used a CCD camera to inspect steel strip as it passes at speeds up to 15 m/s. Data are supplied to a neural network from a digital signal processing board which supplies 18 features of the image. There was no data available to train a supervised network and so a form of unsupervised network was used to classify faults and learn as faults occurred in the strip. The detection of a fault was fairly simple as it corresponds to a deviation in grey scale, however, the network performed categorisation of faults and helped to allow for noise on the data which could cause false classifications. Once a section of

strip had been inspected, it could be graded in terms of quality. These data were then passed to a production scheduling system that optimised the product distribution based on customer specification, and produced more strip where required. The important aspect here for this project was that feature extraction could be used on raw data before training the neural network.

To conclude this section, it has been seen from the literature that model based fault detection and quality inspection has been performed in a number of steel processes, both on-line and off-line. In most cases the principles of designing a property predictive model apply, in that the models need to describe complex interactions and non-linear effects. Neural networks are therefore also dominating this field, replacing existing models in certain circumstances. The problems associated with data acquisition apply once more and in one case this has lead to an on-line learning technique being applied.

2.7 Chapter conclusion

This chapter initially investigated the theory behind a number of data modelling techniques. The multi-layer Perceptron neural network technique was investigated in particular detail since it enables very flexible data based modelling and can easily accommodate for non-linearities and variable interactions.

Literature relating to the prediction of mechanical properties was then investigated, and it was seen that some work to predict the mechanical properties of steel has been undertaken, however, this often only relates to a relatively narrow range of steel types. Often in the literature, the performance of the neural network approach has been compared to that of linear and polynomial techniques. Many authors from a range of model application areas have found the traditional linear and polynomial techniques to provide reduced modelling accuracy, compared with the

neural network technique. The difference in performance is largely thought to be due to non-linearities and variable interactions in the processes.

The next chapter now introduces the principles behind the heat treatment process, together with information specific to the companies involved with the project.

Chapter 3

Heat treatment process

In this chapter the heat treatment process theory is examined in greater detail. Two industrial processes connected with the project as possible data sources are then presented. The BSES process is selected as an initial data source and information gathered from BSES about the process, at the initial stages of the project, is then discussed. Details of variables measured from the plant and material processed are then presented, together with their relative inaccuracies and incompatibilities.

3.1 Brief overview of the steel making process.

Before describing the heat treatment process, it is important to describe briefly the main stages involved in the production of steel. The basic steel making process consists of the following stages:

1 Traditionally, iron ore is made into molten iron in the blast furnace, using coke as a reduction agent and limestone to produce slag, which carries away impurities.

2 From the blast furnace, molten iron is transported to a basic oxygen furnace (BOF) for a conversion (or smelting) process, which produces molten steel. The major element removed from the molten iron in the oxygen steelmaking process is carbon, which is removed via oxidation to carbon monoxide (CO). Other impurities are also controlled in this stage.

3 Alternatively, molten steel can be produced in an Electric Arc Furnace (EAF); this procedure involves the melting of scrap charge by electric arcs. The main heat treatment process modelled in this project is fed by steel

produced from an electric arc furnace. The reactions in the EAF are similar to those in the BOF.

4 Steel makers are increasingly employing a relatively new process known as ladle metallurgy. This involves molten steel from the EAF or BOF being poured into a refining vessel, where the temperature and composition of the molten iron are closely controlled to produce various grades of steels as required by the customer.

5 Traditional steel-making involves the production of ingots, where molten steel is poured into moulds to produce large blocks. After ingots cool they are reheated to a uniform temperature, and shaped into semi-finished sections such as blooms, billets and slabs.

Continuous casting is, however, rapidly replacing the production of ingots. This involves the use of a casting machine that is capable of producing a continuous piece of solid steel, giving higher yield than ingot formation. The intermediate step of rolling ingots into semi-finished sections is also avoided.

6 Following stage 5, further rolling, forging and heat treatment may be required to obtain the correct geometry and properties in the finished product. This is of primary interest to the project and will be considered in section 3.3

3.2 The structure of steel and the effect of alloy addition

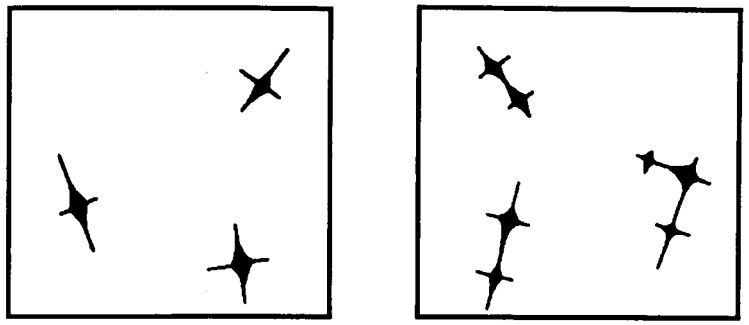
Another feature of steel that is very important when understanding the heat treatment process is its structure. This section presents some basic information to show how steels are composed, however more detailed information can be gained from heat treatment and physical metallurgy text books[53,54]. Much of the information presented in this section has been gained from attending courses within the Department of Engineering Materials at the University. In providing background metallurgical information on the casting and rolling stages, as described in section

3.2.1 and 3.2.2, the major mechanisms believed to affect the mechanical properties of steel are presented. However, it will also be seen that while previous research by metallurgists has enabled discovery of the knowledge presented in this section, it does not allow one to accurately estimate the mechanical properties resulting from a given combination of treatments. The heat treatment process will be considered in section 3.3. Throughout the production route the mechanisms presented here may interact in a complex manner and other more exotic underlying mechanisms not mentioned in this section may also be present.

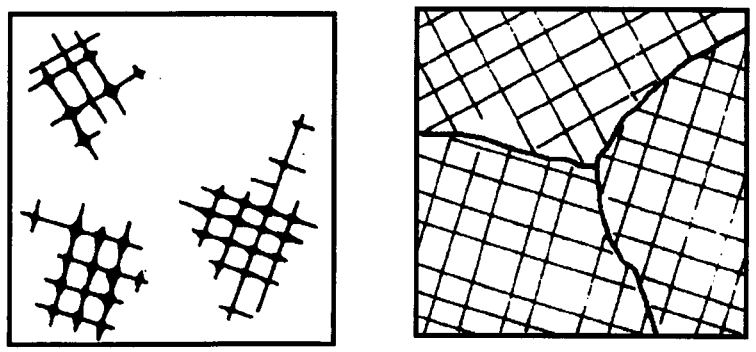
3.2.1 Casting

Metals have a uniform, geometrical arrangement of atoms that is repeated through the material and are therefore said to be crystalline. The regular 3-D atomic pattern is known as the space lattice and the unit cell is the smallest unit of the lattice that retains the overall structure of the lattice. The term crystal structure, refers, therefore, to the size, shape and atomic arrangement of the lattice unit cell and is what varies from one substance to another.

When liquid metal is cooled below its liquid state, the process of solidification occurs. A process of nucleation and growth achieves the solidification of a crystalline structure. When a pure molten metal is cooled just below its freezing point, minute nuclei of the solid form in the liquid. The crystal then grows in a tree-like formation called a dendrite. The main 'trunk' of the dendrite grows quickly in the direction of the fastest heat loss and then secondary and tertiary branches develop in a geometrical pattern consistent with the lattice structure. Each dendrite then continues to grow until it meets other neighbouring ones, at which point the branches thicken to form a totally solid grain of metal. Each grain has a similar lattice structure but usually has a different orientation. This concept is illustrated in Figure 3.1.



(a) formation of nuclei (b) dendrites formed



(c) dendritic growth (d) grain formation

Fig.3.1 Dendritic growth of crystal grains

Grain size is a very important feature affecting the mechanical properties of a metal. The grain size is affected by the rate of cooling of a metal, gradual cooling will result in only a few nuclei being formed and therefore a large grain size, whereas rapid cooling produces many nuclei and therefore a small grain size. When casting molten steel it important to remember the effect of temperature on the grain size since the cooling due to the mould temperature may not be uniform.

A key feature of interest to this project when considering solidification is that it is the impurities in the metal which provide the centre for the growth of the grains (heterogeneous nucleation). The solidification of a pure metal will exhibit no dendritic growth. The reason why this is important, is because the addition of impurities can therefore produce a larger number of grain boundaries, which in turn can prevent a defect from passing through a given metal. This mechanism is termed

grain size strengthening. However, too many impurities can also weaken the material.

Another strengthening mechanism in metals is that of solid solution strengthening. This is achieved by intentionally introducing interstitial or substitutional defects in the crystal lattice. An interstitial defect occurs when an extra atom is present in the lattice, which causes compression of the surrounding atoms (Figure.3.2). A substitutional defect occurs when alloying compounds of impurities cause an atom to be replaced by an atom of a different substance (Figure.3.3). Both defects cause stress fields to form around them and therefore it requires a higher stress to deform the material at the point of these defects.

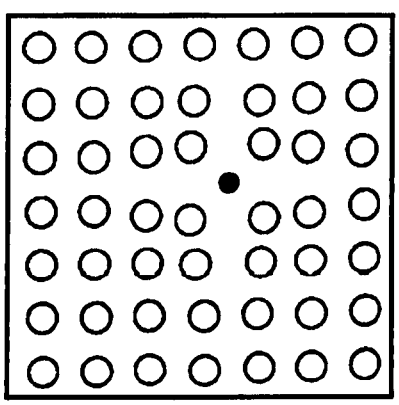


Fig 3.2 interstitial point defect

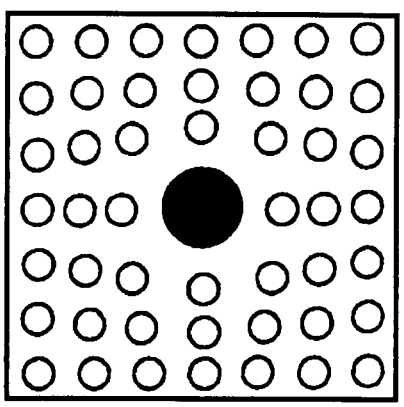


Fig 3.3 substitutional point defect

3.2.2 Rolling

Steel is rolled for two main reasons, to affect the mechanical properties and to produce the required section size or finish. A further strengthening mechanism is present when considering the rolling process, which is the result of another lattice imperfection. This strengthening mechanism is known as strain hardening. Whilst at the rolling stages connected with this project strain hardening should not occur, it is desirable to explain how it is avoided.

An incomplete line of atoms within the lattice causes a linear imperfection, which is termed a dislocation. Dislocations occur as two types, edge dislocations and screw dislocations. If a shear force is applied perpendicular to the plane of the atoms where an edge dislocation is present, the bonds between the atoms in the next row can be broken down such that the original row where the dislocation was is now complete. In this way a dislocation can move through a material. This phenomenon is termed slip, and is demonstrated in Figure 3.4.

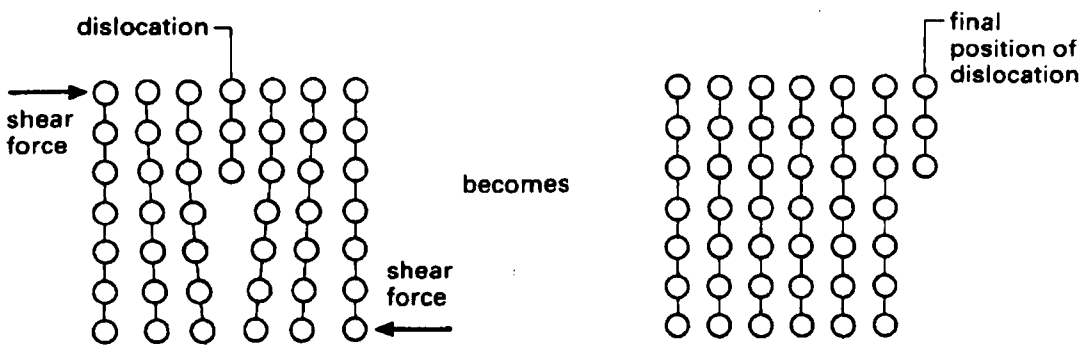


Fig. 3.4. The movement of a dislocation through a metal

The strength of a metal may be increased through the presence of many dislocations since they will block one another and may also be blocked by point defects as mentioned in section 3.2.1.

When work is done to a metal at an ambient temperature such that strain hardening occurs it is termed cold working. As will be seen in section 3.3, when a metal is reheated, recrystallisation may occur, therefore the effect of doing work on a steel above the recrystallisation temperature is quite different as dislocations may be removed. This is termed hot working, and is a method commonly adopted when rolling to produce the required sections prior to heat treatment. Hot rolling with a large reduction in sectional area produces a refined grain structure, which can lead to improve mechanical properties later in the process.

3.2.3 The iron carbon phase diagram

Iron is said to be allotropic, that is it can have more than one crystalline structure. There are three structures that can be formed, these occur when it is cooled below its freezing point of 1535°C, at key temperatures, known as arrest points.
Each of these three phases has either a body-centred-cubic (BCC) or face-centred-cubic structure (FCC), as shown in Figs 3.5 & 3.6.

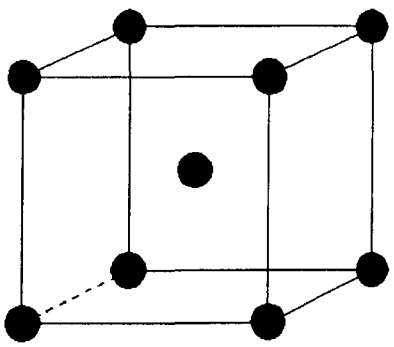


Fig. 3.5 Body-centered-cubic structure

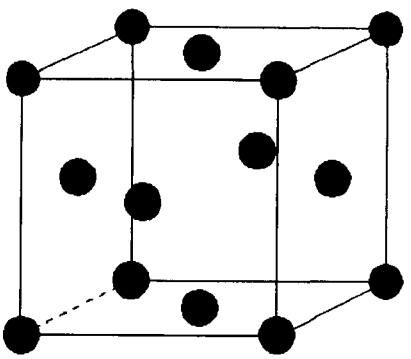


Fig. 3.6 Face-centred-cubic structure

The three phases of iron are:

- δ iron, a BCC structure stable above 1390°C
- γ iron, a FCC structure stable between 1390°C and 910°C
- α iron, a BCC structure, stable below 910°C

The changes in the structure that occur on heating iron do not occur at the same temperatures. The addition of carbon to iron also alters the arrest points, and apart from in the liquid phase, carbon is only partially soluble. When carbon combines with iron, it forms interstitial solid solutions of carbon atoms in iron. In BCC structures there is only limited solubility of carbon, which produces ferrite at low temperatures and δ ferrite at high temperatures. In FCC structures increased carbon solubility occurs and austenite is formed by solid solution strengthening, providing ductile yet increased strength when compared to pure iron. When the solubility in a

given structure is exceeded, a compound called cementite is formed, which is hard but brittle.

An iron-carbon equilibrium phase diagram as shown in Figure 3.7, is used to show the phases present in iron carbon alloys up to 6.7% carbon content. The term equilibrium is used to describe the fact that the phases have had the required amount of time to fully transform. Strictly, Figure 3.7 is not an equilibrium diagram as

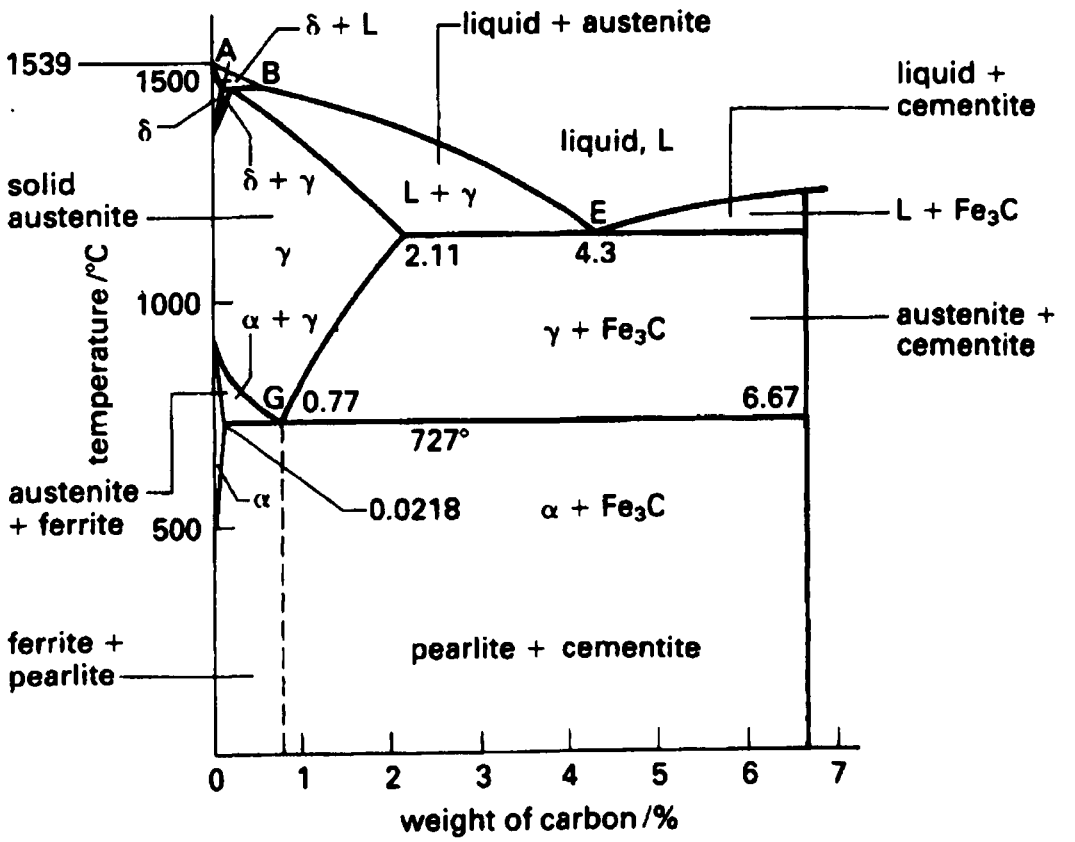


Fig.3.7 The iron-carbon equilibrium phase

cementite can decompose to graphite and so it is called a metastable system. Throughout the diagram there are key points which indicate certain types of reactions between the phases. Point E, at 4.3% carbon and 1130°C is the eutectic point at which liquid alloy solidifies to a mixture of austenite and cementite.

Point G is an important point of the phase diagram since here at 0.8% carbon and 737°C solid austenite transforms into a fine laminar structure of soft ferrite and harder, more brittle cementite, by a process of diffusion. Under a microscope this

laminar structure has a pearly appearance and is known as pearlite. Steels with less than 0.8% carbon are referred to as Hypoeutectoid steels, and in a similar manner those with more than 0.8% carbon are known as Hypereutectoid steels.

The iron-carbon equilibrium phase diagram has been included to demonstrate how phase transformations can occur in plain carbon steels. However in reality it is rare to find commercially produced steels with more than 1.4% carbon. Commonly, engineering steels are hypoeuctectoid. Other alloying additions such as manganese, nickel and chromium are also added to improve the steel's properties. Elements such as phosphorous and sulphur are present in the original ore and are classed as impurities.

3.3 Heat treatment

It has been seen that point defects and other imperfections in the structure of a steel are crucial to the subsequent mechanical properties. It has also been noted that the changes in the crystalline structure are related to the solidification process through dendritic growth and that the grain size and orientation is related to the temperature at which this occurs. Moreover, as the solid steel cools, the equilibrium diagram shows that more phase changes can occur, however because the mechanism for these changes is solid state diffusion, these changes are unlikely to proceed to conclusion. This in turn can cause coring, which is a situation where the composition varies between the grain boundaries and the centre of the grain. Coupled with the volume change that occurs between the FCC. and BCC. structures due to the FCC crystal structure being more densely packed, this causes many internal strains to result in the metal. The structure will tend to re-arrange its atoms such that these strains are relieved, however this can only occur when the atoms are excited into moving, i.e. by the steel being re-heated. For the re-arrangement to occur this heating

needs to occur at a sufficiently high enough temperature, and for a long enough time to allow the transformations to occur. The procedure for heating steel to a specific temperature, for a period of time, is known as soaking. It is a general rule that when the steel's internal structure is strained, it has a higher strength, but is less ductile. Therefore when these stresses are relieved, ductility improves, but the material is weakened.

If a steel containing a sufficiently high amount of alloying addition is cooled fast enough, then non-equilibrium structures of bainite and martensite can be formed.

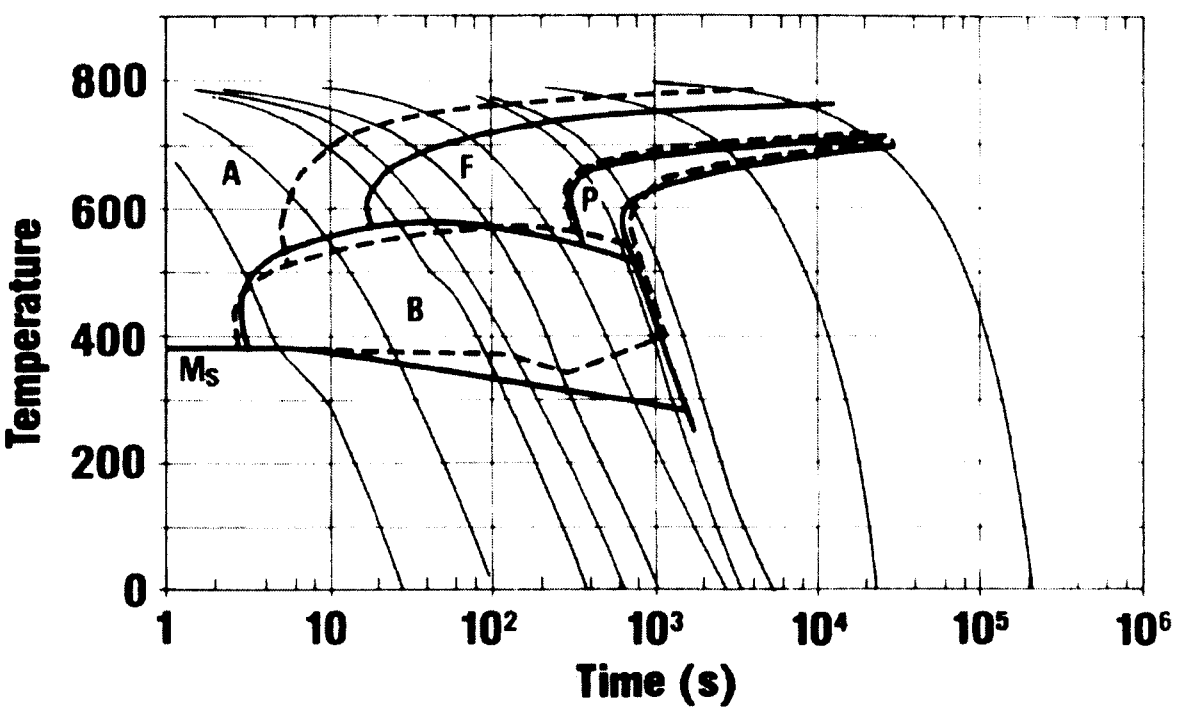


Fig. 3.8. Continuous Cooling Transformation diagram for a 34 CrMo 4 Steel

In chapter two the modelling of a continuous cooling transformation (C.C.T) diagram was described, which was at that stage described as a diagram showing the microstructures which would result from near constant cooling of a given steel. Such a diagram is shown in Figure 3.8. This is demonstrated for a 34CrMo 4 steel although it must be remembered that the phase boundaries will be different depending on alloying addition.

The continuous cooling transformation diagram shows various cooling trajectories for increasing times to the right. The cooling curves cross various bounded regions, which show the microstructures that will be generated. The letters M, A, F, P, B are used to represent the Martensite, Austenite, Ferrite, Pearlite and Bainite regions respectively. In reality, most industrial processes would aim to either produce the martensite or pearlite microstructures, since bainite is a very variable microstructure which is more difficult to control. The martensite microstructure is a highly distorted form of the ferrite BCC microstructure. To obtain sufficient hardness in steel it is common for industrial processes to aim for the martensitic microstructure as it can be later tempered to obtain the required mechanical properties. Figure 3.8 indicates, however, that to obtain a martensitic microstructure, rapid cooling needs to occur, however on an industrial scale this may not be practical with large pieces of material. An additional complication is that if a quench which is too harsh is used, then cracking will occur in the steel due to the thermal stresses. Due to the complications of obtaining a rapid quench, alloy additions to improve the hardenabilty of steel are made, examples of these are chromium, molybdenum, manganese, nickel and occasionally vanadium. These elements act so as to move the phase boundaries, so that martensite can be formed at lower cooling rates. If enough hardenabilty exists, but there are problems of cracking, an oil, or even polymer quench can be used to give a slower cooling rate. This is therefore the hardening stage of the heat treatment process, however, the martensite results in brittle steel that would be impractical for most engineering steels, therefore tempering is required to transform some of the martensite into a tougher structure.

Before describing the tempering process it is first necessary to describe the annealing and normalising processes.

There are two forms of annealing, subcritical and full annealing. Subcritical annealing is used to remove the effects of work hardening and takes place over the temperature range of 650-700°C (no phase changes). It is important that the component being annealed is allowed to soak so that it reaches an even temperature, and it must then be allowed to cool slowly.

The second annealing process, full annealing, is used to reverse the formation of non-equilibrium structures. This means that the steel must be heated to allow all of the bainite or martensite formations to revert back to austenite. Once again it is necessary to allow the components to soak (stay at a given temperature in the furnace), until the transformation to austenite is complete and then cool slowly to give equilibrium structures. Slow cooling may be achieved by allowing the steel to cool in the furnace or in a brick lined chamber.

Normalising is a process closely related to annealing and is used on steels to give a finer grain structure. The very slow cooling of annealing gives a soft, ductile structure but there is also excessive grain growth. A faster cooling rate usually achieved by cooling in air, gives smaller grains and therefore a tougher and stronger material.

The definition of normalising may be complicated by the fact that in industry normalising is a term used for air cooled heat treatments, however through alloy additions it is possible to have an air hardenable steel.

The tempering process, like the annealing and normalising processes is very temperature dependent. The amount of martensite, which may transform is related in a complex manner to the alloying additions made to the cast. The re-heat temperature is important, however, because it shouldn't be so high that too much grain growth occurs, and the steel becomes too weak. After 'soaking' the steel at the required

temperature, the material is allowed to cool at a lower rate than with the hardening stage, in either an air or an oil medium.

From this section and the previous sections so far in this chapter it has been seen that metallurgical theory, even the basic material covered here, can describe the mechanisms of microstructural formation, which lead ultimately to the mechanical properties of the steel. However, this has only been demonstrated for fixed compositions that do not contain complex alloying additions, and has not been directly related to mechanical properties, although this may be done. The problem is that to predict the mechanical properties of steels in this manner (based on physical models), much more knowledge would be needed, which could only be the culmination of time consuming experimental work, which may then not cover an area of commercial production needed. Indeed it is widely accepted that human judgement based on past experimentation, coupled with a metallurgical knowledge, are the techniques currently used to 'predict' which mechanical properties may result from a given heat treatment and alloy addition. The theory does however show that the solution predicting mechanical properties is likely to encompass non-linearities, if one considers the situation where phase changes in the material may be likely to occur. This is the impression given by the 'text book' description of the process, however, the next stage is to look more closely at the actual industrial process involved with the project, and examine under these conditions what other complications may result.

3.4 The industrial process

At the beginning of the project a meeting was held with the Materials Forum members, to establish a suitable process to investigate. The process chosen needed to be potentially beneficial to most Forum members and so those involving obscure

steel grades requiring unusual treatments unique to one manufacturer were to be avoided.

Another function of this initial meeting was to establish how the Forum members interpreted the project brief. An important aspect is that all parties should agree on a common goal, and so it was useful to confirm that what would be initially expected, was a predictive model capable of predicting mechanical properties. At this initial stage it was felt that the prediction of UTS would be the best place to start, and concerns relating to the prediction of impact test results were aired - this will be discussed later in the mechanical test result section.

It was suggested that the batch heat treatment processes of BSES and Aurora forgings would be suitable for the modelling work, these processes were to be investigated so as to establish a suitable starting point for the modelling work. Suggestions from metallurgists present at this meeting were that once a process was selected it would be advisable to firstly concentrate on a specific composition and treatment pattern, in order to investigate the potential of the modelling techniques.

Site visits were arranged to British Steel Engineering Steels and Aurora Forgings, in order to establish the best process to concentrate on first, and the following points were noted in the following sub sections.

3.4.1 British Steel Engineering Steel's heat treatment processes

The heat treatment process at BSES, as mentioned in chapter1, is concerned mainly with mid-low carbon steel, which has been cast into ingots and then rolled into round section bars, or occasionally square section bars. A range of heat treatments are used throughout the sites connected with BSES, these include hardening and tempering, but may also include annealing, normalising and dummy

nitride techniques depending on specific design requirements. Occasionally multiple tempering treatments may also be performed. It was evident from the visit that treatments normally occurred in a pattern of different stages, such that the resulting product qualities were achieved as required by the customer.

Although the steels have been rolled prior to heat treatment, it was decided that, because during the hardening stage the microstructure would become fully austenitic, the effect of the strain hardening effects of the rolling could be ignored since full re-crystallisation would have occurred. Therefore, when modelling any treatment pattern commencing with a hardening stage, one can consider the microstructure to have approximately the same starting point.

During the hardening stage, batches of bars that may individually weigh up to 4 tonnes and have a cross sectional diameter of 400mm, are loaded into large furnaces in a pack of bars as indicated in Figure 3.9.

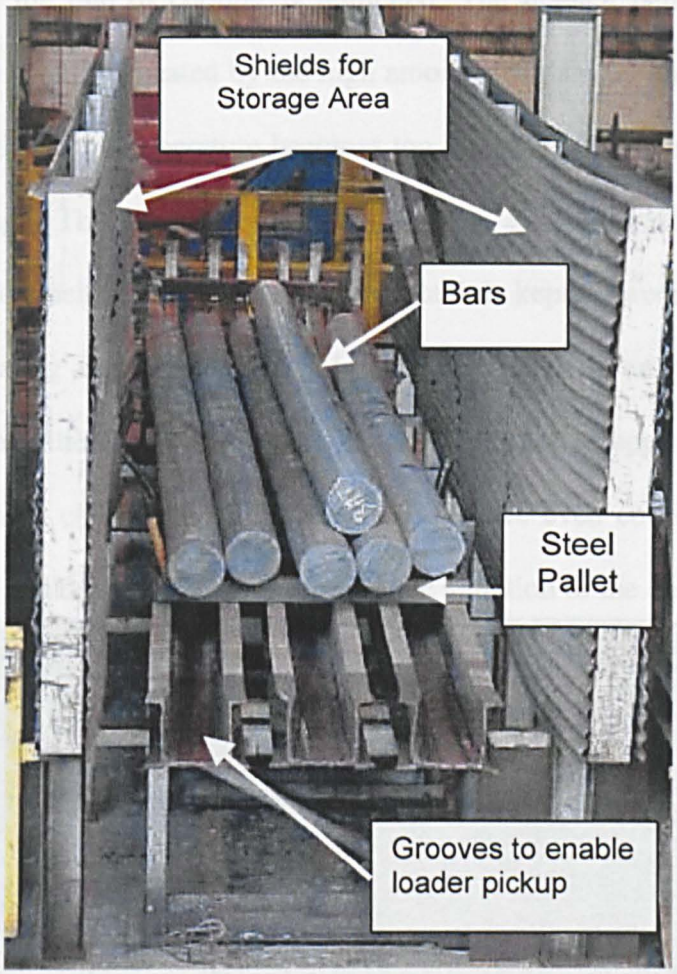


Fig 3.9. Typical arrangement of bars on pallet prior to austenitising treatment

A special loader on a parallel track performs this task using pallet arrangement supporting the bars. During the hardening stage, the bars are soaked at a temperature around 860°C, which may vary slightly depending upon the treatment required. Figure 3.10 shows the removal of the bars from the furnace for transfer to an oil quench tank after the austenitising stage. Furnace soaking times have to allow for the austenite transformation, and are also dependent on bar diameter and the volume of the furnace charge, however a soak time of 5 hours is typical. Once the steel has soaked at its required austenitising temperature it is then either transferred

to a quench bath (for cooling in either water or oil mediums), or is allowed to air cool (for slower cooling), this may also be performed in a brick lined chamber for very slow cooling. Transfer to the required quenching medium must occur as rapidly as possible so that the rate of cooling is regulated. The quench bath temperature is controlled, since it will be heated by the high amounts of thermal energy released by the bars, and if the bath temperature becomes too high the cooling effect will not be significant enough. The water quench tank contains a cooling skid such that the temperature is kept below 38°C. The oil quench tank is kept between 38°C and 80°C and so both heating and cooling control are used. There is an additional safety interest here, since the oil quench media has a flash point of approximately 200°C and also has a high coefficient of expansion. To ensure even cooling, the bars are either moved up and down by the crane to provide agitation or the coolent is agitated.

Fig 3.10. Bars at 875°C being unloaded from the furnace before transfer to an oil quench

The tempering stage, which then follows, involves re-heating the steel to a lower temperature, around 630°C, so as to allow the transformation of some martensite to the softer structures as mentioned earlier. Cooling at this stage is usually performed using air, however oil can also be used. With the steels manufactured in the processes connected with the project, the tempering stage is crucial to the mechanical properties of the resultant material. The hardening stage will tend to mainly produce a fairly uniform structure throughout the range of steel made. For scheduling and specialist treatment reasons, up to five furnaces may be used at each of the heat treatment sites.

The furnaces used in the heat treatment process vary depending upon the treatment that is being performed. The control and heating mechanisms used in the furnaces also vary, however, complex thermodynamic and control aspects are considered for each type of furnace charge to ensure effective heating in the manner required for the treatments. The furnaces used by BSES have gas fired burners which either act from each end of the furnace (resulting in one zone), or in pairs along the sides of the furnace (resulting in multiple zones). The configuration used is dependent upon the length of the furnace, since burners have a limited firing range; therefore a furnace length exceeding 6m would normally be a multiple zone type.

Figure 3.11 shows a typical arrangement of burners and control thermocouples in a multiple zone furnace. The burners are offset so as to provide gas flow through the furnace. The thermocouples are placed in the gas stream at the opposite side of the burner they control. Linings and furnace shape can also affect the heating mechanism, together with the amount of air that is allowed to flow through the furnace. At the high temperatures used for the hardening stages, gas and airflow rates are reduced such that radiant heat is the main mechanism. In the tempering

stage, the temperatures are lower, therefore the heat radiation is lower and the velocity of the burner's gases is the main mechanism used to heat the bars, with

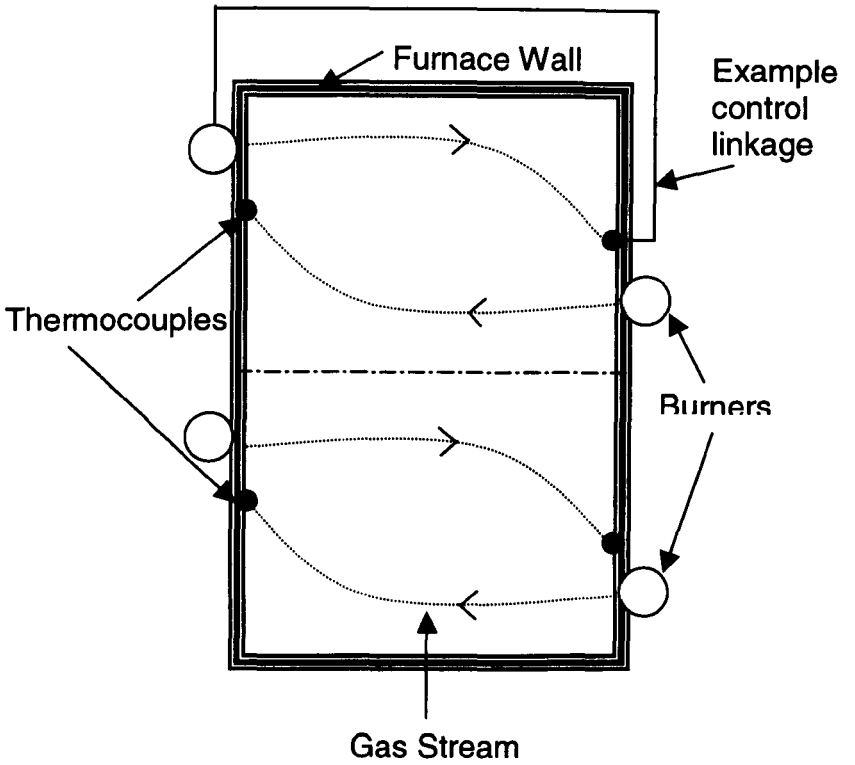


Fig.3.11 Typical control scheme for multi zone heat treatment furnaces

increased gas and airflow. The furnace will therefore run less efficiently in the tempering stage, however this is required to force the heat between the bars.

One important thing to remember is that particularly at the tempering stage, the temperatures throughout the furnace may vary, therefore it is very important where thermocouples are placed for control and temperature measurement. For this reason it is usual to use perhaps 12 thermocouples distributed throughout the furnace for temperature measurement, and then a control thermocouple in the gas stream per burner in the furnace.

There are two types of calibration performed with the furnaces, the first is the thermocouple calibration and the second type is a dummy charge calibration. The thermocouple calibration is typically made to within $\pm 0.5^{\circ}C$ at $1000^{\circ}C$. The dummy

charge consists of either a frame or scrap bars, containing thermocouples, to look at the temperature distribution with respect to the heating of the bars. This can be used to formulate improved control or burner settings. The overall furnace temperature provided by the control system is ± 5°C below 750°C and ± 10°C on temperatures above 750°C.

When the required treatments are completed, mechanical property testing is performed along with a visual inspection for surface defects (for example cracking or warping). A full description of the procedures used for obtaining the mechanical property test results will be provided in section 3.7. Depending upon the customer order, the test results obtained from the heat treated product must lie within the range dictated by the z-card, which is a type of specification document, stipulating key features which the end product must fulfil.

The production method described above is known as the 'as treated' product supply, however steel may also be supplied 'as rolled', to customers wishing to perform their own heat treatment. In the 'as rolled' case, a certificate has to be provided with the steel supplied, to confirm that under certain treatment conditions, the mechanical properties required by the customer can be achieved. This is done via a laboratory scale heat treatment that is performed on a sample of the steel, which is going to be supplied without treatment. The hardening and tempering operations in this laboratory process are similar to those described in the plant, except that the furnaces and heat-treated sections are smaller.

Throughout both laboratory and plant based processes, a database is used to record information relating to the metallurgical analysis of a particular cast of steel, the subsequent heat treatments including temperatures and quench mediums, and the mechanical test results and conditions relating to the tests. This information can be recalled on a chronological, metallurgical or specific batch basis, and has been held

in this form since 1986. Although process instrumentation is connected to the computer systems throughout the works, the real-time process information used for control is not logged automatically to the database. Information about each heat treatment is instead entered at discrete intervals throughout a given product's production. Because customer requirements vary, not all-mechanical tests are always performed.

3.4.2 Aurora Forging's heat treatment process

The Aurora Forgings heat treatment process has two main sites of interest from a modelling point of view. The first is the Meadowhall works and the second is the Parkgate works, both are situated in South Yorkshire in England. A third site, at River Don, specialises in drop stamp and press forgings, some of which are for the automotive industry, these processes do not however involve heat treatment. Representatives present at the first visit to the site explained that the Parkgate works specialise in stainless steel and titanium alloys using open die forgings and batch heat treatment. The Meadowhall works has several specialist forging processes, however the steels used here are closely related to the compositions of BSES. Despite the heat treatment equipment of the Parkgate works being newer and possibly providing more uniform results, it was advised that the Meadowhall plant may be of more interest due to the applicability of these alloy steels to the rest of the Forum.

Aurora is not a steel manufacturer, instead it buys steel in an 'as rolled' form, 80% of purchases are from BSES. This steel is then used to make various parts for Rolls Royce, the petrochemical industry, the M.O.D, mining machinery manufacturers and producers of large earth moving machinery such as Caterpillar and Komatsu. Depending on the application the steel ordered will have to meet certain mechanical testing requirements when it is treated. Figure 3.12 shows the

interaction between the steel supplier and Aurora (the purchaser and forging supplier). Steel is purchased with a test certificate that is proof that the properties ultimately required by the forger can be achieved, this is done by the lab treated sample from the main bar. The forger then performs their forging operation and then heat treats the forged component. They then test a sample of the completed forging to prove to their customer that it meets their requirements.

The Meadowhall works houses a 3,500 tonne and 5,500 tonne CNC controlled back extrusion press, which enables hollow cylindrical forgings to be made thorough a multistage process. Prior to the forging process the billet needs to be heated in a furnace at 1200-1500°C, this softens the billet, allowing it to be deformed easily. Several re-heats may be necessary to enable the desired shape to be achieved. When the forging is complete, it is allowed to air cool, before the heat treatment process.

In a similar way to BSES, the forging process commonly consists of a hardening and a tempering stage. The hardening stage is operated at temperature ranging from 800-900°C. The forged components are put into the furnace in batches,

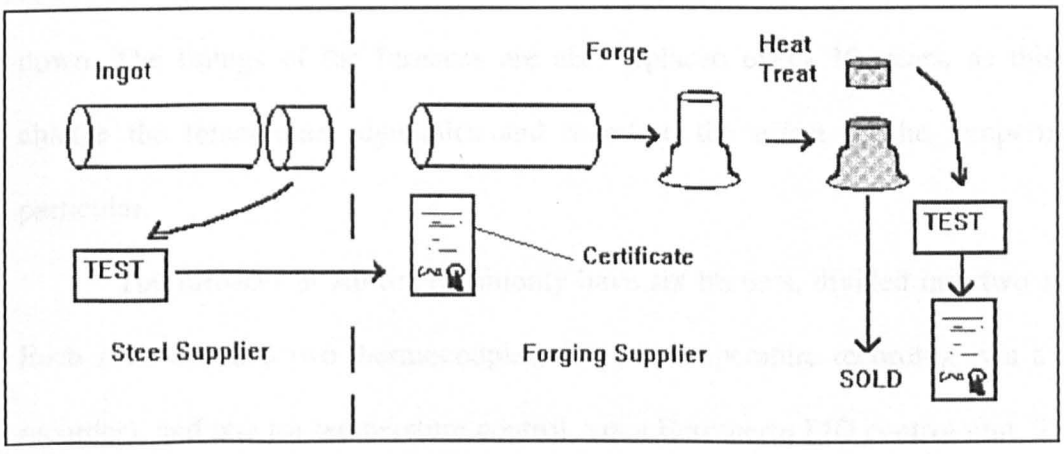


Fig. 3.12. The interaction between forger and steel supplier

however unlike the BSES process the configuration of this loading is dependent on the geometry of the components. When the components have soaked at their

hardening temperature they are transferred to a quench bath containing oil, water, or a polymer quench. The quench tanks contain heat exchangers to allow cooling when the temperature becomes too high. Agitation of the quenchent is provided via a pump as opposed to the movement of the bars as with BSES. All quenching media are also monitored for quality, in particular the polymer quench, which requires its concentration to be checked daily with a refractometer along with quarterly checks of the long chain polymer lengths as these can break down with use.

An unusual feature of Aurora is that, due to some component shapes, vertical as well as horizontal heat treatments can be performed. This is usually required for long components where distortion is critical, it should be remembered that these components are very size critical and changes in size due to heat treatment can be a problem.

Apart from one electric furnace, all furnaces at Aurora are gas fired. Different furnaces are used for hardening and tempering. The hardening furnaces tend to have greater convection, providing a fiercer heat, whereas the tempering furnaces tend to provide less convection. Furnace lining may also vary with some having brick lining and some having fibre linings, this affects the rate at which they heat up and cool down. The linings of the furnaces are also replaced every 10 years, as this will change the temperature dynamics and therefore the effect of the tempering in particular.

The furnaces at Aurora commonly have six burners, divided into two zones. Each zone contains two thermocouples, one for temperature recording (via a chart recorder), and one for temperature control, via a Eurotherm PID control unit. This is a similar arrangement to that of BSES. Additionally there is a 'police' thermocouple, by way of a safety mechanism, to override control of the two zones if the temperature becomes too high. Charts of the furnace temperature are kept with

reference to a batch number to allow investigation into possible heat treatment problems.

All furnaces are calibrated to within ±5°C at 500°C and ±10°C at 700°C every 3 months. This calibration is performed using a dummy charge as with BSES, which enables the thermocouples to be adjusted. Due to furnace dynamics variations, certain regular orders are often treated in designated furnaces to ensure reliable results.

When the heat treatment is completed hardness testing is first performed, as a go, no-go test. This simple test, described in section 3.7.4 is first used because further tensile and impact tests can be expensive to perform. A Brinell hardness test enables a cheap, rough prediction of the tensile properties of the material. If the hardness properties are sufficient then further testing is performed.

As with the BSES process, details pertaining to the steel analysis forging and heat treatment were available in a database format.

3.5 Selection of an appropriate process

This project was now in the fortunate position of having two potential processes from which to draw data. However it was decided that initially only one process should be investigated, due to the incompatibilities which the data from each process may have because of differences in the processes and working practices. It was therefore necessary to decide which process to concentrate on in order to build a suitable model. It was decided that the BSES process would be chosen for the following reasons:

- The bar form prior to rolling would provide a more uniform mechanical history (Aurora's components vary far more in geometry from batch to batch, depending on the customer, which will also affect the microstructure of the untreated steel).
- BSES appeared to have data available on a standard geometry of product throughout a range of treatment sites, which also perform a similar working practice.
- A wider range of steel analyses are treated within the BSES plants
- The laboratory treatments would cover a range of analyses under one 'plant' which may also have closer control and therefore could produce a source of 'pure' data.
- BSES were keen to provide data that could first be selected by an expert in their processes, thus enabling the data collected to meet the needs of the modeller.
- The Swinden Technology Centre would enable metallurgical support in the form of evaluation of the modelling process as it develops whilst also providing help in a technical context.

The next step was now to establish in what form the data were available and what types of steels should be concentrated on first.

3.6 The BSES data

In this section, an introduction into the variables contained in the available data is presented, together with some points to note about the use of these variables from a modelling perspective. Much of this section is part of the knowledge acquisition process undertaken at the beginning of the project. It was essential to gain a detailed picture of the process procedures and problems in order to construct an accurate model. This knowledge was developed through meeting metallurgists and

specifically a test house manager at BSES Stocksbridge. A common problem with this process is that a person who is an expert on a process, may be too close to it to explain the assumptions or motivation behind what may be a basic procedure that may be important in the context of the resulting model. It was therefore important to question all aspects of the process in order to extract this information. Jackson[55] provided a useful insight into this problem.

3.6.1. The 'MET' database

At BSES all data relating to previously heat-treated steels is kept on the 'Met' database. This database tracks orders from cast to batches of heat-treated bars, using a variety of codes.

When a cast of steel is made, it is given a Cast Number, the cast may be larger than the amount of steel which can be heat treated in a single batch, and so each batch of heat treated steel from a cast is allocated a heat treatment batch number. It is therefore possible for several heat treatment batch numbers to relate to the same composition code, however this does not mean that the treatment profile needs to be the same. The allocation of cast and heat treatment batch codes is important, because it allows for checks to be made relating to the treatments given to certain orders, and in the event of a problem, allows the user to reference furnace temperature logs and test samples pertaining to the treatment. In chapter 5 it will be seen that these codes have also been of assistance in the modelling process.

3.6.1.1 Process and product variables contained in the database

The MET database contains a number of variables relating to the process, the cast, and batch of heat-treated steel in the process, these are shown in table 3.1, together with an explanation of their significance.

Variable	Explanation	
Cast	The cast number (as explained in Section 3.4.1)	
Year	The year the cast was made	
Comp	An alpha numeric code relating to the composition of the cast	
Piece_id	A unique identifier for tracking individual bars through the plant	
Position	The position of the bar in the ingot	
Specimen Depth (mm)	The depth in the treated bar from which a mechanical test sample is taken	
Specimen Orientation	The orientation can either be transverse or longitudinal in relation to the length of the bar	
Size (mm)	Original size of the bar	
Forge Size (mm)	The size of the bar heat treated (normally the forge size will equal the size for plant based treatments, but will be significantly less than the original bar for laboratory treatments.	
Heat Treatment Batch Number	An alphanumeric code relating to the batch in which the piece was treated (as explained in Section 3.4.1)	
Location	The location at which the treatment was performed	
C	Carbon	The chemical composition of the metal in the heat-treated item. This includes alloying additions and residual levels of impurities present after the steelmaking process. This is measured as a % weight.
Si	Silicon	
Mn	Manganese	
P	Phosphorus	
S	Sulphur	
Cr	Chromium	
Mo	Molybdenum	
Ni	Nickel	
Cu	Copper	
N	Nitrogen	
Nb	Niobium	
Ti	Titanium	
Al	Aluminium	
V	Vanadium	
W	Tungsten	
Z-Card	An alphanumeric code relating to a set of properties, test conditions and heat treatment schedules which the finished product must have.	
loc*x*	The location at which a specific treatment was performed (if different to the overall location)	The treatment schedule may incorporate up to 6 treatment stages (x = 1:6)
treat *x*	The treatment type performed	
temp*x*	The temperature at which the piece was soaked	
cool *x*	The quench medium used to cool the piece	

Table 3.1. The product and process variables available in the MET database

The variables relating to the mechanical test results are shown in Table 3.2. These relate to the tests described in Section 3.7. The data relating to the variables in this table is generated by the routine testing laboratory, and is performed on all heat treatment batches.

Variable	Explanation
Elongation gauge length 4D or 5D	The specimen length for the elongation test sample (either 4 or 5 times diameter of specimen)
.1% Proof stress (N/mm^2)	The percentage proof stress of the treated sample
.2% Proof stress (N/mm^2)	
.5% Proof stress (N/mm^2)	
Yield Stress (N/mm^2)	The Yield Stress of the sample
Reduction of Area (%)	The percentage reduction in area during elongation
Elongation (mm)	The elongation of the relevant specimen diameter
Hardness	The Brinell hardness value of the sample
Impact 1 a (Joules)	The impact energy required to fracture the sample type used, three measurements are taken for reliability. A second set of impact results may also be present (b).
Impact 2 a (Joules)	
impact 3 a (Joules)	
Temperature a (°C)	The temperature of the specimens in the impact test (may be different for a second set of impact results.
Type (a)	The type of impact test used for the current set of impact results
Units (a)	The method used to measure the impact energy (Backed off or absolute)
Test Reference	A unique code relating to a set of tests
Result Status	A flag to signify if the sample passed or failed the specifications set in the Z-Card

Table 3.2. The mechanical test result variables in the MET database

The variables in Tables 3.1 and 3.2 can be downloaded from the MET database in a variety of spreadsheet type forms, for the required compositional range, treatment batches or cast numbers required. In this way the variables are the columns and the data points are rows of the matrix.

3.7 Mechanical properties testing

This section describes the mechanical tests used to obtain the mechanical testing results for the heat-treated steels. It is necessary to understand the mechanisms behind the tests in order to understand their limitations and sources of errors. The section also explains why certain test types are incompatible and why certain customers require different test types to others. Information in this section is the results of visiting the test houses at BSES, but also from background reading from a number of texts[56,53,57,58].

3.7.1 Introduction to mechanical properties and their testing

Mechanical properties can be divided into static and dynamic properties. A static property is independent of the loading rate at which a force may be applied to test piece, however a dynamic property is dependent on this. Strength and hardness are examples of static properties, whereas fatigue, creep and impact resistance are examples of dynamic properties.

Before commencing the explanation of the tests it is worth first stating some definitions to explain the meaning of the properties tested for. Although these definitions are basic knowledge, it is surprising how often they are misused. Of the static properties:

- Strength is the ability to resist a force without breaking; three kinds of loading which might test a materials strength are tensile, compressive and shear.
- Elasticity is a material's ability to return to its original shape after being deformed.

- Plasticity is the readiness to deform to a stretched state when a load is applied, where plastic deformation is permanent even after the load is removed.
- Ductility is the ability to be drawn out longitudinally (therefore it must have plasticity).
- Hardness is the resistance to wear or indentation.
- Malleability is the ability of a material to be stretched in all directions.

Of the dynamic properties:

- Creep is a slow plastic deformation under a prolonged load.
- Fatigue is where a failure can result in a material where a load is applied repeatedly (whereas the material might be able to withstand that load if it was applied a lower number of times).
- Toughness is the ability of a material to withstand sudden loading.
- Brittleness implies lack of ductility or toughness.

Not all of the above properties are measured directly in the test house at BSES, however all of these properties are important when designing engineering steels and therefore from the tests made these properties must be inferred. Some tests therefore relate directly to one property and provide an indirect indication of another property at the same time. An example of this has been seen already in the way that Aurora forgings used a hardness test to provide an indication of Ultimate Tensile Strength.

3.7.2 Tensile testing

Tensile testing is one of the most important forms of mechanical property testing, it results in the determination of a number of values, namely the Ultimate

Tensile Stength (UTS) the Proof Stress (PS), the Yield Stress of the material (YS), and the Elongation and Reduction of Area of the specimen. Tensile testing has to be performed on a designated test piece, this is commonly determined by BS 18:1987 'Tensile testing of metals'. The test piece is stipulated to have the form shown in Figure 3.13, with a uniform central gauge length and shape, which will both affect the results obtained. The Figure shows d the diameter if the test piece, l_c the gauge length, l_o the parallel length and r the radius at the shoulder.

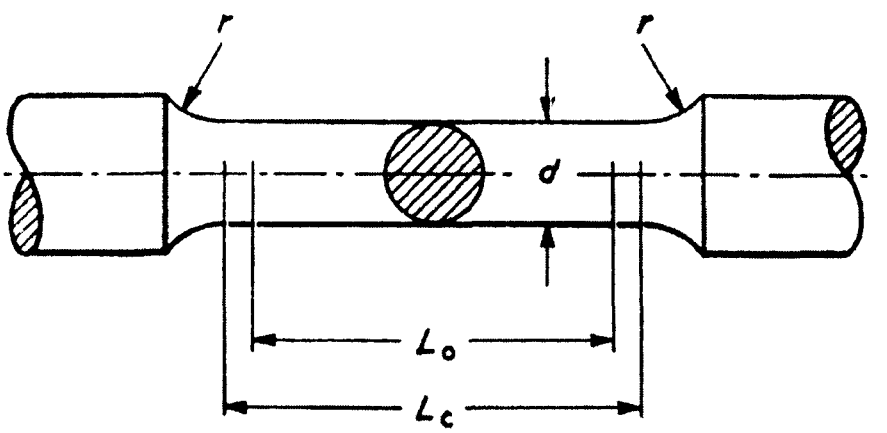


Fig. 3.13 British Standard tensile test piece with circular cross section

The British Standards sets out a range of proportions that can be used. One important aspect of the test piece is that if there is a non uniformity in its construction then premature failure can result, this is easy to spot for dramatic defects, however may be less obvious for slight detects which may cause inaccuracies in the results. The testing machines used at BSES incorporate a hydraulic mechanism to apply an increasing tensile force along the length of the specimen, which is clamped at either end by a gripping arrangement. Modern equipment uses load cells to measure the strain and extensometers to measure the extension of the specimen. The actual features measured are load and extension, however this can be readily translated into stress and strain, where stress is a nominal value of unit stress, which is the load

divided by the original cross-section area of the parallel proportion of the test piece. In a similar manner, strain is the extension divided by the original gauge length. An example of a tensile stress-strain graph, such as might be generated from a low alloy steel is shown in Figure 3.14.

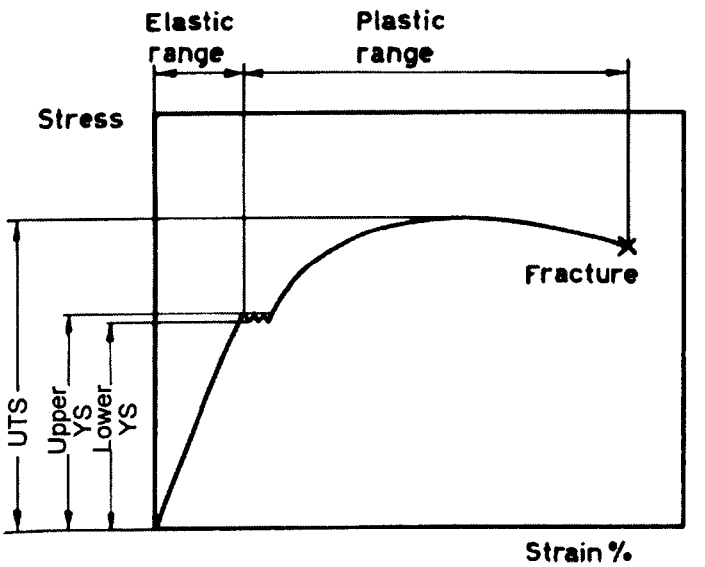


Fig. 3.14 Stress-strain curve as exhibited by a low alloy material

The load is increased until the yield point is reached, until which point elastic deformation is occurring. At the point where plastic deformation occurs, the specimen undergoes internal structural changes, with many dislocations being formed at stress points which travel through the material as observable lines known as Luders bands. At the upper yield point a relatively large extension of the test bar takes place with a constant load value, and thus the stress reading will fall to what is the lower yield point. This may appear as an oscillatory point as the load is increased and the yield proceeds. Depending upon the specifications used the lower or upper yield point may be considered. Not all materials exhibit a clear yield point, and indeed at BSES this is the case with many steels containing high alloy contents. Figure 3.15 shows the Stress-strain curve for such steel.

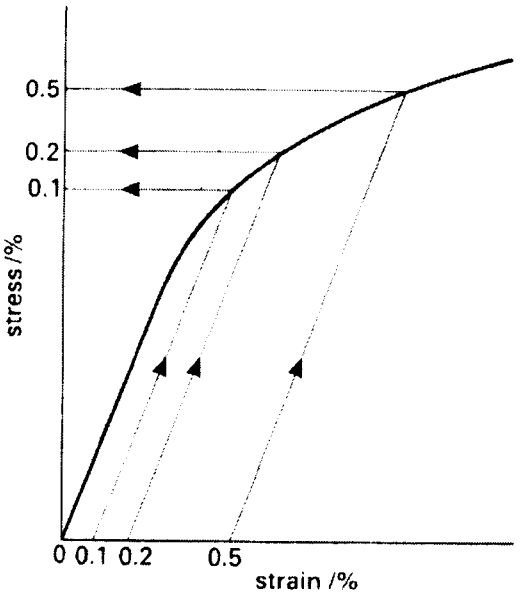


Fig. 3.15 Stress-strain curve for a high alloy steel or material without a clear yield point.

The proof stess is often used instead of yield stress in situations where there is no clear yield point. It important to consider therefore that composition may be highly correlated with the type test used here. The proof stress may be determined by applying a load to the specimen such that when it is removed a permanet elongation of say 0.2% strain remains, this is termed the 0.2% Proof stress. Fenner[57] describes a variety of methods in which this can be done. Another technique, however is to draw a line parallel to the elastic slope (fig. 3.15), through the point corresponding to 0.1% strain to intersect with the stress strain curve. This would give the 0.1% Proof Stress. 0.2 and 0.5% Proof Stress values are also reported by BSES where necessary.

After the yield point (Fig. 3.14), if present, the load will continue to increase again, and the rate of strain is commonly increased since work hardening is now taking place and extensions of up to 20% may be reached. In either case, the point of maximal stress will be reached; this is the ultimate tensile strength of the material. Beyond this point the material will narrow or 'neck' and the nominal stress falls. Eventually fracture will occur at the end of the plastic range. Fractures may be

classed as either ductile or brittle. Brittle fractures occur by rapid propagation of a crack through the material after little plastic deformation. A ductile fracture would occur after much more plastic deformation and exhibit a 'cup and cone' effect. The rate of loading can affect the type of fracture that occurs, if the loading is too fast then a brittle fracture may occur. It is usual for a metallurgist to examine the fracture under a microscope if the results are unusual, since this may enable the determination of a fault in the steel's composition.

The ultimate, yield and proof stress measurements considered so far are concerned with the strength of the material, however the elongation and reduction of area also provide a guide to the ductility of the steel. The elongation and reduction of area are measured as percentage changes in the gauge length or diameter of the specimen after fracture, and are dependent on the specimens used. For this reason two elongation results relating to different gauge lengths cannot be compared. It is common practice to mark the specimen with a sharp punch along its length prior to the tensile test, such that after fracture the pieces can be re-joined for the elongated length and final diameter to be recorded. This procedure carries obvious inaccuracies of measurement and alignment.

3.7.3 Impact testing

Impact testing produces some notoriously complex results due to the multitude of standards that exist and is argued by some to be imprecise due to the variety of results that can occur due to slightly less than perfect test conditions. It is also important to note that the impact tests described here are used as an indicator of toughness and are used by BSES for quality control rather than the indication of fracture toughness during a particular application. An example of this is in the cases of pressure vessels and similar safety critical applications; here it is usual to perform

an impact test on a similar section thickness as would be used in the real application. The fracture in this case will often propagate from an initial fatigue crack, which is generated artificially prior to the main impact test. This type of test is primarily to investigate the severity a crack might have on a material in its application. The types of tests described below do still vary based on the application, in the sense that the impact toughness may be investigated at sub zero temperatures. However, most test variations are due to customer preferences of the test type used, which is commonly dependent on the country from which the order stems.

The principle of impact testing is to measure the energy necessary to fracture a standard notched bar specimen, by an impulse load. There are two commonly used types of test, and both are used by BSES. In both test types a striker swings in an arc striking an appropriately mounted specimen. Figure 3.16 shows the arrangement

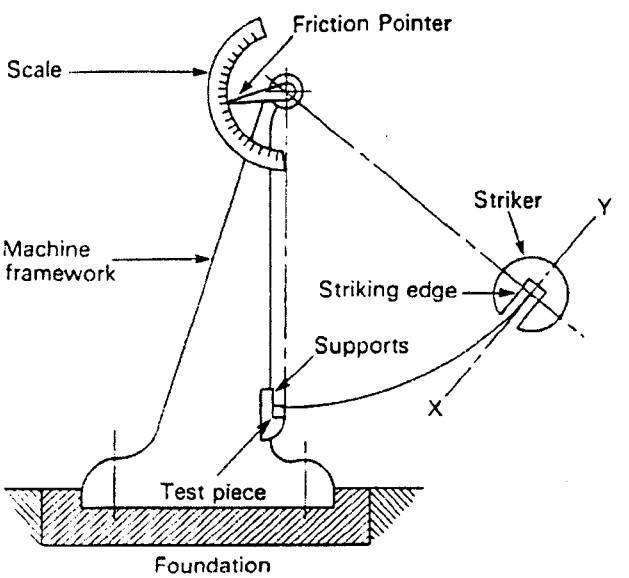


Fig. 3.16 Diagram of standard impact testing arrangement

used to perform the test.

The energy absorbed by the specimen is measured by the angle of displacement of the pendulum after the fracture. The striker angle, shape, test piece geometry and rate of loading effect the results obtained, therefore standardised

equipment and specimen sizes are used. The Charpy specimens tested at BSES exist with either V or U shaped notches, and may have a depth of 2, 3, or 5 mm. The Izod specimens have a V shaped notch, with a fixed depth, and are often favoured by Germany and France as a standard test type.

The ways in which the two specimens are supported are quite different, Figure 3.17 shows the cantilever arrangement of the Izod specimen and the 3-point beam arrangement of the Charpy test.

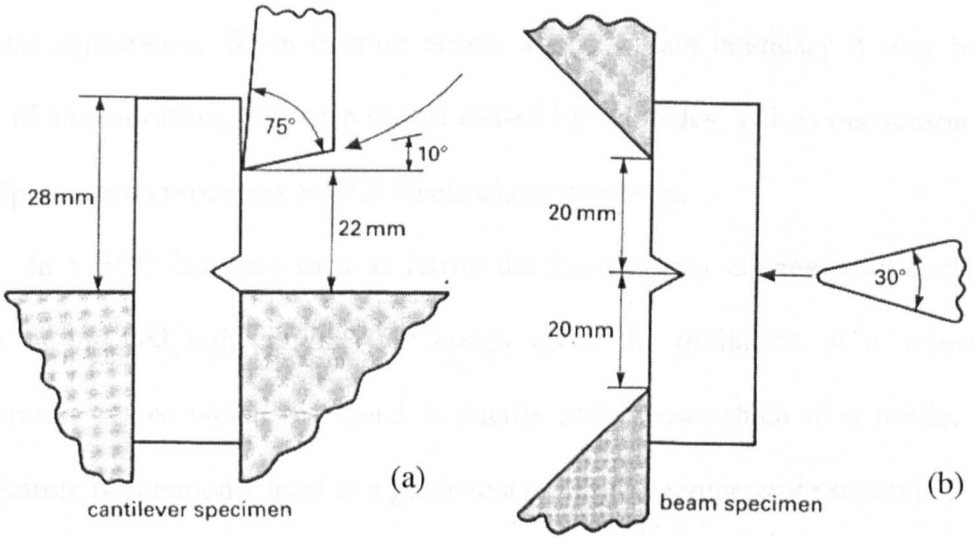


Fig. 3.17 (a) The Izod and (b) Charpy V test specimen positions.

The energy required to fracture the specimen is measured in Joules for both methods, however, a mechanical interpretation of Figure 3.17 would show that the Izod and Charpy test types are not compatible. This is important from a modelling perspective since data containing both test types would need to be separated as there is no conversion from one test type to the other. Moreover, from discussions with the test house manager at BSES and from consulting Thelning[53] it is apparent that Charpy test results of different sizes and notch shapes may not be compared either. This is due to the differences in fracture profiles that the shape changes cause and the behaviour of the fractures at low temperatures. At reduced temperatures the impact

strength also reduces. With a U notched sample this reduction appears to be quite continuous with falling temperature, however with a V notched sample the values tend to undergo a transition between high values on an upper shelf and then a steep incline to the low values at the low temperature.

Two types of fracture can occur, brittle and ductile fracture. The brittle fracture occurs after propagation of a crack after little or no plastic deformation. The brittle fracture can occur along grain boundaries or preferred cleavage planes. Because of the difference in the orientation of the planes, the brittle fracture will have a shiny granular appearance. When fracture occurs along a grain boundary it may be the result of an embrittling film such as that caused by sulphides. This is one reason why de-sulphurisation processes exist in steelmaking processes.

In a BCC structure such as ferrite the fracture may change from ductile to brittle at reduced temperature, this brings about the definition of a 'transition' temperature above which the metal is ductile and below which it is brittle. This temperature is commonly used as a guide to a steel's low temperature suitability.

A further complication of the charpy impact test is that there are two methods

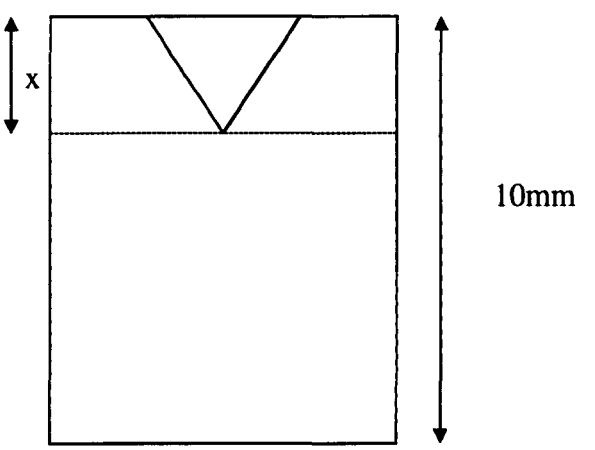


Fig 3.18 Notch depth in Charpy sample.

in which the impact energy can be calculated. The first method is termed the 'absolute' calculation and the second is termed the 'backed off' calculation. Figure 3.18 shows a Charpy test piece with a 10mm x 10mm cross sectional area. The

impact energy for the absolute method would take into account the fracture across the whole of the 10mm section size. The backed off value, however, takes into account the fracture energy across the length at the center of the specimen, which results in a higher impact energy due to fracture occurring over a shorter distance. Results at BSES are recorded in terms of both backed off and absolute values, however unlike the test types it is possible to convert from one to the other using the relation:

$$Absolute\ value = Backed\ Value * (10\text{-}x) \tag{3.1}$$

3.7.4 Hardness testing

A variety of hardness tests exist, these include the Vickers, Rockwell, Knoop and Brinell Tests. All tests are measuring the materials resistance to indentation in one form or another. Often the test type used is specific to the type of material being tested. The hardness data generated in the MET database is that from the Brinell test. The Brinell hardness test uses a hardened steel or tungsten carbide ball to make an indentation into a flat test specimen. The ball usually has a diameter of 10mm and a 3000Kg force is used to make the indentation. The specimen must be at least 8 times as thick as the indentation made. If multiple tests are performed on the same specimen they must be 40 mm apart (for reasons of work hardening). The diameter of the indentation made is measured using a microscope. The following relation is then used to calculate the Brinell hardness, where F is the force used, D is the diameter of the ball and d is the diameter of the indentation.

$$HB = \frac{2F}{\pi D[D - \sqrt{(D^2 - d^2)}]} \tag{3.2}$$

A rough estimate of the tensile strength of the sample can be found by multiplying the Brinell hardness by 3.

3.8 Accuracy of variables contained in the database.

Having established which variables were collected in the MET database, another important task was to establish how accurate they were. This was done through consulting plant engineers and the test house manager. This enabled the following table of measurement tolerances to be devised.

Variable	Tolerance
Composition	$\pm 0.004*(wt)^{-0.5}$
Hardening Temperature	$\pm 10^{o}C$
Tempering Temperature	$\pm 5^{o}C$
Tensile Strength	$\pm 1\%$
Size	$\pm 1mm$

Table 3.3. Accuracy of measured variables in the MET database

At this initial stage of the project it was also necessary to establish the degree of accuracy a predictive model would have to achieve in order to be useful in its application. The accuracy of the UTS model was primarily considered and was to be defined by a standard deviation of model error of $30N/mm^2$.

3.9 Chapter conclusion

In this section the main metallurgical and practical issues of commercial heat treatment processes have been examined. The processes connected with the project have been discussed and a suitable initial data source selected. A number of problems to do with incompatibility in the variables have been examined by exploring the test methods. These had to be overcome in the following modelling stages. Finally, the accuracy of the variables in the MET database at British Steel

have been examined, this being important in establishing the accuracy to which the predicted values may be obtained and is discussed later in the project. The next chapter deals with the initial collection of the data in spreadsheet format and its initial analysis and modelling. The data were collected for a constrained group of steels in order to allow initial experimentation and rapid data familiarisation. It was also apparent that the data may not always be fully accurate and indeed contained some faulty points. The following chapter considers data handling, investigation, initial modelling and 'cleaning' of these faulty points.

Chapter 4

Familiarisation and initial modelling of heat treatment data

4.1 Introduction

This chapter details the development of a suitable data set from the industrial process data of BSES. Initially a small data set pertaining to a single grade of steel was evaluated for consistency and data familiarisation, however this was later extended to a much larger set covering a wide range of steel types. The modelling abilities of linear and polynomial techniques were evaluated at various stages of the data sets development. As the data grew, an increasing amount of incompatibilities within the data set were seen. A variety of methods were examined to tackle this problem, in order to increase the accuracy and usability of the predictive model. Finally, having noted spurious points in the data set, a structured methodology to 'clean' the data set, with the aim of further improving its representation of the process is presented. Much of the work in this chapter, although based on the prediction of UTS, is aiming to develop a large, accurate set of data which may be applied to any of the process' outputs.

4.2 An introductory data set

Having established the structure of the process data and mechanical test results as seen in chapter3, the next stage was to select a data set from the MET Database. There is a very wide variety of steels available in the MET database, and so it was important to agree upon which types should be included in the data set. At initial project meetings with BSES representatives, it was suggested that in the long-term, it may not be possible to construct a single model covering a wide range of

steels, and that the data may need to be 'decomposed' into sub sets relating to key grade types. The metallurgists felt that the decomposition may be required to overcome the problem that, as steel grades change, so do the underlying mechanisms which produce the mechanical properties in a steel. The additional complications of varying sites, quenches and test types were also highlighted at this stage, and so it was agreed that an 'introductory' data set, which would largely avoid these complications, would be beneficial to aid data and process familiarisation. A data set relating to a grade of steel classed as 3%CrMo was therefore initially selected. The process experts felt that the 89 batches of heat treatment and test data relating to this grade should show relatively 'straightforward' trends typical of the process. The term 3%CrMo is used to illustrate that the steels in this group typically have a 3%Cr addition, together with an addition of molybdenum.

4.2.1 Simplifications of the 3%CrMo data set

The complications of multiple sites, quench types and treatments are reduced in the 3%CrMo data set since it only contains data from the Lab or West Bank works of British Steel. The hardening quenches, believed to be more significant to the mechanical properties than the tempering quench, are also all of the oil type. A further simplification is that the data set relates only to one hardening and then one tempering treatment. This also eliminates the problem of dealing with multiple tempering treatments. Initially, only the UTS output was considered, since it was felt that this would have a more predictable behaviour.

All variables in the 3% CrMo data-set were checked for missing values and also to make sure the range of the variables were within the max and min limits expected by the experts. Later in this chapter it will be seen that this form of basic validity check can be expanded to locate a variety of faulty data points.

4.2.2 Initial investigations into the 3%CrMo data set

The first investigation was to look at the basic statistics of the data. This could only be performed on continuous variables which exhibited some variance in the data set. Additionally, it was recognised that the 'forge size' variable (see Table 3.1) should be used instead of size, since it related to the actual size of bar heat treated in both lab and works based situations. The Max, Min, Mean and Standard Deviation values were found for each variable. The mean and standard deviation were calculated as:

$$\mu = \frac{\sum x}{n} \quad (4.1) \qquad \text{and,} \qquad \sigma = \frac{\sqrt{n\sum x^2 - \left(\sum x\right)^2}}{n(n-1)} \quad (4.2)$$

Variable	Max	Min	Mean	SD
Depth	15	12.5	12.691	0.516
Forged Size	280	30	118.267	64.882
C	0.315	0.235	0.253	0.017
Si	0.29	0.18	0.239	0.032
Mn	0.66	0.5	0.554	0.028
P	0.023	0.006	0.011	0.005
S	0.034	0.002	0.013	0.009
Cr	3.46	3.04	3.229	0.107
Mo	0.61	0.51	0.538	0.017
Ni	0.3	0.12	0.206	0.055
Cu	0.19	0.09	0.142	0.030
N	0.013	0.008	0.010	0.001
Nb	0.004	0.001	0.002	0.001
Ti	0.002	0.001	0.001	0.000
Al	0.032	0.016	0.024	0.005
V	0.01	0.006	0.008	0.001
W	0.03	0.01	0.015	0.007
Temp1	910	895	900.170	1.599
Temp2	700	570	601.591	17.079
UTS	1295	716	994.538	117.015

Table 4.1 Basic statistics of 3% CrMo data set *(n=89)*

Because the data set relates only to one grade of steel, there is very little variation in some variables. However, simply looking at the statistics of the data did not explain whether the variance in a given variable was significant; it may have been due to the

tolerances involved with the measurement and control of the elements in the cast instead of relating to an intentional addition. Moreover, it was realised that certain elements are more 'potent' and 'better controlled' than others, but the effect that these tolerances have on the output is not yet known.

Having discussed Table 4.1 with a metallurgist it was apparent that C, Si, Mn, Cr and Mo are the only compositional variables which represent intentional additions in the 3%CrMo steel. The variation in the other variables is due to variation in the scrap metal put in to the electric arc furnace to make a particular cast, and is not 'controllable' within the ranges found in the data via purification processes. It is possible that these 'residual' variables may not have much effect on the mechanical properties at the levels they are present in the data set, indeed within this narrow composition range even the variables which do represent an addition may have little effect. It is important to note that in a wider range of steel grades, intentional additions of what are in this case termed the residual elements may be made. It can also be seen from the table that there was little variation in the test depth and the austenitising temperature (Temp1), however the variations in site and tempering temperature (Temp2) were significant.

Part of the familiarisation process was to identify variables that were important in the prediction of UTS. One method of doing this was to look at the correlation of input variables with the UTS test results. When looking at the correlations it is important to remember that the data are from both the lab and the plant based sites and also that, whilst the most important quench, the hardening quench, is of only one type, the tempering quenches consist of water or air types. From examining the data set, however, it was apparent that the second quench type strongly relates to the treatment location, i.e. in this case the lab treatments use an air cool after the tempering stage, whereas the works treatment mainly use a water cool.

Realising that this may cause an amount of unexplained variance in the data when these variables are not taken into account, together with the problem that there may be non-linear or interaction effects in the data, the correlation of the relevant input variables with the UTS were examined. The following formula was used to calculate the correlation coefficient:

$$\rho_{X,Y} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_j - \mu_x)(y_j - \mu_y)}{\sigma_X . \sigma_Y} \quad \text{where } -1 \leq \rho_{xy} \leq 1 \tag{4.3}$$

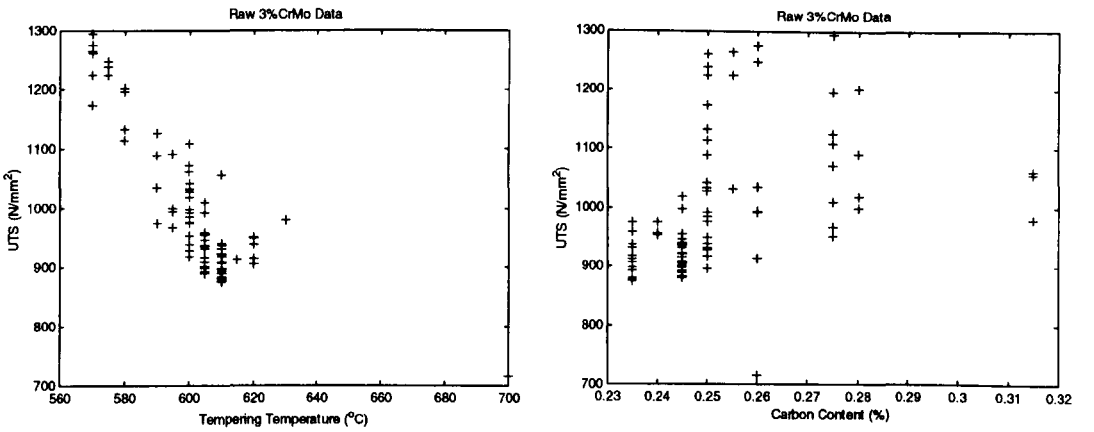
This yielded the following correlation values with UTS.

Variable	ρ_{UTS}
Temp2	-0.828
Forged Size	-0.642
C	0.414
Ni	-0.376
P	-0.352
Cu	-0.337
V	-0.259
W	-0.250
Nb	0.184
Cr	-0.176
Si	-0.157
Al	-0.156
Depth	0.137
N	0.136
Temp1	-0.090
Ti	0.067
Mn	0.053
S	-0.045
Mo	0.044

Table 4.2 Continuous variable correlation with UTS for 3%CrMo data

The variables in Table 4.2 have been arranged in order of decreasing correlation with UTS, therefore it is apparent from the use of this statistic that the tempering temperature has most effect on the UTS, followed by the forged size and then carbon. It was noted that the effect of tempering temperature and size were negative in relation to the UTS whereas the carbon variable had a positive correlation. This

confirms knowledge presented in chapter 3, where it was seen that at higher temperatures, more excitation is provided to the atoms in the steel and more martensite is transformed into softer structures, therefore the UTS of the material reduces. The size has a negative effect because as the section size gets bigger, the volume of steel that is hardened at a fixed depth increases. Hardenability is a term used to describe the ability of a steel to be hardened to a certain depth, however, this is dependent on composition, and is therefore basically fixed in this case. However even with a fixed hardenability, as the section size increases, the volume of steel that needs to be hardened at fairly constant test depth (Table 3.1) also increases. The thermodynamic effects this causes in relation to the heating and quenching of the steel during the hardening stage means that as the steel bar gets larger, at a near constant test depth it also gets softer. A metallurgist examining the coefficients did, however, comment that the effect of section size on UTS for the 3%CrMo grade, should not perhaps be as great as illustrated here, since it was a high hardenability steel and therefore the section size should have less effect. The interstitial strengthening mechanism of carbon is also confirmed by its positive correlation with UTS. As the correlation of each variable continues to reduce, so theoretically does the usefulness it has in the predictive model, since it also carries a noise penalty.



(a) (b)

Fig. 4.1 3%CrMo data scatter plots of (a) tempering temperature vs. UTS and (b) carbon content vs.UTS

Figure 4.1 shows the graphical difference in the relative correlation of tempering temperature and carbon with UTS, using an XY scatter plot of the raw data. Whist the correlation of carbon is still quite high at 0.4, its effect on the UTS is much less apparent when viewed as raw data.One problem in constructing a model is deciding upon the number of variables to include as inputs. For example the effect of vanadium on UTS is known to be positive, but very non-linear in that it doesn't have much effect until a concentration of 0.1%. In this example where there was no vanadium addition a negative correlation resulted, this may well be due to noise in the data at the low levels seen in the data set. Methods of determining the optimum variables to include in a model will be discussed later in this chapter. However, for this constrained data set it will be seen that the UTS could be predicted with reasonable success for this group of steels, using a least squares linear model with tempering temperature forged size and carbon content as inputs and UTS as the output.

4.2.3 The linear modelling program

A program to perform the least squares estimation, as described in chapter 2, was written in the MATLAB programming environment. This was to enable frequent models to be built for evaluation purposes. The program is capable of loading a randomly ordered data set in matrix form, with a single output variable as the last column of the matrix. The data were then separated into a modelling data set, which was used to find the least squares model coefficients, and a testing set, used to investigate the ability of the coefficients to predict new values. The testing set chosen was one third of the total data set available. Throughout the work in thesis, histograms have been used to confirm that the data distribution of the test sets is representative of the training sets.

Plots could then be generated of the measured and predicted values for each example in the data set, the measured vs. predicted values and the error values of the coefficients for the model data and the test data.

The data to be modelled were normalised with zero mean and unit standard deviation throughout the project unless otherwise stated. The purpose of this was that the input variables to the model for this type of data have very different ranges, and only by normalising the data can the model coefficients be investigated for significance.

4.2.4. Linear modelling of the 3%CrMo data

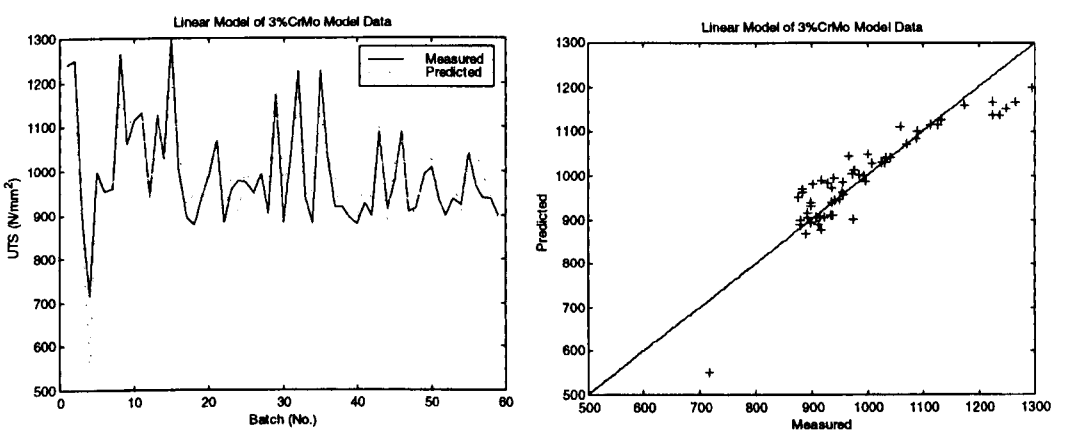
The program was used to model the 3%CrMo data with the three inputs of tempering temperature, size and carbon content. The following normalised variable coefficients were generated:

Variable	Coefficient
Carbon	0.413
Tempering Temperature	-2.560
Forged Size	-0.062

Table 4.3. Normalised linear model coefficients for 3% CrMo data

The coefficents in Table 4.3 essentially represent the significance of the various inputs to the linear model for the prediction of UTS. It is apparent from these coefficients that, although the previous correlation coefficients showed forged size to be more important than carbon content, the model coefficients now show the converse is true. This may be due to the fact that the univariate correlation analysis did not detect the significance of carbon when it is acting with tempering temperature.

Figure 4.2 shows the measured and predicted and measured vs. predicted graphs for the model and test data sets.



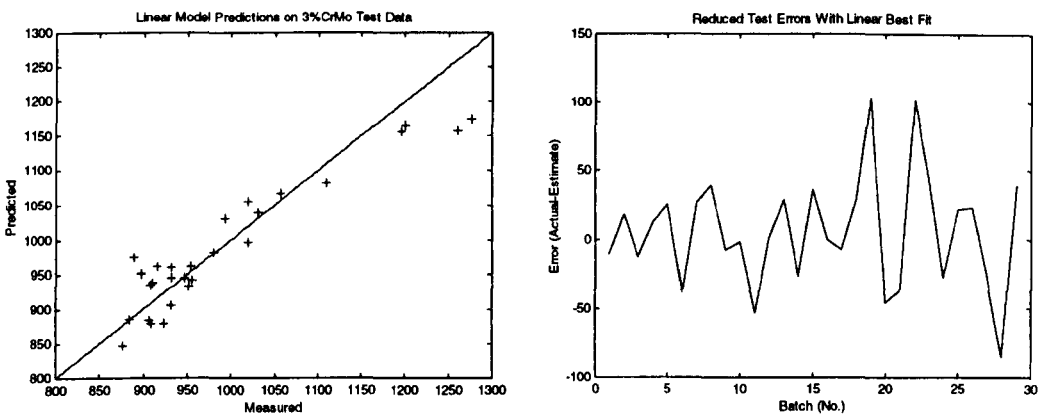
(a) (b)

Fig. 4.2 (a) Measured & Predicted and (b) Measured vs. Predicted Model set estimates of 3% CrMo Data

These graphs show the same information in different ways, however it was decided that since the number of data points would be increasing, the measured vs. predicted plot would give a clearer view of the fit of the data over a range of output values. On a smaller data set the measured and predicted plot can be helpful in visualising which

batch of data a particular result relates to. With the measured vs. predicted plot, a perfect model fit to the measured data would mean that all points lie along the 45° 1:1 line. Both plots show a reasonable fit to the original data, however one data point which was also present in Figure 4.1 seems to be of a much lower tensile strength than all of the others and is creating a large error between the measured and predicted value.

When the model coefficients were used to predict the UTS for new 3%CrMo test data the results shown in Figure 4.3 were obtained.



(a) (b)

Fig 4.3 (a) Measured vs. predicted plot of test set predictions and (b) absolute errors on predictions of 3%CrMo test set.

It can be seen from Figure 4.3(a) that the predictions were generally good over the range of UTS results measured. The error or residual values of the predictions can be calculated as:

$$error = (measured - predicted) \tag{4.4}$$

The errors per batch of data are shown in Figure 4.3(b) which shows that the largest errors were about 100N/mm^2 and because this error was positive, it means that the model was predicting a value that was too low in these cases. Initially, in order to provide an idea of the significance of a given error, it was felt that the largest

percentage error would be a reasonable estimate of model performance. The percentage error is given by:

$$percentage\_error = \frac{error}{measured\_value} \tag{4.5}$$

This was an initial idea since, if a predictive model were to be used in reality one would want to know the worst possible prediction performance it would provide. It will be seen later that this measure was not used alone, and was eventually dropped in preference to more significant measures. Using this measure, however, the worst case percentage accuracy for the model data was 77.1% and for the test set was 91.8%, this is mainly due to the outlying data point present in the data set. This will be discussed further in chapter 5. This accuracy was felt to be reasonable, especially considering that the data, although constrained to only one grade of steel, may have had some variation present due to the Lab or West Bank site it stemmed from. Indeed, by ordering the data by treatment site the variation between the Lab and the Works treatments may be seen (Figure.4.4).

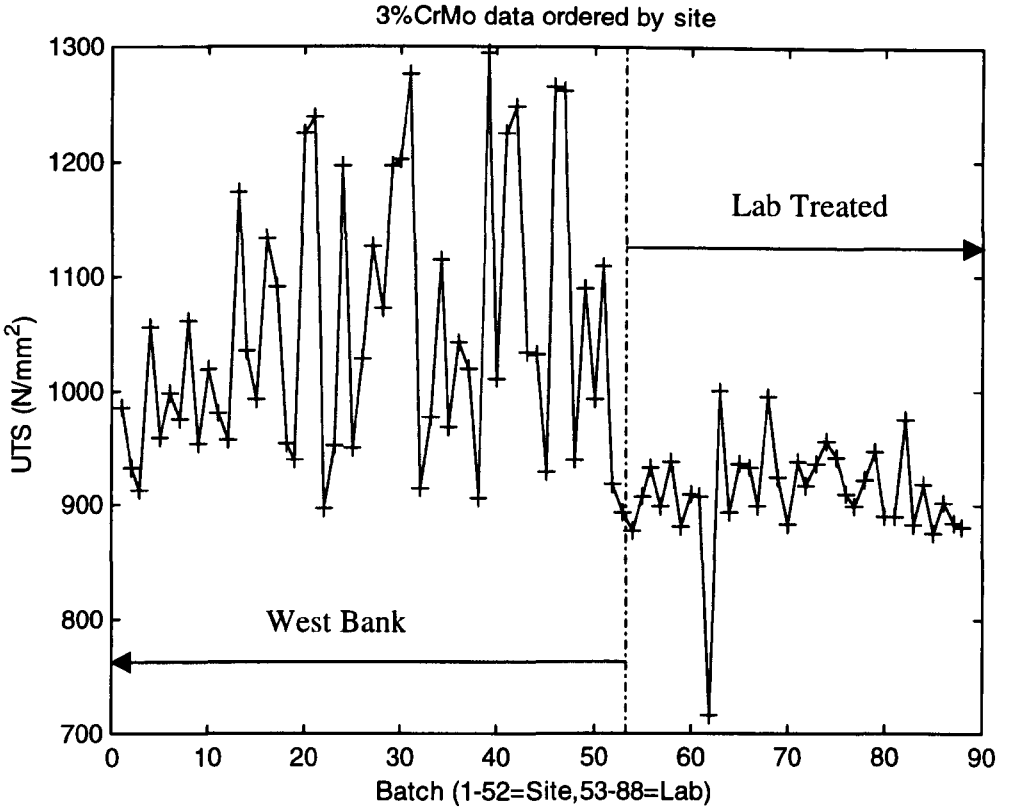


Fig 4.4 UTS values of 3%CrMo data ordered by treatment site.

For this constrained data set the difference between the two sites was very apparent, however as the variabilities of treatment type and composition increase as the data numbers increase it will be seen that this is not the case for a wider range of data. This graph shows that generally the lab data contains lower UTS values than that of the West Bank works treated data. This does not necessarily show a 'systematic' difference in treatment, since a systematic difference would only be present if the same treatments had been performed on the same steels at both sites producing different results. Process differences between the two locations are quenching type after the tempering stage and the section size of the specimens.

4.3 Expanding the data set

It has been seen that a model can be built using a linear technique with only three variables as inputs to the data. This grade of steel upon which this model was based was very tightly controlled and has reasonably straightforward underlying process mechanisms. In order to enable investigation into other process variables in hope of developing a more useful model, it was decided that a larger data set covering a wider range of steel grades, locations and treatment types, should be investigated now initial familiarisation had taken place. The new data set contained 2040 examples of heat treatments performed on a wide range of steel grades in 1996. This was confirmed by the fact that the data covered 112 different z-cards, where a z-card can be thought of as a specification of a steel type. The test house manager at BSES selected these steel types from over 6000 examples so that they represented frequently made grades but with significant variation in composition. The data stemmed from 6 different sites; West Bank, Roundwood, Whithams, Special Steels, Stocksbridge Lab, Pearsons.

The data did not contain variables relating to the test depth and gauge length of the test samples, this information was derived from the z-Card relating to each example, as this defines the test conditions. A macro in Microsoft Excel was used to make these additions.

This new data set also contained all of the test results in Table 3.2, however not all were present for all examples in the data set as the test types used are often steel grade dependant.

4.3.1 Variables in the 1996 data

The 1996 data set contains results from a variety of impact test types which, as was seen in chapter 3, cannot be compared. The increased number of sites was

expected to make a systematic difference, but may also only treat certain types of steel in certain ways.

Multiple tempering treatments were also present in the data set. This should have had an accumulative softening effect. The model needed to operate with a single tempering stage for the purposes of prediction and so examples with multiple treatments were not used unless they could be converted into a single tempering stage. Other treatments such as stress relieving and normalising were also present in the data set and would also not present relevant data to the harden and temper model.

Because we were dealing with a variety of steel compositions, the sample size and test depth should now have been important, as the hardenability of the steels may have varied depending on the composition. In particular there were now steels containing high aluminium and vanadium contents, which should have significant non-linear effects on the steel's mechanical properties.

An additional process variability is that the quenching media used may be either oil or water at the hardening stage and oil, water or air at the tempering stage.

The problem was now how to deal with all these variations when constructing a single model. It is important to utilise as many treatment examples as possible, however it was realised at this stage that if all the examples were 'thrown together' in a model based only on the continuous variables, then the model would not accurately represent the process. This is because it would not take into account the information relating to the sites or quench types used, moreover it would be very difficult to build a model which could take data relating to a differing number of treatment stages, without modifying the data. It was also realised that impact test results could not be converted and therefore this data should be kept separate in relation to the modelling of this output.

It was realised that the neural network techniques may be capable of forming 'sub models' for data sets that are inherently different. However, for reliable and meaningful model predictions, it was understood from the literature review in chapter 2 that it is often unwise to treat the neural network technique as a black box technique. For this reason various methods of overcoming the incompatibilities in data were investigated. The effectiveness of these techniques were investigated initially on linear and polynomial models as well as neural models. This was because, although the project title suggests the use of a neural network for the solution of this problem, a basic regression model may have been adequate if a suitable method for decomposing the problem existed.

Finally, another variable now present in the data set was a Pass or Fail variable, to indicate whether the treated steel in its final heat treated state passed or failed its specification requirements set out in the z-card. At first one would think that this could be taken as an indicator of examples that should not be included in the model since 'if the steel failed its tests then something must have gone wrong'. From discussions with the test house manager, however, it was apparent that with tight specification boundaries put on the mechanical test results, a steel could fail a test simply because it wasn't given quite the right treatment. This highlights the need for a reliable predictive model as, particularly when an uncommon grade of steel is being treated, a certain amount of 'educated guesswork' is clearly required and sometimes it is incorrect. Despite the steel failing its mechanical specifications, assuming nothing went wrong in the process, then that heat treatment example would represent a useful data point with which to train or test the model. The pass/fail statistic cannot therefore be viewed as one for locating faulty heat treatments.

4.3.2 Methods for decomposing the data

Initially it was decided that only data with a single tempering stage should be used as this would still give 1214 examples, whilst avoiding the problems of multiple tempering stages and other treatment types invalidating the data. The problem of the incompatibilities of the impact and elongation results would be ignored as this is only relevant for the development of the impact and elongation models, and would reduce the number of data points available for the initial investigation of the UTS model.

Having made this initial simplification, the problem of site and quench types was still present, therefore the following ideas for decomposing the data further were recognised:

- To find some way of categorising the microstructure of the steel after the hardening and tempering stage of the process, as this is may indirectly summarise the effects of site or quench type as well as being a function of composition and bar geometry. Although the type of microstructure would not allow one to calculate the mechanical properties of the steel, it is recognised as being an important factor. It may therefore be possible to use it as a method of decomposition to put 'similar' types of steel together. This seemed an unlikely method to use because unlike many rolling data sets, the microstructure of the heat-treated steels is not measured and recorded.

- To decompose the data according to the site and then for each site decompose the data further based on the quenches used at each stage. The results will then be many sub models dependant on the sites or quench types used. Because certain types of steel are treated in certain ways this may also be indirectly decomposing the data by steel grade.

4.3.2.1 Microstructural decomposition

The BS metallurgists were consulted to see if there would be any way of breaking the process into stages based on the variables included in the data set. It was discovered that the ratio of two of the mechanical test results, the Proof Stress and the Tensile Strength, could be used to estimate the microstructure of the steel after the hardening stage. It was apparent that, although in general the microstructure after the hardening stage would be martensitic, this was not always be the case, and for example, some bainite microstructures were also present. As was described in chapter 3 the bainite microstructures are very different to the martensite ones and are considered less controllable, they do however exist in some steel grades and treatments. If present in the 1996 data then their underlying transformation mechanisms would be very different to those of the martensitic steels and would be difficult to include in the same model.

The following criterion to locate martensitic steels within the data set was used:

$$\frac{PS}{UTS} > 0.8 \qquad (4.6)$$

On inspection of the data set two complications arose, the first being that while there were UTS measurements available for each example, there was not always a proof stress measurement available, so this further reduced the number of data points available. The second problem was that, as seen chapter 3, there are at least three methods of proof stress measurement available. It appeared that the most common method used was the 0.2% Proof Stress and therefore this was used to maximise the number of data points available.

It was found that 734 out of the 1214 complete, single tempering temperature examples had a PS/UTS ratio greater than 0.8, this reduction in data numbers was obviously also a function of the number of 0.2 % PS measurements available.

On investigating the statistics and correlation of output to continuous input variables, together with advice from metallurgists on significant variables in the data set, it was found that the addition elements of Mo, Cr, Si, Ni and Mn were all now expected to be important with respect to the mechanical properties developed. Vanadium also would be a very important element where additions of it were made, however this was not indicated by correlation analysis since there were only 3 vanadium additions present in the 1996 martensitic data set.

It was apparent that correlation analysis was now of limited worth in indicating variables which would be important in a model to predict the mechanical properties. One example is that the correlation of carbon content with UTS was very low, and it is widely accepted that, with carbon varying between 0.2-0.5 wt % as in this data set, a significant effect on the UTS would be seen. For this reason variables used for the construction of a model were heavily based on expert knowledge of the process. It was decided that experimental re-modelling with a variety of input variables could be used to determine the optimum variable configuration when a suitable method of

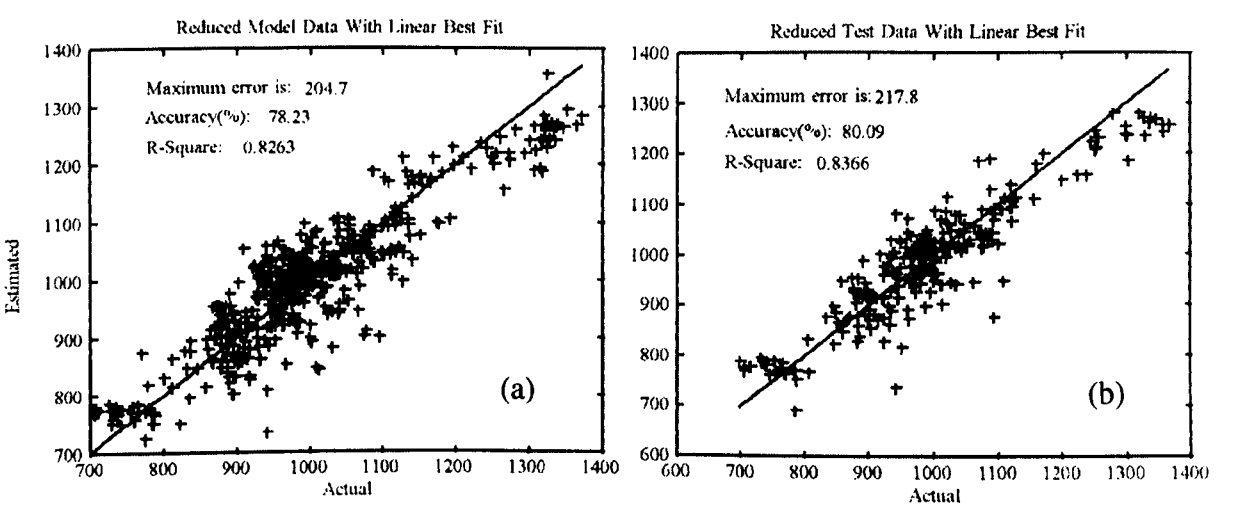


Fig 4.5 (a) Linear model data of martensitic subset (b) Linear test predictions of martensitic subset.

data composition became clear.

A linear model was therefore constructed on the 1996 martensitic data set to predict the UTS using Specimen Size, C, Si, Mn, Cr, Mo, Ni, & the tempering temperature as inputs. Note the absence of an austenitising temperature since this is only important in determining the microstructure properties at the hardening stage, together with the quench which cannot be included as a continuous variables.

As part of the Matlab modelling program written, the accuracy of the models could be printed on the measured vs. predicted graphs. It can be seen that the maximum error for the model set was 204.7 and for the test set was 217.8. This yielded 'worse case' percentage accuracies of 78.2 and 80.09 for the model and test sets respectively. Compared with modelling just the 3% CrMo data the model accuracy had improved and the test set accuracy had deteriorated. It was apparent that through the decomposition, the outlying data present in the 3%CrMo model data set was now not present in the martensitic model data. This was thought to explain the improvement in worst case accuracy of the martensitic data set. The deterioration in test set accuracy indicates that the linear model, which now covered a more complex (although still very refined) range of steels, had a limited ability to cope with the increase in process complexity.

Through discussions with representatives of the Technology Centre it was decided that an additional statistic would be included to more objectively evaluate the fit of the model. The problem is that the worst case accuracy was very sensitive to outlying results (it is determined by the worst case error). The new statistic used was therefore the r-Square statistic, which is the square of the Pearson product moment correlation coefficient calculated from the measured and predicted values. This can be interpreted as the proportion of the variance in y attributed to the variance in x and is given by,

$$r = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{\left[n\sum X^2 - (\sum X)^2\right]\left[n\sum Y^2 - (\sum Y)^2\right]}} \quad (4.7)$$

The r-square value varies between 1 for a perfect fit to 0 for no fit.

It can be seen from Figure 4.5 that with an r-Square value of ~ 0.82 for both the model and the test set that the linear model of the martensitic data provided a reasonable fit.

A polynomial model was then developed with the same input variables and martensitic data set as with the previous linear example. Identical points were placed in the test set for comparison purposes. This model took the multiple regression form of,

$$y_m = B_0 + B_1u_1 + B_2u_2 + \ldots + B_nu_n + B_{n+1}u_1^2 + B_{n+2}u_2^2 + \ldots + B_{2n}u_n^2 \quad (4.8)$$

Theoretically the polynomial form of the regression model allows greater modelling flexibility than the linear form, and so if there was a polynomial relationship in the data which could not be modelled by the linear model, then the modelling accuracy should improve. The results of this technique are shown in Figure 4.6.

The results show a slight improvement in the worst case accuracy and the r-

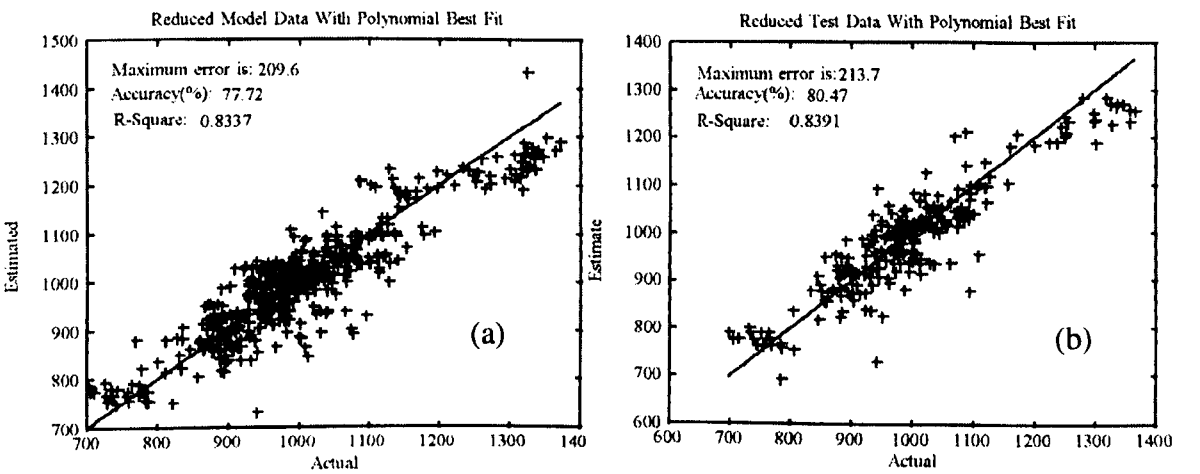


Fig. 4.6 (a) Polynomial model data of martensitic subset (b) Polynomial test predictions of martensitic subset.

squared values of the polynomial regression model, however this was not a very significant improvement given the increase in model complexity. The literature often recommends the use of parsimonious models (i.e. the simplest model, which gives the best performance) and in this case this would still favour the linear model.

4.3.2.2 Site and treatment based decomposition

The next method of decomposition was site and treatment based. With the martensitic decomposition method alone, only two treatment pattern stages could be considered, however with the site and treatment based method this was not the case. For example, if for a given site a three stage treatment is regularly made, then a model may be constructed taking into account the temperatures throughout that treatment. A tree diagram was constructed to visualise the number of treatment patterns that were undertaken for each site, however this diagram proved to be extensive. This was due to the fact that many treatment patterns were for 'one-off jobs'. It was decided therefore to concentrate on the main patterns in the martensitic data, with the aim of further increasing the modelling accuracy for each subset. This yielded the following groupings:

Site	Total number of points	Treat / Quench (1)	Treat / Quench (2)	Treat / Quench (3)	Treat / Quench (4)	Treat / Quench (5)	Subset size / Ref No.
West Bank	345	HR (Oil)	TM (Air)	-	-	-	21 (No1)
		HR (Oil)	TM (Water)	-	-	-	34 (No.2)
		HR (Oil)	TM (Air)	SR (Air)	HR (Oil)	TM (Air)	22 (No.3)
		HR (Oil)	TM (Water)	HR (Oil)	TM (Water)	-	30 (No.4)
		HR (Water)	TM (Water)	SR (Air)	-	-	54 (No.5)
Lab	162	HR (Oil)	TM (Air)	-	-	-	112 (No.6)
Special Steels	39	HR (Water)	TM (Air)	-	-	-	30 (No.7)
Whitha-ms	37	HR (Oil)	TM (Air)	-	-	-	26 (No.8)
Round-wood	151	HR (Oil)	TM (Air)	-	-	-	43 (No.9)
		HR (Oil)	TM (Water)	-	-	-	74 (No.10)

Table 4.4. 10 data sets relating to the site and treatment based decomposition of the martensitic subset.

Each data set was randomised and then separated into 2/3 model and 1/3 testing data. Once randomised, the same training and testing data were used for each modelling method. At this stage in the project it was decided that as well as the linear and polynomial methods, a neural network technique would also be used to model the data set.

To enable rapid experimentation, initially, a commercial package called 'NNmodel' that executed a MLP network with a selection of training algorithms was used. The package allows experimentation through a spreadsheet based data driven system, enabling inputs to the network to be readily varied along with the network parameters. It was felt that by using such a package, one could quickly ascertain whether the data available were of suitable quantity and quality for neural modelling, whilst providing a comparison to the traditional modelling techniques. It was

envisaged that once a suitable data set decomposition had been established, if the neural network technique provided the best performance, then a variety of networks architectures would be tried along with the possibility of programming such a system ones self.

For the purposes of the modelling of these data sets, a MLP network with one hidden layer containing 6 hidden layer neurons was used. Although it was recognised that the architecture of the network should be individually suited to the data set in question, at this stage this fixed semi-optimal configuration was useful for comparison purposes. In a similar manner, rather than using a validation set to 'guarantee' the prevention of over-fitting, experimentation resulted in 1000 training iterations being used for each neural model. The weights of the neural network were initialised to small random values before beginning training as described in chapter 2. This initialisation can make small differences in the weight values to which the network converges. For this reason, so as to provide a comparison between results, the weights of the network were always initialised using the same random seed, such that this variation was eliminated from all the experiments.

The Matlab programs written to perform the linear and polynomial regression were used as previously described and the following results in Table 4.5 were obtained. The inputs used for all modelling techniques were bar size, C, Si, Mn, Cr, Mo, Ni, V, and the tempering temperatures (as many as present in the treatment pattern). Vanadium has been included as an additional input here in order to account for the few vanadium steels present in this data set. Exclusion of these steels and/or the vanadium input parameter did however appear to have little effect.

Data Set	Linear				Polynomial				Neural			
	Model		Test		Model		Test		Model		Test	
	R^2	Acc %	R^2	Acc %	R^2	Acc %	R^2	Acc %	R^2	Acc %	R^2	Acc %
1	0.97	98.0	0.59	73.0	0.99	99.5	0.39	91.8	0.91	69.8	0.69	91.8
2	0.89	95.4	0.53	96.5	0.97	97.8	0.02	35.2	0.68	87.8	0.59	95.1
3	0.99	94.5	0.98	95.3	0.99	97.6	0.80	88.0	0.99	96.6	0.99	94.4
4	0.81	96.8	0.54	88.0	0.81	97.1	0.00	77.7	0.83	96.8	0.06	91.0
5	0.96	93.2	0.91	87.6	0.98	92.9	0.41	22.2	0.94	90.1	0.95	90.2
6	0.93	91.9	0.87	78.5	0.94	91.9	0.80	78.1	0.93	92.4	0.86	77.5
7	0.45	93.4	0.02	76.5	0.97	97.8	0.14	74.3	0.68	92.9	0.25	87.4
8	0.63	95.2	0.01	6.1	0.64	95.4	0.01	7.0	0.49	95.2	0.50	95.4
9	0.98	95.3	0.80	89.3	0.99	98.4	0.63	85.9	0.97	96.3	0.63	87.8
10	0.69	95.0	0.87	93.8	0.93	96.9	0.04	46.9	0.86	95.4	0.74	87.9

Table 4.5. Results of modelling the site and treatment decomposed martensitic subset with linear, polynomial and neural techniques.

The first thing that is evident from the results in Table 4.5 is that the worst case percentage accuracy does not always fall when the r-square statistic is low. This can occur if there is little spread in the predicted and measured values (i.e. little error) but the predicted and measured values are not correlated. The predicted values are therefore not related to the inputs very well, but are confined to a certain range.
In this sense the R-square statistic provides a much more realistic measure of the model's fit to the measured data.

The polynomial model has a very good model set performance, however it can be seen that the test set performance is quite spurious, with some very low R-square values. The polynomial modelling technique is notorious for providing such results as it can produce an over-parameterised model, which produces unreliable predictions.

Overall the neural models perform most favourably, however both the linear and neural model's performances are close and are better at some data sets than others. One data set in particular provides low R-square values for all modelling methods, being data set 7, Special Steel's hardening and tempering process with water and air quenches after each respective stage. This poor result may be due to the

fact that there is a greater variety of steels manufactured at this site, and yet a low number of data points from which the models train.

The most significant performance improvement provided by the neural model is that seen in data set 8. The test set accuracy of the linear and polynomial models is very low, however the neural model is significantly better at generalising to the test set. Closer investigation revealed that an outlying data point was present in the test set for this data set and that the neural network appeared to be less affected by it than the linear and polynomial methods.

The problem with decomposing the data in this way is that the number or data points in each subset can become very low. Linear regression methods may be capable of functioning on low data numbers, however one would typically aim to use data in the order of 1000's for training a neural network, particularly one which has 9 inputs as are used in this case.

4.3.2.3 Combining data from more than one site

Having decomposed the data set and observed the modelling performance that could be achieved, the next stage was to see whether by using the neural network technique the data set could be combined. The linear and polynomial techniques were still used to evaluate whether, as the data set was increased, the neural model was capable of forming 'sub models' due to it's additional degrees of modelling freedom.

Values for the martensitic harden and temper treatments with oil and air quenches respectively for all the sites were combined into a single data set. This resulted in a total of 202 data points. The test sets from each of the individual data sets (numbered 1,6,8 & 9) were kept as the test sets of the combined data set so that

any outlying points previously in the model or test sets would still be in the same area of the data partition.

Table 4.6 shows the statistics that this data set now covers. When compared with Table 4.1 it can be seen that the alloy content range, and therefore the grades of steel which the data covers is significantly increased from the original 3% CrMo data set. It is desirable to try to produce a single model that covers a wide range of alloy contents since if many, narrow range models are developed it would be difficult in application of the models to interpolate between grades when predicting the mechanical properties.

	Max	Min	Mean	SD
Bar Size	290	25	116.5	53.4
C	0.45	0.26	0.37	0.05
Si	0.35	0.18	0.27	0.04
Mn	0.93	0.36	0.62	0.12
Cr	3.13	0.63	1.08	0.45
Mo	0.57	0.16	0.37	0.16
Ni	2.71	0.11	1.54	0.89
V	0.27	0.002	0.012	0.04
Temp	695	520	591.6	36.5
UTS	1358	801	1046	122.6

Table 4.6 Statistic of the combined site HR(Oil) TM(Air) martensitic data.

The linear, polynomial and neural models were trained with the inputs and parameters previously described in the site and treatment based models. The results of the model and test sets are shown in Table 4.7.

	Linear		Polynomial		Neural	
	Model	Test	Model	Test	Model	Test
R^2	0.799	0.826	0.836	0.829	0.845	0.885
Acc %	84.9	86.2	83.6	85.5	87.5	87.8

Table 4.7 Results of modelling the combined site HR(Oil) TM(Air) martensitic data set.

It can be seen from these results that the performance of the neural network is better than the linear and polynomial techniques, however this improvement is not very marked. Moreover it can be seen that the combination of the data from the individual sites has in fact improved the model accuracy especially in the case of the neural network. It was postulated that it might be possible to include additional inputs to the neural network so as to help the data from each site to be offset by the network. The aim of this was to encourage a separate 'sub model' to be formed for each site whilst also allowing all of the data to contribute to a useful model offering effective interpolation.

An additional input was therefore added to the data set, which contained the entry 1, 2, 3, or 4 depending on which site the heat treatment examples came from.

The linear, polynomial and neural networks were then retrained with the same inputs and data sets as before, but with the additional input of site, the results being shown in Table 4.8.

	Linear		Polynomial		Neural	
	Model	Test	Model	Test	Model	Test
R^2	0.799	0.823	0.843	0.821	0.860	0.913
Acc %	85.2	86.4	83.0	83.7	85.6	87.3

Table 4.8 Results of modelling the combined site HR(Oil) TM(Air) martensitic data set with the addition of a single input variable to denote site.

The results show that there is little effect on the linear and polynomial model's performance, however there is a small improvement in the neural model's performance. Because there are still a low number of data points, it was decided to

request more data from BSES to see if a further improvement in the neural model's performance could be developed.

4.3.3 Time at tempering temperature

In an attempt to determine possible reasons other than bar size or quench type for systematic differences that may result between sites, it was considered that the tempering time between sites might be different. On investigation, although these data were not recorded as part of the data set available, it was confirmed that key sites hold the steel at the soak temperature of the tempering stage for different amounts of time. It should be noted that the operator at each site will generally work with a given duration of tempering time and will adjust the tempering temperatures to obtain the required mechanical properties. The model is combining data from various sites and so it was considered that it may benefit from a variable relating to the duration of the tempering temperature soak.

The soak time for each site where known, together with the site's code is shown in Table 4.9.

Site Code	Site	Tempering Time (Hr)
1	West Bank	3
2	Lab	1
3	Whithams	unknown
4	Roundwood	1.5

Table 4.9. Soak time and site code for HR(Oil) TM(Air) data.

Holloman and Jaffe[59] have made an investigation into the relation between tempering time and temperature. The work was undertaken in 1945, and concentrated initially on the principle of predicting the properties of a quenched steel. It transpires that this was done through working out the relationship between time and tempering temperature in order to obtain a given hardness. Although the hardness predictions were made on one base composition of steel which varied only in carbon content,

and would not be applicable to a range of alloy steels, the relation between tempering time and temperature may still be useful.

The hardness relation quoted in the paper can be written as:

$$Hardness = f[T(c + \log t)] \tag{4.9}$$

where T is the absolute tempering temperature, t is the soak duration in hours and c is a coefficient relating to the steel. The tempering parameter allows a linear approximation to be made so that the function f is linear.

The coefficient c may be determined via a variety of methods, the first of which is to use some past data relating hardness to tempering temperature and time. However this technique was not possible given the distribution of the data available, since it did not contain data for all grades over a range of tempering times, so two other options remain. The first remaining option is to use the 'rule of thumb' value of c, which is 19.5 for ordinary carbon and alloy steels (0.25-0.42 %C) and 15 for tool steels (0.90-1.20 %C). All our steels were of the first type.

The second remaining method of finding c was to use the following relation,

$$c = 21.3 - 5.8(\%C) \tag{4.10}$$

The authors claimed that the calculation of c from this method is more accurate than the rule of thumb, but less accurate than finding the coefficient experimentally. The authors also claimed that the relationship between time and temperature holds whether the initial microstructure is martensite (with or without retained austenite), banite, or pearlite.

From reading the paper, ideas relevant to the final application of the model were also developed. For example, one might be able to provide an option for the tempering temperature relating to a change in time so that if, for processing reasons, the material should not be tempered at a certain temperature then a comparative

treatment could still be made. In a similar manner it may be more economical to perform a treatment at an elevated tempering temperature, but for a reduced duration of time, so as to provide the same treatment, but at a reduced cost.

4.3.3.1 Experimentation with tempering parameter

From Table 4.9 it can be seen that the tempering time for the Whithams site was undetermined, therefore the examples in the martensitic harden and temper data with oil and air quench respectively which relate to this site were removed. This left 176 data points in the data set. It was decided that experimentation would be carried out on this small data set and, if an indication of model performance improvement was seen, then the technique could be applied to a larger data set when available.

Experimentation to determine the effect of adding the tempering parameter was carried out in a variety of configurations. In each configuration the data rows were kept in the same random order, and the modelling parameters were kept constant (initial conditions, hidden layers etc.). In all, five new data sets were configured and modelled with each technique. The contents of which are described in the points below.

- Data set 1 :

  Initially, the data set with the Whithams examples removed was remodelled so that a benchmark accuracy could be determined with the linear, polynomial and neural techniques.

- Data set 2:

  The tempering parameter was simply added to data set 1 as an additional variable. The fixed form of constant c was used, the value of which was 19.5, for ordinary carbon and alloy steels.

- Data set 3:

  Assuming that the temperature information was now described by the tempering parameter the tempering temperature was removed from data set 2.

- Data set 4:

  This data set is similar to that of set 3 except that c was determined from equation (4.10) dependent on the carbon content of each batch rather than as a fixed value.

- Data set 5

  Finally, the bench mark data set was used but with the addition of the tempering time as an extra variable (no tempering parameter was present).

The results of modelling these five data sets are indicated in Table 4.10.

Data Set	Linear				Polynomial				Neural			
	Model		Test		Model		Test		Model		Test	
	R^2	Acc %	R^2	Acc %	R^2	Acc %	R^2	Acc %	R^2	Acc %	R^2	Acc %
1	0.84	85.7	0.86	84.0	0.87	85.0	0.84	79.8	0.95	92.1	0.89	81.3
2	0.84	87.8	0.85	86.5	0.94	93.1	0.86	82.2	0.95	95.5	0.85	85.4
3	0.84	87.7	0.85	86.5	0.87	84.3	0.85	83.1	0.94	91.0	0.86	81.8
4	0.84	87.6	0.86	86.8	0.87	84.3	0.87	84.7	0.95	91.1	0.87	81.5
5	0.84	88.0	0.86	86.3	0.87	85.0	0.84	79.8	0.94	92.2	0.87	82.2

Table 4.10. Results of modelling the five data sets for the inclusion of a tempering parameter in the HR(Oil) TM(Air) martensitic data set.

As with the results in Table 4.8 the neural network technique produces the most accurate predictions. However, with regard to the tempering parameter there appears to be little improvement in the performance regardless of the configuration used, in fact with the neural network the addition of the tempering parameter causes a lower test set R^2 statistic than the benchmark set.

From examining the tempering parameter analytically, it can be seen that the actual variation in the variation of the tempering parameter for the same steel tempered between 1 and 3 hours is very small. For example at 968°K and using

c=19.5 in equation (4.9), the tempering parameter for a 1 hour soak is 18876, and for a 3 hour soak is 19338 yielding a difference of only 462.

It was, however, expected that the tempering parameter may be useful in the future if enough two stage harden and temper data were not available since it may be possible to use it to convert two tempering stages into one at an elevated temperature.

The conclusion of this experimentation was that the addition of the tempering parameter or indeed the tempering time did not improve the prediction accuracy of models, and therefore would not be used in this form any further.

At this stage the performance measures used were once more reviewed. It was decided that instead of using the worst case accuracy, the standard deviation of the residuals would provide a good estimation of model error, that would not be offset by outlying points as much as the worst case percentage error. The standard deviation assumes a normal distribution of the model errors.

4.3.4. The 1995 heat treatment data.

The 1995 data set was acquired from the MET database. This contained a total of 3971 batches, however this included a variety of treatment patterns as with the 1996 data. It was noted that the 1995 data contained fewer treatments at Roundwood and more treatments at West Bank. It was also noted that the maximum carbon levels were now higher with the 1995 data, together with extended manganese and nickel levels.

To ensure that the modelling results from each year's data were consistent it was decided to keep each year's data separable. This was achieved by allocating a code to each example in the data set for each year. The first two digits of the code pertained to the year from which the data came, for example 96 or 95. The following four digits related to the row number which the example for each year has once it has

been randomised in the data set, for example 951002 would be the 1002'nd example in the 1995 data set. In this way, tracking of each example was ensured from the reduced variable model set, to the fully described original spreadsheet containing useful data about the example which was not used in the model's construction (for example the heat treatment batch number).

Initially the 1995 data were modelled separately to see if a similar accuracy to that of the 1996 data could be obtained.

4.3.4.1 Extending the 1995 martensitic data set to cover multiple quench types

At first it was intended to select just the harden and temper data with oil and air quenches respectively, however it was realised that with only one other data set potentially available, this would result in a small data set. Moreover, because certain grades of steel are given particular quenches, restricting the model to one quench type would result in a narrow range model, for example only high alloy air hardenable grades of steel are hardened with an air quench after the hardening stage.

It had already been established that decomposition of data into different treatment patterns results in many models with low numbers of data points which are difficult to interpolate between, and so the aim was to find ways of linking data from the individual subsets of quench type into a single model.

In section 4.3.2.3 the addition of a variable to denote the site to which the data pertained had enabled an improved accuracy in the modelling of data from a combination of sites. In a similar way it was anticipated that codes could be allocated to the data set to denote quench type after the hardening and tempering stages.

Initially a single variable code was allocated for each quench type in the two stage process. The effects of adding this code in a variety of ways was first investigated on

harden and then temper data of the 1995 martensitic examples. When this subset was separated from the main 1995 data set 584 examples remained. This was a substantial number of examples and should also allow the effect of an increased number of data points on the neural network to be seen against the other modelling techniques.

The codes allocated for each quench type are shown in Table 4.11. Because the 1995 data set had not been previously modelled a benchmark data set was made. The following data set configurations were therefore formed:

- Data set 1:

  This data set contained no site or quench codings, and was intended as a benchmark. The standard model inputs shown in Table 4.6 were used.

- Data set 2:

  Here the site codes were added as described with the 1996 data, this helped to see if an improvement could be seen for the 1995 data as was observed with the 1996 data.

- Data set 3:

  This data set was as with data set 2, but with the addition of two variables to denote the quench at each stage of the process.

Quench Type	Quench Code
Oil	1
Water	2
Air	3

Table 4.11 Quench codes allocated for each type of quench.

The four data sets were modelled with each technique, with fixed data set order and training parameters in the case of the neural network. The results are shown in Table 4.12. Note the new use of the SD of model error instead of worst case percentage error as described in the previous section.

Data Set	Linear				Polynomial				Neural			
	Model		Test		Model		Test		Model		Test	
	R^2	SD of E	R^2	SD of E	R^2	SD of E	R^2	SD of E	R^2	SD of E	R^2	SD of E
1	0.71	74.6	0.81	59.5	0.72	71.0	0.83	55.1	0.73	72.1	0.70	79.4
2	0.72	73.9	0.82	56.9	0.75	70.5	0.84	55.4	0.76	68.4	0.84	54.6
3	0.72	73.2	0.82	55.7	0.77	66.8	0.84	56.0	0.86	51.9	0.85	52.1

Table 4.12. Results of experiment to determine the effect of adding a quench type code to the input of the model when combining all two stage martensitic data for the 1995 data set.

The results in Table 4.12 show several interesting points. The first is that without coding, the linear and polynomial models are more accurate than the neural one. This is different to the 1996 data and on investigation it was found that faulty data points in the 1995 were to blame. The detection and removal of these faulty data points will be investigated further in chapter 5, however it was suspected that the linear and polynomial models were less affected by the outlying data points than the neural model which had more flexibility to model faulty data. Another interesting feature of the linear and polynomial models is that the test sets are consistently more accurate than the model sets, whereas for the neural model the converse is true.

The linear and polynomial models do not appear to exhibit any significant differences with inclusion of the site and or quench parameters in the training set, however the neural model does benefit significantly from the addition of the site parameter, for both the model and test sets. The inclusion of the quench parameter further improves the neural model's accuracy, however, this is less substantial than the improvement seen from the site code. The results for the addition of site and or

quench code demonstrate the neural model to be more accurate than the traditional regression techniques. From discussion with the metallurgists at BSES it appears that the use of a quench code may be increasingly useful if the data set was further extended to include data from a range of microstructures after the tempering stage.

4.3.5. Extending the 1996 martensitic data to cover multiple quenches

Having demonstrated that inclusion of site codes and quench codes resulted in an improved accuracy in the 1995 data pertaining to a variety of quench types, this technique was applied to the 1996 data. The inclusion of additional quench types in the 1996 data set resulted in an increase in the number of examples available, from 220 to 445, statistical analysis showing that this covered a wider range of manufacturing conditions and grades. The results in Table 4.13 were obtained when this extended data set was modelled.

	Model		Test	
	R^2	SD of Error	R^2	SD of Error
Neural	0.92	37.5	0.89	43.9
Linear	0.83	55.1	0.76	64.3
Polynomial	0.86	50.4	0.80	58.0

Table 4.13 Modelling of the 1996 data martensitic data set when extended to include multiple quench types with the addition of 2 indicator variables.

The SD of error was not used in the results prior to the extension of the 1996 data set to include multiple quench types shown in Table 4.8. It can be seen, however, that the neural network results of extending the model with the aid of the quench codes has, for this year's data, actually improved the R^2 statistic of the model. It is considered that this is because, although the model is covering a wider range of data, it is more accurate due to the increased data numbers.

4.3.6 Combining the 1995 and 1996 data

At this stage it was suspected that both sets may include some faulty data points, particularly with the 1995 data set, where it had been observed that the SD of error was much higher than with the 1996 data. However, before investigating how to deal with these faulty points the effect of combining the two data sets was investigated. There was a chance that, by combining both years worth of data, the modelling accuracy might improve due to the extended data numbers as was the case with extending the 1996 data set.

The martensitic two quench data from each year were combined such that the examples in the test sets of the separate models were now forming the test set of the combined model. This resulted in a total of 1029 examples, which was split 2:1 for model and test sets. Therefore, any outlying points previously in the model or test sets would still be in the same division so that this would not influence any change in accuracy.

Neural, linear and polynomial models were constructed as previously described with the descriptive codes for site and quench type. The results are shown in Table 4.14.

	Model		Test	
	R^2	SD of Error	R^2	SD of Error
Neural	0.81	59.4	0.85	50.9
Linear	0.72	72.8	0.77	63.2
Polynomial	0.75	67.8	0.78	62.0

Table 4.14. Results of modelling the combined 1995 and 1996 martensitic two-stage data set.

For the neural model, the results show that there is a slight improvement in accuracy over modelling just the 1995 data with similar input variables, however when compared with modelling just the 1996 data, the results indicate a deterioration in model performance. It is difficult to say at this stage if the model has improved or whether the 1996 data set is simply improving the statistics of the 1995 data when

the combination is made. What is clear is that neural performance is consistently improved over the linear and polynomial models, particularly with the increasing diversity of the data. However, even though this is the case, on examining the measured-vs-predicted graphs in Figure 4.7 of the model and test sets it is clear that even the neural model contains a number of high residuals that need to be improved. Also, a target standard deviation of residuals of 30 N/mm^2 set by BSES has not yet

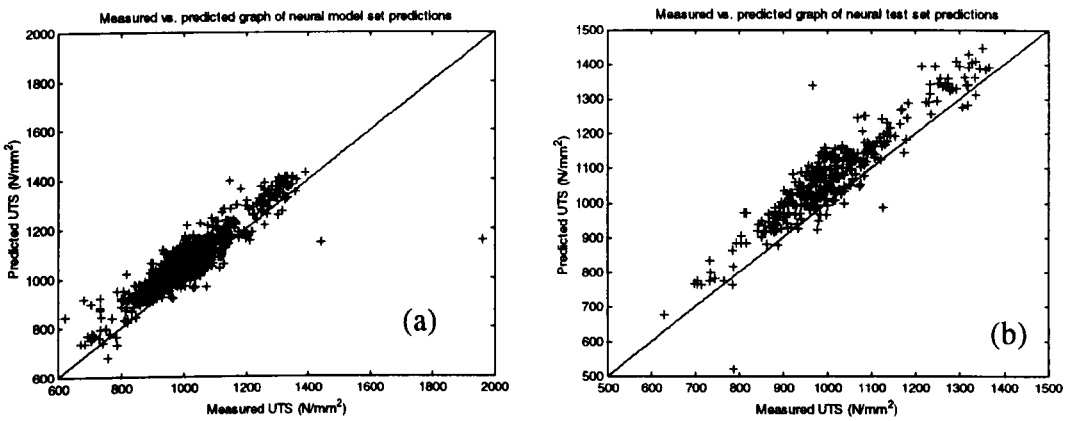


Fig. 4.7 Measured vs. predicted graphs of martensitic 2 stage, multiple quench type, multiple site, 1995 and 1996 combined (a) model data and (b) test data, with neural network technique

been reached. Overall, the distribution of the predictions for both the model and test set also appears to show a bias, such that the model is consistently over predicting. Throughout the experimentation in this chapter a fixed training strategy has been used for the neural model in respect to the number of training iterations and hidden layers. This was to avoid additional sources of variation other than the data set. However, as was described in chapter 2, it is widely accepted that, particularly as the data set becomes more complex, the neural model should be trained with a variety of architectures, together with a validation set, to achieve an optimum generalisation performance.

4.4 Chapter conclusion

Initially, this chapter dealt with process data familiarisation. The significance of key variables in the process was noted, and this resulted in the definition of a number of input variables for the development of a predictive model which are ultimately dependant on the data they describe. The main aim of the work has been to develop a data set capable of representing a range of quench types and site locations. The idea of categorising the microstructure of the steel after the hardening stage, to only include those steels with a martensitic microstructure, was enabled through the generation of a new variable obtained from expert knowledge of the process. It has been seen that further decomposition of the data to avoid a mixture of site locations or quench types results in reduced data numbers and segregated models which would be difficult to interpolate between. Experimentation has revealed that the neural network technique allows descriptive variables to be introduced, thus successfully representing a range of site and quench types within one model. This result should ultimately produce a model that covers a wide range of steel grades, enabling interpolation between the data in a more desirable way. The prediction of UTS test results has been used to investigate the development of this combined data set. It has also been demonstrated that the accuracy of the neural technique is superior to that of the linear and polynomial regression techniques, particularly as the data set becomes more diverse and carries a larger number of examples. The accuracy of the 1995 martensitic model was seen to be poorer than that of the 1996 data, which was thought to be caused by a greater range of steel grades, but also by data points that may be faulty. Similar points may also be present in the 1996 data set. Outlying predictions can be seen on the measured vs. predicted graphs as those lying far away from the 45^{o} line. These may relate to faulty data points, however the expectation was also that some faulty data points might have been modelled,

especially by the more flexible neural technique, so as to produce low model set errors, but high errors in the predictive case. Through the experimentation, agreed measures of statistical performance have also been established, and a target value standard deviation of residual of 30 N/mm^2 has been set.

The following chapter now investigates the techniques that were used for locating faulty data points and correcting them where possible. The effect of this 'data cleaning' on the model accuracy is demonstrated. The chapter also introduces another data set comprising steels from the 1997 data set, resulting in an increased, cleaned data set, which can then be modelled in a more extensive manner.

Chapter 5

Data cleaning

5.1 Introduction

Following a recognition that there may be some faulty data points present in the 1995 and 1996 data set, this chapter describes research and the development of techniques to enable such points to be detected and in some cases corrected. Initially, an investigation into the possible reasons for faulty data points with relation to the process was made through consulting experts at BSES. A literature review was then undertaken to investigate existing methods established to detect and correct faulty data points. As a result of this literature review combined with expert process knowledge, a structured procedure for the detection of outliers was then developed and validated through iteraction with BSES. Corrections of these faulty points have been made where possible. Finally, the improvement in modelling accuracy resulting from this data cleaning is demonstrated on a data set, which includes heat treatment examples from the 1995, 1996 and 1997 database records.

5.2 Possible reasons for outlying data points

One may think that outlying data points are faulty data points, and indeed often in the literature review, which follows, it was seen that papers spoke of removing outlying data points. An outlying data point can be defined as a sample that is statistically different to the rest. Martens[60] defines an outlier as a sample that carries a high statistical leverage. When investigating the possible reasons as to why 'outlying' data points might occur in the heat treatment process, it was evident that not all outlying points are faulty. And indeed some faulty data points may be 'inlyers' that is, they may be faulty, but as an individual point may not be statistically

different from the rest. When a faulty data point is referred to in this thesis it means a measurement that has a tolerance from the actual value which is greater than the measurement tolerances established for that point.

The reasons for outlying data points in the heat treatment process can be classed as shown in the following remarks. It is expected that, with many modern industrial processes using similar systems for data collection and retrieval, these categorisations would also hold for other processes.

- Data handling errors

  These are errors related to the way in which data are stored, sorted or retrieved. In the case of the MET database, data relating to the process are stored in more than one database. In such cases, there is scope for errors to result in the way the retrieval is made. Moreover, the resulting data set may contain a section of data that has been retrieved multiple times due to the way a search is carried out. This would result in repeated values. Repeated values are not faulty data points, however they may create a model which does not represent the true underlying process, since they affect the prior probability of the model. Another problem that can occur with data handling is when data is sorted according to an index in order to enable the merging of variables. For example, if the order of the index relating input variables to output variables becomes corrupted then a whole set of faulty data may result.

  When sorting data sets, as was done for the work in chapter 4 in a spreadsheet application, it should also be remembered that there is always a chance of invalidating data with accidental keystrokes.

- Measurement faults

As seen in chapter 3, the data are the result of many measurements taken throughout the process. In the case of bar size or test depth, the instrumentation is robust to malfunctions, however the measurements may be prone to human error, which may generate faulty results.

The composition measurements are determined via specialist equipment that is regularly calibrated, but is obviously still open to malfunction. Specimen preparation may also play an important part here in terms of producing faulty results.

Temperature measurement is complex since it introduces the concept of measurements taken to control the process. They do appear to be quite robust, although it was seen in chapter 3 that the furnaces comprise a set of control and a set of measurement thermocouples. The temperature measurement entered in the database is that prescribed by the metallurgist, which under normal conditions would be reached within the tolerances stated. The measurement thermocouples are closely watched to make sure that the temperature throughout the furnace is as desired. Because several thermocouples are used, a malfunction in one thermocouple would not greatly jeopardise the overall representation of the temperature in the furnace. Faults in the control thermocouples would affect the temperature within the furnace, however it is unlikely that this would not be noted by the measurement thermocouples and the necessary action taken.

- Process faults

This category can comprise many features, the most important of which are factors which may vary from treatment to treatment, but are not logged in the data set to be modelled. Even if a feature of the process is monitored locally and is acceptable, a combination of events relating to variables not logged may still

generate a faulty data point in that it does not relate to the data recorded in the database. This would include faults that cause variations in the furnace atmosphere or burner intensity. Time at temperature, from the hardening aspect, where it is important the bar has soaked thoroughly and from the tempering aspect where a particular soak temperature must be maintained for a specified period of time are two areas where problems may occur. On the cooling side of the process, delays in advancing the hot bars to the quench tanks and problems which cause significant variation in the temperature of the quench tanks may all generate faulty data points.

- Typographical /transcription errors

  These relate to human generated errors when recording results manually or transferring data to the database via a keyboard. Faulty data would be generated in both cases, however this may not always generate vastly outlying points, for example if temperature was incorrectly entered as 675 instead of 657.

- Incorrect treatment prescription

  This is a situation where the metallurgist may decide to use a specific treatment in order to obtain a particular set of mechanical properties, as set out by a z-card. Particularly with rare grades (where a predictive model is required), because the process is complex, the treatment recommended may be incorrect and this could result in the situation where that example may be very different to the rest. This would generate an outlying data point, but this may still be a valid one and indeed useful to the model, provided that the treatment was carried out as logged.

5.3. Literature on 'data cleaning'

A literature review was made on subjects relating to outlier and faulty data point detection and correction. In total only eleven relevant publications were found. Occasionally, authors of modelling papers may note the possible occurrence of problems caused by outlying data points, but it was rare to find any methodology for dealing with such anomalies.

Of the papers located, not all were directly related to a static industrial process model. Those that were not, tended to relate to time series problems such as that by Simoudis, Livezey and Kerber[61], who investigated a data mining package, which claimed to include methods for removing outlying data points. The data were from a financial problem and many of the methods of outlier detection relied on the rate of change of variables. Two dimensional scatter plots were also used to examine variables believed to have a fixed relationship. The paper shows that the techniques used in the package do not function efficiently without the interaction of an expert knowledge about the system. No procedures for dealing with outlying data points are discussed, and all outlying points are treated as faulty and hence the definition of false positive is introduced to describe points detected as outlier which are not faulty.

The next paper, by Wu and Cinar[62], is also related to a dynamic process. It is concerned with developing a knowledge-based input / output modelling system. It claims the system can provide autonomous outlier detection and the development of a parsimonious model. The outlier detection methods were based around two techniques, the first was a leave k-out validation method, where high residual points may be excluded, and the second was an outlier criterion. The outlier criterion involved data points being removed based upon a statistical measure of noise, which was known for the arbitrary data set used in this application. Although not stated, one

would assume that expert knowledge was used to set parameters relating to outlier thresholds for both methods.

The next four publications relate to learning algorithms that have been developed to cope with outlying data points. The first entitled 'Clearning' by Weigend, Zimmermann and Neuneier[63], details an unusual approach that combines cleaning data with learning. The authors introduced the concepts of the data modifying the model and then the model modifying the data. Clearning was used in conjunction with a pruning algorithm (where weights in the neural network were removed when they had little variance compared to their value). The technique modified the standard back-error propagation rule to accommodate for a cleaning term where, as well as weights being modified, the input data were also modified to reduce the cost function. Although the clearning technique may work well for some data sets, it was considered that there might be complications in data sets containing an uneven data distribution. In such a situation, there is a possibility that an example (correct or faulty) in an area of low data density may simply become modified towards an area of high data density but lower error, regardless of what its true values are or should have been. A similar technique[64], not directly related to data cleaning but carrying similar concerns, related to an adaptive neural network where new data that generated high residuals on prediction by the neural network were automatically included in the training set. The assumption is that the high residuals showed the data point to be novel, and did not account for the fact that it may be faulty.

The third publication relating to a modified learning algorithm is by Danuser and Sticker[65], who applied a robust generalised least squares algorithm to a specialist pattern recognition application. This does not directly relate to the application in this

thesis, however the concept of inlying data points is noted in terms of faulty points which are within the max and min limits of the variable ranges.

The fourth publication in this category, by Kosko[66], suggests that the standard back-error propagation algorithm, using the least squares training error, is sensitive to large training error and thus sensitive to outliers. Robust backpropagation was therefore suggested, where instead of having a least squares training error that amplifies large training errors, one can use the following error that treats the large errors and small errors linearly:

$$E_i = \sum_{j=1}^{P} \left| y_{ij}^{pred} - y_{ij} \right| \tag{5.1}$$

This approach gave improved outlier tolerance, but at the cost of longer training times.

The next three publications concentrated on the pre-processing aspect of outlier detection, that is data set preparation prior to learning. The first of these papers by Famili et al[67], considered the reasons for the occurrence of faults in detail, but offered little advice as to how these can be detected or corrected.

The second publication, by Guyon et al[68], contained a section on data cleaning in relation to inductive learning for data mining. This introduced the concept of an 'information criterion', which essentially means the value (rarity) of a data point. They proposed two data cleaning techniques using this criterion, one fully automatic and one semi-automatic. In the fully automatic technique, an algorithm removed points that had a high information criterion (those that are most surprising). The authors recognise that the danger of this method was that by automatically removing such examples, valuable data points which may genuinely carry large amounts of information may be excluded. The semi-automatic technique was more realistic in that an expert was used to investigate points with a high information

criterion, which in turn could decide whether or not to exclude a data point. This therefore acted as a filter, preventing an expert having to investigate all the data.

The third paper that classified outlier detection as pre-processing of data is by Qin and Rajagopal[69]. The authors mentioned two types of outlier, obvious ones and non-obvious ones. He defined obvious outliers as points that exceeded maximum or minimum limits for that particular variable. The non-obvious outliers were defined as the converse, lying within the limits of the data, which was earlier termed 'inlying faulty points' in section 5.2. The authors also noted that outlying data points generally caused high model errors, something which was noted in this project and which started the investigation into outlier detection techniques. In relation to non-obvious outlier detection techniques, work by Wold et al[70] is cited. This work relates to the multivariate statistic analysis technique of PCA, mentioned in chapter 2 of this thesis, for its capabilities to visualise the underlying process functions of a high dimensional system through the projection of the higher dimensions onto a reduced number of variables. This method was traditionally used to monitor changes in processes, however in this reference it was applied to detecting non-obvious samples that were statistically different. Qin also discusses the possibility of correcting outlying data points, and suggests that they may be treated as missing values, which can be replaced with a mean or otherwise interpolated value from either the input or the output of the data set.

The final paper in the literature search considered a very different reason for data cleaning. Hernadez and Stolfo[71] were concerned with the merge/purge problem; that is when merging data from multiple data sources (realising when examples have been previously encountered). This appears to be a considerable problem when dealing with marketing data such as that used for mailshot and customer relation applications. Although their application was unlike the one in this project, especially

considering their data was often alphanumeric, it raised interesting points in relation to repeated values when data are merged from multiple sources.

With this literature in mind, together with the particular features associated with the heat treatment application, a structured method for the detection of outliers was developed and is described in the following section. It was felt that much of this work could be readily applied to other industrial processes, and as a result the following work was published[72] at a conference entitled 'Intelligent Processing and Manufacturing of Materials'.

5.4. Structured method of outlier detection

It was felt that, rather than just considering outlier detection as something that should be performed as a pre-processing stage, it should be used throughout the

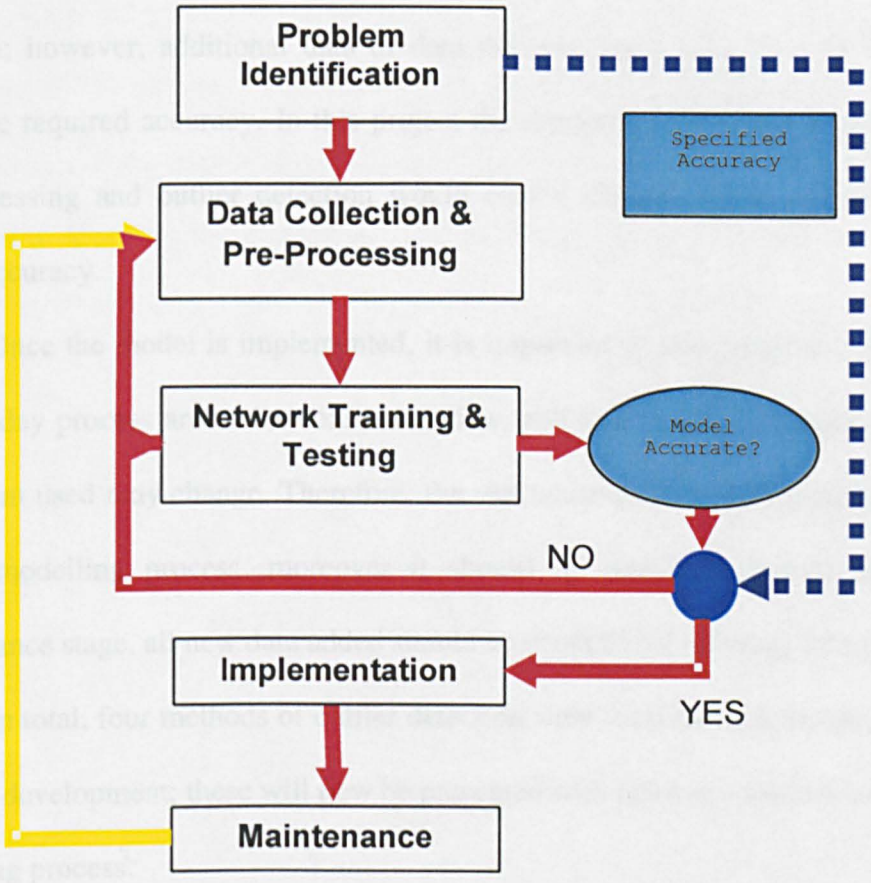


Fig 5.1. The main stages of a neural model's development

model's development. The basic stages of a neural model's development are shown in Figure 5.1. From Figure 5.1 it can be seen that the neural modelling process consists of 5 main stages. Progress through the problem identification and data collection stages has already been described in chapters 3 and 4, however there are also aspects of these stages which are important from a data cleaning point of view, as will be seen. The second part of the second stage is data pre-processing, some of which has been described in chapter4. As seen from the literature review, outlier detection is often performed as a data pre-processing stage and this indeed can also be done with some of the techniques developed in this project. The network training and testing stages generate a model accuracy, which, if it meets the specified accuracy, will indicate that the model may be ready for implementation. If the accuracy of the model does not meet the specified accuracy then modifications need to be made by repeating either or both of stages three and four. Sometimes re-training the model with a different algorithm or network architecture may be effective; however, additional data or data pre-processing may also be needed to reach the required accuracy. In this project the expectation was that improved data pre-processing and outlier detection would enable the attainment of the required model accuracy.

Once the model is implemented, it is important to also remember that many modern day process are under constant review, and that products manufactured and equipment used may change. Therefore, the maintenance stage forms the final stage of the modelling process, moreover it should be remembered that during the maintenance stage, all new data added should be checked for outlying data points.

In total, four methods of outlier detection were used through the stages of the model's development; these will now be presented with reference to each stage of the modelling process.

5.4.1. Problem familiarisation

The data cleaning process really began in the problem familiarisation stage in the form of information acquisition. Minimum and maximum variable boundaries were defined for variables, together with expected process behaviour, such as how one variable may relate to another, for example carbon content with hardness. Advice was sought on physical rules governing the data, while knowledge of indicator variables was also gained. Indicator codes are variables that are not used directly as the input to the model, but may play an important part in its development. For example, when considering a batch process, each example will often carry a code relating to its manufacture (a batch number) which may also contain a code relating to the location of its production. Whilst appearing abstract at first, this information may prove useful for outlier detection and correction as will be seen in the following sections. Knowledge of known process faults can also be gained, together with indications in the data to watch out for. Other important information gained at this stage was that of the expected model performance, as seen from Figure 5.1, this was important in knowing the amount of pre-processing and data cleaning that may be required.

5.4.2. Pre-processing

During the data pre-processing stage three methods for outlier detection were utilised.

5.4.2.1 Basic outlier detection

The first of these methods was 'basic detection', having found the maximum and minimum values of the input variables; an obvious place to start was to investigate points that exceeded these limits, Qin and Rajagopal[69] used a similar approach. Although this had partly been done during the data acquisition stage, on repetition, after the 1995 and 1996 data sets had been constructed, a number of points were investigated for each data set because they exceeded these limits. Building on this principle, known physical relations were also investigated, for example the correlation between UTS and Hardness. As described in chapter 3, these are two mechanical property tests, which have an approximately linear relationship. Through graphical inspection, any points that did not fit within limits of this linear correlation were investigated. An example of this is shown in Figure 5.2, where graphical inspection shows a number of data points to be outliers.

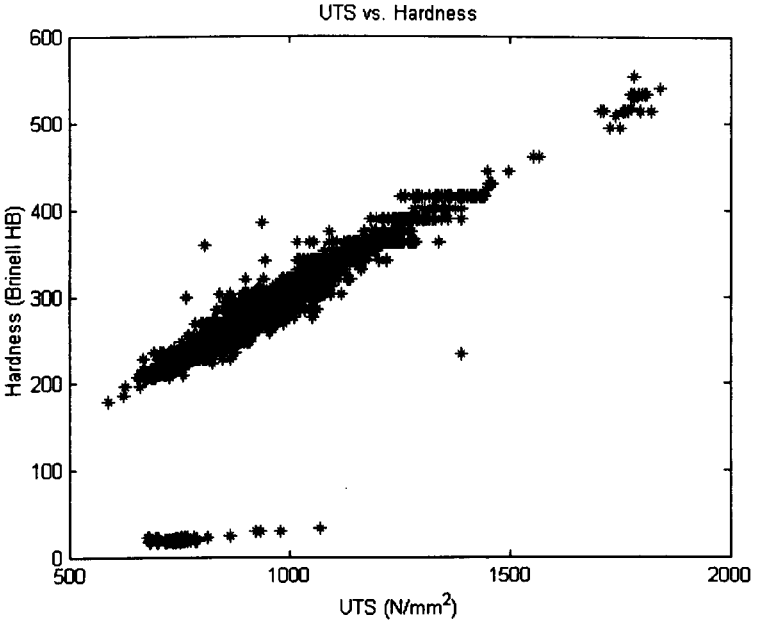


Fig 5.2 Graphical inspection of the relationship between Hardness and UTS can be used to locate outlying data points.

On analysing the outlying data points detected, it was found that the basic outlier detection method was useful for locating typographical and large process or measurement errors.

5.4.2.2. 'Sames' outlier detection

The second method of outlier detection used in the data pre-processing stage was the 'sames' method of outlier detection. The idea behind this method was to use the data to check its own integrity. The principle of this technique is that in the industrial process, the likelihood is that a similar size and type of steel may have been manufactured before under similar heat treatment conditions. Therefore, by picking one input vector in the data set, and searching for a similar ones, the mechanical test results of two or more similar examples can be compared.

This method is not only useful for finding outlying data points, it can also locate repeated values, which may also invalidate the data set.

The 'sames' method is easily automated, and in this case a program was written in Matlab to perform the search. The first stage of the automated 'sames' procedure is illustrated in the Figure 5.3.

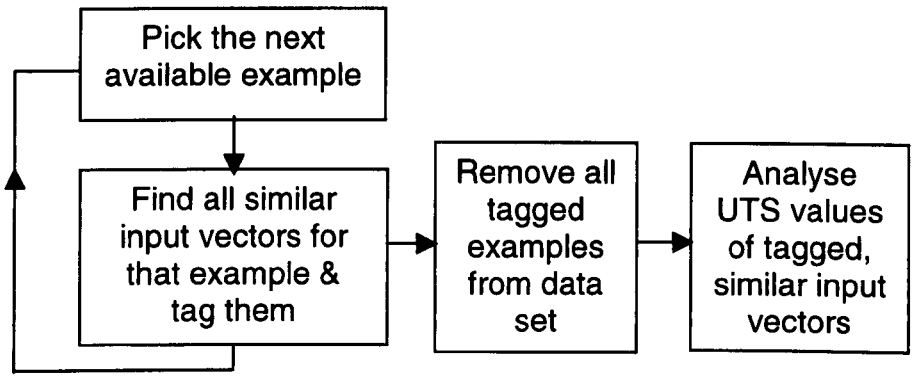


Fig 5.3. The first stage of the 'sames' outlier detection procedure

The loop continues until all the examples (not previously tagged) have been checked for similar input entries. The inputs to the data sets were site, size, test depth, C, Si, Mn, S, Cr, Mo, Ni, Al, V, hardening temperature and tempering temperature, together with the codes relating to the quench type used after the hardening and tempering stages. It should be noted that the inputs in the data sets for the 1996 and 1995 and the newly acquired 1997 examples had been increased after consultation with the BSES metallurgists. It was suggested that the variables Al, S and hardening temperature should be added in order to include any variables, which may affect the UTS beyond just the martensitic data set. The data relating to two treatment stages, with a mixed microstructure was included in the cleaning process, in case at a later date it was required for modelling purposes. Table 5.1 shows the number of examples available in each data set.

Data Set	Number of two stage mixed microstructure examples
1995	2094
1996	1216
1997	3038

Table 5.1. The number of harden and temper treatment examples available when the martensitic restriction was removed.

The test depth variable added was believed to be important from a hardenability point of view, and likewise was the hardening temperature and its quench type. The additional alloying elements of S and Al were added as it was envisaged that with some microstructures these may be important. There wasn't the same danger of introducing additional input variables into the 'sames' test as with building a model, since if not many similar input variables were found then the match criterion could be widened. This could be done by finding close input vectors instead of actually similar input vectors, in the form of a clustering algorithm, where the distances for each variable relied upon the significance of each variable to the model.

Having found a large number of examples to have similar input vectors the UTS values of the similar input values were then analysed. Expert knowledge was important here, since an idea of the process variation was needed in order judge the amount of variation that should normally be present in the UTS values of a group of identical steels treated in a similar manner. It was estimated that any group of similar input vectors should have UTS values within 40 N/mm^2 of each other. This meant that any variation within 40 N/mm^2 could be put down to process variation, but a deviation above that level may have indicated a process or data point invalidity.

The groups of similar input vectors were therefore categorised in the following way based on the difference between the UTS values within each group:

1 Groups having zero difference in their UTS values

2 Groups having a difference less than 40 N/mm^2 but more than zero.

3 Groups having UTS values with a difference greater than 40 N/mm^2 for the same input vectors.

In total, 109 examples over the three data sets fell into category 1. It was surprising at first that even two examples may have had an exactly equal UTS value. This is because even if the same test sample had been formed into two specimens and tested twice, then the likelihood is that the measurement tolerance associated with the UTS result would have resulted in a slightly different result. The UTS test ID codes for each group were examined and were found to be identical for each group; these groups therefore represented repeated entries in the data sets. For each group of similar inputs, only one example was retained, with the rest being deleted from the data set.

There were 1976 examples in category 2. In this category, one is not looking to find faulty data points, or indeed duplicate entries as with case 1. However, it was found that repeated test results for the same heat treatment, which were not duplicate entries, were present and needed to be dealt with. Because there were a large number of examples in this category, having made an initial investigation, a rule based correction method was utilised.

Upon investigation of the groups of similar input vectors, it was evident that the indicator codes, mentioned in section 5.4.1 were now very important. When investigating the cast number for each example, it was found that the cast number for each group of similar input variables was the same. This was to be expected, since it would be unlikely that one could obtain exactly the same composition in two separate casts given the nature of the steel making process.

Because the quantity of steel which may be produced from one cast exceeds that which may be heat treated within one batch, it was found that for a group of similar input vectors, a range of results would be present, pertaining to the same cast but different heat treatments. This was one of the reasons why the variation of up to 40 N/mm^2 was occurring. Each of these individual heat treatments represented a

valid result about the same composition, however it was also found that for some groups there may be two or more results with different test ID (therefore not a duplicate entry), but which pertained to the same heat treatment batch. After consideration, it was obvious that this related to the situation where for some large batches, more than one test sample was taken, and both entries were present in the database. It was decided that these multiple test results on the same heat treatment should be avoided, as they would unfairly represent the occurrence of one heat treatment based upon what was really a distribution due to the testing procedure used. It was therefore decided that a rule could be constructed to automatically remove these multiple tests on the same batch, and replace them with an average value based on the multiple test results. There was now only one entry in the database for each heat treatment performed.

Finally 370 examples were found to belong to category 3, these were similar input vectors which had a difference in UTS greater than 40 N/mm^2 (that which was attributed to the process variation). One group found in this category is shown in Figure 5.4. It can be seen that the majority of the values lie around 840 N/mm^2, however one example appears to be significantly different with a UTS of 785 N/mm^2.

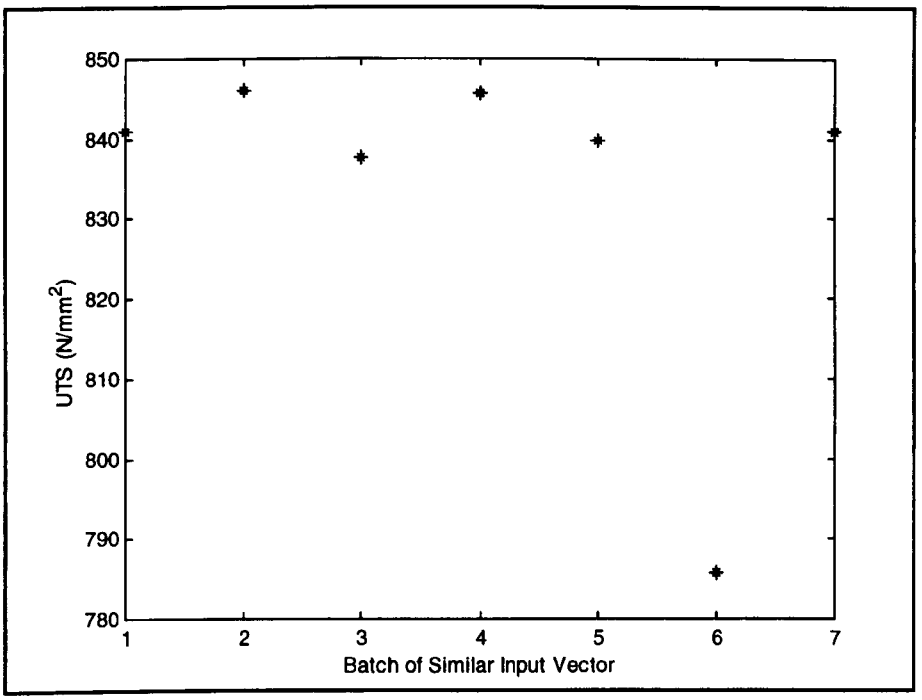


Fig. 5.4. A group of similar input vectors with a difference in UTS values exceeding 40 N/mm^2

The examples present in this category were referred to the test house manager for expert interpretation. Although there were 370 examples included in this category, because the 'sames' check had shown that these tended to fall into groups of several points where there would be 1 obvious outlier, investigation was fairly rapid. Heat treatment records were investigated, together with test records. Corrections were made where possible, or, where there were other examples in that area of the input space, the faulty examples were simply deleted. One interesting point was that some batches with a deviation slightly over 40 N/mm^2 were found not to be faulty. This was because although on average the process variation is within 40 N/mm^2, for some compositions that are harder to control, there may be greater variation due to process conditions without any specific malfunction. In chapter 7, detailed analysis of the UTS model highlights this phenomena.

To summarise the use of this method, it has been useful in finding repeated entries, and ambiguous data resulting from process variations or file handling errors.

5.4.2.3. Multivariate data analysis

Potentially, multivariate statistical analysis methods, such as PCA, can be utilised in two ways. The first is that because PCA reduces a large number of variables with some redundancy into a set of lower dimensional, new variables, which are orthogonal, the noise associated with many correlated inputs may be reduced. The effectiveness of this will be investigated in chapter 6.

The second way in which PCA may be used, more directly for data cleaning, is that, by viewing the many dimensions of the model inputs as a lower number of principal components, examples that are statistically different may be revealed, with the potential that these may be outliers. This technique is analogous to the numerous applications of PCA within industry for process monitoring and fault detection[73,74,75,76,77,78,79,80,81]. Within these applications, PCA was used to provide a summarised view of the high number of process variables, in order to aid a process drift or fault to be identified more easily.

The first example of this technique relates only to the 1996 harden and temper martensitic data with oil and air quench which was lab treated. Principal component analysis was performed as described in chapter 2, on the data set containing the variables of bar size, C, Si, Mn, Cr, Mo, Ni, V, tempering temperature and UTS. It should be noted that the UTS can be used as an input to the PCA calculation, since it may contain outlying data points, and will be correlated with other input variables, therefore forming part of a new variable. The data were normalised by dividing by the mean. The PCA program was then implemented from a set of functions available in the Matlab statistical toolbox.

The principal components were arranged in order of most significance and by investigating the principal component vectors, one gained an appreciation of which

variables most affected each principal component. The weights that corresponded to the first 3 principal components are shown in Table 5.2. These are the linear combinations of the original variables that generate the new variables. The largest weights in the first column indicate which variables are most important in the 1st principal component, in this case molybdenum & nickel. The most important variable in the second principal component is carbon. It should be remembered, however that each principal component is the weighted sum of all the variables.

Variable	1st Principal Component	2nd Principal Component	3rd Principal Component
Size	0.3155	-0.0207	0.0816
C	-0.0363	-0.6608	0.1570
Si	-0.0072	-0.2013	0.4847
Mn	-0.3327	-0.4110	0.0453
Cr	-0.3228	0.3818	0.3499
Mo	0.4197	0.2228	-0.0090
Ni	0.4480	0.0010	-0.2068
V	-0.3277	0.3706	0.2995
Temperature	-0.3205	-0.0922	-0.4259
UTS	0.3120	-0.1104	0.5434

Table 5.2. Weight values for the first three principal components of the Lab HR(Oil) TM (Air) martensitic data.

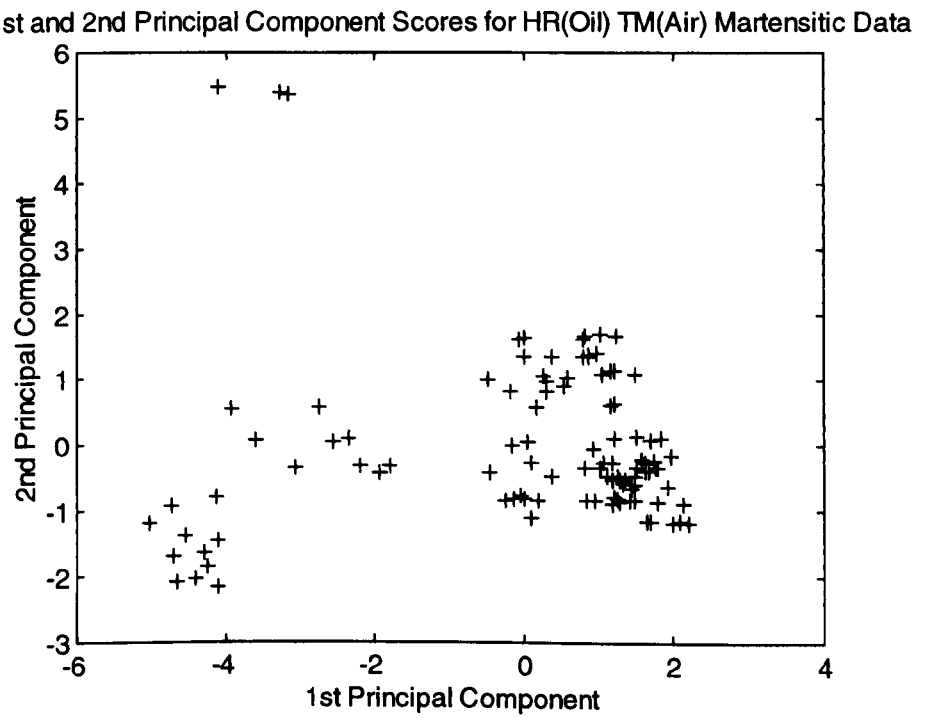


Fig 5.5 Scores plot for 1st two principal components generated from the HR(Oil) TM(Air) martensitic data.

The data set was then projected onto the new co-ordinate system defined by the first two principal components. These two principal components were then plotted against one another as a scatter plot, which is shown in Figure 5.5.

Figure 5.5 shows three main clusters of data along the 1st principal component, and three points that are outliers on the second principal component. By identifying which examples these scores related to, it was found that the distribution of the points was due to normal variation in the data. In particular, the 3 points that were outliers on the second principal component related to very low carbon values which, due to the negative term of carbon in the 2nd principal component, caused a high positive score.

The variance explained by each principal component can be investigated to determine how much variation in the data the first two principal components describe. Figure 5.6 shows that with the first two principal components approximately 60% of variance in the data was explained.

For this data set, only data points that were statistically different from the rest had

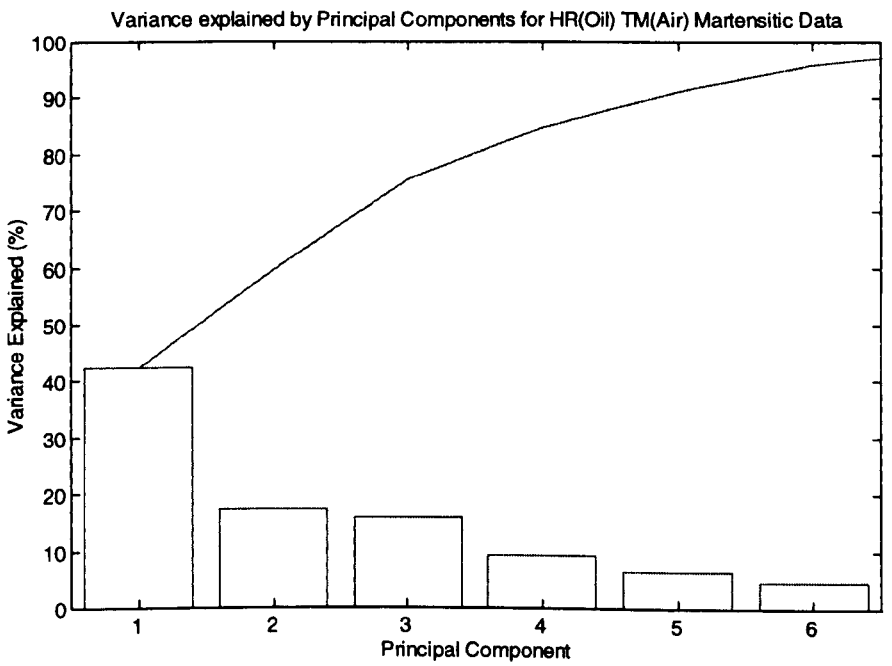


Fig. 5.6 The variance explained by each principal component of the HR(Oil) TM(Air) data.

been located, which did not represent faulty data points. In a further experiment, the data from the 1995, 1996 & 1997 examples with harden and temper treatment stage, multiple quench types, sites and locations were investigated with the PCA method. The data set consisted of the variables test depth, size, C, Si, Mn , S, Cr, Mo, Ni, Al ,V, hardening temperature, tempering temperature & UTS. These were similar to those used for the structured analysis of the combined data sets to account for the mixed microstructures, but with the addition of UTS and removal of the binary codes. The scores plot shown in Figure 5.7 was obtained.

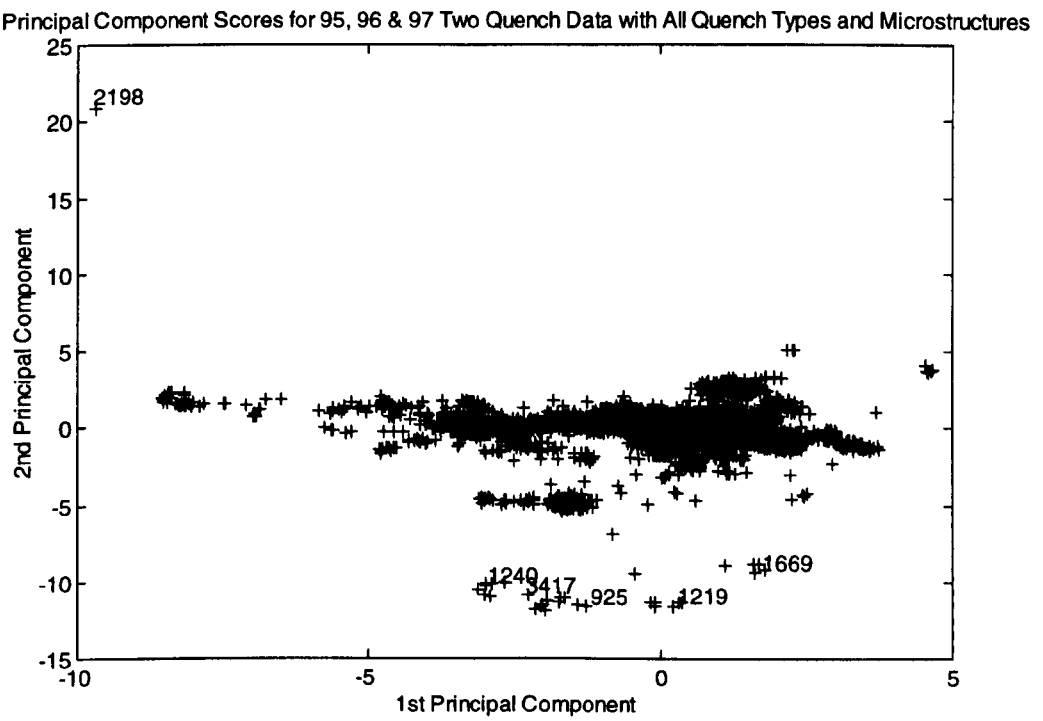


Fig. 5.7 Principal component scores for combined 95, 96, & 97 data sets covering all quench types, sites and microstructures.

When the examples relating to the outlying points of the scores plot were investigated, it was found that the point marked 2198 at the top left of the plot was a faulty data point. This example had a large typographical error in the hardening temperature resulting in an entry of 150°C instead of 800°C. The outlying points at the bottom of the plot were found to relate to vanadium-containing steels, which behave very differently due to this potent addition. As with the previous data set,

there was nothing wrong with most of these points except that they were statistically different. The conclusion of this work was that the PCA technique could help visualise the underlying process features in a reduced number of variables. However, the only faulty points which were readily located were those which related to what Qin termed 'obvious outliers', which could also be detected by a max/min basic detection method. The technique did readily help with the location of faulty inlying data points. The problems here are similar to the data cleaning technique in the literature review which selected data points with a high 'information criterion', it shows more what is different than that which is actually faulty.

5.4.3 Network training and testing

During the network training and testing stage, a model-based outlier detection procedure was utilised. In the literature review it was noted that outlying data points generally cause high model errors.

5.4.3.1 Model-based outlier detection

There are two reasons why a high residual may result. The first is that the modelling technique may not be capable of fitting the data. A graphic example of this is when a linear technique is used to model non-linear data; residuals will be present, not because the data is faulty but because the modelling technique is simply not flexible enough. Realistically, even with a flexible technique like neural networks, a level of residual will always be present because of noise in the data meaning that regularisation is required to prevent overfitting so as to provide good generalisation. The second reason why large residuals may be present is because the data may not fit the model, even though the model provides a good representation of the process.

There are then two possibilities in this case, the first is that the data are correct but are statistically different to that previously seen by the model and the second is that the data is faulty in some way.

Provided that a model covering a diverse range of examples was utilised, it was found that the highest residuals from both the model and the test set could be used as a basis for finding outlying data points.

There is a danger that if a faulty data point lies within a sparse area in relation to the model data set, that with little else to contradict this faulty point, the neural network would tend to fit the faulty data point. However, by constructing models a repeated number of times with different model and test set partitions, this problem may be avoided.

Initially, model-based outlier detection was performed with the 1995 and 1996 martensitic data using the linear and polynomial modelling techniques to investigate whether the modelling technique used would affect which examples were modelled with a high residual. The inputs to the model were Size, C, Si, Mn, Cr, Mo, Ni, V and tempering temperature. The output variable of UTS was used to locate high residuals. It was found that the examples relating to high residuals in the linear model and test sets tended to have high nickel contents. It was also noticed that the predictions for these points were always higher than the true value (the model was over predicting). From discussion with metallurgists it became evident that from experience they had found nickel to have a non-linear effect which reduced in strength at high addition levels. This illustrates the first reason why high residuals may result (because the modelling technique is not capable of fitting the data); therefore there was not necessarily anything wrong with these data points. After this experimentation it was decided that the neural technique would be used for all model-based outlier detection work.

Initially, the performance of the technique on the 1995 and 1996 martensitic, multiple-quench type data sets was observed. A neural model of each data set was constructed with the inputs described above as for the linear model. It was noted in chapter 4 that outlying data points might have been the reason for reduced model accuracy in the 1995 data set compared with the 1996 data set. It was therefore decided that an original, random ordered data set would be constructed for both years' data before and after data cleaning. The high residuals of the model and test sets were then analysed and corrected where archive records were available. When a replacement value was not available for a faulty data point, the offending point was deleted. A partition of 2/3 to 1/3 was made for the original model and test data set for each year's data. When the re-modelling of the 'cleaned' data was performed, the order and partition of the data was preserved so that when re-modelling with similar parameters was performed, a reasonable comparison of the model performance before and after cleaning with the model-based technique could be made.

From the 1996 data set, 20 points were found to have a residual greater than two standard deviations from the mean; of these: 7 were found to be faulty, 5 corrections were made and 2 deletions.

From the 1995 data set, 25 points were identified as outliers and 11 of these were found to be faulty; 1 deletion and 10 corrections were made.

The results of the modelling performance before and after outlier detection for the neural model are shown in Table 5.3.

Data Set	Before Data Cleaning				After Data Cleaning			
	Model		Test		Model		Test	
	R^2	SD of Error	R^2	SD of Error	R^2	SD of Error	R^2	SD of Error
1996	0.90	36.2	0.89	44.1	0.91	35.7	0.88	39.0
1995	0.86	51.9	0.83	54.7	0.91	38.4	0.903	42.0

Table 5.3. 1995 & 1996 mixed quench martensitic model performance before and after data cleaning.

It can be seen from Table 5.3 that the performance of both years' models has improved, particularly for the 1995 data set, whose performance is now much closer to that of the 1996 model.

To improve the efficiency of this technique, a file handling system was established whereby an example that had been checked, corrected or removed could be tracked. This was facilitated by the addition of a column in the spreadsheet containing the letters A, R, or C, where these denoted:

- Altered (where a data point was faulty and could be altered)
- Removed (where no correction of a faulty data point was possible)
- Checked (where the data point in question was not faulty, but had been checked for validity).

This notation helped, because if a point that contained the letter 'C' recurred as an outlying data point, it would not be referred for checking by the expert. Moreover, points which persistently re-accrued as outliers, but which were correct, indicated examples that the model could not cope with, and could therefore be used to target future data acquisition. The notation also provided a safeguard against typographical errors when correcting values, since if an altered point was present as a high residual, its allocated value could be checked. Points containing the removed flag could be filtered out when constructing a data set, but were still available for reference if required.

The use of the model-based outlier detection method was then extended to cover the mixed microstructure and mixed quench 1995, 1996 and 1997 data sets. The high residuals were selected from the model and test sets generated from a model of each of the years data, with multiple-quench types and sites, and without the martensitic constriction. Each year's data were treated separately at first. The criterion for a high residual was set as those greater than 2 standard deviations from

the mean. The procedure was performed iteratively as illustrated in Figure 5.8. The inputs to the model were those mentioned for the data set used with the structured detection algorithm.

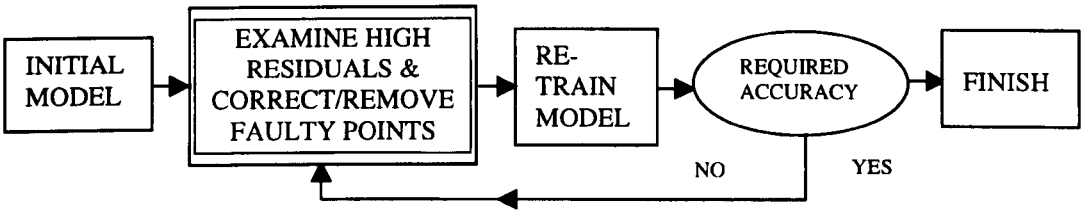


Fig 5.8. The iterative approach of model based outlier detection.

Having performed 3 iterations on the individual years' data sets, re-randomising the data contained in the model and test sets each time, the three data sets were combined, and outlier detection was performed for a further 3 iterations. The data set resulting now covered a range of microstructures after the hardening stage, and contained some 5711 examples. 194 faulty data points were found in the uncleaned data set. An effect of 'diminishing returns' was seen on the last iterations of the single and combined outlier detection, where there was initially a large performance improvement with the removal of some dramatic outliers, which was followed by a lesser improvement as more 'subtle' mistakes were detected.

Where possible, expert knowledge was applied to mistakes which may have related to a number of examples, such as one error where results pertaining to a certain Z-Card were entered in KSi (Kilograms per square inch), instead of N/mm^2! Using such knowledge, bulk corrections could be made using a conversion factor.

5.5. Review of inputs and network training procedure used to model the combined data set.

Before experimentation to determine the effect of data cleaning on the modelling accuracy, it was decided that, having added the additional 1997 data, a

range of experiments would be conducted to monitor the effect of the additional input variables suggested to allow for the martensitic microstructures in the data. These additional input variables of hardening temperature, Al & S were added to the combined data set to account for the additional range of the model. From discussions with metallurgists it was also expected that the effect of quench medium after the tempering stage may have little effect on the UTS model's accuracy, and so the removal of this variable was also attempted in the experimentation. Additionally, having read a section on binary coding by Tarasenko[15], it was decided to investigate the effects of using a series of binary codes to describe the sites at which the steels were treated instead of a single number.

The method used to train the neural network was also varied slightly to allow for the varying complexity of the data sets, by using a validation set. This involved calculating the RMS (Root-Mean-Squared) error of the model and validation set at intervals of the networks training. When the validation set's error started to rise, signifying over-fitting, the training could be stopped. In this way each data set could be trained to its maximal point possible and would not be limited by a fixed number of training iterations. At this stage, the number of hidden layer neurons was still kept at 6 and the fixed weight initialisation procedure was maintained for comparison purposes.

The experiments in data set formulation were performed for the 1995,1996 and 1997 data sets combined for a range of sites using the neural network technique. The following points describe the configuration of the data sets in the experiment.

- Data set 1:

  This contained all suggested inputs i.e. depth, size, C, Si, Mn,S, Cr, Mo, Ni, Al, V, temping temperature, hardening temperature and the new binary form of the site and quench inputs (for both quench stages). With the binary coding scheme,

one bit was allocated to each of the 6 sites now present in the combined data set. Each bit formed a separate input variable in the data set. The quench codes functioned in a similar manner with each binary input representing a separate input. Both schemes are shown in Tables 5.4 and 5.5.

Site	Input1	Input2	Input3	Input4	Input5	Input6
Pearsons	1	0	0	0	0	0
Whithams	0	1	0	0	0	0
WestBank	0	0	1	0	0	0
Special Steels	0	0	0	1	0	0
Roundwood	0	0	0	0	1	0
Lab	0	0	0	0	0	1

Table 5.4. Binary coding scheme used for treatment site locations.

Quench Type	Input1	Input2	Input3
Air	1	0	0
Water	0	1	0
Oil	0	0	1

Table 5.5 Binary coding scheme used to describe quench types after hardening and tempering stages

- Data set 2:

  This data set was similar to data set 1 but with the standard, single variable, numerical codes to represent each quench stage and the site location.

- Data set3:

  Similar to data set 1 except that the sulphur input was removed from the model

- Data set4:

  Similar to data set 1 except that the aluminium input was removed from the model.

- Data set5:

  Similar to data set 1 except that the second quench type binary codes (after the tempering stage) were removed.

- Data set6:

Similar to data set 1 except that the hardening temperature was removed.

The results of the modelling of these 5 data sets are shown in Table 5.6. for the modelling and validation data sets.

Data Set	Model		Validation	
	SD of Error	R^2	SD of Error	R^2
1	34.1	0.953	38.4	0.931
2	37.7	0.938	42.6	0.911
3	34.6	0.951	38.7	0.929
4	34.8	0.951	38.8	0.928
5	34.1	0.953	38.4	0.930
6	35.2	0.942	39.8	0.919

Table 5.6. Results of modelling the 6 data sets developed for the review of inputs to the combined model experiment.

Looking at the results of data sets 1 & 2, it can be seen that the binary input scheme does improve the performance of the model significantly. When comparing the data sets 3 and 4 with data set 1 it can also be seen that the suggested additional variables of S and Al have benefited the model, albeit to a lesser degree than changing the quench and site coding. Although the change in accuracy from the removal of these values was small, it should be remembered that this is over all of the grades of steel in the model. It is expected that for certain examples relating to additions of these elements, the effect of including these elements would be more dramatic.

When comparing the results of data set 5 to data set 1, it can be seen that the effect of removing the quench type after the tempering stage of the model is negligible. Therefore, as a result of this review, only the binary variables relating to quench type after the hardening stage were used.

Finally, the results of data set 6 show that the effect of removing the hardening temperature from the model causes a reduction in accuracy, which

therefore justifies its inclusion in the suggested set of input variables for the UTS model.

5.6 Experimentation to determine the effect of data cleaning on the model accuracy

Having investigated the effects of the additional variables in the data set, the inputs to the model were reviewed and it was decided that an experiment to show the effect of the data cleaning on the model's accuracy could now be performed.

The statistics of the 5711 cleaned data set's input and output variables are shown in Table 5.7.

Variable Name	Type	Min.	Max.	Mean	SD
Test Depth	Input	4	140	16.08	9.35
Bar Size	Input	8	381	156.4	83.95
Treatment Site	Input	Binary codes represent 6 locations			
C	Input	0.12	0.63	0.39	0.06
Si	Input	0.11	1.87	0.26	0.04
Mn	Input	0.35	1.75	0.76	0.22
S	Input	0.0005	0.21	0.02	0.012 636
Cr	Input	0.05	3.46	1.04	0.45
Mo	Input	0.01	1.0	0.26	0.14
Ni	Input	0.02	4.21	0.79	0.86
Al	Input	0.005	1.08	0.04	0.09
V	Input	0.001	0.27	0.008	0.023
Temperature at Hardening Stage	Input	820	980	856.9	16.9
Type of Quench at Hardening Stage	Input	Binary codes represent 3 quenches; oil water or air.			
Temperature at Tempering Stage	Input	20	730	604.9	70.7
Ultimate Tensile Strength	Output	516.2	1841	929.1	156.1

Table 5.7. Statistics of the cleaned 1995,1996 and 1997 combined, mixed microstructure, site and quench type data.

An indication of the increased range of grades covered by the data set can be gained from comparing the statistics of the variables in Table 5.7 with Table 4.6, covering just the martensitic harden with oil and temper with air quench. A much wider variable spread has been obtained, representing a wider range of grades and treatments covered within this single data set.

5.6.1 Experimental procedure and results for demonstrating the effect of the data cleaning technique

In order to show the effectiveness of the data cleaning process the following experiment was devised, thus demonstrating a predictive model's generalisation on an unseen test set both before and after cleaning.

When training and evaluating the neural network performance, data were partitioned into equal training, validation and test sets[15]. As previously described, the validation set was used to prevent over-fitting of the training data, by stopping further training when the validation set error started to rise. Having performed the data cleaning a test set was randomly selected from the data that had not been deleted by the cleaning process. These points were removed from the cleaned and uncleaned data sets; the cleaned test set was reserved for testing. The remaining data in the cleaned and uncleaned sets formed the training and validation sets. Due to deletions in the cleaned data, points were randomly deleted from the uncleaned set to make the number of data points equal thus not biasing the standard deviation calculation used in the results. The training and validation sets were constructed five times to show the effect of certain data points falling in the training and validation sets. The training was performed with a MLP containing 6 hidden layer neurons, using back error propagation with gradient descent and momentum. The weights were initialised randomly each time the network was trained.

The five models resulting from the cleaned and uncleaned data were then used to predict UTS values on the unseen test set. With a relatively low proportion of faulty points in the original data, it can be seen from the results shown in Fig.5.9, that the cleaned data have a lower standard deviation of residuals for both the training and test sets. Additionally, it can be seen that the predictive accuracy of the cleaned data is more stable with the different random ordering of the training and validation sets.

A further investigation, which highlighted the performance of the data cleaning

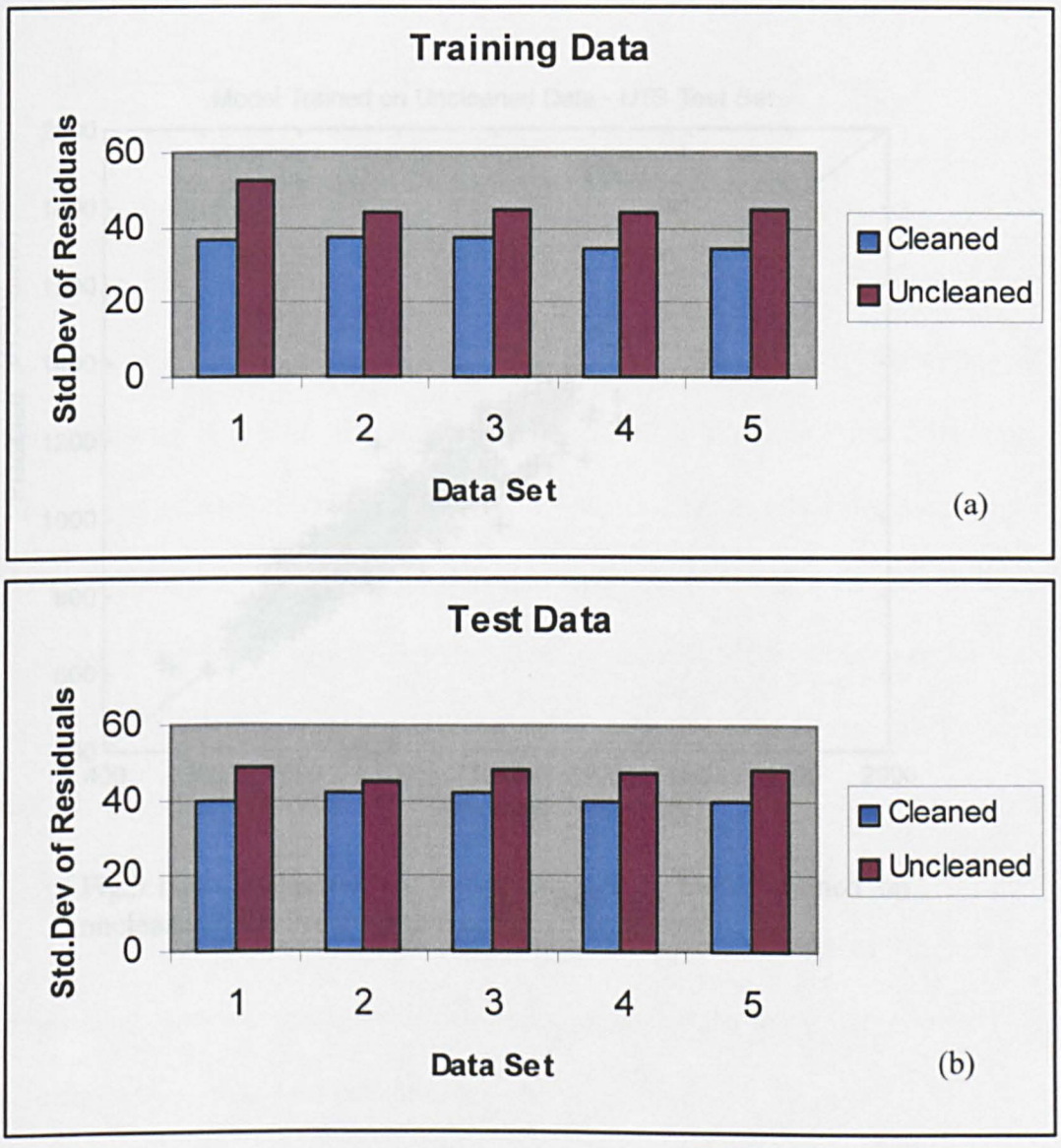


Figure 5.9. Results of experiment to determine effect of data cleaning on (a) model set and (b) unseen test set.

techniques, was to investigate the measured vs. predicted graphs of the individual models presented in the results. Figure 5.10 shows the test set predictions of two models predicting the same clean test data (identical measured values). The model shown in Figure 5.10(a) was trained on uncleaned data and the model shown in Figure 5.10(b) was trained on cleaned data. It can be seen than the model trained on uncleaned data has mis-predicted a number of tensile strengths above the 1400 N/mm^2 area, and generally has a poorer fit overall. The model trained on cleaned data has a closer fit and does not experience problems with the examples relating to a UTS around 1400 N/mm^2.

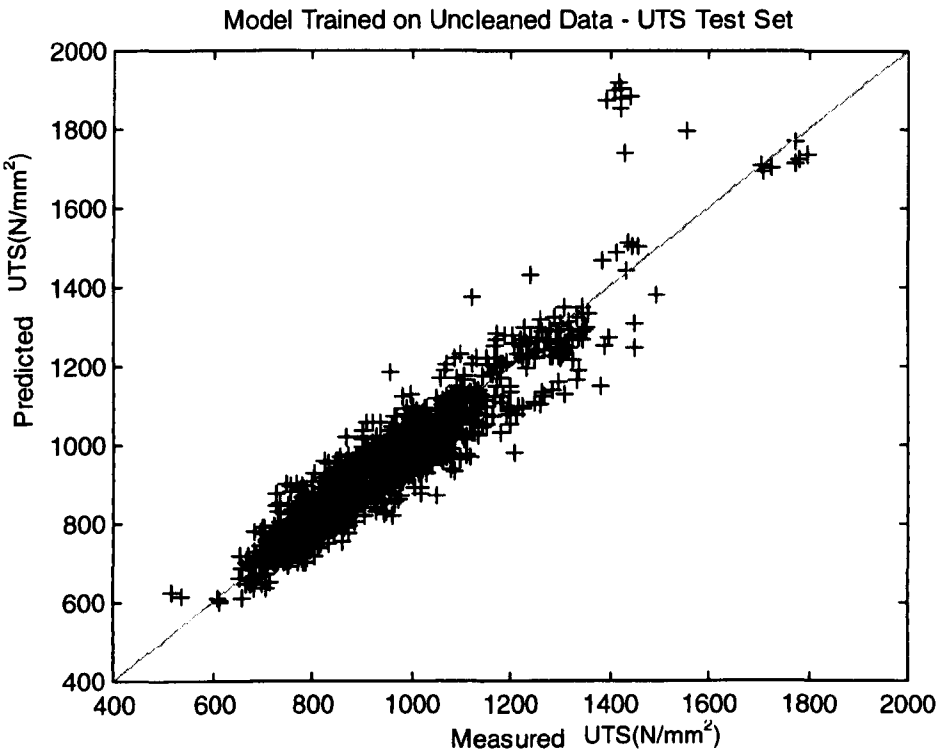


Fig.5.10(a). Measured vs. Predicted plot of model trained on uncleaned data. Predictions made on unseen test set.

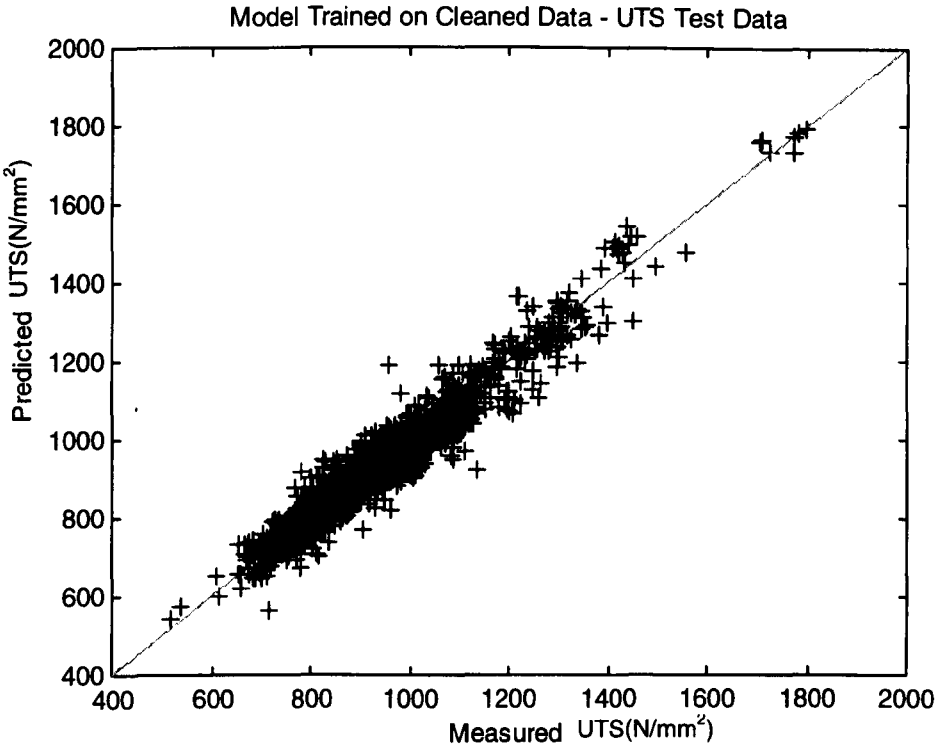


Fig.5.10(b). Measured vs. Predicted plot of model trained on cleaned data. Predictions made on identical unseen test set.

Upon examination of the uncleaned data it became apparent that a number of examples relating to a grade of steel had been entered in the wrong units, having a faulty value of around 1800 instead of a true value which should have been in the region of 1400N/mm^2. The uncleaned model had learnt this as a feature of these types of steel and when tested on them again, produced incorrect results for that grade. Overall, other faulty points also contributed to the poorer performance seen throughout the uncleaned model's range.

One further point which should be noted about both the uncleaned and cleaned models in Figure 5.10 is that, having trained the network using the validation set, there is no longer a 'bias' in the distribution of the points along the measured vs. predicted graphs as was seen in Figure 4.7.

5.7 Chapter conclusion

This chapter has seen the development of a structured technique for the detection of faulty data points within the BSES data sets. Research into current work on outlier detection revealed that only a small number of authors had tackled outlier related issues, often only providing information on why outlying data points were present as opposed to how they might be detected and corrected. The literature review did, however, generate a number of ideas towards possible methods of outlier detection, whose implementation was described throughout the chapter. The description of all the techniques used is related to the entire model development process, as opposed to existing authors who consider data cleaning purely a data pre-processing stage. One technique, the 'sames' checking method is particularly novel and has also solved the problem of repeated values within the data set.

Having cleaned all the two stage data relating to the BSES processes, it was decided that with the addition of other input variables, the martensitic constraint placed on the modelled data in chapter 4 could now be removed, in order to model a range of microstructures after the hardening stage. This was important since some important steel grades contain non-martensitic microstructures after the hardening stage and are now accommodated by the model. A review of the effect these additional variables have on the prediction accuracy was then made, which also saw the introduction of binary input variables instead of numerical codes to denote site and quench type. The review also utilised an improved training technique, and showed that all the additional inputs were justified, but that the second quench descriptive variables could be removed as they had little effect.

The expansion of the data set resulted in a wider range of steels and treatments being represented by the model, which was demonstrated with statistical measures on the cleaned data set.

The effect of the data cleaning on the prediction accuracy was then investigated in the form of an experiment, where it was seen that the model accuracy is very dependant on the quality of the data. The work has demonstrated that even if a small group of points are faulty, a significant deterioration in performance will result.

The project aims to model a wide range of steel grades accurately. The ideas of data decomposition have now come full circle, through the use of expert knowledge for the selection of variables and data cleaning, a wide range of microstructures have now been modelled with improved accuracy over the decomposed data sets considered in chapter4.

Having developed a clean data set with suitable input variables to encompass a range of treatments, the next chapter now shows the use of more advanced neural modelling techniques. These techniques are used with the data set developed to produce a UTS model, the primary project aim, which meets the specification accuracy requested by BSES, and also a range of other mechanical property prediction models such as proof stress, reduction of area, elongation and impact properties. Previously acquired data cleaning knowledge is then applied to these additional output variables and the resulting model performances are evaluated in detail.

Chapter 6

Improvements and extensions to data modelling

6.1 Introduction

This chapter uses the cleaned data, developed in chapter 5, to generate models capable of predicting a range of mechanical properties. Initially, improvements in network training techniques and architecture are considered, with the aid of a brief literature review into key topics considered important in producing models of improved accuracy.

Having previously kept most neural model parameters constant, the effects of the number of hidden layer neurons and hidden layer initialisation weights are now considered. Modular and ensemble techniques are investigated as a result of the literature review in a variety of configurations. Their relative performances are evaluated on a fixed test set.

Having generated a UTS model of increased accuracy, the data set is expanded to include additional outputs and an investigation is made into other incompatibility problems which result from this.

Experimentation in network architectures and training techniques led to the use of the Matlab neural network toolbox to produce much of the experimentation in this chapter, as this provided more flexibility than the commercial package previously used in this project. Additionally, given the growth in the data set size, multiple processor Unix machines at the University were used to produce models more quickly.

6.2. Improvements to the MLP neural modelling technique

Having generated a cleaned set of data it was found in chapter 5 that a further improvement was needed in the model's standard deviation of residual value in order to meet the specification of $30N/mm^2$ set by BSES at the original project meeting. Having previously only used a 'fixed' set of parameters to train the network with a simple back error propagation rule with a momentum term, it was decided that investigations should be made into more advanced techniques of network training. Already in chapter 5, the use of a validation and test set had led to an improved distribution of data on the measured vs. predicted graph and so similar techniques to 'tailor' the network to the data were needed.

Literature relating to methods of improving the MLP networks performance was therefore investigated in order to establish a suitable technique to use.

6.2.1. Finding the optimum number of hidden layer neurons

Throughout much of the literature reviewed in chapter 2 and indeed in the texts of Tarrasenko[15] and Bishop[12] it was seen that the number of hidden layer neurons used in a single layer network plays a very important part in determining the model performance. In the early stages of the work, experimentation with this parameter led to 6 hidden layer neurons being used even when the data set had been expanded to include data relating to many different grades of steels. The main reason for this was because the network parameters were being kept constant in order to determine the optimum data set configuration.

If too many hidden layer neurons are used, it is likely that the network may behave like an associative memory, storing individual training points rather than learning the underlying function. This would be similar to the effects of over-training a network. The converse of this is that if there are not enough hidden layer neurons,

then the network will tend to under-fit the data and therefore not describe the underlying process functions effectively.

It was decided that an experiment should be performed to determine the effect of the number of hidden layer neurons on the cleaned data set. In order to accurately determine the optimum number of neurons it was realised that a range of models would need to be produced for each incrementing number of hidden layers used, each with different initialisation weights. This was because the initialisation weights are believed to affect the final value to which the network converges, and if one is trying to find an optimum solution this needs to be varied.

The Matlab neural network toolbox was used to write a script which would alter the number of hidden layer neurons used by the MLP with back error propagation with momentum from 1 to 33 neurons, ten initialisations being made for each configuration. In most literature and texts where such experimentation has been performed, it was noted that the MSE was used as a statistic to measure model performance and so, as this was readily available through training in the Matlab package, it was utilised. The model inputs and output established in Table 5.7. were used initially. The network was trained using the validation set early-stopping criterion and the results of the Training, Validation and Test sets were logged for each of the 320 models trained.

The results of this experimentation are shown in Figure 6.1, which shows the average result of the ten initialisations made. It can be seen that there were some fluctuations in the results, despite taking the average of ten initialisations. As the number of hidden layer neurons are incremented the training (or model) error falls steadily. The validation and test set errors also fall to begin with, however after 10 hidden layers a plateau is reached which, after 17 hidden layer neurons, begins to increase. Close inspection reveals that the lowest validation and test set error was

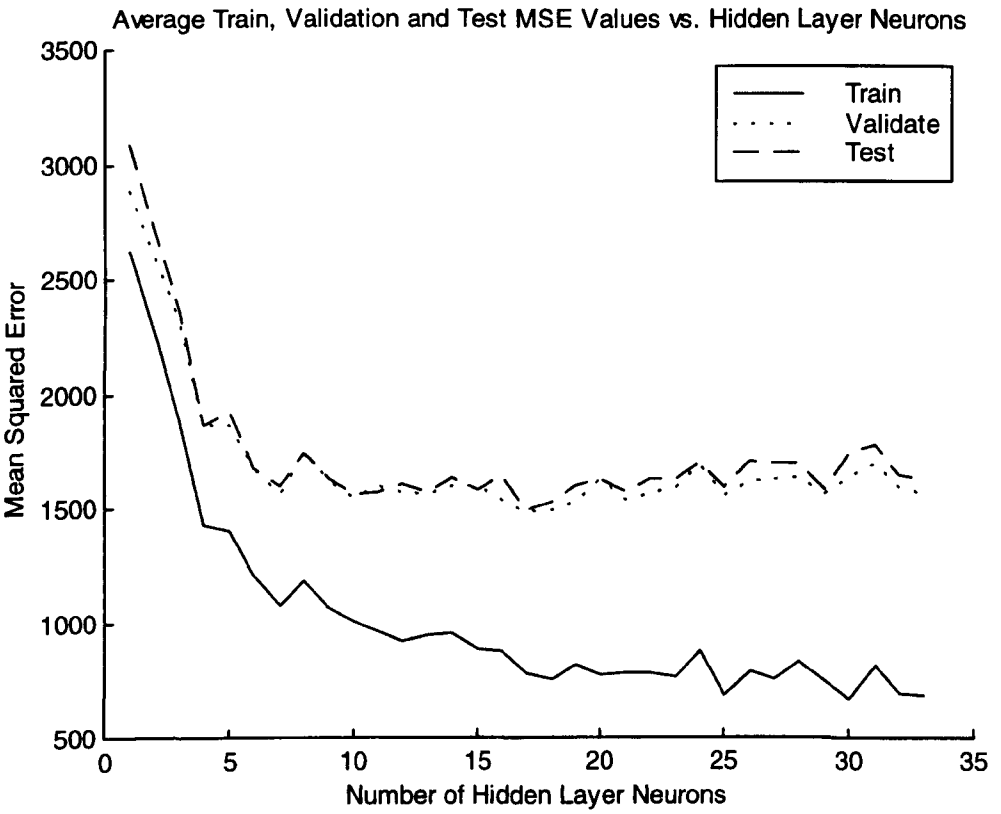


Fig.6.1. The effect on the average MSE value of ten networks trained with varying hidden layer neurons on UTS data.

reached at 17 neurons, however there is minimal improvement on the performance at 10 neurons. To promote a parsimonious model and reduce computational intensity it was decided that 10 hidden layers would be sufficient for this data. A noticeable effect was also seen from the initialisation weights on model accuracy, however this will be discussed further in section 6.2.4.

If upon detailed investigation of the model, it appears that it does not generalise effectively, another method to prevent overfitting is to use a 'regularisation' technique, which involves altering the training algorithm in some way to generate a smooth mapping which does not fit the training data noise. This enables one to construct a network with a generous number of hidden layers and yet still not over-fit the training data. One popular method as suggested by Bishop[12] is the weight decay method which involves adding a penalty term Ω such that the squared error function becomes:

$$\tilde{E} = E + \mu\Omega \tag{6.1}$$

where:

$$\Omega = \frac{1}{2}\sum_i w_i^2 \tag{6.2}$$

This sort of technique is particularly useful when one is trying to model unfamiliar data, with little time to experiment and investigate the effects of hidden layer neurons on generalisation performance.

One further form of regularisation which is also claimed to have other advantages in 'automated' neural network training is that of the 'Bayesian framework' for training the MLP. This will be discussed further in chapter 7 from a different perspective than model accuracy. Literature such as that by Penny[82], suggest that although the Bayesian technique of regularisation may offer an 'automated' approach to training a model with good generalisation, techniques such as cross validation and model ensembles may provide equally good results.

6.2.2. Improved training algorithms

It appears that there are two main reasons why improved training algorithms exist, these being for increased accuracy in the final model, and improved training speed.

In chapter 2, the addition of a momentum term was shown to improve model accuracy, by helping to avoid the problems of local minima, and represents a historical advance in improving the final accuracy to which a MLP may converge.
It has already been mentioned in chapter 2 that the selection of a learning rate which is too large will result in an unstable network, and that if the learning rate is too small then the network will take too long to converge. It is easy to see that a network is diverging and so the use of an algorithm that incorporates an adjustable learning rate could therefore be considered a speed rather than an accuracy improvement.

When considering improvements in training speed, there is a range of algorithm improvements that exist. These take two approaches, heuristic and those that utilise standard numerical optimisation techniques.

Of the heuristic algorithms, the use of a variable learning rate has already been considered. Another major method that falls into this category is that of the conjugate gradient[12]. With basic gradient descent, the direction of each step is given by the local negative gradient of the error function and the step size is fixed. The conjugate gradient method came about through a realisation that moving along a search direction given by the local negative gradient vector is not necessarily the optimal strategy; instead, a sequence of successive search directions such that each is 'conjugate' to all previous directions is used. This forms the basis of the conjugate gradient optimisation algorithms[83,84,85].

Of the numerical methods, Newton's method is the first[12]. Newton's method is an alternative to the conjugate gradient methods for fast optimisation, where local quadratic approximation can be used to directly obtain an expression for the location of the minimum point of the error function. This gives the basic step of:

$$w_{k+1} = w_k - \mathbf{H}_k^{-1}\mathbf{g}_k \tag{6.3}$$

where $\mathbf{g}$ is the gradient at step k and $\mathbf{H}$ is the Hessian matrix (second derivatives) of the performance index at the current values of the weights and biases. Bishop explains that the exact computation of the Hessian matrix and moreover its inverse for non-linear networks is computationally demanding and becomes prohibitive if performed at each stage of an iterative algorithm. Quasi-Newton methods have therefore been developed to overcome this computational intensity. Instead of calculating the Hessian directly and then calculating its inverse, they build up an estimation of the inverse over a number of steps, utilising the first order derivatives. The Levenberg-Marquart algorithm is similar to the Quasi-Newton methods in that they were designed to approach a second order training speed without having to compute the Hessian matrix. However, the Levenberg-Marquart algorithm was specifically designed to minimise the sum of squares error function. Under this condition the Hessian can be approximated as:

$$\mathbf{H} = \mathbf{J}^T\mathbf{J} \tag{6.4}$$

And the gradient can be computed as:

$$\mathbf{g} = \mathbf{J}^T\mathbf{e} \tag{6.5}$$

$\mathbf{J}$ is the Jacobian matrix, which contains first order derivatives of the network errors with respect to the weights and biases.

Hagan and Menhaj[86] provide a full description of the algorithm, however the weight update can be written as:

$$w_{k+1} = w_k - [\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}]^{-1}\mathbf{J}^T\mathbf{e} \tag{6.6}$$

The Matlab neural network toolbox contains an implementation of the Levenberg-Marquart algorithum. The variable λ is decreased after every step where there is a reduction in the performance function, and is increased only when a tentative step would increase the error function. The aim is therefore to keep λ small and thus stay towards a newton-like update, whereas if λ is large the standard gradient descent rule is used more. It is claimed that the algorithm provides between 10-100 times improvement on the computational time of the standard gradient decent backpropagation method.

The Levenberg-Marquart algorithm was applied to the 5711 examples heat treatment data set in this project with inputs as established in Table 5.7 and the output of UTS. It was found that a training time in the order of 10 minutes as opposed to 3 hours could be obtained when the algorithm was used instead of the standard back error propagation with momentum algorithm.

The drawback of any method that utilises the Jacobian matrix is the high memory requirements of the algorithm. The size of the Jacobian matrix is W x N where W is the number of weights and N is the number of samples in the training data set. However, having access to computers with relatively large amounts of memory, coupled with a relatively small network size meant that this was not a problem. If this was ever a problem, then reduced memory algorithms have also been developed where the Jacobian matrix is split into sub matrices which can be computed in turn, but this does however impose a significant computational overhead.

6.2.3. Pre-processing with PCA

The use of PCA for data analysis has already been considered in chapter 2 from a theoretical point of view and chapter 5 when applied to the data cleaning problem. It was considered in chapter 5 that another method of improving the model performance might be to use principal component analysis to transform the input data matrix to the neural network, so as to remove any redundancy in the input variables, and therefore any unnecessary noise penalty. The technique was mentioned to have three effects on the data set:

1 It orthogonalises the components of the input vectors

2 It orders the resulting orthogonal components (principal components) so that the greatest variation is in the first component

3 By setting a cut-off point in the variance explained by each component it can eliminate those components which contribute least to the variation in the data set.

It was decided that an experiment would be performed to establish whether the PCA technique could improve model accuracy (and yet reduce its complexity) when applied to the cleaned data set. Having recently varied the number of hidden layers, and changed to the Leveberg-Marquart training algorithm it was decided that ten benchmark models would be constructed using these techniques in order to establish a set of average results before PCA was applied to the model inputs. Then, having performed the principal component analysis, experimentation into the number of hidden layer neurons required to model the 'reduced' data set was performed. Having selected the required number of hidden layer neurons though experimentation, the

model with PCA inputs was then trained 10 times, this being done with the new algorithm with varying initialisation weights.

The PCA algorithm was implemented within the Matlab neural network toolbox environment, where it is possible to select the number of components kept as inputs to the model as a function of the 'minimum fraction of variance explained'. This was initially set to be 2%, but in order to monitor the variable reduction this gave, the level was then later increased to only include those components accounting for 10% or more variation in the data set.

The results are shown in Table 6.1 for the three configurations used in the experiment, using the average of the 10 initialisation's standard deviation of residuals as the performance measure for each data set.

Data set	Minimum variance explained	Number of principal components	Training set SD of residual	Validation set SD of residual	Test set SD of residual
1	0%	N/A (Full (22))	36.4	39.7	40.1
2	2%	16	45.3	55.2	55.4
3	10%	2	99.4	98.6	97.1

Table 6.1 Results from experimentation with PCA inputs to neural model trained on the Levenberg-Marquart algorithm.

It can be seen from Table 6.1 that even when a large number of principal components are used as the input to the model, the performance of the training, validation and test sets deteriorates significantly. This deterioration is worsened when only the two most significant principal components are kept as inputs to the model. This suggests that the variation in the data set is distributed throughout the variables, and that there is little redundancy. Figure 6.2 shows the variance explained by the first 10 principal components. When this is compared to Figure 5.6, for the harden and temper martensitic data with just oil and air quench, it can be seen that the variance explained by the first two principal components is now significantly less than the

60% explained before. This is believed to be due to the additional complexity that the

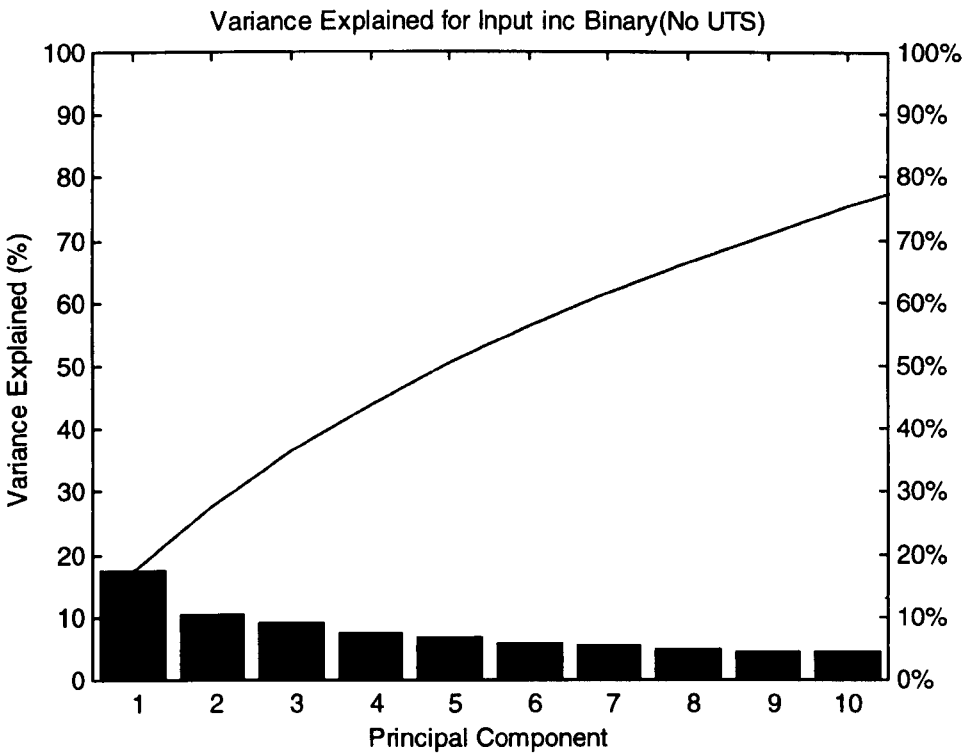


Fig 6.2. Variance explained by the first 10 principal components of the cleaned data set.

multiple quench types and microstructures now bring to the data set. It should also be remembered that the standard principal component analysis algorithm can only establish a 'linear' relation between two or more variables and therefore will be less effective on a data set with increased non-linearities. Non-linear methods of this technique are, however, a developing field[87,9].

Following this experimentation it was decided that the PCA method of data pre-processing would not be used in this case.

6.2.4. The effect of random weight initialisation on network training

It has already been mentioned in both chapter 2 and the data set development chapters 4 and 5, that the random initialisation of the network weights affects the final convergence of the neural model. Previously the weights have either been kept

constant or an average of several initialisations has been taken to allow for the effects this may have.

Table 6.2 shows the standard deviation of the test set residuals obtained for 10 different initialisations of the same network, with fixed training, validation and test sets.

Initialisation Attempt	SD of Residual of Test Set
1	36.1
2	41.6
3	41.8
4	38.8
5	36.8
6	36.7
7	37.3
8	36.1
9	35.2
10	39.8

Table 6.2. The variation in SD of residual from making random weight initialisations.

It can be seen that there is a significant variation in the test set accuracy based on the effects of random initialisation. Tarassenko suggests that many networks should be generated with varying weight initialisations and, where data is sparse, with different training, validation and test set partitions. Then the network that provides the best performance in relation to a pre-defined set of criteria should be selected. He warns against the use of the test set as a means of selection since this may also become part of the optimisation procedure (unlike the validation set, the test set should be unseen in the respect that it should play no part in the training of the model).

One paper, concerned with reliable roll force prediction[43], used more than one model (in this case a neural and a mathematical model) to make a model prediction. This claimed to increase the reliability and accuracy of the model produced, it was

therefore decided that literature relating to the combination of several models would be researched, the results of which will be discussed in the following section.

6.2.5 Combining models

The literature search revealed a number of papers that considered the combination of models, specifically neural models.

Two literature reviews on the combination of neural networks have been performed by Sharkey[88,89]. These reviews reveal that there are in fact two main architectures for combining neural networks for regression. The first is the ensemble of networks and the second is the modular approach. Various authors have claimed improved results with certain data sets when modeling both techniques, and indeed the main reason for combining networks is to gain a performance improvement. A secondary advantage of combining networks, particularly in the case of the modular approach, can be the problem simplification that task decomposition brings. The function of the individual network in each topology is quite different.

When combining a set of networks using the modular approach, each network is trained on a different portion of the data such that it represents a different function of the process. This 'task decomposition' may either be automatic (via an algorithm) or explicit (based on expert knowledge). An example of a modular approach relating to the data in the project would be to train a network for each site of steel production. In a sense, through the decomposition already described in this project, the modular approach has already been followed to some extent.

Sharkey notes that the relationship between the networks in a modular topology may be successive, co-operative or supervisory. In a successive arrangement, the first network may be used to pre-process the data before passing to the next network. A co-operative relationship is typically where a gating network is

used to combine the predictions from certain modules. An example of a supervisory approach is where one module might be trained to select the parameters of a second net, on the basis of observations of the effect of various parameter values on the performance of that net.

The idea of the co-operative relationship between modules brings us to the concept of an ensemble arrangement. The function of each of the networks in an ensemble is identical (to represent the whole of the input space). When one wishes to use the ensemble to make a prediction, it is made up of a weighted sum of the predictions of all of the networks. The ensemble technique does however rely on the fact that, despite learning data about the same problem domain, each network should still be different. If each network produced identical predictions, then there would be little advantage in combining networks. Instead, the differences in each network lead to a robustness in the overall prediction, and may be obtained by the following techniques:

- Varying initial conditions
- Varying training data (but from the same problem)
- Changing individual network architectures
- Training each network with different algorithms.

The papers revealed by the literature search all outlined examples where the ensemble approach had been followed. However, enough information was gathered from Sharkey's paper to enable the modular approach to be performed on the cleaned data set with a variety of decomposition methods. This will be investigated further in section 6.2.5.1.1

Theoretical proof that diversity of networks can lead to reduced generalisation error can be seen in papers written by Sollich and Krough[90] and

Bishop[12]. Within these publications, the increased accuracy that may be obtained from an ensemble network is illustrated by considering the bias and variance of the predicted values on the test set. This is considered for the usual case where one is trying to approximate a target function f_0 from R^N to R, where only noisy samples of the target function can be obtained. The inputs to the network are drawn from a distribution $P(x)$, and it is assumed that an ensemble average of k independent predictors is denoted as:

$$\bar{f}(x) = \sum_k w_k f_k(x) \tag{6.7}$$

where w_k is a weight representing the strength of 'belief' in each network, which has a positive value, where all the weights total to one.

For an input x the error of the ensemble (x), the error of the kth predictor $\varepsilon_k(x)$, and its 'ambiguity' $a_k(x)$ may be defined as:

$$\varepsilon(x) = (y(x) - \bar{f}(x))^2 \tag{6.8}$$

$$\varepsilon_k(x) = (y(x) - f_k(x))^2 \tag{6.9}$$

$$a_k(x) = (f_k(x) - \bar{f}(x))^2 \tag{6.10}$$

Within these publications the ensemble error is written as:

$$\varepsilon(x) = \bar{\varepsilon}(x) - \bar{a}(x) \tag{6.11}$$

where,

$$\bar{\varepsilon}(x) = \sum_k w_k \varepsilon_k(x) \tag{6.12}$$

which is the average error, and,

$$\bar{a}(x) = \sum_k w_k a_k(x) \tag{6.13}$$

is the average of their ambiguities, which as equation (6.10) shows, is the variance of the output over the ensemble. When averaged over the input distribution, and therefore $y(x)$, the following generalisation is suggested:

$$\varepsilon = \overline{\varepsilon} - \overline{a} \tag{6.14}$$

Hence Sollich and Krough separate the generalisation error of the average of the ensemble into two terms, the weighted average of the generalisation errors of the individual predictors and the weighted average of the ambiguities. The relation therefore shows that the more the predictors differ, the lower the error will be, provided that the individual errors remain constant.

Much of the literature concentrates on methods for generating ensembles which provide a low generalisation error, but whose individual members have errors which are uncorrelated for the reasons described by equation (6.14).

Varying training data is one method whereby a set of uncorrelated error networks can be generated. Krough and Vedelsby[91] explored the possibility where a no-overlap technique of training and test set selection was made for an ensemble. They therefore considered the average test set error for generalisation purposes, which resulted in a fixed test set. They claim that this not only generates diverse networks but also allows the entire data set to be used for training and testing, in a similar manner to cross validation techniques.

Further experimentation where individual networks of an ensemble were trained on randomly selected data is given in a paper by Opitz and Maclin[92]. This was actually applied to a classification problem, but the methodology is also applicable to regression problems. This paper terms the random selection of data sets (which may be overlapping) as 'bagging', and also examines the performance obtained from another technique which is termed 'boosting'. When training using the

boosting technique, the probability of points being selected for the next ensemble member is increased for points which yielded high errors in the previous model's training set. In this way it is claimed that the performance of the overall ensemble is improved by boosting the representation of points which have a high training error. Although the basic philosophy behind this technique seems sensible, the technique raises some concerns when one also considers that data points with high residuals also tend to be faulty outliers. One would not wish to train a model based on such points, as their existence in future data may be unpredictable. The author confirms this suspicion when noting that the technique of bagging is probably appropriate for most processes, but that the boosting may produce improved results on some data sets. Drucker et al[93] also examined the boosting technique, by comparing it to the bagging method. The computational cost of training the networks was also taken into account on an OCR (Optical Character Recognition) problem as part of the evaluation. It was found that for a large data set, some form of boosting was best.

The next group of papers all concentrate on techniques which modify the cost function of the individual neural networks, to incorporate a term which encourages decorrelation of network errors. The first of these papers, by Rosen[94]
considered a sequential co-operation between the networks, where, after an initial network was trained, each subsequent network had to obtain an optimal performance not only in terms of error but also in terms of decorrelation in training errors. This scheme has been successfully demonstrated with reference to a noisy sine function and another one-dimensional non-linear function, and appears to provide improved performance over ensembles that comprise independently trained networks. The author does suggest that this method is most advantageous when there is insufficient data to train each individual network on a disjoint subset of training patterns.

One problem with this approach, however, is that if a term is added to the standard sum-of squares cost function to penalise correlation in errors between networks, then the balance between this and the 'error' part of the function is critical. A term was added to the modified cost function presented in the paper which, the author explains, did need some adjustment to prevent very decorrelated networks being produced, which have a high error. Liu and Yao[95] also investigated learning with decorellation, but this time performed the training of the networks simultaneously, they term this idea CELS (Co-operative Ensemble Learning System). The authors suggest that with simultaneous training, an opportunity is provided for networks to co-operate and specialise. One important result shown through experimentation was that an ensemble generated with independently trained networks produced an ensemble prediction with reduced variance, however the bias was unaffected. However when training with the CELS method, the authors have shown a reduction in bias as well. The problem of choosing the amount of importance of decorrelation in respect of each networks training is, however, still an issue. The authors would also like to resolve problems such as varying the architecture of individual networks, so as to provide an optimum set of nets, as well as dynamically determining the number of ensemble members needed. The final paper in this group of 'active' decorrelation papers is by Opitz and Shavlik[96], who detail a method of ensemble generation that uses genetic algorithms to search explicitly for a highly diverse set of accurate trained networks. This works by generating an initial population and then testing the diversity of them, they are then arranged in order of a fitness function consisting of accuracy and diversity as entries. Then a genetic algorithm forms new candidate networks by making changes to the topology of existing candidates based on the present population. When combined with expert knowledge, this algorithm was shown to provide improvements over the bagging method.

Two papers relating to the ensemble technique in a different way are by Hansen et al[97] and Hasse-Sorrensen et al[98]. These papers are primarily concerned with a technique called LULOO (Linear Unlearning Leave One Out). This method has been developed to overcome the computational expense required to train using the leave-one-out cross validation procedure on larger data sets. Instead it is assumed that the unlearning of a single example only affects the network weights slightly, and under this hypothesis they use approximation techniques to estimate the change of network parameters. It was also shown in the earlier paper that if the networks produced from the cross-validation technique were pooled into an ensemble with linear combinations, the generalisation performance is identical to that of a single network trained on the full set of data.

The review also shows that weights of the linear combination of each component network can be determined by several methods. Sharkey's review on ensemble networks[89] showed that very often, equal contributions of each network can be used to establish the network output. Krough and Vedelsby[91], however, found the optimal combinations of network weights using gradient descent minimisation. In a similar manner, Igelnik et al[99] used least squares regression to solve the problem of optimal weight calculation.

Several authors have also postulated that regularisation methods, when applied to each component network, may not be appropriate. Naftaly et al[100] suggest that through 'overtraining' a network a smaller bias but larger variance is obtained in an individual network's prediction. However, when combined as a standard ensemble, the bias of the ensemble is unaffected, but the variance is reduced. In this way, it is suggested that overtraining might be beneficial. Sollich and Krough[90] also considered this phenomenon but suggest its use in large ensembles.

Naftaly et al[100] also showed that even through varying just initial conditions, a significant reduction in the variance of the ensemble could be seen.

Having made the review of the techniques of combining neural networks, and investigated the reviews of Sharkey, it was concluded that deciding upon the method of combination may be quite data set dependant in relation to its size, quality and target function. Ueda[101] showed that under some situations, combining neural network estimators does not always increase the generalisation performance – this varies depending on the data set.

It was therefore proposed to combine networks using a variety of methods, in order to establish a technique suitable for the heat treatment data.

6.2.5.1 Methods of combining neural network models

Within this section, experimentation into both modular and ensemble approaches is described. The results were evaluated on a fixed test set to enable an evaluation to be made into the best technique for the final application. The best technique may not however end up as the most accurate, since retraining and implementation of the models also had to be considered.

6.2.5.1.1 Modular decomposition of steel data

The first idea was to investigate the effects of performing the modular approach on the full, cleaned data set, with two methods of decomposition: site and composition based. Previously, as the data set was expanded, aspects of site-based decomposition were investigated, however at that stage, decomposition resulted in small data sets and was therefore thought not to be useful, and the results of modular decomposition were not investigated across a data set of varying sites.

Composition based decomposition was considered in the sense that data sets had been restricted to steels pertaining to a martensitic microstructure after the tempering stage, however this did not relate explicitly to the types of steel present in the data set. One problem with this second method was that, although a compositional analysis was provided with each example in the data set, there was no reference to the 'family' of steels to which the example belonged, for example, a CrMo (chrome-molybdenum) steel or a NiCr (nickel-chrome) steel. It was concluded from metallurgists that the 'family' to which the example belonged would influence the steel's mechanical property behaviour extensively. By decomposing by 'families' of steels the idea was therefore to allow each network to specialise on a different type of steel. The problem of allocating a general steel type to each example was overcome by the knowledge that each example was allocated a composition code, this code in itself, however did not directly relate to a generic type of steel. Instead, a unique list of composition codes were generated, for steels within the database, which was then passed to a metallurgist for expert interpretation, and the grouping of the codes into families of steels. The codes have a pre-defined alphanumeric format, which relates to the most important alloy additions within the steel. Once a 'family name' was allocated to a set of composition codes, each example within the cleaned data set was allocated a number based on the expert information obtained, to allocate it to a family of steels. Initially, the groupings proposed by the expert were quite detailed, for example 1% CrMo, 2%CrMo and 3%CrMo (where the percentage represents the chromium addition), instead of just CrMo steels. Then, some groupings were merged so a to allow variation within the data set. This amalgamation of some sub groupings resulted in 15 families of steel, shown in Table 6.3.

Category	Group Number
Cr	10
CrMoV	11
CrMoAl	12
CrMo	13
CrV	14
C	20
Mn	30
MnMo	31
MnV	32
MnCr	33
MnNiCrMo	34
MnNiMo	35
NiCr	40
NiCrMo	41
NiCrMoV	42
SiCr	50

Table 6.3. Categorisation of examples within the cleaned data set based on steel 'family'

Having categorised the steels as in Table 6.3 it was realised by the metallurgists that the SiCr steel, of which there was only one example, was in fact not a suitable type to include in the data set, and so it was removed from the cleaned data sets.

For each approach of modular decomposition (site and composition based), the data sets were then constructed so as to enable a comparison between the two methods. A test set was separated from the cleaned data containing 1/3 of the examples present. This enabled the performance of each decomposition method to be compared on the same data. The remaining data were then used for training and validation of the individual models in the modular approach. The number of data points present for each site is shown in Table 6.4.

Site	Number of Training/Validation Set Examples
Pearsons	15
Whithams	314
Lab	1195
Special Steels	516
Roundwood	202
West Bank	1517

Table 6.4. The number of examples present for training and validation in each model of site

Despite the increased number of data samples within the full, cleaned data set, the number of examples present for the Pearsons model was still very small. Although the test set and validation set selection was random, a check was made to ensure that a proportional number of examples related to each model. However, it was decided that models would be constructed for all sites, to enable the overall performance to be evaluated. The models for each site were trained using the inputs of size, depth, C, Si, Mn, S, Cr, Mo, Ni, Al, V, hardening temperature, tempering temperature, hardening Quench Type and UTS as the output.

Each network was trained using the Levenberg-Marquart algorithm, with 10 hidden layer neurons, using the validation set to enable early stopping (to prevent overfitting). Each model was constructed ten times with different random weight values, to enable an average performance to be assessed. The average standard deviation of the residuals for each model of each site is shown in Table 6.5. for the model validation and test sets.

Site	Model SD of residual	Validation SD of residual	Test SD of residual
Pearsons	0.01	76.8	48.6
Whithams	18.11	32.2	30.8
WestBank	21.8	26.7	28.6
Special Steels	22.4	29.1	34.5
Roundwood	16.0	49.1	34.9
Lab	28.2	42.5	41.4

Table 6.5. The model, validation and test results for each site's model.

It can be seen that the worst results relate to the Pearsons model, which has an unreasonably low number of examples, meaning that the model set obtains an outstanding accuracy, which does not generalise well. The best accuracy and generalisation is obtained with the modelling of the West Bank data.

The overall performance of the multiple models on the test set was evaluated, by 'gating' the test set input to the appropriate model based on the example's site code. This was performed for each of the 10 random initialisations and a SD of residual of 34.3 was obtained.

When the data sets for the second method of modular decomposition were constructed it was realised that, once again, the number of examples in some categories was insufficient to train a suitable network, even if cross-validation techniques were employed. And, so, as the numbering of the groups in Table 6.3 may suggest, the 'families' were further amalgamated into groups relating to the main alloy addition of the steel. With the removal of the SiCr steel as mentioned earlier, this resulted in four categories of steel, namely Cr, C, Mn and nickel based.
The number of training and validation points relating to each composition type is shown in Table 6.6.

Composition Category	Number of Training / Validation set
Cr	1963
C	176
Mn	97
Ni	1524

Table 6.6. The number of examples for training and validation in each composition category

This table shows the uneven distribution of the data. However, despite composition categorisation, the distribution in relation to the hyperspace represented by the input variables of the model may not be so uneven: this will be further investigated in chapter 7.

The model for each composition category was trained ten times with different random weight initialisations, to enable an average result to be obtained. Initially this was performed with a network containing 10 hidden layer neurons, using the

Levenberg-Marquart algorithm. The validation set was used as a means of regularisation with the early stopping criterion.

The inputs to the individual models were the same as with the site based models, but with the addition of the 6 variable binary site codes since this was no longer part of the decomposition. The average results for each model type for the training, validation and test sets are shown in Table 6.7.

Composition Model	Training SD of residual	Validation SD of residual	Test SD of residual
Cr	34.0	37.6	37.6
C	12.8	18.8	24.8
Mn	28.8	52.1	54.3
Ni	30.3	38.7	37.9

Table 6.7 Results of modular, composition based decomposition approach for average individual model.

It can be seen that the worst generalisation once again relates to the model based on the smallest number of training examples. However the best model performance does not relate to the composition category with the most examples. The implementation of this method is more difficult than the site-based model since, when a previously unseen input is applied to the model, one will need to somehow direct the input to the appropriate model, based on which composition type it is closest to. This is not such a problem with the site-based method because one will know which site the proposed steel was treated at. However with composition, a composition code may not have been generated at the experimentation stage which the model was being designed for. It was therefore expected that, if this method proved to be the most successful, some sort of distance function (for example the Euclidean distance) could be used to work out which steel model (or combination of models) the unknown examples' mechanical properties should be calculated from. More research into this particular subject area is considered in chapter 7, in relation to a slightly different problem area associated with the project.

At this prototype stage the composition codes were known for the test set examples and so a simple algorithm was used to pass the examples to the appropriate model. The average, overall test set error obtained was a SD of residual of 33.8, this is slightly more accurate than the site based approach.

It was considered that with both methods of modular decomposition a less complex model, containing fewer hidden layer neurons, may be sufficient or indeed more accurate, since the decomposition may mean that each sub model was performing a simpler task. However, through experimentation this was proven not to be the case with 10 hidden layers being the optimal amount for both setups.

6.2.5.1.2. Ensemble modelling of steel data

The next experimentation to investigate the effect of combining neural models with the cleaned steel data was to consider the ensemble approach. As was seen from the literature review (section 6.2.5), this does not involve decomposition of the problem domain, but involves generating diversity in the component networks.

It was decided that 3 different methods of generating ensemble diversity would be investigated.

1 Varying the initial conditions for a fixed set of data.

2 Varying the training and validation set members, whilst also varying the initial conditions for each ensemble network.

3 Varying the architecture and initialisation conditions of the networks in the ensemble.

The same test set which was reserved for the testing of the modular network would be used so that the performance of the modular networks could be compared with that of the ensemble approach.

The data remaining after the extraction of the test set used for the modular approach were randomly organised into training and validation sets for the first experiment. Ten models, each with different random initialisation weights, were trained using the Levenberg-Marquart training algorithm. The variation in the training, validation and test set results obtained (over the 10 initialisations) is shown in Table 6.8.

Model	Training SD of residual	Validation SD of residual	Test SD of residual
1	29.6	39.8	45.8
2	28.9	40.8	42
3	28.9	40.1	40
4	28	37.3	39.5
5	28.4	39.2	41.4
6	29	37.2	37.1
7	29.5	38.4	36.9
8	28.3	39.9	39.4
9	30.4	40.3	40.1
10	28.3	38.4	37.4

Table 6.8. The SD of residuals for the training, validation and test sets of the 10 models created with different initialisation.

When predictions of each test sample were made using the mean of all of the ensemble members, the SD of residuals of the resulting predictions was 34.6. This is slightly less accurate than both the modular methods of decomposition that had SD of residuals of 33.8 for the composition approach and 34.3 for the site based approach. Note from Table 6.8 however, that this is more accurate than the most accurate test set predictions of any of the unitary models. Therefore the combination of the models has reduced the variance of the predicted values from the values measured. An investigation into how other statistical measures are affected will be shown in chapter 7.

Next, the experiment to determine the effect of varying the training and validation partition was carried out. The fixed testing data was removed as before, however, on this occasion, the examples present in the training and validation sets were randomly re-selected with each of the ten models in the ensemble, which were each initialised with different random weights.

The results of the training and validation sets are shown in Table 6.9 below.

Model	Training SD of residual	Validation SD of residual	Test SD of residual
1	30.9	40.5	41.7
2	30.9	36.6	39.5
3	33.0	38.2	42.4
4	32.6	41.6	43.3
5	31.3	37.1	40.3
6	34.2	40.2	40.7
7	32.1	39.9	40.0
8	33.6	39.8	42.3
9	35.0	38.5	41.8
10	29.6	37.1	41.2

Table 6.9. The SD of residuals for training and validation sets with different data selection and initialisation for each model.

The results show a greater variation in SD of residual than in Table 6.8 without the variation in training and validation sets. When the average of the ensemble of these networks was used to predict the test set values, a SD of residual of 36.1 was obtained. This is the worst accuracy obtained with any of the combination methods so far and, contrary to the literature review, shows poorer performance than the random initilisation alone.

The next experiment with the ensemble approach was to vary the architecture and the initialisation weights of the ten models. From knowledge of Figure 6.1, it was decided that the number of hidden layers would be varied from 9 to 18. This was done for a fixed training and validation set partition. The results of this experimentation for the training and validation sets are shown in Table 6.10.

Number of Hidden Layer Neurons	Training SD of residual	Validation SD of residual
9	32.1	36.2
10	36.8	43.0
11	31.6	37.9
12	30.3	39.4
13	34.9	43.9
14	36.6	44.4
15	30.2	38.1
16	31.2	38.3
17	34.1	41.9
18	26.7	38.6

Table 6.10. Results of ensemble members with hidden layer neurons varying between 9 –18 units.

One interesting point from the results in Table 6.10 is that, unlike the graph showing the effect of hidden layer neurons on MSE error, the training SD of the residual is lowest for the configuration with 18 hidden layer neurons. This may not actually be a genuine effect of network complexity however, since the error appears to fluctuate for all numbers of hidden layer neurons. This effect is almost certainly due to the effect of the random initialisation of each model and is why, when determining the number of hidden units required, one takes an average of several initialisations for the same configuration. When the average of the ensemble of these networks was used to predict the test set, a standard deviation of residual of 34.8 was obtained. This is very similar to the accuracy obtained for varying just the initial conditions.

6.2.5.1.3 Selection of an appropriate method for improving neural modelling performance

From the literature review it was evident that a variety of methods have been established for combining neural network predictors, the primary reason for doing this being to increase model accuracy. Some of these techniques have been

investigated with respect to the cleaned steel data, and their performance has been evaluated on a fixed test set. The modular approaches have used expert knowledge for task decomposition, and the ensemble approach has been attempted using some of the methods for diversity generation outlined by Sharkey[89]. Active decorrelation methods have not been investigated in this project, partly due to development time constraints, but also because the improvement in accuracy seen for any the ensemble methods, leaves the accuracy of the model very close to that set out in the project aims. Furthermore, it is envisaged that increasing the ensemble members would lead to a greater reduction in variance of the predicted values, however a collection of 10 networks was used for practical reasons.

When selecting an appropriate method to use for further model development it was decided that the accuracy of all the methods was a significant improvent on a single unitary network. Although the modular approaches were more accurate in terms of lower SD of residual, it was decided that the ensemble approaches were more appropriate for retraining and effective implementation. Because certain sites treat key grades, by decomposing a modular network by site it would restrict the compositional range that the model could predict for each site. Similarly there appeared to be a great deal of problems in effectively classifying new compositions to groupings set out in the compositional-based model approach. When deciding which approach to use, one has to ask if the model needs to cover a wide interpolatable range, or if an increased accuracy and lower interpolation ability would be more appropriate. Given the nature of the project brief and the small reduction in accuracy of the ensemble approaches it was therefore decided that the ensemble approach would be more appropriate.

The next stage was to decide which ensemble approach to use. There was very little variation in the ensemble methods results, apart from when the training

and validation data was varied, which may be a feature of the distribution of the data. Despite the multiple initialisation method not being the most accurate method (or the least) with respect to standard deviation of error, it is found that it carries the benefit of assessing the reliability of the predictions that are made as a function of data density. This development is further discussed in chapter 7. Therefore it was decided that the first ensemble method of multiple initialisations on a fixed data set would be used for model construction.

6.3 Developing models for the prediction of other mechanical test sets

Having established an optimal modelling technique and input data set using the UTS test results as an output, it was decided that further models could now be constructed to predict the other mechanical test results described in chapter 3. These additional test results are; impact energy, proof stress, reduction of area and elongation.

Separate models were constructed for each output variable. However, because not all types of mechanical test results were present for all examples in the data set, the number of training/validation and test examples varied depending on the test result modelled. When the inputs to each model were considered, expert knowledge to determine extra variables was gathered as will be described in the following sections.

During the data cleaning stage (chapter5), examples were tagged with a code to disclose whether they had been altered, removed or corrected. This information was now useful when constructing the new models for the additional output variables, since UTS results are present for all examples, and any faults in the inputs of the UTS model would therefore also affect the other output variables. Altered values present in the UTS data, which also corresponded to one of the new test

results, were therefore used. Examples removed due to un-correctable faults (such as process faults) were also removed, where present, from the data sets of the additional test results. This was under the assumption that these faults had affected the properties of the steels produced.

The prior data cleaning knowledge would remove faulty points relating to process errors and correct data where possible, however it was anticipated that there would still be an additional source of error relating specifically to the individual test results. For example, a process fault relating to a data sample manifesting itself with a high residual may only be highlighted through prediction of impact test results. It was also considered in chapter 3 that the mechanical test itself might also be a source of error. For this reason, after each data set had been cleaned with prior knowledge, an investigation was made to see if it should also be cleaned using the model-based method of outlier detection (investigating high residual values). A decision for this was made based on the accuracy level of the model after cleaning with existing knowledge and if used, was stopped if the proportion of faulty points found on a particular run of the model based detection method was very low. Repeated values were also removed from each respective data set, using the 'sames' method as described in chapter5. In this way both prior and data set specific knowledge was used to clean the data generated by additional test results.

The following sections now detail specific considerations, which were taken into account for the construction and modelling of each data set. These considerations included input selection and incompatibility problems that may be present between different test types on the same mechanical property. When modelling each data set, it was important that, although the process characteristics should be similar, the procedures for hidden layer selection should be followed, as initially described in section 6.2.1. The difference now, however, is that because the

ensemble approach is being utilised, it was felt that the optimum number of hidden layers used should be found for the MSE of the ensemble prediction. This is already averaged over 10 initialisations and therefore multiple runs of ensembles were not felt to be necessary.

6.3.1 Impact energy

The impact test results of a material encompass a variety of problems, due to the fact that, not only is the impact property highly non-linear in relation to the steel's composition and heat treatment regime, but the test types used may also be incompatible. The impact test was described in section 3.7.3, where it was mentioned that the results from different fracture profile and test depths, could not be compared or converted. Upon investigating the number of samples utilising each test type, it was decided that only the Charpy 2mmV samples would be used for data set construction, as these occupied the maximum number of samples. An explanation of these test conditions is presented in section 3.7.3.

Impact test results are logged in sets of three due to the variable nature of the test, for each set of results, the average value was taken as the impact result to be used for training and prediction. Additionally, two sets of results may be present for the same sample, due to customer requirements (for example a test at both room and sub-zero temperature may be required).

There were fewer impact results available than in the UTS data set, so when a second set of impact results had been obtained this information was entered into the data set as a new sample, by copying the input information and appending the average of the second result to it.

The next problem to be overcome was that, even though all the test types were now of type Charpy 2mmV, the units used to measure the impact energy were

in the two forms of absolute and backed-off. This problem is discussed in section 3.7.3, where equation 3.1. is presented to allow the backed off impact values to be converted into absolute ones. Having performed this conversion, the uncleaned data set consisted of 2042 examples. Data cleaning based on past knowledge (UTS sample faults) was then applied, together with investigation of max/min limits and removal of repeated values. It was found that the original uncleaned data had an impact energy range from 4-341 Joules, however practically no impact energies should exceed 300 Joules.

Specific outlier analysis then followed, with two iterations of the model-based detection method. In total, 60 points were examined and 23 specific errors were found. The final cleaned data consisted of 1748 examples.

The next stage was to determine if any changes should be made to the impact model compared with those needed for UTS model. After discussion with metallurgists it was decided that no other alloy addition inputs would be relevant, however the test temperature at which the impact test was performed should be included as an input.

The variables used for the construction of the impact model, together with the statistics of the cleaned data are shown in Table 6.11.

Variable Name	Type	Min.	Max.	Mean	SD
Test Depth	Input	5.5	69.8	16.8	6.59
Bar Size	Input	11	381	185.1	82.8
Treatment Site	Input	Binary codes represent 6 locations			
C	Input	0.12	0.52	0.39	0.064
Si	Input	0.11	0.38	0.260	0.032
Mn	Input	0.41	1.75	0.833	0.242
S	Input	0.0008	0.052	0.019	0.008
Cr	Input	0.1	3.25	1.081	0.295
Mo	Input	0.02	0.99	0.242	0.101
Ni	Input	0.03	4.21	0.428	0.567
Al	Input	0.003	0.047	0.027	0.004
V	Input	0.001	0.26	0.008	0.025
Temperature at Hardening Stage	Input	820	980	863.2	17.68
Type of Quench at Hardening Stage	Input	Binary codes represent 3 quenches; oil water or air.			
Temperature at Tempering Stage	Input	190	730	634.3	46.65
Impact Test Temperature	Input	-59	23	5.712	23.32
Impact Energy	Output	4	251.6	90.80	33.11

Table 6.11 Impact model data and variable content.

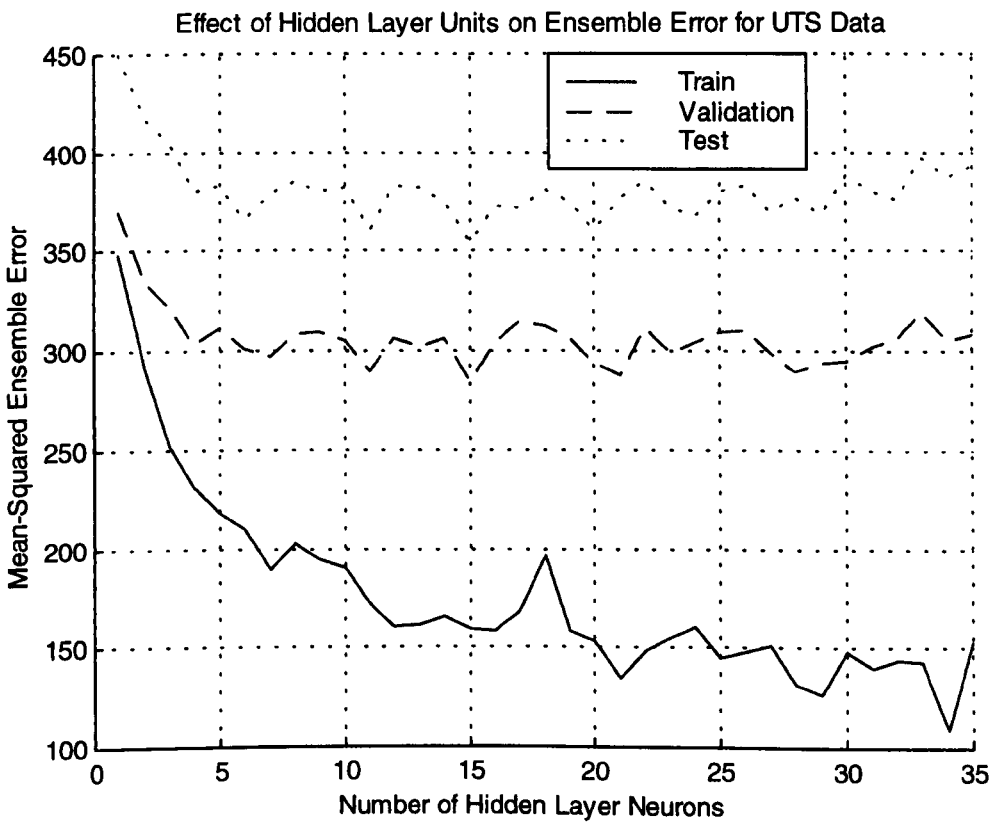


Fig. 6.3 The effect of hidden layer neurons on MSE ensemble error for Ch2mmV impact data.

The data was re-randomised and experimentation to determine the optimum number of hidden layers was performed. The number of hidden layers was varied from 1 to 35, and the MSE of the ensemble prediction was logged, these results being plotted and are shown in Figure 6.3. The first thing that was evident from the graph is that unlike Figure 6.1 (for the average effects ten predictors), there is an offset between the validation and the test set. This was thought to be due to the relatively small range of the variable predicted (as indicated in Table 6.11), and also the complexity and sparsity of the data set. It was anticipated that, because one is now looking at the ensemble error, the random initilisations of the ten component networks would be sufficient to remove fluctuations in the results due to network initilisations. However, it is evident from Figure 6.3 that this was not the case. It was thought that because the impact data was known to have a number of sparse regions, the effect of random intilisations would be greater. More networks may therefore be required to smooth the MSE error curve so as to make a decision on the number of hidden layer neurons used. The training time required, however, for 10 networks with 1 to 35 hidden layer neurons, was already 17 hours on a multiple processor Unix machine, and it was not felt that more extensive experimentation was feasible. Another factor influencing this decision was that the classical overfitting features were not shown in Figure 6.3 and 6.1. One would expect both for increasing training iterations and number of hidden layers, that the training error would continue to fall and the validation error to reach a turning point. It was thought that this could be due to the fact that the validation set was already preventing overfitting even with a high number of hidden layer neurons. Therefore, due to the randomness of the results, rather than using the graphs to pinpoint a specific number of hidden layers required, it was decided that they should instead be used to determine a reasonable number of

hidden layers needed to provide an accurate result. It was therefore decided that 11 hidden layer neurons should be sufficient. Note that although the test error curve is plotted for information in this thesis, it was not used to evaluate the number of hidden layer neurons required, as its purpose is to evaluate the performance of the final model on an 'unseen' data set. It is interesting, however, that even though the predictions of each ensemble are comprised of ten random initilisations, those points which have higher or lower error results (a spike in the curve), are carried through the validation and test sets. This means that when the error curve is flat as it is in this graph, that by selecting a low error validation point, one is probably as much selecting a good network due to random initialisation as hidden layer effect. The test results do, however, show that this produces a model network with a better generalisation error.

The standard deviation and r-square values of the final cleaned impact model and an impact model trained and validated on the original uncleaned data are shown in Table 6.12. The test result for both sets was selected from the clean data set as points that had not been deleted from the unclean data set as with the UTS example in chapter5.

Impact data set	**Training**		**Validation**		**Test**	
	SD of error	**RSQ**	**SD of error**	**RSQ**	**SD of error**	**RSQ**
Cleaned	13.2	0.84	17.1	0.74	18.3	0.71
Uncleaned	17.7	0.76	22.0	0.57	19.4	0.63

Table 6.12 SD of ensemble error and r-square values for impact models trained and validated on cleaned and uncleaned data.

The measured vs. predicted plots produced by these models on the cleaned test set are shown in Figure 6.4. It can be seen from these plots that once the data are cleaned, the fit of the predicted points is significantly improved. It was thought that the fit of the uncleaned model might have been significantly affected by faulty points

above 300 Joules in the original uncleaned data that were anticipated to have placed a skew on the distribution of the data. The graphs do however show that even with

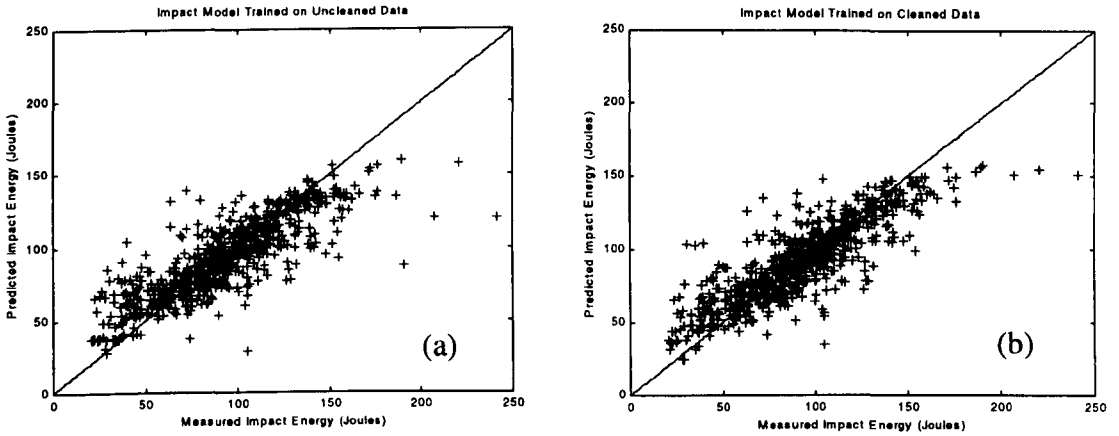


Fig 6.4. Measured vs. predicted plots for impact models trained on (a) unclean ed and (b) cleaned data.

the cleaned data, there is still a significant amount of spread about the Measured = Predicted line, this being a feature of the difficulties involved with modelling the impact data. The modelling of the highest impact properties is improved through data cleaning but is still not perfect. The r-square value of 0.71 for the cleaned model on the test set is, however, acceptable given the nature of the data.

6.3.2. Proof Stress

Proof Stress (PS) is a tensile test result as described in section 3.7.2. The Proof Stress result is obtained from the same test (and specimen) as with the UTS results. It was also explained in section 3.7.2 that for the BSES steels in the data set, there is no yield point and so the PS can be measured in a variety of ways, for example the 0.1, 0.2 and 0.5% stress from the stress-strain curve. As with the impact results this causes incompatibility problems in the data and so it was decided that only the 0.2% Proof Stress results would be modelled since these related to the majority of results. A 0.2 Proof Stress result was not available for all examples and

the original uncleaned proof stress data set with repeated values removed, consisted of 4015 results. Data cleaning based on existing knowledge was applied. However, having investigated one iteration of the model-based, specific outlier detection method, it was not felt necessary to perform additional data cleaning. The final cleaned data set therefore consisted of 4003 examples. The statistics of the cleaned data set are shown in Table 6.13. It should also be noted from Table 6.13 that the inputs to the proof stress model are identical to those used for the impact model except that the impact test temperature is no longer required.

Variable Name	Type	Min.	Max.	Mean	SD
Test Depth	Input	4.00	90.0	16.3	10.08
Bar Size	Input	8.00	381.0	156.9	87.13
Treatment Site	Input	Binary codes represent 6 locations			
C	Input	0.14	0.63	0.39	0.06
Si	Input	0.11	1.87	0.26	0.04
Mn	Input	0.35	1.72	0.75	0.17
S	Input	0.00	0.05	0.02	0.01
Cr	Input	0.05	3.25	1.02	0.41
Mo	Input	0.01	1.00	0.25	0.14
Ni	Input	0.02	4.18	0.76	0.86
Al	Input	0.01	1.08	0.03	0.05
V	Input	0.00	0.27	0.01	0.02
Temperature at Hardening Stage	Input	820.0	980.0	856.6	16.78
Type of Quench at Hardening Stage	Input	Binary codes represent 3 quenches; oil water or air.			
Temperature at Tempering Stage	Input	170.0	730.0	602.9	68.8
Proof Stress	Output	336.9	1418	760.0	178.6

Table 6.13 Proof Stress model data and variable content.

Experimentation into the number of hidden layer units needed to accurately model proof stress was then undertaken in the usual manner. The results from this experimentation can be seen in Figure 6.5. There is much less fluctuation in the results with the proof stress data than with the impact data shown in Figure 6.3. This

was thought to be attributable to the comparatively more uniform data distribution of the proof stress data (sparse areas would not be generated through test temperature as with impact results). It was decided that 11 hidden layer neurons should be used to model the proof stress data. As with the impact data there was not a significant rise in validation error at high numbers of hidden layer neurons and this was likewise thought to be attributed to the effective function of the validation set.

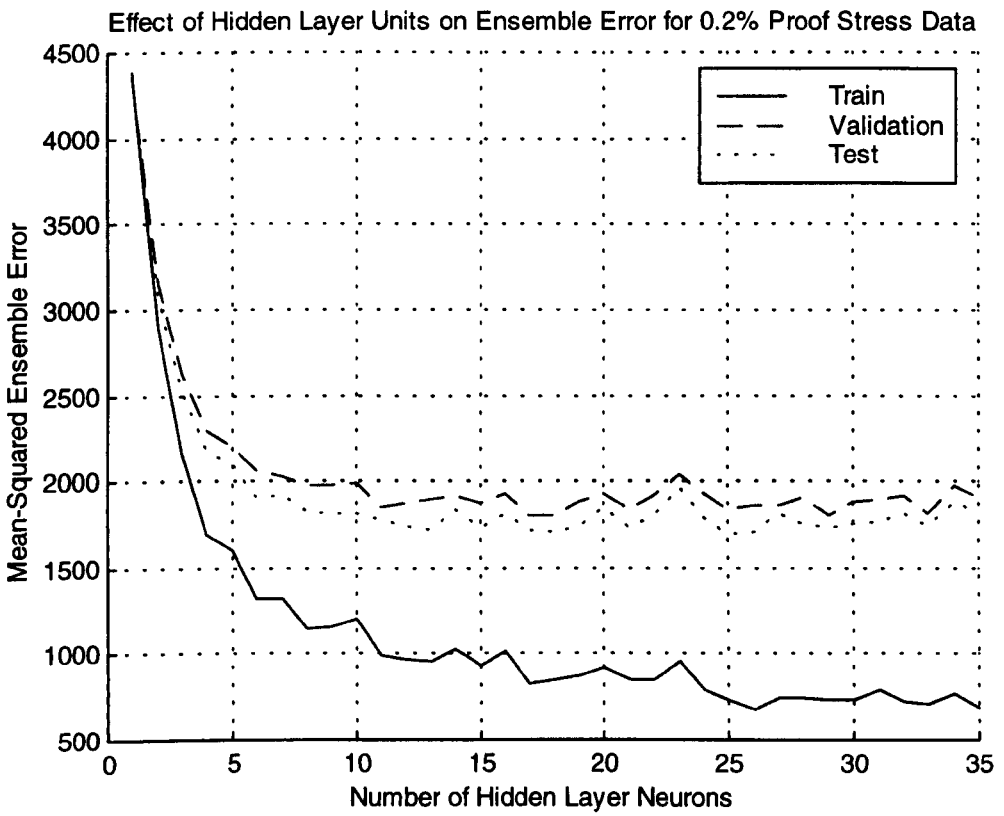


Fig 6.5 The effect of hidden layer neurons on MSE ensemble error for 0.2% proof stress data

The standard deviation and r-square statistics of models trained on clean and uncleaned proof stress data with 11 hidden layer neurons are shown in Table 6.14. These results once again show an improvement in the model trained on cleaned data compared with that trained on uncleaned data. This improvement is less dramatic than the UTS examples in chapter5, however, Figure 6.6 shows that the effect of this cleaning is noticeable on the test set predictions of both models, particularly around the 850 and 1200 N/mm^2 regions. It should also be remembered that this

improvement in prediction accuracy has been achieved solely with past data cleaning knowledge obtained from the UTS data set.

0.2% PS data set	Training		Validation		Test	
	SD of error	RSQ	SD of error	RSQ	SD of error	RSQ
Cleaned	32.6	0.97	43.6	0.94	41.4	0.95
Uncleaned	33.9	0.96	46.2	0.93	43.5	0.94

Table 6.14 SD of ensemble error and r-square values for proof stress models trained and validated on cleaned and uncleaned data.

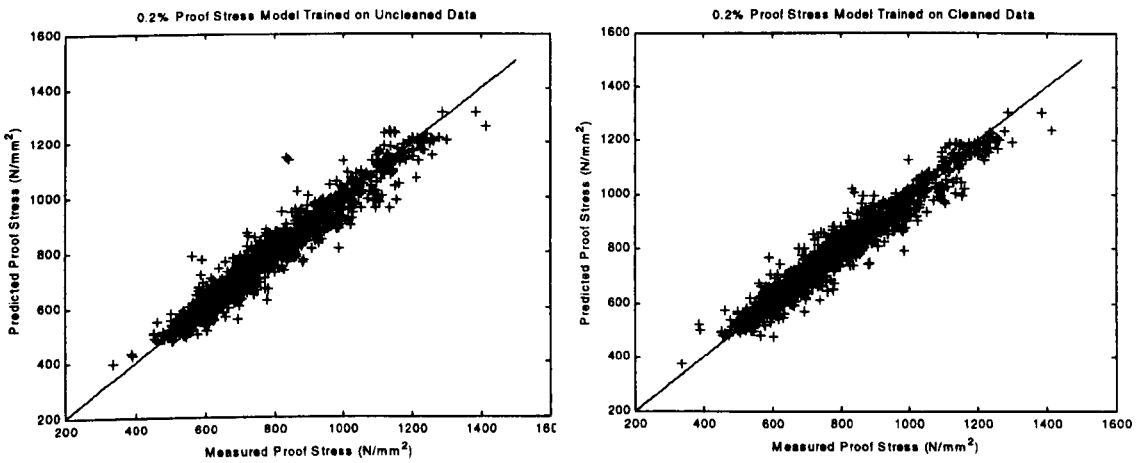


Fig. 6.6 Measured vs. predicted plots of proof stress models trained on (a) uncleaned and (b) cleaned data.

6.3.3 Reduction of Area

Reduction of area (ROA) is part of the tensile testing procedure described in section 3.7.2. The reduction of area is measured as the percentage change in the diameter of the specimen after fracture. The original uncleaned data set consisted of 5573 data points (with repeated values removed). Prior cleaning knowledge was then applied, followed by one run of the model-based detection method, where 11 examples were removed and 1 was corrected. These outliers tended to have low values. The final cleaned data set consisted of 5559 examples.

It was felt that the existing input variables would be sufficient for the modelling of ROA. The statistics of the cleaned data set for the variables used are shown in Table 6.15.

Variable Name	Type	Min.	Max.	Mean	SD
Test Depth	Input	4.0	140.0	16.2	9.4
Bar Size	Input	8.0	381.0	156.8	84.4
Treatment Site	Input	Binary codes represent 6 locations			
C	Input	0.12	0.63	0.39	0.07
Si	Input	0.11	1.87	0.26	0.04
Mn	Input	0.35	1.75	0.77	0.22
S	Input	0.00	0.21	0.02	0.01
Cr	Input	0.05	3.46	1.04	0.45
Mo	Input	0.01	1.00	0.26	0.14
Ni	Input	0.02	4.21	0.78	0.86
Al	Input	0.01	1.08	0.04	0.09
V	Input	0.00	0.27	0.01	0.02
Temperature at Hardening Stage	Input	820.0	980.0	857.1	17.02
Type of Quench at Hardening Stage	Input	Binary codes represent 3 quenches; oil water or air.			
Temperature at Tempering Stage	Input	20	730	604.9	71.54
% Reduction of Area	Output	21.8	79.4	60.4	6.42

Table 6.15 Statistics of the cleaned data set for the modelling of Reduction of Area.

Having cleaned the data set, experimentation to determine an appropriate number of hidden layer neurons was performed. Figure 6.7 shows the effect of the number of hidden layer neurons on the mean squared ensemble error. It can be seen that a minima in the validation error is reached later than in the other data sets, and it was therefore decided that 13 hidden layer neurons would be used to train the ROA model. This graph also has a greater gap between the validation error and the test set error, indicating that the model does not generalise as well to an unseen data set. The results of this training, validation and testing for models trained and validated on clean and unclean data can be seen in Table 6.16.

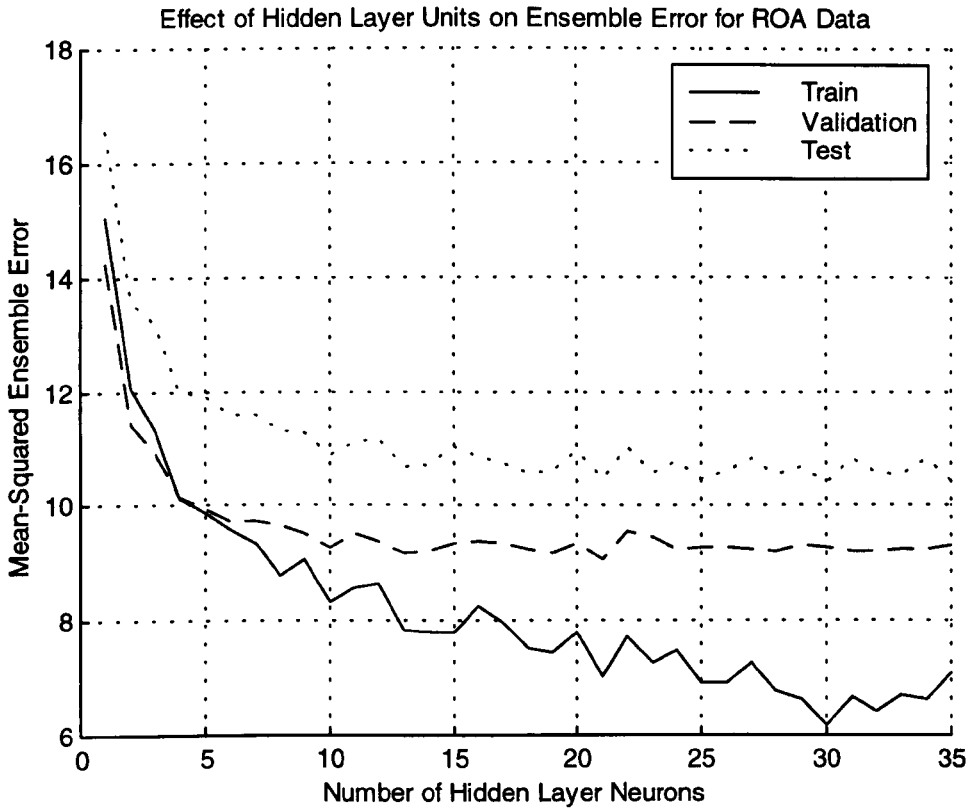


Fig. 6.7 Effect of hidden layer neurons on the mean-squared ensemble error of the ROA model.

ROA data set	Training		Validation		Test	
	SD of error	RSQ	SD of error	RSQ	SD of error	RSQ
Cleaned	2.82	0.828	3.05	0.758	3.27	0.748
Uncleaned	2.79	0.827	3.29	0.741	3.34	0.734

Table 6.16 SD of ensemble error and r-square values for reduction of area models trained and validated on cleaned and uncleaned data.

It appears that the data cleaning has not had much effect on the reduction of area model and in the case of the training data the uncleaned model is actually more accurate than the cleaned one. The valiation and test sets do, however, show an improvement in accuracy. A possible explanation for this is that the training data may have either fitted some faulty data, or that the random data set selection has resulted in mainly clean data in the training set.

The measured vs. predicted plots generated for the test set predictions of the models trained on cleaned and uncleaned data can be seen in Figure 6.8. These do show a slight improvement in the distribution of the points for the cleaned predictions. What is most striking, however is that the low value ROA points are modelled poorly, even though they have been checked in the cleaned data set for accuracy. The distribution of the ROA is obviously skewed towards the higher values, and this was thought to have affected the accuracy of the low-end values. It was also thought that the effect of the data cleaning was less noticeable given the sparse area around the low values of ROA and the poorer fit generally of the model.

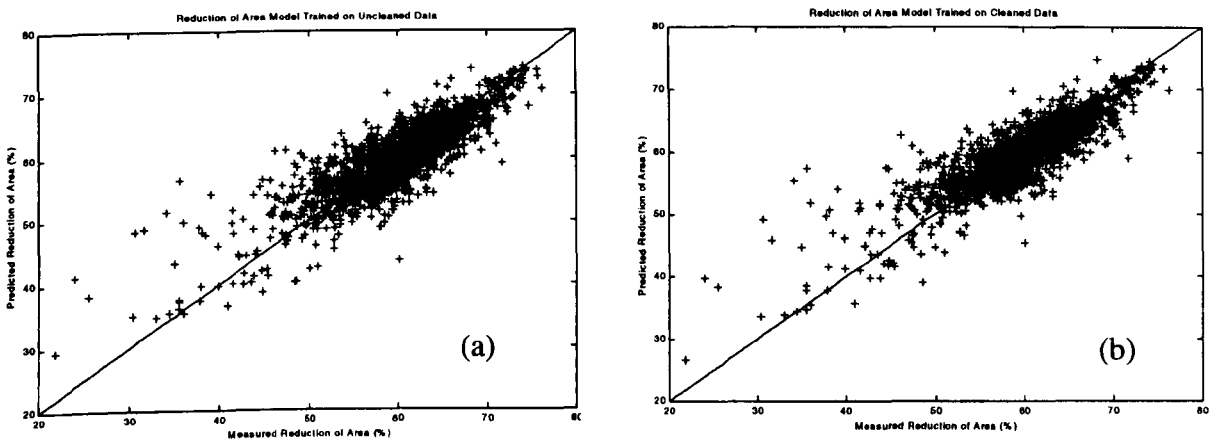


Fig. 6.8 Measured vs. predicted plots for ROA models trained on (a) unclean and (b) cleaned data.

6.3.4 Elongation

Elongation is the final property derived from the tensile test, described in section 3.7.2, it being the percentage change in gauge length after fracture. Unlike reduction of area, however, it is dependent on the gauge length used for the specimen, which may be defined as either 4 or 5 times the diameter of the specimen. The specimens used for data collection in this project are all of identical diameter, however the gauge length does vary. It is not possible to convert a result from one gauge length to another. However, it was decided that there were two possible

procedures for modelling the elongation results. The first was to only model 4 or 5 D results. The second method was to include an additional input to enable both gauge lengths to be modelled, with the hope that information from one gauge length may also benefit the other. Even though the measurements are not compatible, it was anticipated that a 'mapping integral to the network' might be generated as with the site code, so as to benefit the overall model from the acquisition of more data.

Before experimentation into the best method to use proceeded, the data cleaning approach was applied. The uncleaned data set covering both gauge lengths (but with repeated values removed) consisted of 5710 data points. Data cleaning based on existing knowledge was then performed, followed by one run of the model-based detection method, where 3 points were removed and 2 were corrected.

Specimens with a gauge length of 5D are most common in the elongation data set and so the effect of modelling a data set comprising of only the 5D specimens (4623 examples) and a data set comprising of both specimen lengths (but with an additional input) was investigated.

It is difficult to compare the performances of the two models, because one cannot use identical data sets for training, validation and testing, when one is trying to compare the use of additional data to the model. However, it was felt that the experiment did provide a guide to the performance change of modelling 4&5 D specimens together. The results of the two modelling approaches can be seen in Table 6.17, and relate to ensemble predictions where each component network has a nominal number of 10 hidden layer neurons.

Data set type	Training		Validation		Test	
	SD of error	RSQ	SD of error	RSQ	SD of error	RSQ
4D&5D data	1.45	0.79	1.37	0.79	1.73	0.75
Just 4D data	1.27	0.74	1.3	0.71	1.72	0.62

Table 6.17. Results of modelling elongation data with and without mixed gauge length specimens.

The results show that the SD of residual for the combined gauge length model is higher (model is less accurate), however the r-square values are also higher (model shows a stronger correlation between its measured and predicted points). This may not be entirely relevant, because the data sets are obviously different, however the experiment has demonstrated that there is not a significant performance degradation from modelling both data gauge lengths, and suggests that it may benefit the overall fit of the model. Given the aim of the project, it was decided that the combined model would be developed further.

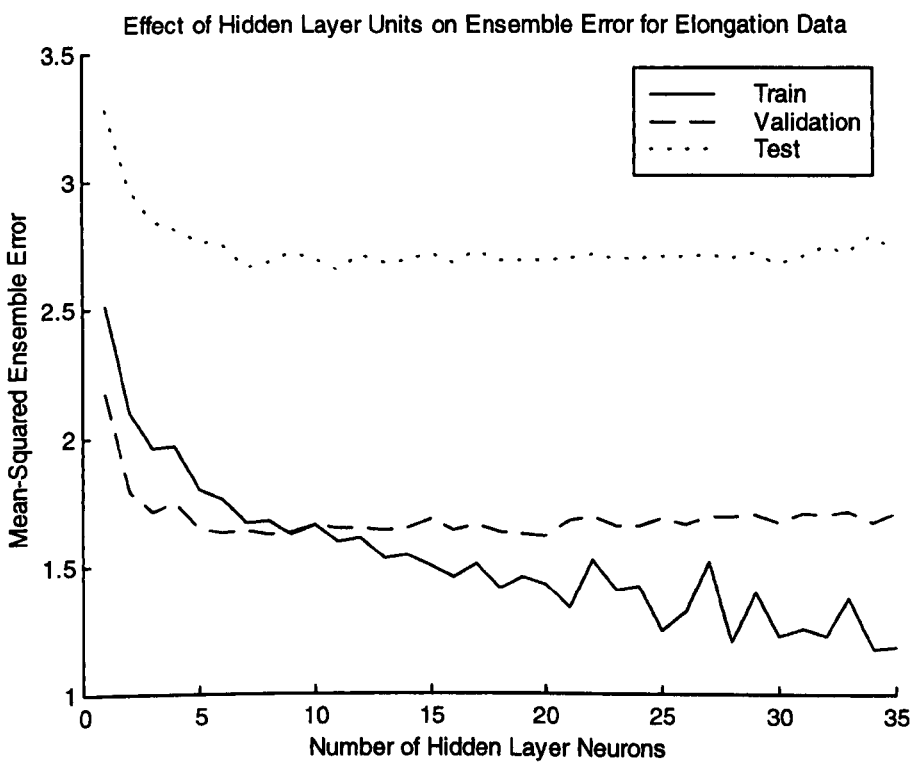


Fig 6.9 Effect of hidden layer neurons on the mean-squared ensemble error for elongation models

Experimentation was therefore conducted to determine the optimum number of hidden layer neurons needed to model the combined guage length data set. The results of this experimentation can be seen in Figure 6.9. It can be seen from Figure 6.9 that 9 hidden layer neurons appear to be an appropriate number to use. An elongation ensemble model was therefore constructed using this architecture and the results for this and a model trained and validated on unclean data can be seen in Table

6.18.

Elongation data set type	Training		Validation		Test	
	SD of error	RSQ	SD of error	RSQ	SD of error	RSQ
Cleaned	1.44	0.81	1.38	0.81	1.61	0.76
Uncleaned	1.39	0.85	1.53	0.79	1.61	0.76

Table 6.18. Results of modelling elongation data with cleaned and uncleaned data.

These results are similar to those for ROA (Table 6.16) in that for the training set the accuracy of the model actually appears to be better for the uncleaned data. The validation set shows an improvement for the cleaned data, however the test set shows no change, this being confirmed from the measured vs. predicted test set graphs in Figure 6.10.

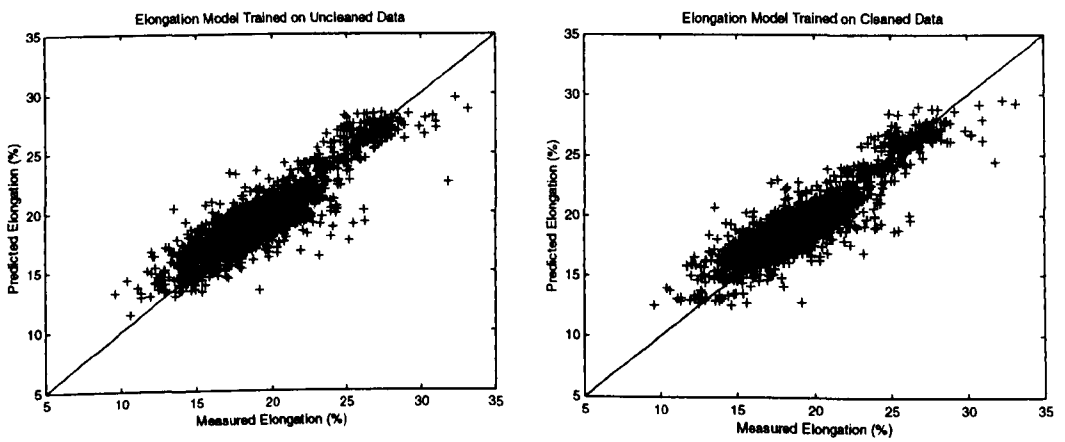


Fig. 6.10 Measured vs. predicted plots for elongation models trained on (a) unclean and (b) cleaned data

These results were disappointing, however it was thought to be due to the small number of points affected by the specific cleaning process. For this data set and indeed the Proof Stress and Reduction of Area data, there were no 'catastrophic' faults such as with the UTS data, and therefore it was expected that the cleaning process would have less effect. The effect of the cleaning process on these additional test results has, however, been evident from the validation set in all cases, and was still felt to be a significant benefit in most cases.

6.4 Effect of ensemble error on UTS hidden layer units decision

Although the development of a UTS ensemble model was detailed earlier in this chapter, the decision as to the appropriate number of hidden layers was based on the average predictions of 10 individual networks. It was therefore decided that the hidden layer experiment should be repeated with the mean-square prediction error of the ensemble, rather than the average of the individual networks, to see if a different decision area results.

A graph showing the effect of hidden layer units on the mean-squared ensemble prediction error was obtained as with the other test results in this chapter, and can be seen in Figure 6.11.

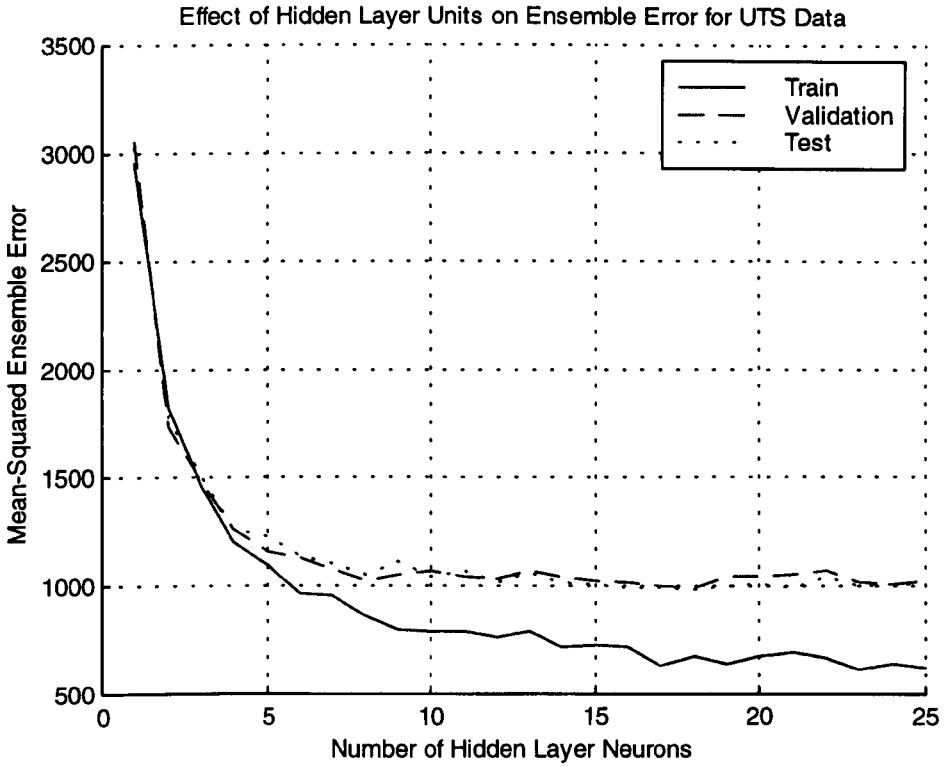


Fig 6.11 The effect of hidden layer neurons on the mean-squared ensemble prediction error.

The first noticeable difference between Figure 6.11 and the original graph (Figure 6.1, (pp. 175)) showing the average of 10 network's predictions is that the minimum mean-squared error is lower with the ensemble. Additionally, the error curve is also smoother; there is less variance in the predictions based on random initialisation. The plot seems to still indicate that 10 hidden layer neurons are appropriate. One interesting feature of this experiment was, however, that when the UTS model was re-trained with a new set of initialisation weights for each component network, but with the same architecture as before, a similar test set SD of residuals was obtained to that in section 6.2.5.1.2. The results of this, together with ensemble statistics for the training and validation sets are shown in Table 6.19. Uncleaned performance has not been included here since this has already been shown in chapter 5.

UTS MODEL	Training		Validation		Test	
	SD of error	RSQ	SD of error	RSQ	SD of error	RSQ
Cleaned	27.8	0.97	33.2	0.96	34.4	0.95

Table 6.19 Statistics obtained from re-modelling UTS data with 10 hidden layer neurons.

The measured vs. predicted plot for this final version of the UTS model is shown in Figure 6.12.

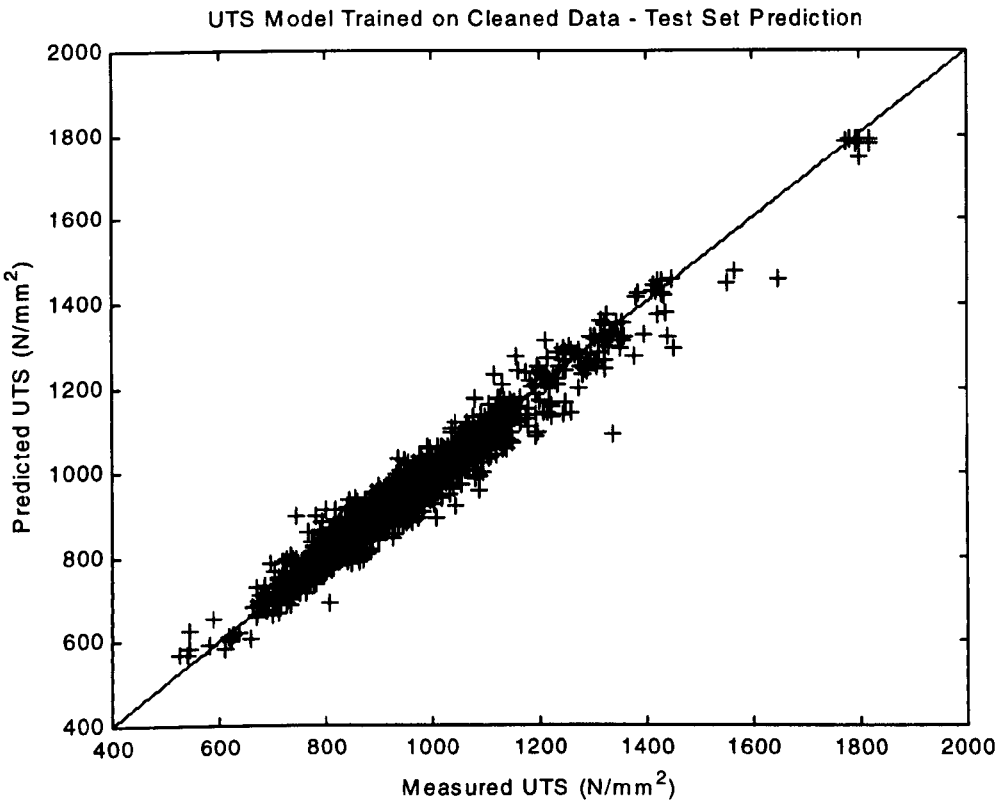


Fig. 6.12 A measured vs. predicted plot showing the test set predictions of the final UTS model

6.4 Chapter conclusion

This chapter initially investigated some methods for improving the accuracy of the neural network technique. Initially, training parameters such as the number of hidden layer units had been kept constant in an attempt to compare the effects of data set changes, however, experimentation into ways of improving training was now

performed. Experimentation to determine the optimum number of hidden layer neurons was initially performed using the average prediction error from a set of 10 networks. This was used to obtain the minimum number of hidden layers needed to accurately model the UTS data initially, given that a more complex network architecture may over fit the data and generalise less well. Improvements in the training algorithms were then investigated, and the Levenberg-Marquart algorithm was utilised for its increased speed of convergence - this did however have a memory overhead. The effect of random initialisations on the network accuracy were then investigated and shown to be significant, therefore it was decided that for reliable predictions, network predictions should be combined. Research into possible ways of combining neural networks was then made, and revealed that two main methods existed – ensemble and modular techniques. Experimentation with these techniques was carried out, in several forms, which for the modular techniques utilised expert knowledge of the steel types described by the data. The results of these experiments showed that although one form of the modular technique was slightly more accurate, it was practically restrictive in terms of usage. Therefore, an ensemble technique which makes its predictions based on ten networks which are given separate random initialisations was used. This technique also has benefits in terms of prediction reliability assessment, which will be discussed in the following chapter.

These improved modelling techniques were then applied to the other test results available in the original data set, which are impact, proof stress, reduction of area and elongation. Investigation into network architecture and input variable selection have been made in each case, however given that the ensemble architecture was being adopted, the ensemble prediction error was utilised instead of the mean of the individual network predictions. Data cleaning knowledge gained from the UTS data set has been applied to these additional test results based on the Altered,

Corrected and Removed coding scheme developed in chapter 5. Additionally, where appropriate, model based data cleaning has been performed to find errors specific to each test. The effect of this data cleaning was noticeable for the impact and proof stress data sets, but was less significant for the reduction of area data set and made little difference for the elongation set. It was felt that the quantity and severity of outliers in the data set affected the 'impact' of the data cleaning technique. Additionally, unlike UTS, these additional test results were generally not modelled with a high r-square statistic, and therefore the effect of faulty points may have been less significant. The data cleaning technique was, however, felt to have been beneficial when applied in this manner to the additional output variables.

The next chapter now investigates methods of exploring the function of the neural network. This is applied to each model developed so as to validate the effect of each variable from a metallurgical perspective, rather than a statistical one. The development of a Graphical User Interface (GUI) is then detailed, allowing the UTS model to be evaluated in its industrial context. Feedback from this is then discussed and further improvements to the GUI are described. One important issue in an industrial context is that the model should be safe and dependable, but black box models are notorious for producing spurious predictions under certain conditions, and so a technique to 'supervise' the function of the model is developed.

Chapter 7

Information extraction from the neural model

7.1 Introduction

The work in this chapter investigated techniques for extracting information from the models that have been constructed in the previous chapter. Initially some techniques that allow an insight into the function of the model were investigated. These include a method for determining the sensitivity of the output of each model to its inputs, and a method for visualising the effect each input has on each output over the range covered by the model. Once again, expert knowledge was used to allow this experimentation to provide useful results.

Results from analysis of the models are presented and then work focuses on how the models might be used in an industrial context. Research into potential users' needs led to the development of a Graphical User Interface (GUI), which allows users to make predictions and investigate the effects of variables within the UTS model.

An evaluation version of the graphical user interface was then released into the industrial situation for which it was designed, to enable feedback on possible improvements. This feedback resulted in a series of suggested improvements to make to the GUI.

One improvement that was very important was that there is a need to determine the reliability of a prediction produced by the neural model. This is crucial to a neural model's usage within an industrial context, particularly when the input space of the model has a non-uniform distribution as with this project. Research into several methods for performing this was undertaken, however a spin-off from the ensemble technique proved to provide the best solution.

The benefits of this reliability indicator are then demonstrated, and opinions of metallurgists sought to ensure that the information it provides is representative of the problem domain.

An additional use of this technique was then investigated, which aims to aid someone constructing a model to determine the quantity and location within the input space of additional data which might be required to make the model more reliable.

7.2. Methods of model analysis

This section looks at two methods of model analysis, which were utilised in the project for assessing the functionality of the models developed. There is a difference between the functionality and the accuracy of a model. That is that, although a black box model such as a neural network may be capable of accurately predicting, UTS for example, the way in which it used the input variables to perform this calculation may not be consistent with the theory behind the process. Therefore, by looking at the way in which the individual or combined input variables contribute to the output of the model, one can begin to satisfy oneself that the model is valid and has a greater chance of performing well on previously unseen data. This is also an important process when the model is to be used within an industrial context since it allows understanding and confidence in the ways a certain model might work.

7.2.1 Data set based sensitivity analysis

When a linear model is developed it is straightforward to determine the inputs which the model regards as important, by looking at the linear model coefficients which relate to the individual inputs. This has already been seen in chapter 4. With a neural model, however, interpretation of the network weights is somewhat more

complex, and so an empirical method for determining the model output's sensitivity to it's inputs was used. The work was based on methods employed by commercial neural network packages such as Neural Fusion's NNmodel program to determine network sensitivities. The method used the data set from which the network was trained to determine the effect on network output of varying each variable in turn by a small amount (+/- 1% of the variable range) for all examples. Although the training set is typically used, validation, and test sets or a combined data set could also be utilised. The variation in output variable was noted for each of the variables for each of the examples, and then the average effect of each variable was calculated as a percentage of the total variation for all variables.

After consideration, it was decided that this method would provide an overall impression of the importance of each variable, given that the data set used was representative of the problem domain (this was determined by comparing histograms of the training, validation and test sets for a certain model). It was, however, also considered that the binary input variables should not be included in the calculation since these variables firstly did not vary continuously and, secondly, a change in site, for example, was represented by a simultaneous change in all variables. The effect of site and quench will therefore be analysed later in this chapter.

An algorithm to perform the sensitivity analysis was written in Matlab, the operation of which is described by the block diagram in Figure 7.1. Once the average difference for each variable was calculated over all samples, a normalised sensitivity value was obtained for all variables by dividing all results by the sum of the modulus of all sensitivity values for each variable.

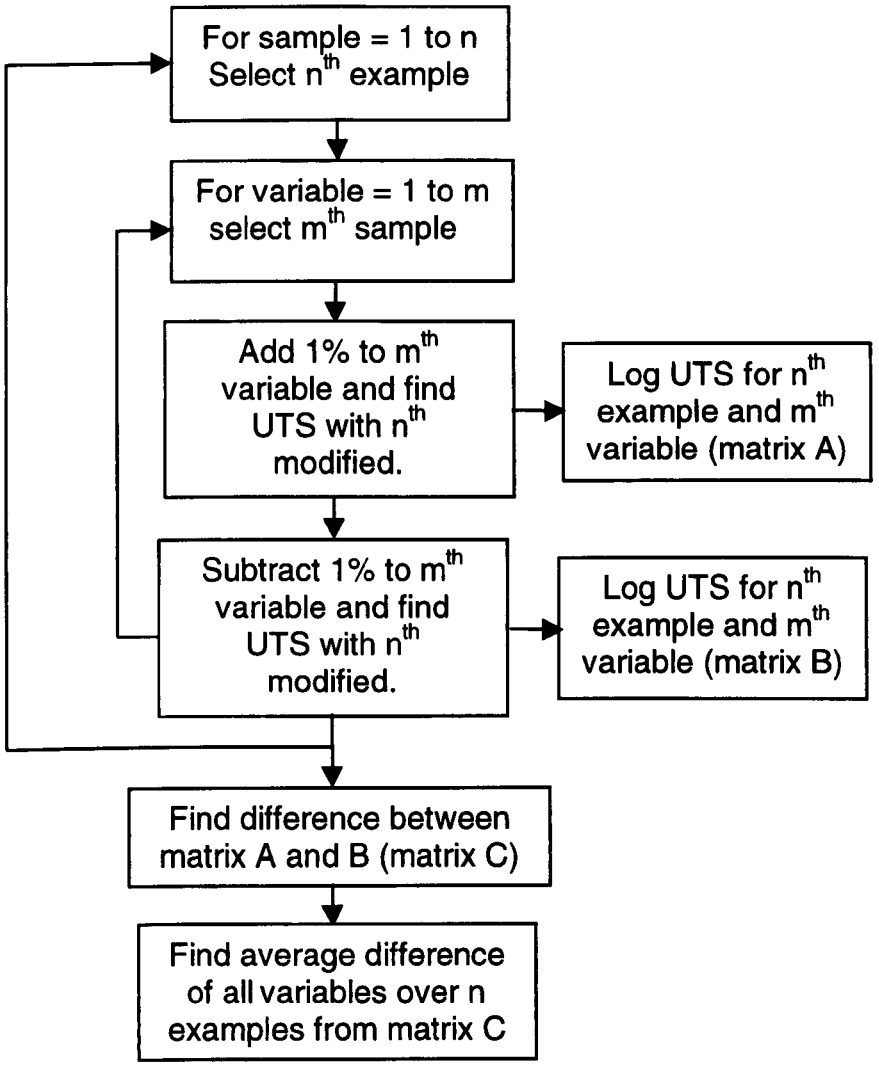


Fig. 7.1 Diagrammatic representation of a program to determine the sensitivity of a neural model's output to its inputs.

There are, however, two considerations to take into account when using this method. The first of these is that the effects of interactions between the input variables are not taken into account. The second is that the sensitivity of the input variables will not be the same for two input samples representing two different grades of steel, this is why the sensitivities produced by the method are only a guide over the distribution of the data set used to make the calculation. One method for visualising the effect of interactions between two input variables will be seen in section 7.2.3. Although the significance of certain input variables may change depending upon the levels of all the input variables, it was felt that an overall data set would provide a useful

validation method of the model function. This is because a metallurgist familiar with the plant from which the data originated, would have an overall idea of the general significance of the variables in the process. Analysis of variable effects at a fixed composition (and therefore input level) will be investigated in section 7.2.2.

The input sensitivities of each mechanical test result model were calculated as has been described. The results of this together with their interpretation are as follows.

The UTS model's sensitivities are shown in Table 7.1. In this table, the variables and their results are arranged in order of decreasing significance.

Input Variable	Average Sensitivity
Tempering Temp.	-0.3367
C	0.147211
Cr	0.123117
Mo	0.115782
Ni	0.079006
Depth	-0.0423
Size	-0.03995
Mn	0.03714
Al	0.026736
S	-0.02102
Si	0.016413
V	0.009841
Hardening Temp.	0.004791

Table 7.1. Sensitivity results for UTS ensemble model

It can be seen that the most significant variable is tempering temperature; carbon, chromium and molybdenum and nickel then follow this. This order concurs with metallurgical knowledge in that Tempering temperature is thought to be the most significant variable in the process and carbon should be the most significant alloying element. The variables chromium, molybdenum and nickel are were also expected to be high in the sensitivity analysis since these variables work together to affect the hardenability of the steel. It can also be noted from Table 7.1, that some variables

have negative terms and some positive. To obtain correct 'polarity' of results it is important that the results obtained in Matrix B are subtracted from Matrix C (see Figure 7.1). The negative values also concur with metallurgical knowledge because, for example, tempering temperature is known to soften the material. Increasing test depth and forged size also reduces the tensile strength of a sample since the quenching effect at increased depth and volume will be reduced. Another variable that has a negative effect on UTS is sulphur, which is considered an impurity, which typically occurs at a residual level but is widely known to reduce the strength of a material. Metallurgists regard vanadium to be a very potent element, having a significant effect on UTS. However, only a small number of samples within the data set (127 out of 5710), contained a vanadium addition. This meant that, overall, the effect of vanadium was not found to be as strong as would be expected if all the examples had been vanadium containing. From running the sensitivity simulation with different data sets (i.e. with different data selected for the training set), it was found that slightly different order could be obtained for some variables which had a sparse distribution, such as vanadium. This is obviously due to the fact that the number of samples within the data set represents many different types of steel, and therefore when there is a slight change in the distribution of the training data, the average sensitivity results will also change. The method could be made more 'stable' by using all samples available for the sensitivity calculation, however the existing method using just the training data took around 5 hours to run on a multiple processor Unix machine. It was, found that the high significance variables which were well represented in the data set were stable in their order, this was because they commonly occurred across many grades of steel. This fact should, however, be remembered when considering variables with low sensitivities. The sensitivity could

also be calculated on a particular grade of steel rather than a distribution as will be seen in section 7.2.2.

The sensitivity results for the impact model are shown in Table 7.2.

Input Variable	Average Sensitivity
Temp2	0.289632
C	-0.17539
Cr	0.101467
TempA	0.092869
Size	-0.06855
Mn	0.046936
Al	0.043362
V	-0.04118
Mo	-0.0404
Size	-0.03178
Temp1	0.026028
Ni	-0.01834
S	-0.01431
Depth	-0.00975

Table 7.2. Sensitivity results for impact ensemble model

These results show that tempering temperature is the most significant variable affecting impact properties, as was confirmed by metallurgists, but that carbon, chromium and Impact Test Temperature are also important. The importance of the impact test temperature was described in chapter 3, it will be seen in section 7.2 how this can be used to generate an impact transition curve, which shows the effect a low temperature has on impact properties. The final feature of the results in Table 7.2, is that the effect of carbon, and indeed several other alloying elements which would normally promote a higher UTS, on impact energy is negative, this is because toughness tends to reduce in higher strength materials.

The sensitivity results relating to the proof stress model are shown in Table 7.3. It can be seen that tempering temperature is the most significant variable as one would expect, followed by the hardenability variables of chromium, molybdenum and nickel. carbon, which also will have a significant effect, comes after these other

alloy additions. Close inspection of the contribution values relating to molybdenum, nickel and carbon, reveals that these additions have similar significance in the prediction of Proof Stress. All results so far show that hardening temperature has low significance, this is because the majority of steel compositions are fully martensitic after the hardening stage as described in chapter 4, and therefore the effect of hardening temperature is not as important for these steels. Adding hardening temperature to the models did, however, benefit the accuracy of the model and therefore is significant for some grades of steel.

Input Variable	Average Sensitivity
Temp2	-0.22962
Cr	0.14227
Mo	0.1089
Ni	0.092735
C	0.090764
Depth	-0.08751
Al	0.085947
Mn	0.065499
Size	-0.05098
V	0.034443
Si	0.007005
Hardening Temperature	0.002237
S	-0.0021

Table 7.3. Sensitivity results for the proof stress ensemble model

Table 7.4 shows the sensitivity results for the elongation model. Once again, tempering temperature is the most significant variable as would be expected, but as with the impact model its effect is positive, i.e. a steel with less martensite is softer, tougher and more ductile. carbon, chromium and molybdenum are then the next significant variables, but they negatively effect the elongation properties. This is because these variables positively affect the UTS of the material and so they negatively affect the ductility of the material. The vanadium sensitivity appears to be higher for this model than for others previously investigated, however process knowledge that only a relatively small number of steels contain vanadium tells us

that this is also a feature of how many vanadium-containing examples lie within the experimental data set.

Input Variable	Average Sensitivity
Tempering Temperature	0.389907
C	-0.11247
Cr	-0.10772
Mo	-0.08114
V	-0.07011
Al	0.061272
Depth	-0.05178
Ni	-0.05054
S	-0.0317
Mn	-0.02994
Hardening Temperature	-0.00548
Si	-0.00402
Size	-0.00394

Table 7.4. Sensitivity results for the Elongation ensemble model

Finally, sensitivity results for the reduction of area model are shown in Table 7.5. Like the impact and elongation results these also show tempering temperature to be the most significant effect, which is positive due to the increase in ductility this causes. Many of the classical strengthening variables such as C also now have a negative effect on reduction of areas since these reduce the ductility of the material.

Input Variable	Average Sensitivity
Tempering Temperature	0.8168
C	0.7615
Depth	-0.5444
Al	-0.4306
S	-0.2544
Mn	-0.1353
Mo	-0.0737
Si	-0.0588
V	-0.0452
Ni	-0.0369
Cr	-0.0278
Hardening Temperature	-0.0183
Size	0.008

Table 7.5. Sensitivity results for the reduction of area ensemble model

7.2.2 Composition-based variable effect analysis

Having performed sensitivity analysis based on a distribution of compositions, it was decided that a more detailed method, which would allow visualisation of the non-linearity of the input variables, would be used to keep all inputs constant whilst varying each input variable independently. The output of the model with this variation could then be plotted so that the effect of the input variable could be seen on the output of the model. It was decided that initially, each variable could be varied throughout its max and min range for the model. The only problem was deciding the levels at which the remaining variables should be set. At first it was thought that the mean or median values could be used to as inputs for the variables not under variation, however it was found that because these values did not actually represent a valid composition the results were difficult to interpret. Instead, it was

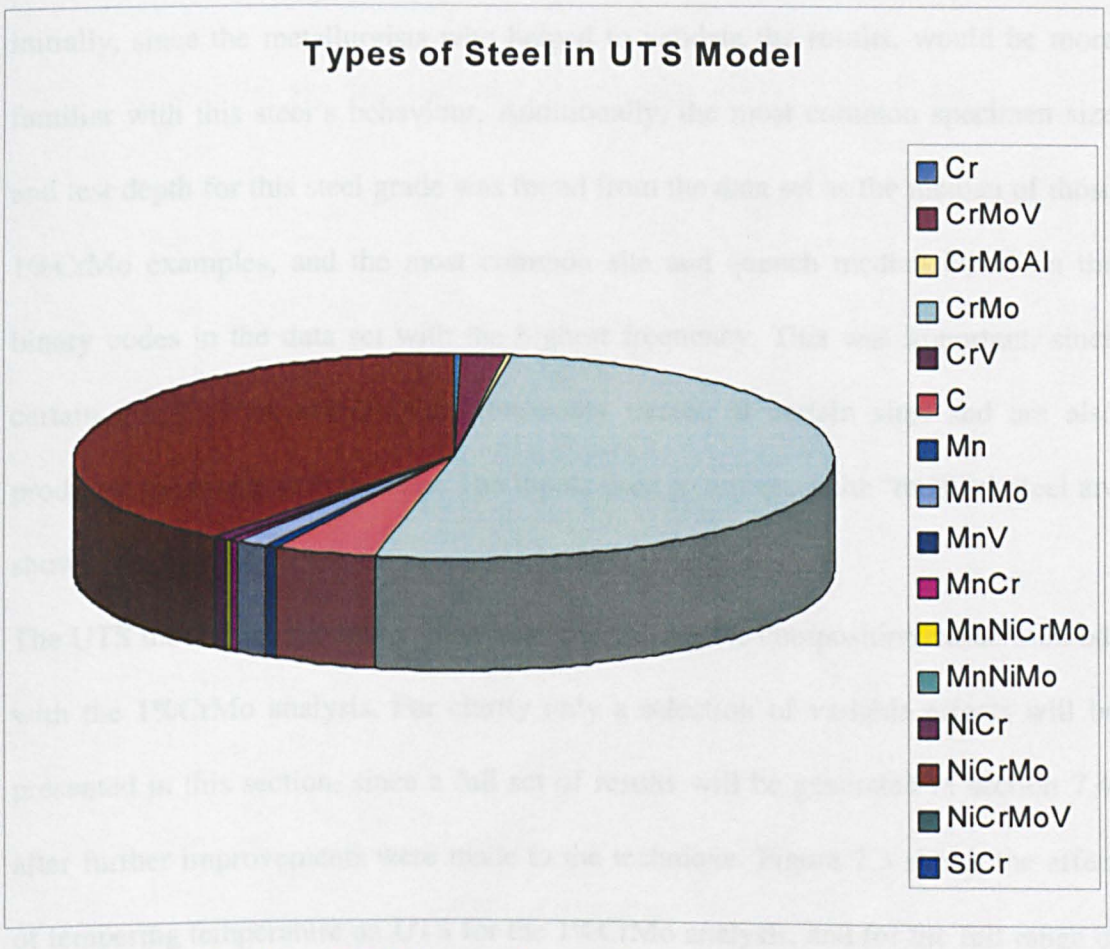


Fig. 7.2 Classification of UTS data set into steel type

decided that the expert composition data gathered for use in the modular decomposition section of chapter 6 should be used. This information is summarised in Figure 7.2, which shows the relative proportions of each family of steel in relation to the data collected from BSES. It can be seen that the main steel grades in the data set are of types CrMo and NiCrMo. The CrMo steel type occupies the largest proportion of the data set, and represents a family of grades containing additions of chromium and molybdenum, of which there can still be sub-categorised a number of steel types. Of these sub-types, the most common compositional form is that of the 1%CrMo steel (containing 1% chromium). The data set obtained from BSES contained a variety of 1%CrMo steel examples, depending upon the precise composition obtained over each case. In order to obtain the most common 1%CrMo composition, the median of these examples was found, and this was then used as the base composition for the variable effect experiment. A popular composition was used initially, since the metallurgists who helped to validate the results, would be more familiar with this steel's behaviour. Additionally, the most common specimen size and test depth for this steel grade was found from the data set as the median of those 1%CrMo examples, and the most common site and quench median based on the binary codes in the data set with the highest frequency. This was important, since certain grades of steel are more commonly treated at certain sites and are also produced in certain section sizes. The inputs used to represent the 'median' steel are shown in Table 7.6.

The UTS model was the first to be investigated using the composition-based method, with the 1%CrMo analysis. For clarity only a selection of variable effects will be presented in this section, since a full set of results will be generated in section 7.4, after further improvements were made to the technique. Figure 7.3 shows the effect of tempering temperature on UTS for the 1%CrMo analysis, and for the full range of

the model. It can be seen that effects are non-linear, the gradient of the curve re-iterates the findings of the sensitivity analysis, showing tempering temperature to have a significant effect.

1% CrMo ANALYSIS	
Variable	**Value**
Site (1-6)	3
Size (mm)	180
Test Depth (mm)	12.7
C (%)	0.41
Si (%)	0.27
Mn (%)	0.78
S (%)	0.023
Cr (%)	1.08
Mo (%)	0.22
Ni (%)	0.19
Al (%)	0.027
V (%)	0.005
Hardening Temp (°C)	860
Tempering Temp (°C)	630
Impact Test Temp (°C)	20

Table 7.6 The 'median' model inputs representing the 1%CrMo analsyis

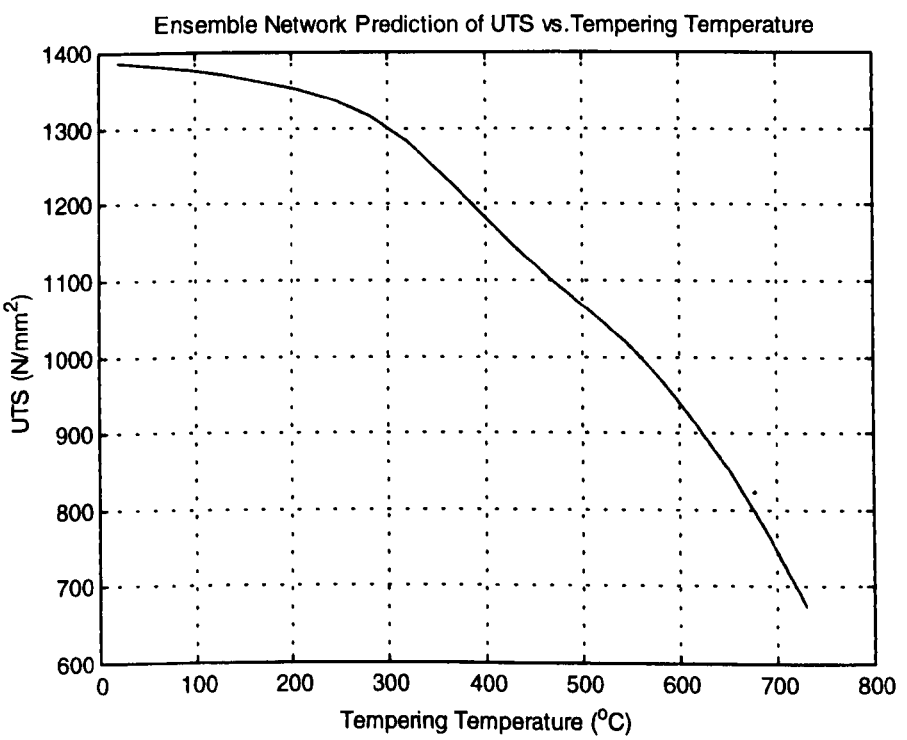


Fig. 7.3 The effect of tempering temperature on UTS.

The effect of carbon was also seen to be significant and can be seen in fig 7.4, which shows it to have a more linear effect than tempering temperature.

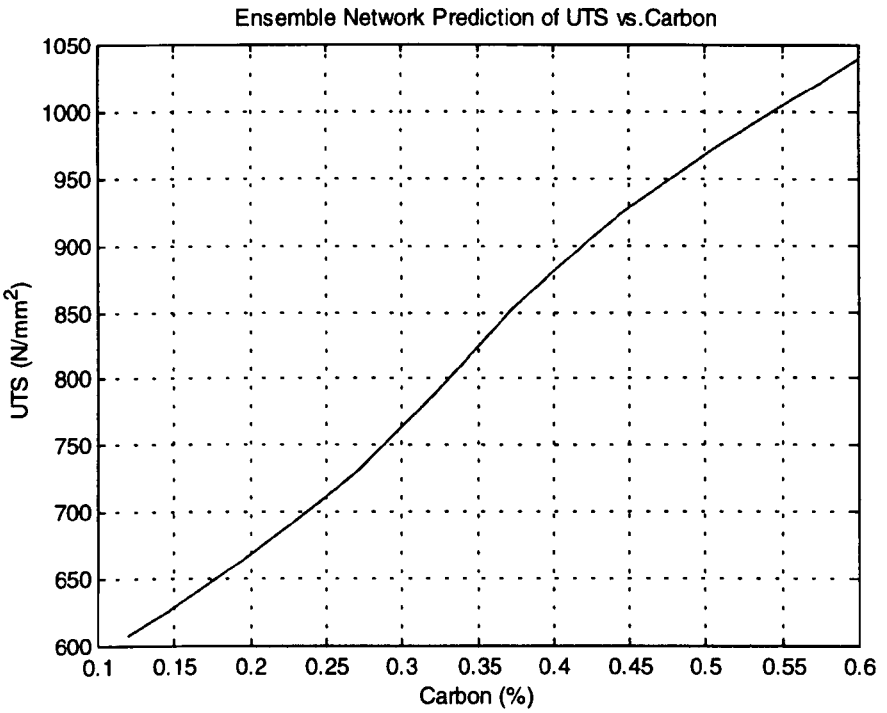


Fig. 7.4 The effect of carbon on UTS

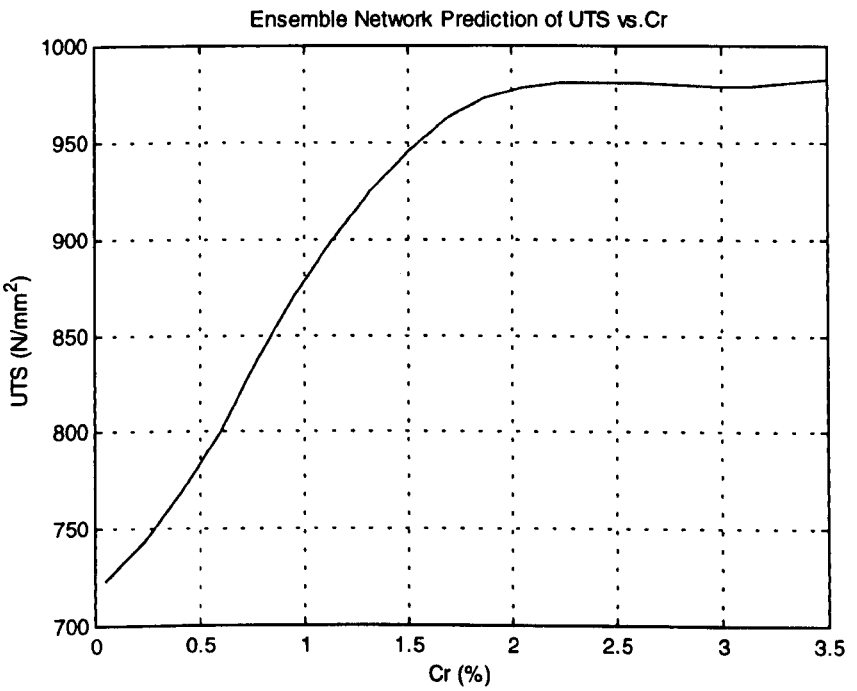


Fig. 7.5 The effect of chromium on UTS

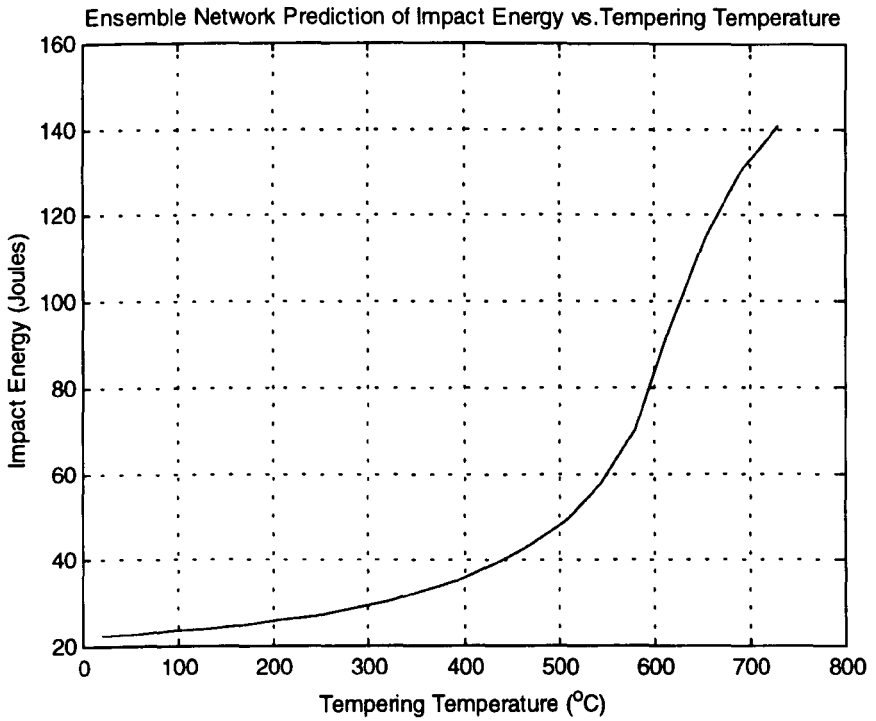


Fig. 7.6 Effect of tempering temperature on impact energy

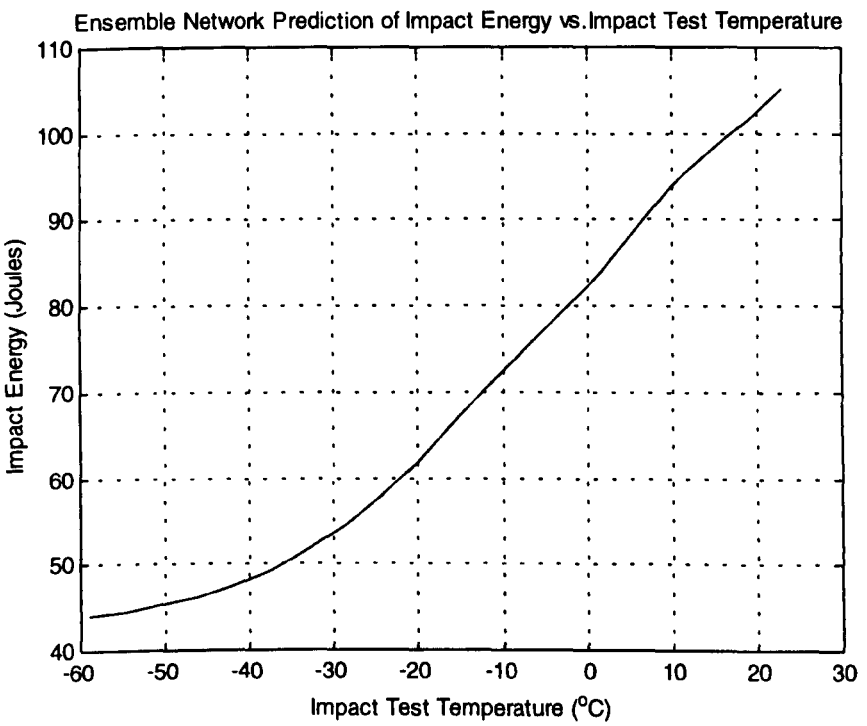


Fig. 7.7 Effect of impact test temperature on impact energy

One very interesting effect was that of chromium on UTS, shown in Figure 7.5. This shows chromium to have a non-linear effect, which reached a plateau after

about a 2% weight addition. This is interesting, because chromium additions are made primarily to increase the hardenability of the material, and yet it seems to have little effect beyond this 2% point. It was, however, noted by metallurgists that most additions made within the data set were made up to about the 1.5% chromium level. In section7.4.3, further analysis of this graph will be made, which explains why this phenomenon occurs.

Some interesting results were also seen with the impact model, Figure 7.6, which shows the effect of tempering temperature on Impact Energy. This is a very non-linear effect, which shows the reduction in impact energy produced from a reduced tempering temperature.

Another plot obtained from the impact model is the effect of test temperature on impact energy, shown in Figure 7.7. This is a classic curve used by metallurgists to determine the transition temperature of a material, which is the point of transition between ductile and brittle fracture, as explained in chapter3.

7.2.3 Interaction effects between variables

An extension of the composition based variable effect method, was to investigate the effect of varying two model inputs at once. An example of this was investigating the effects of varying tempering temperature and carbon content at the same time, the results of which were plotted on a 3-dimensional graph, which is shown in Figure 7.8. The graph shows that the strength of this 1% CrMo steel is greatest at low tempering temperatures and high carbon contents (as was agreed by metallurgists). Conversely high tempering temperature and low carbon content produce low UTS. More detailed analysis also reveals, however, that at low carbon contents, the effect of tempering temperature is quite linear, however at high carbon contents, the effect is somewhat more non-linear, the gradient of the curve above

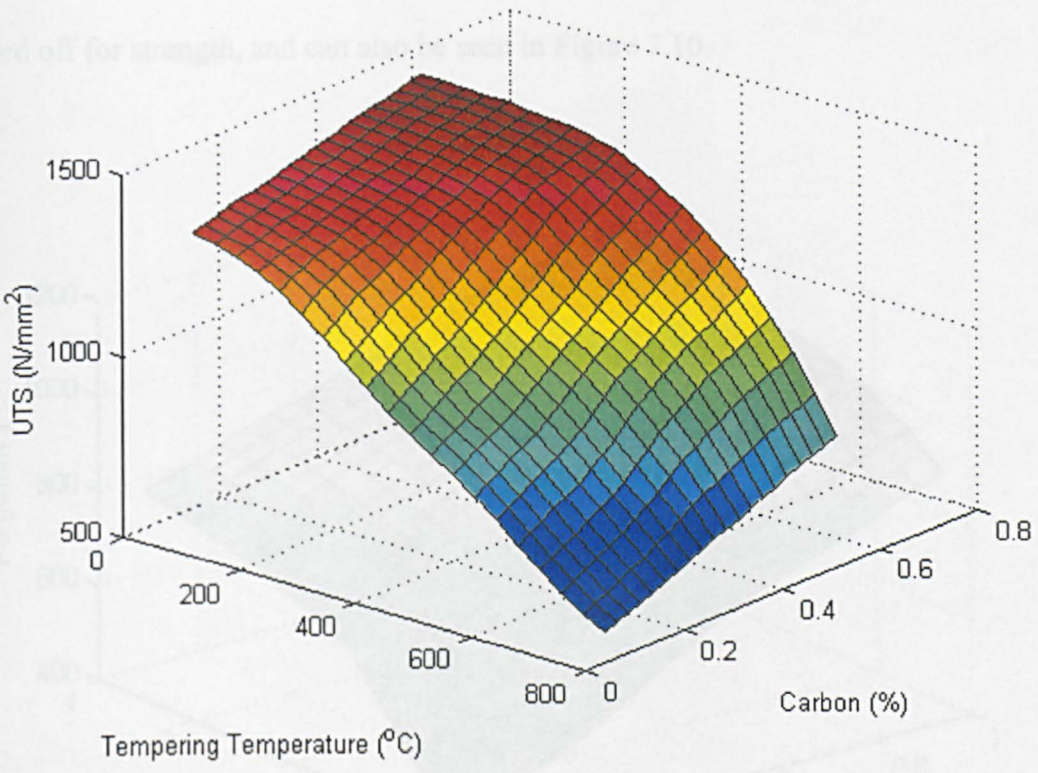


Fig 7.8 Interaction effect of tempering temperature and carbon content on UTS

500°C at high carbon content is much steeper than at the low carbon content. This is important when measurement and control tolerances are taken into account since a ±5°C difference in tempering temperature will have a greater effect with a 1%CrMo steel containing greater carbon content.

Another example was generated from the Proof Stress model, where the elements chromium and molybdenum were seen to be important in previous sensitivity analysis. The interaction effects between these two variables can be seen in Figure 7.9. The graph shows how the two variables work together to produce a higher Proof Stress, the effect at low values for both variables is also much more pronounced than at higher levels.

The final example of variable interaction is that between carbon and tempering temperature on reduction of area. It was noted in the sensitivity analysis section that the effects of carbon and tempering temperature are reversed for the

reduction of area model, because ductility is (when described in a very basic way) traded off for strength, and can also be seen in Figure 7.10.

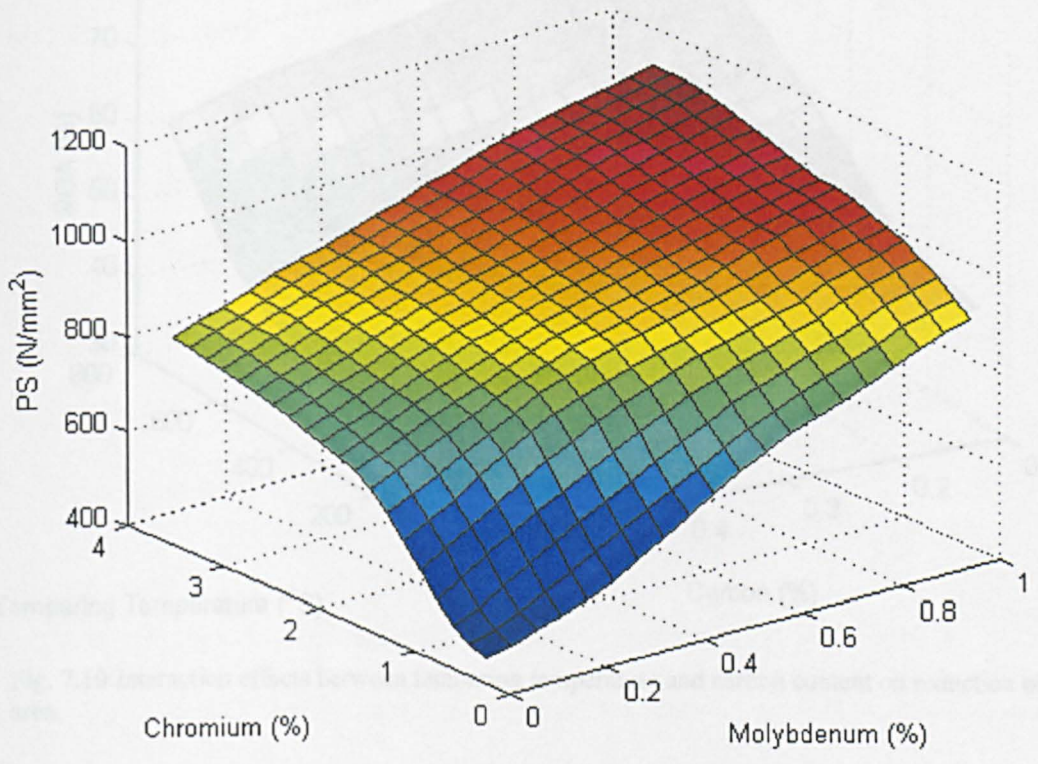


Fig. 7.9 Interaction effects between chromium and molybdenum on Proof Stress.

Many more of both the interaction effects and single variable effects could be demonstrated for just the 1%CrMo steel, in order to validate the model. Moreover the effect of changing the base composition (the 'median' analysis) to another grade of steel or indeed a new grade of steel would allow further insight into the models, and indeed this was the purpose of project.

To allow the UTS model to be evaluated, and eventually be used in an industrial situation, it was decided that ultimately a Graphical User Interface (GUI) would need to be developed to allow potential users to interact with the neural models developed.

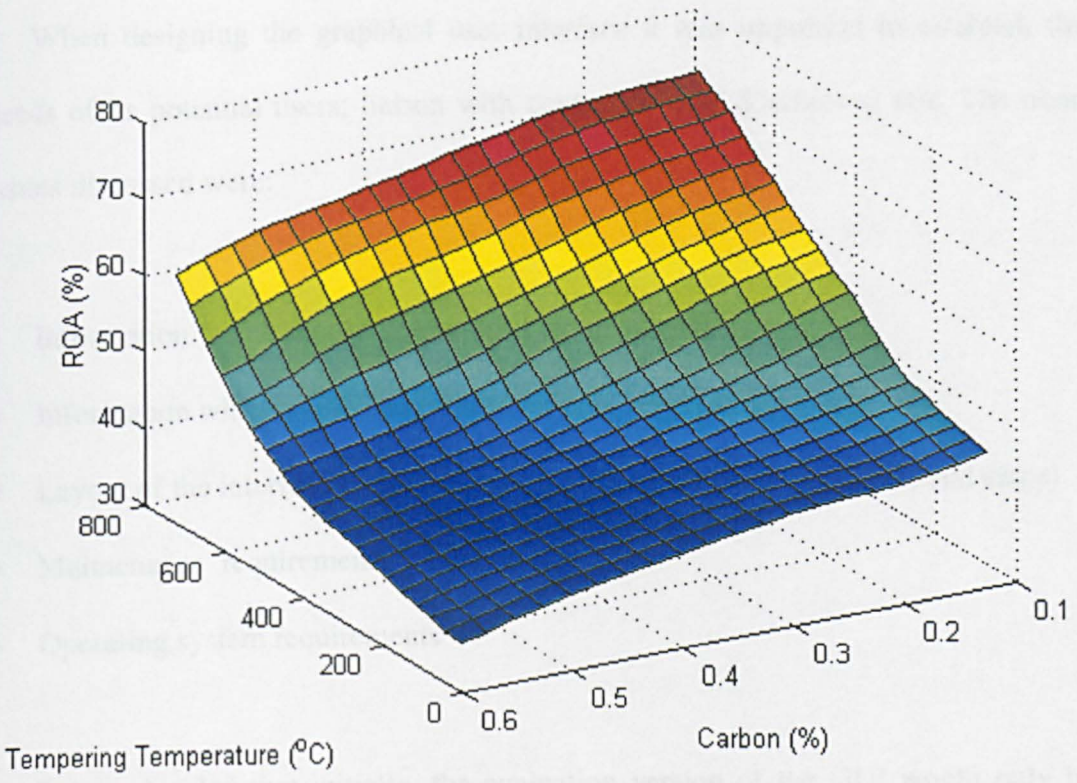


Fig. 7.10 Interaction effects between tempering temperature and carbon content on reduction of area.

7.3. Development of a graphical user interface

Before describing the development of the graphical user interface it is important to note that the work which is described in this section was undertaken before the ensemble method was employed for neural network prediction. The feedback which resulted from it partly motivated the investigation of ensemble methods, and the usage of a ten-network ensemble trained with different random initialisation weights. It was however felt appropriate that discussion of the graphical user interface should be made after model analysis methods were described, to enable an insight into its design. There are, obviously slight differences between the results of the best single network in the GUI, and ensemble predictions in use in section 7.2, these differences being due to the increased accuracy of the ensemble method.

7.3.1. GUI design considerations

When designing the graphical user interface it was important to establish the needs of its potential users; liaison with contacts at BSES achieved this. The main issues discussed were:

- Information which should be provided by the interface
- Information which should be entered into the interface
- Layout of the interface (including methods of information extraction and entry)
- Maintenance requirements
- Operating system requirements

It was decided that initially, the evaluation version of the GUI would only be used to provide UTS predictions, this was partly because the UTS model was one of the first to be ready for use in the project, and was potentially of the greatest value. Having seen results similar to those in earlier sections of this chapter, it was agreed that the interface should be capable of providing UTS calculation (i.e. one test result value for a given input vector), together with individual variable effects on a given base composition.

It was also decided that all input variables to the model should be able to be entered through the graphical user interface. In particular it was decided that continuous input data such as composition or size should be able to be entered directly, whereas binary values such as site or quench type should be entered through toggle switches. Additionally, rather than entering the six binary site codes or three quench codes, a linguistic term should be displayed in the toggle arrangement, allowing the user to select say, the 'lab' as the treatment site or 'water' as the quench type.

Sketches of potential layouts were made and discussed, and a single screen layout was designed, such that all inputs, variable effects and calculated results could be viewed at the same time.

Maintenance of any predictive model is important, since processes are continually changing and if a model is not updated, then it will begin to become unrepresentative of the process as described by Myllyskoski[31]. For this reason it was decided that a 'slot in' neural network module to the network would be preferable, such that the neural model could be updated without having to alter the entire user interface program.

Finally, the graphical user interface was required to work as a stand-alone package (i.e. without the need for Matlab to be installed on the system, for example), which should be capable of running under the Windows 95/98 or NT operating environments.

7.3.2 GUI construction

The GUI was developed while the commercial package NNMODEL by Neural Fusion was still being utilised for model construction. An added benefit of this package was that models could be exported to Visual Basic and Visual C++ programming environments. It was therefore decided that this evaluation version of the GUI would be constructed using visual basic, as this would enable short development times, and allow modifications to be made readily. The exportation of models from the NNMODEL package to interface with the GUI also meant model maintenance could be achieved without disruption to the software that would be installed to provide the GUI.

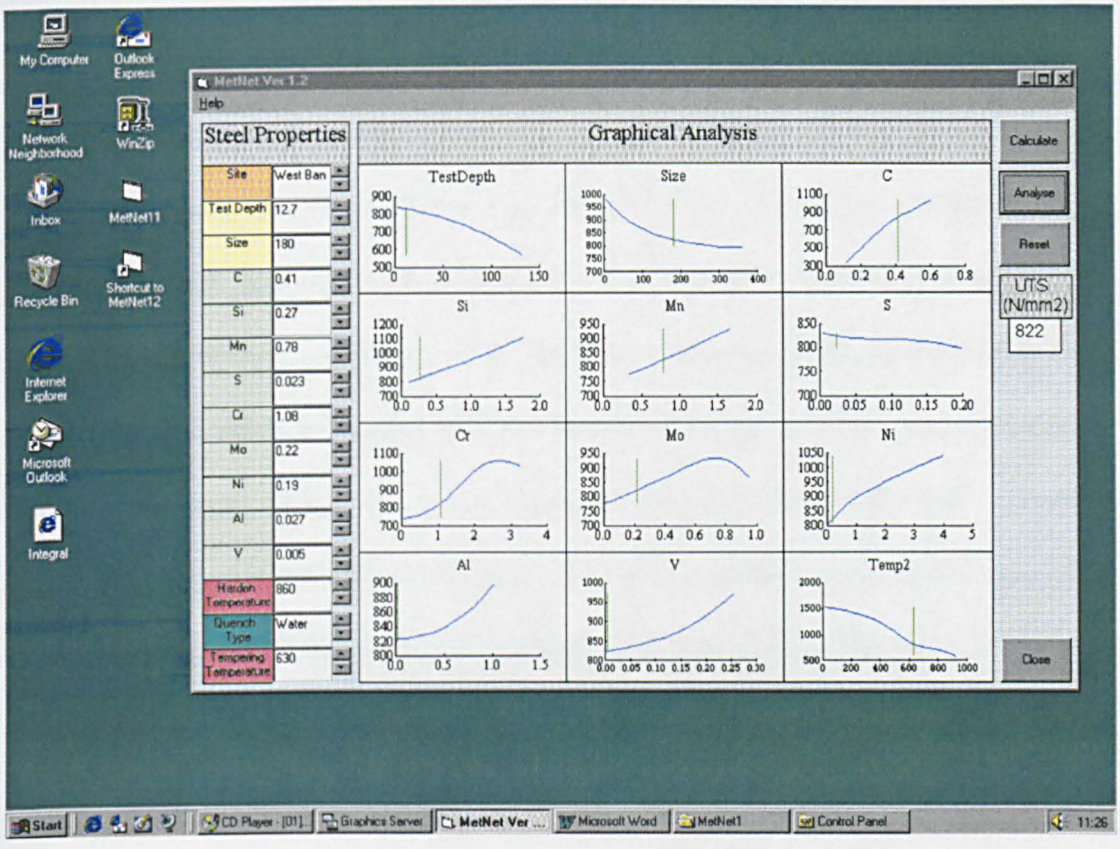


Fig. 7.11 Screen shot of Graphical User Interface to enable interaction with UTS model

The software was written and distributed to three metallurgists at BSES in order to enable evaluation, and has successfully been used on both Windows 95 and NT operating environments. A screen shot of the completed user interface is shown in Figure 7.11. The screen shot shows the interface in use within a window environment. The area to the left of the GUI enables input selection, and then the variable effect graphs are generated in the central region. The generation of these graphs uses a loop to vary each input variable from its maximum to its minimum value. It was found that 20 predictions per graph were adequate to obtain a smooth curve, without taking an unreasonable amount of time (the Visual Basic program was found to run relatively slowly when performing loop operations). Two buttons were implemented to allow the user to either 'Analyse' a set of inputs by creating the variable effect graphs, or just 'Calculate' a single UTS value if required. During the

development of the program, four other requirements also became apparent in the design. The first was that markers were placed on the variable effect graphs to highlight at what point the inputs fitted were in comparison to the graphs generated, these can be seen as vertical green lines in Figure 7.11. Secondly, a scaling option was added, such that all graphs generated could either fit the max and min of the UTS values predicted, or, have user defined fixed axes to determine the significance of one variable effect compared with another. Thirdly, the program was designed to default with input values set to the 'median' analysis inputs described in section 7.2.1. This set of input values relates to the most commonly made grade specimen type, and treatment. Therefore, it was intended that the user would most commonly be interested in investigating a variation around those inputs and would therefore have to make minimal input to the interface in order to satisfy their query. Finally, it was necessary that all model inputs were limited to the maximum and minimum values of the training data used to construct the model. This is because it is widely accepted that neural networks should not be used for extrapolation, and attempts to achieve such have been seen to result in predictions of poor predictive accuracy. One such finding of this was concluded by Bhadeshia et al[21], who tried to extrapolate beyond the data set for impact toughness predictions on C-Mn steel.

7.3.3 Feedback

There were three main types of feedback that resulted from the evaluation version of the model. The first type related to the usability of the GUI, the second type related to the accuracy of the model on steels currently being treated in the plant, and the third related to a more experimental investigation which was made on more abstract steel grades.

The following points were made in relation to the usability of the GUI:

- The system ran at an acceptable speed using a Pentium 133 PC. The calculations and drawing of the graphs was a little slow, however this did not need to be done for each prediction.
- The screen layout appeared acceptable with the graphs showing the effect of each element being very useful.
- The TAB sequence between fields is not in order and does not move logically down the screen.
- When within a field, any old values within that field are not overwritten when inputting new values.
- When a test depth was at the mid radial position the value had to be calculated and entered each time. It would be easier if an option of Mid-Radial could be selected which would automatically select the correct depth from the size.
- It was very time consuming entering a cast analysis each time that the cast was changed. Ideally, capability for a cast number to be input should exist and then if the cast had been used before its input values could automatically be set.

It was felt that all of the points raised by the users were valid and could be easily accommodated with future versions of the GUI; most relate to 'bugs' in this early version of the software. It was however felt that a gain in speed could be achieved by coding the interface in Visual C and that this may further be enhanced if only selected plots were updated during the 'Analyse' function. This would be helpful because graphs of, for example, the effects of residual elements are not always significant and by viewing fewer graphs, a larger scale could be used to aid clarity.

The next stage of the feedback concerned the accuracy of the neural model, when it was predicting commonly treated steels at the plant. The performance of the single neural model was compared to that of an existing linear model developed by

British Steel specifically for predicting tensile strengths at one plant, named 'West Bank'. This linear model will be referred to as the 'West Bank predictor'. The results of both predictors' performance were presented by a heat treatment at the plant, for a number of popular steel grades. The statistic used to measure the performance was (%) error, which was defined as:

$$\text{(Predicted Result – Actual Result) / Actual Result *100} \qquad (7.1)$$

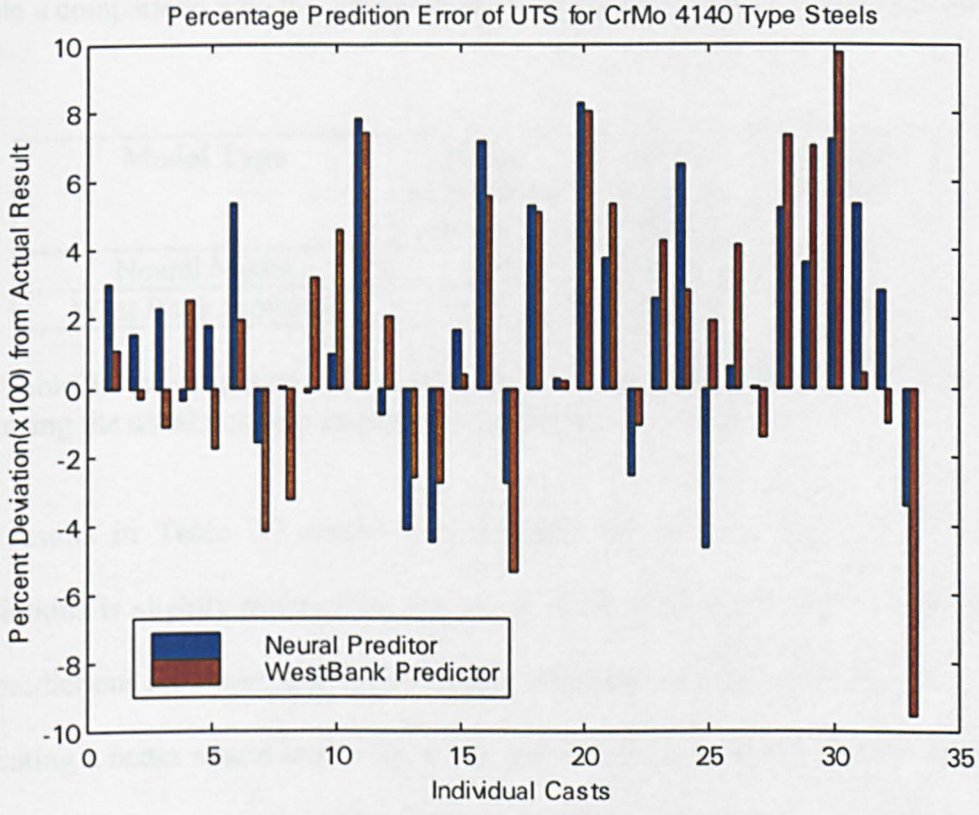


Fig. 7.12 Comparison of neural predictor with the West Bank linear predictor by BSES for Cr-Mo 4140 type steels.

The results using this performance measure are shown in Figure 7.12. It can be seen that, on the whole, neither predictor seems significantly worse than the other. When looking at the maximum and minimum percentage errors generated by the models, the linear model has a wider range of percentage prediction error (-9.50% to 9.78%

as opposed to –4.66% to 8.32% for the neural model). It was however noted that both models tend to over predict (predict higher than the actual result). The mean percentage error was used by the metallurgist as an overall indication of the prediction accuracy of each model, this was found to be 1.76% for the neural model and 1.51% for the West Bank predictor. This was disappointing at first since the performance of the neural model appeared to be slightly worse than that of a linear model, however, when a fuller investigation of the results was made, using SD of error, and r-square statistics, the following results in Table 7.7 were obtained. Note that the predicted UTS values have been converted to Psi (pounds per square inch) to enable a comparison with the actual results which were measured using these units.

Model Type	**Mean percentage error (%)**	**SD of residual (PSI)**	**r-square statistic**
Neural Model	1.76	5018.9	0.88
West Bank Predictor	1.51	5681.7	0.84

Table 7.7 Investigation of Neural model and West Bank predictor' accuracy, using the usual r-square and mean-percentage error statistics.

The results in Table 7.7 shows that although the 'bias' of the neural model's predictions is slightly higher than that of the West Bank predictor, the variation of the predictions are lower and the correlation of predicted and actual values is higher, indicating a better neural model fit. It was felt that because the West Bank predictor was specifically designed on West Bank data that this might be a reason why the predictions obtained from it had a lower bias that those of the neural model trained on data from a range of sites. After this evaluation, it was decided that the performance of a neural model trained only on West Bank data would be investigated. Such a model was developed in chapter 6, section 6.2.5.2.1, as part of the modular decomposition investigation. This 'West Bank neural model' was used

to predict the same examples used in the previous evaluation. The results of this, together with those obtained from the existing West Bank predictor are shown in Table 7.8.

Model Type	Mean percentage error (%)	SD of residual (PSI)	r-square statistic
West Bank Neural Model	-0.84	6384.2	0.82
West Bank Predictor	1.51	5681.7	0.84

Table 7.8. Investigation a Neural model trained solely on West Bank Data and the West Bank predictor's accuracy, using the usual r-square and mean-percentage error statistics.

The results in Table 7.8 were interesting because they show that the neural model trained specifically on West Bank data, and then tested on test data from the plant has a much lower bias than with the neural model trained on more data from a range of sites (Table 7.7). The results from the West Bank predictor in Table 7.8 are obviously the same as those in Table 7.7 since the same data test data was used, but are included to aid comparison. It can be seen that the neural models bias is now significantly less than that of the West Bank predictor, however the standard deviation of its predictions are now higher and the r-square statistic is lower. This indicates that although the neural model trained on just the West Bank data has a lower bias, the model has suffered in terms of variance, perhaps from the limited training set.

Since the evaluation version of the software was released, the ensemble approach has been applied to the neural network predictor. This was still trained on the entire data set (for all six sites), but has increased accuracy over the standard neural network model as shown in Table 7.9. It can be seen that compared with the standard neural network, the ensemble predictions have a slight reduction in average percentage error, and a significant reduction in standard deviation of residual, accompanied by an increase in r-square statistic. This shows the ensemble technique

is beneficial for more accurate predictions, but does not solve the problem of bias error as well as a model trained solely on data from West Bank.

Model Type	Mean percentage error (%)	SD of residual (PSI)	r-square statistic
Ensemble Neural Model	1.67	4207.1	0.90
West Bank Predictor	1.51	5681.7	0.84

Table 7.9 Investigation the Ensemble Neural model trained on the entire data set and the West Bank predictor's accuracy.

The conclusions from this investigation were that the statistics used to investigate a model's performance are very important. If a low bias is required then a model trained only on data from that plant would be the best solution, however it should be recognised that this model may have more variance due to a reduced training set. The range of situations handled by the model are also important, since if

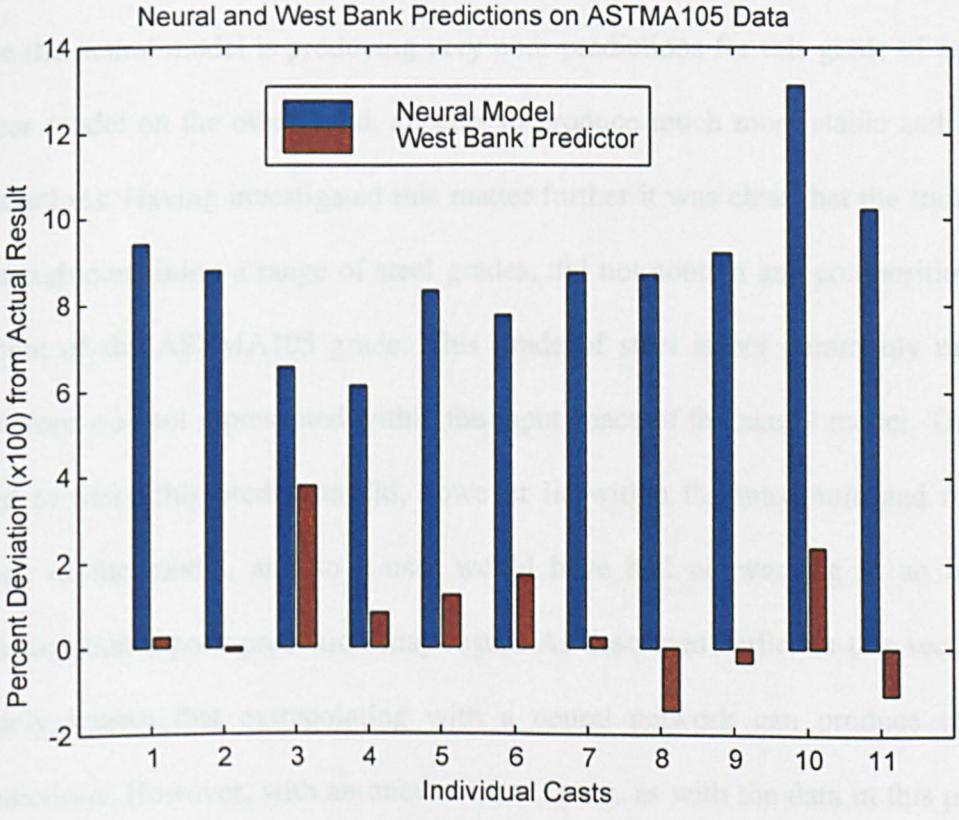


Fig. 7.13 Comparison of neural predictor with the West Bank linear predictor by BSES for ASTMA105 type steels (not within neural model's training set)

a model which is supposed to be broad ranging is trained on data from only one site it may predict less well in other situations. Having investigated the possibility of various decompositions previously in this project, it was decided that a model covering a wider range of steel grades and sites was to be developed. The aim of the work in this project was therefore to develop a wide-ranging model, rather than one dedicated to a particular site since its primary purpose was intended to provide predictions for a wide range of steel types.

The final part of the feedback looked at the performance of the original neural network model (as distributed to BSES), when predicting steels which are not described by the training data of the model. Such a group of steels was those of type ASTMA105 which are a group of very low carbon steels not found in the training data at all, but ones which lie within the max and min ranges of the training data. The graph in Figure 7.13 shows the percentage error of the standard neural and West Bank predictors on data from this class of steel. It is clear from the graph that in this case the neural model is producing very poor predictions for this grade of steel. The linear model on the other hand, appears to produce much more stable and accurate predictions. Having investigated this matter further it was clear that the training set, although containing a range of steel grades, did not contain any compositions close to that of the ASTMA105 grade. This grade of steel is not commonly made and therefore was not represented within the input space of the neural model. The inputs used to make this prediction did, however lie within the maximum and minimum range of the model, and so a user would have had no warning in an industrial situation that a poor prediction may result. As described earlier in this section, it is widely known that extrapolating with a neural network can produce unreliable predictions. However, with an uneven input space, as with the data in this project, it has also been found that, when within a sparse area of that input space, spurious

results can also occur even when not technically extrapolating. It therefore appears that model accuracy is a function of data density. This was a major problem of the neural model and one that needed resolving. It was understandable that the model might make poor predictions on an unusual grade of steel, but there must be some indication of when this was going to happen, particularly when the user may not have knowledge of the types of steel used to train the model. For this reason the next section considers the development of a reliability framework, to allow an indication to the user of the reliability of a prediction produced by the model.

7.4. Reliability framework

Research into possible ways of illustrating to the user the reliability of the network predictions was sparked by the discovery that spurious predictions could be made by the neural network predictor when they were away from a populated region of the input space.

7.4.1. Literature connected with neural model prediction reliability

One solution, which claims to provide an estimation of prediction error, is that of Mackay's Bayesian framework for backpropagation which was the subject of a review paper[24]. The technique was mentioned in chapter 6, for its ability to produce regularised networks. The technique does, however, require the calculation of a term called 'evidence', which measures how probable a model is given the data set it was trained on and assuming equal prior probability. Penny[82] mentions that the technique assumes a Gaussian approximation to the posterior distribution and notes that with a finite number of data points, the approximation breaks down. He also notes that the approximation may be tested by looking at the correlation between evidence and test

set error (in theory a model with high evidence should have a low test set error), however in several experiments he shows that this is not always the case. At this stage in the project it was already suspected that the error distribution on the data set might not be uniform. It was felt that extensive research would need to be performed in order to satisfy oneself that the Bayesian technique would be applicable to the data sets described in this project. Additionally, the literature search indicated that other techniques could also be used to provide an indication of prediction reliability. One such paper which indicated this is by Cho et al[43], who recognises that neural networks are prone to unpredictable behaviour in what they term a 'novel' environment. A variety of techniques for improving the reliability of the model were suggested in this paper, the first was to train an additional neural network to predict the error of the original network, however this technique was unsuccessful in their particular case. Another approach was to detect the novelty of a point input into a predictive network. This is based on the assumption that a more novel input vector would result in a less reliable prediction as was shown by Pican et al[44] who defines the sparse area of the input space as the 'aberrant domain'. Pican did this by investigating the effect that varying binary values (used to denote steel type) had on prediction error, and then used a linear model for predicting areas of the model corresponding to high error values. This method would however be more complex when the entire input space was considered. Finally, an auto-associative technique such as that described by Krammer[102] was suggested as a popular technique for novelty detection. This works on the principle that it is possible to train a neural network with the same inputs and outputs, such that the network learns to predict its own input space. If a common input is given to the network, a similar prediction should result, however if a novel input is applied, the auto associative network should produce a very different output to that input. The authors also note that using

a committee of networks (such as with the ensemble technique) is another good way of improving prediction reliability, this has already been seen within this thesis and was demonstrated in chapter 6.

Having seen the literature, three approaches to further improve the reliability of the model were investigated, all of these approaches aim to inform the user when an input may result in an unreliable prediction.

7.4.2 Experimentation into techniques for improving prediction reliability

Initially, it was decided that the key to identifying the reliability of the model's prediction would be to see how close a particular input vector was to examples in the training set. The problem of finding how close a particular input vector is to the examples in the data set is that, as has already been explained, the input space is of high dimension. It is therefore difficult to visualise how close a particular input might be to others in the training data set. The PCA technique has already been mentioned in chapters 2, 5 and 6 for its ability to reduce high dimensional data. Even though the technique had been of limited use previously, it was decided that it would be used to try and visualise the input space. It was decided that one way of investigating the ability of the PCA method to visualise the input space would be to plot the first two principal component scores, with a colour coding for each point to represent the main alloy addition the steel it represented contained. Having performed classification of the training data previously this information was readily available. In addition to colour of the point relating to each example in the data set, symbols were also assigned to see if sub-categories could also be seen on the PCA chart.

It was decided that only the composition of the examples should be used for this experiment, since the categorisation was known for this composition. If tempering temperature or other treatment variables were included, then the appearance of the plot would change and the distribution colour and symbols relating to composition would have less meaning.

Table 7.10 shows the colour and shape coding used to generate the scores plot of the 1st and 2nd principal components, which is shown in Figure 7.14.

Steel Type	Colour and symbol type	
Cr	Yellow	+
Cr Mo V	Yellow	o
CrMoAl	Yellow	x
CrMo	Yellow	*
CrV	Yellow	>
C	Green	+
Mn	Blue	+
MnMo	Blue	o
MnV	Blue	x
MnCr	Blue	*
MnNiCrMo	Blue	>
MnNiMo	Blue	s
NiCr	Magenta	+
NiCrMo	Magenta	o
NiCrMoV	Magenta	x
SiCr	Black	+

Table 7.10. The colour and symbol-coding scheme used to denote steel grade on PCA plot.

Figure 7.14 shows that if one looks at the distribution of the colours of the steel types the technique seems to be showing the distribution of the compositional space quite well. There is some overlap, however, and the separation of the sub categories of steel is not very defined. One of the problems with this experiment is not explicitly how far in a distance sense one steel grade is from another, and indeed, two steel

grades may be very different in a compositional sense, but may lay close together in the input space. The PCA plot is thought to be good at representing the main steel grades, because it is has been concluded from discussions with metallurgists that alloy additions are made in respect of each other. In other words, there might be a correlation between, say, an addition of a certain level of chromium and molybdenum. One other explanation for the overlap in the plot, however, is that the variance in the composition has not been sufficiently explained by the first two principal components to enable an interpretation of the input space. It was found that the first two principal components only accounted for about 50% of variance of the data. One interesting point is that the PCA plot was generated before the SiCr steel was removed from the data set. The SiCr steel was removed from the data set in the

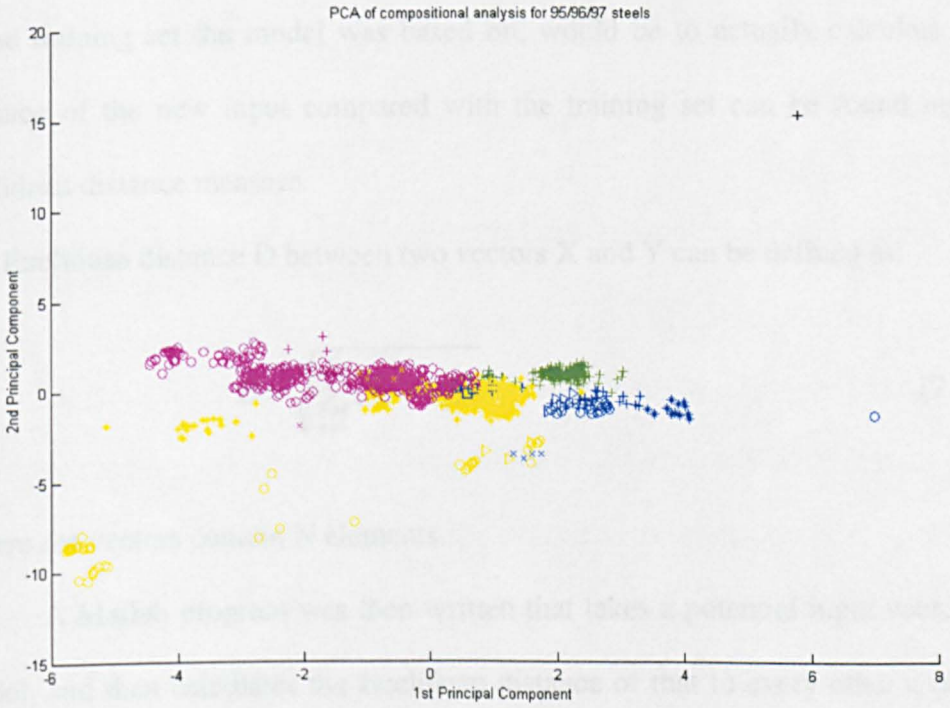


Fig. 7.14 A principal component scores plot of composition for the 1995,1996 & 1997 data sets.

final cleaned data set, because it was considered that this type of steel had vastly different behaviour to all other types selected by the metallurgist for inclusion in the

database. The point at the uppermost region of the PCA plot corresponds to the SiCr steel type.

It was decided that the PCA method was useful for visualising the distribution of the input data to a certain extent. However, because the first two principal components did not explain a larger amount of variation within the data set there were therefore uncertainties concerning the relation of the PCA scores to a distance measure of the input space (for example the Euclidean distance). Additionally, even when this relationship was determined, there would still be additional problems in automatically finding the distance of a model input to the clusters produced by the PCA chart.

Having considered a technique of 'visualising' the distance between inputs, it was decided that a better approach to determining how close a new model input was to the training set the model was based on, would be to actually calculate it. The distance of the new input compared with the training set can be found using the Euclidean distance measure.

The Euclidean distance D between two vectors X and Y can be defined as:

$$D = \sqrt{\sum_{n=1}^{N} (x_n - y_n)^2} \tag{7.1}$$

where the vectors contain N elements.

A Matlab program was then written that takes a potential input vector to the model, and then calculates the Euclidean distance of that to every other example in the data set. In order for the distance measure for each variable to be meaningful, it was necessary to scale all the variables with a minimum of −1 and a maximum of 1. This provides each variable with an equal significance in the distance calculation. The result of the program is that a vector of distances between the input vector and each of the data set examples is produced. Initially, the 'input' vector to the program

was selected at random, and in this example related to a MnCr steel, a histogram from the distances between this example and the other examples in the data set is shown in Figure 7.15.

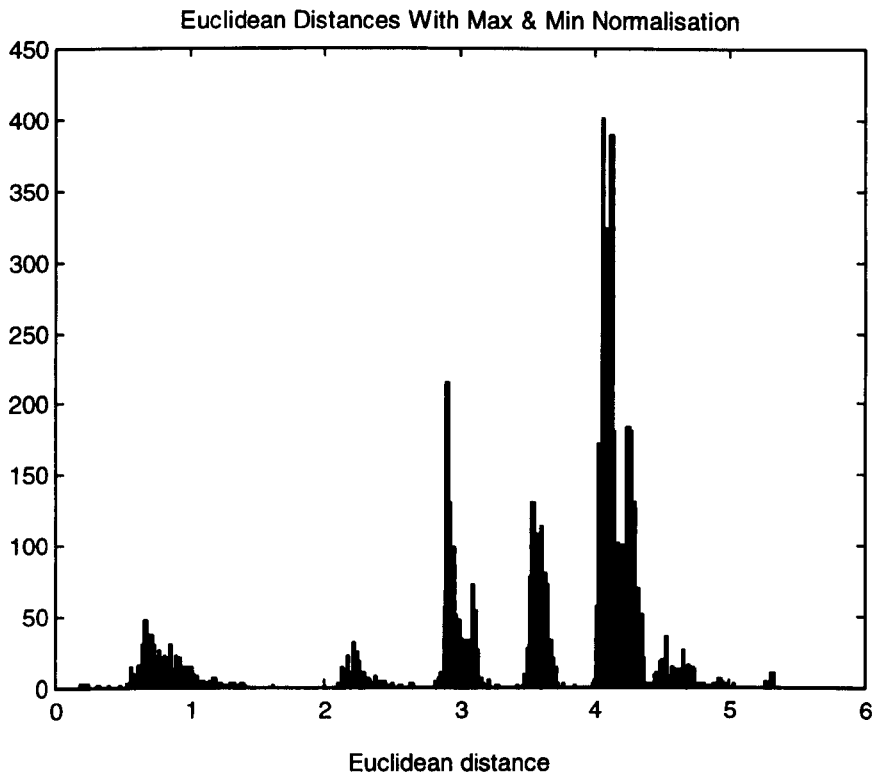


Fig. 7.15 A histogram of Euclidean distances between a MnCr steel example and the remaining data set.

Figure 7.15 shows that there are a number of 'clusters' of examples away from the example selected. The histogram also shows that the frequency of these clusters, i.e. the density of the data varies as well as the distribution. It was therefore thought because the weight update of the neural network is a function of error and the frequency of that error that, rather than just taking into account the distance of the input to the training data, distance should perhaps be weighted against frequency as well. Moreover, given that certain variables are believed to be more important that others in the calculation of UTS, then if the distance/density measure is used as an indication of reliability, each variable could also be weighted in terms of significance. The concept of this approach seemed very exciting, however, it was

found that a large amount of computation power was needed to perform the Euclidean distance calculation for every member of the training set. Whilst this was acceptable in off-line experimentation, it was thought that during online experimentation this would be prohibitive. One potential solution to this problem would be to perform clustering of the input data based upon a method such as the K-means method (described in chapter 2). The weighted distance to each cluster centre could then be calculated to provide an impression of the input space, at a vastly reduced computational intensity given that the histogram in Figure 7.15 suggests that less than 10 clusters exist with respect to a given data point. One problem that would still remain with this technique, however, is that the significance of the binary codes is difficult to gauge. Indeed with an insight into the project it was realised that certain grades of steel are made across a range of sites, whereas others may be restricted to one site. In other words, the relationship between site or indeed quench and prediction reliability may not be straightforward.

It was then realised that there was a more direct method of indicating the reliability of the predictions produced by the various models, which was integral to the neural network technique used to produce the final models in this project. As described in chapter 6, the final technique used to model the cleaned data was that of an ensemble of 10 individual networks, which were trained on the same data but with different random initialisation weights. The diversity between the ensemble members in this arrangement was therefore occurring due to the amount of significance the initialisation weights has on the final solution of a given area of the weight space. It was expected that this would therefore relate to the density and quality of the data on which the ensemble of networks was trained. This hypothesis was shown empirically on a two dimensional example initially. Ten networks containing 3 hidden layer neurons were trained on 16 data points generated from a noisy sine function for a

fixed number of training iterations. The mean and the standard deviation of the ensemble predictions were then investigated and are shown in Figure 7.16 together with the sine function and the data points generated.

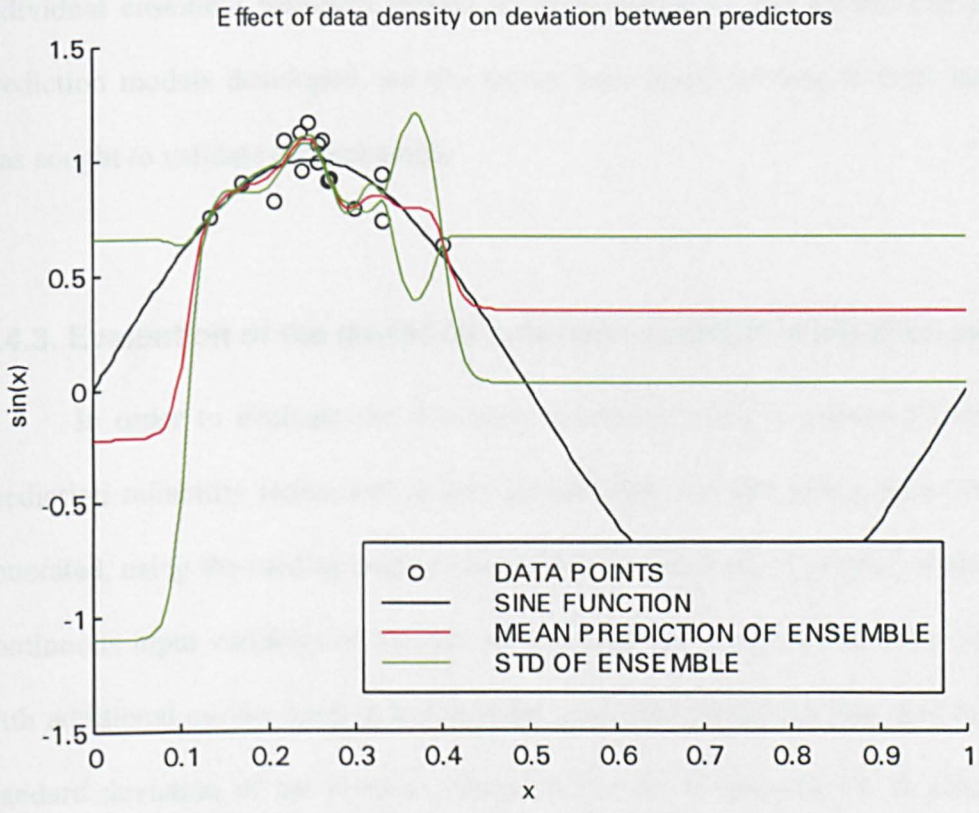


Fig. 7.16 A two dimensional example of the effect of data density on the deviation between predictions of 10 ensemble members.

Figure 7.16 shows that there is clearly a greater prediction deviation between ensemble members, where the data density is low in particular at each extreme of the populated region of the sine function and at the x=0.35 region. If one considers the mean prediction of the ensemble it is clear that there is an amount of overfitting, however the network was trained with a fixed number of training iterations. It is accepted that the deviation between ensemble members may vary with factors such as the number of individual networks in the ensemble and the architecture and training conditions of each net since, for example, networks which overfit their data may also be more effected by random initilisation weights. However, it was decided

that this technique may provide a good indication of the reliability of predictions for the ensembles trained in chapter 6 since it is a true indication of the uncertainty between the predictiors which provides valuable information about the mean prediction. It was decided that the behaviour of the standard deviation of the individual ensemble networks should be investigated for the mechanical property prediction models developed and the expert knowledge relating to their indication was sought to validate the technique.

7.4.3. Evaluation of the deviation between ensemble predictors method

In order to evaluate the deviation between ensemble members' method of prediction reliability indication, it was decided that variable effect plots should be generated, using the median analysis in Table 7.6. A full set of graphs relating to the continuous input variables of the five mechanical test result models were generated with additional curves relating to the mean ensemble prediction plus and minus the standard deviation of the residual, these are shown in appendix B. A selection of these graphs is presented in greater detail within this section to demonstrate the effectiveness of the technique.

Figure 7.17 shows the graph relating to the effect that tempering temperature has on UTS, this is the same as Figure 7.3 except that the deviation between predictors is now also shown. It can be seen that the deviation between predictors for tempering temperatures above 500°C is very narrow, particularly around the 630°C region, which relates to the most common tempering temperature for the 1%CrMo composition. The statistics of the cleaned UTS data was shown in Table 5.7. this showed that despite the tempering temperature range provided by the data being between 20-730°C, with a mean of 604°C the standard deviation of the tempering temperature was just 70. There are therefore only a small number of examples in the

data set (let alone relating to the 1% CrMo analysis), which have a tempering temperature lower than the 500 region. This data sparsity relating to the low tempering temperatures is indicated in Figure 7.17 by the deviation between the individual ensemble members.

Another example of the deviation between predictions relating to the distribution of

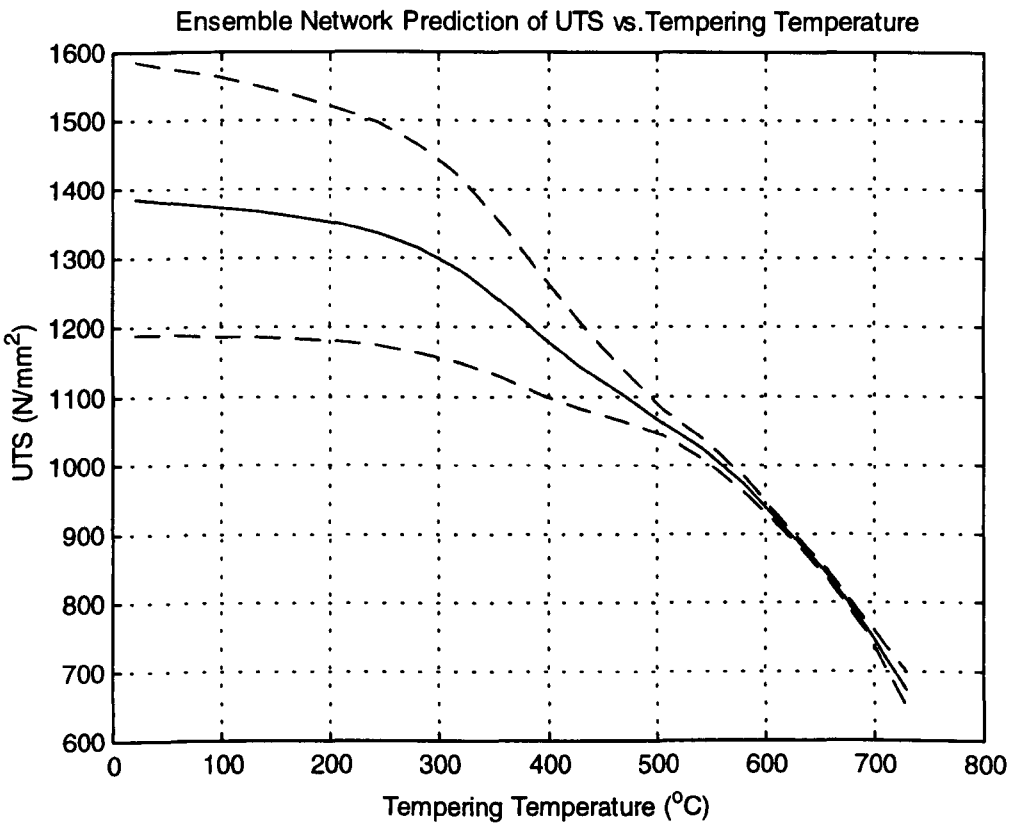


Fig 7.17 Effect of tempering temperature on UTS with SD of deviation between ensemble members marked as dashed-lines

the UTS data is shown in Figure 7.18. This shows the effect of carbon content on UTS and also shows the fact that for a 1% CrMo steel type, the data set contains examples with a good distribution of carbon additions. It can, however also be seen that the minimal amount of deviation between members exists at 0.41% C, which relates to the most popular carbon addition for a 1%CrMo steel. It can also be seen that at the extreme values of carbon addition the deviation between predictions is greatest. Metallurgists have commented that the 'error bands' produced by the

standard deviation of individual predictions of the ensemble, indicate the steel-making ranges used for the data in the models training set. The steel-making range is the range of additions and temperatures applied to a particular steel type.

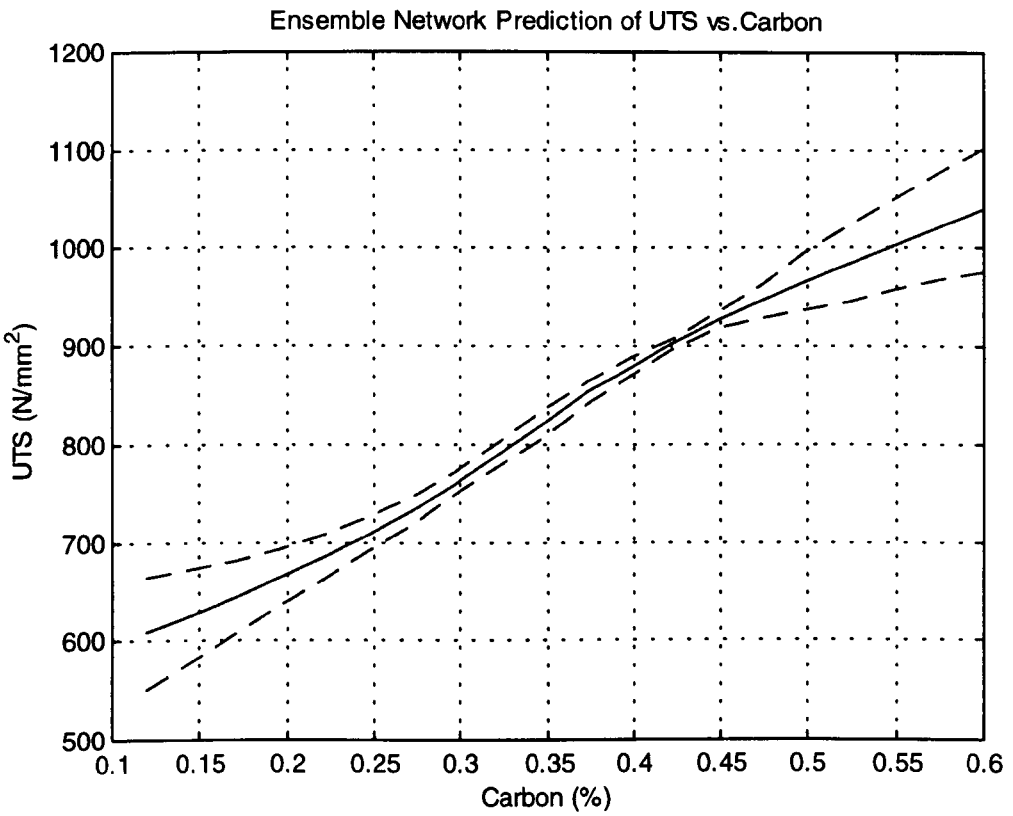


Fig. 7.18 Effect of carbon content on UTS with SD between individual prediction members marked as dashed lines.

The importance of having an indication of model reliability is particularly demonstrated with the next graph Figure 7.19, which shows the effect of chromium on UTS. It has previously been commented that the graph in Figure 7.5 shows an interesting feature that the effect of chromium appears to reach plateau beyond an addition of 2%. It can be seen from Figure 7.19, that this effect may not be as genuine as it first appeared. When the error bands are taken into account, it is realised that there are very few additions above 2% and therefore the predictions above that point will not be very reliable. The information provided by the error bands is therefore crucial to an accurate interpretation of the variable effects.

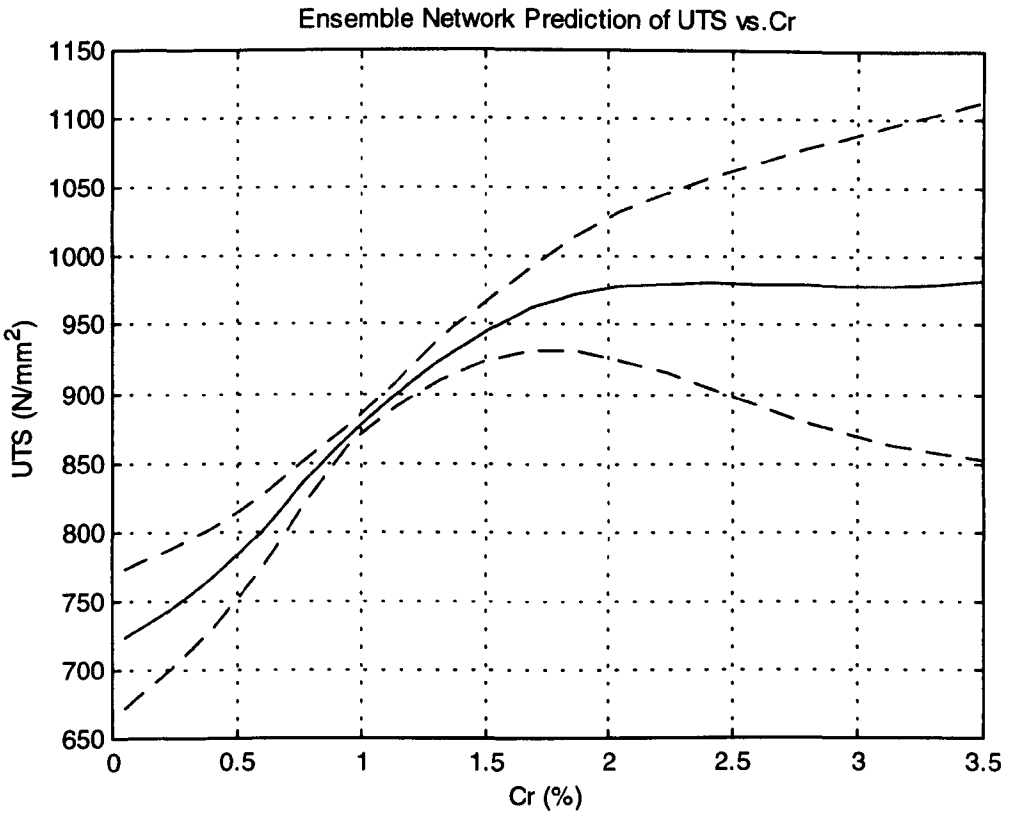


Fig. 7.19 Effect of chromium content on UTS with SD between individual prediction members marked as dashed lines.

The final graph shown in Figure 7.20 demonstrates a feature of the impact model, which was well known through knowledge to the data set, this is that the number of sub-zero impact test results available at the time of model construction was very limited. These points were, however included to provide useful information to the model, however, it was suspected that the reliability of the predictions generated by the model in this region would be reduced. The graph demonstrates this reduction in model reliability, and shows that although the user could accept a general downward trend, it would be preferable to limit predictions to the room temperature area for the 1%CrMo inputs, if the best reliability is to be achieved.

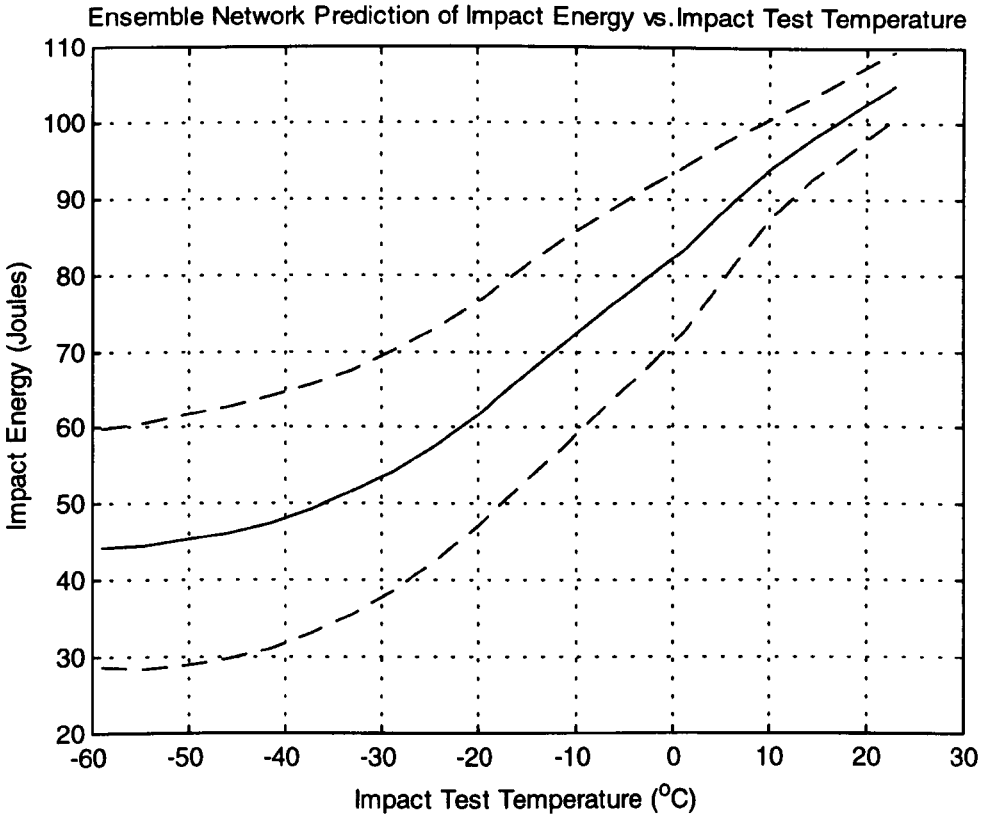


Fig. 7.20 Effect of impact test temperature content on impact energy with SD between individual prediction members marked as dashed lines.

7.4.4. The effect of binary codes on predicted values

So far, evaluation of variable effects has been limited to those variables which can be varied continuously, however it would also be interesting to note the effect that the binary variables have on the prediction.

Details of this investigation have been limited to just the UTS model in this section, however, it will be demonstrated that variation of the binary codes can be quite difficult to interpret, particularly if the deviation between predictors is not taken into account.

Using the 1%CrMo analysis, the quench codes were varied such that an oil water and then air quench was input into the model. The resulting mean ensemble predictions and their standard deviations are shown in Table 7.11.

Quench Type	Mean UTS prediction (N/mm^2)	SD of ensemble predictions
Water	890	6.9
Oil	805	9.7
Air	893	72.2

Table. 7.11 Effect of quench change on 1%CrMo input vector

After discussion with metallurgists, it was ascertained that if the same quenches were applied to a particular steel, then one would expect the quenches to act in the following order of decreasing cooling effect:

1. Water - Greatest
2. Oil
3. Air - Least

From chapter 2 it is known that the greatest cooling effect (after the hardening stage) will cause the greatest hardening effect in the end product (assuming that the tempering temperature and other parameters remain constant). However, from Table 7.11 it is apparent that this is not the case. If just the mean prediction is used, then it appears that the air-cooled steel has the highest strength, even though a metallurgical and thermodynamics understanding would suggest it to have the least effect. The reason for this is that the Air cooled steels occur very infrequently within the data set and relate only to one grade of steels, in fact there are only 53 air cooled steels within the whole data set of 5709 points. This anomaly is caused because Air-cooling is only actually applied to one type of steels within the data set. These steels are termed 'high-hardenability' steels and have a nickel content of 3-4%. A high hardenability means that the steel is easily hardened, and therefore it can develop a high tensile strength without water or oil quench. The examples of this class of steel have a high tensile strength and because this is the only example relating to the binary code for

air cooling, when the 1%CrMo steel is input with an air quench (which never happens in the real world because it is not a high hardenability steel) a higher prediction is obtained than that which would be expected. There is no easy way of knowing what tensile strength an air cooled 1%CrMo steel should have, however one can expect that the neural network prediction is unreliable by inspecting the standard deviation of the individual predictions of the ensemble. The water quench and oil quench have very low deviations, however the air cool causes a much larger deviation between the predictors indicating that the predictions are unreliable and that there is very little information regarding this area of the model.

The effect of site was also investigated for the 'median' inputs of the 1%CrMo analysis. The site input to the model was varied throughout the six possibilities available in the training data and the results are shown in Table 7.12.

Site	Mean Prediction (N/mm^2)	SD of predictors
Pearsons	892.8	17.6
Whithams	881.5	7.2
WestBank	890.6	6.9
Special Steels	875.8	7.5
Roundwood	934.7	19.6
Lab	909.8	9.9

Table 7.12. Effect of varying site code on mean and standard deviation of predicted values

The effect of varying the site codes is a lot harder to interpret that that of the quench codes. Ideally any site should be able to manufacture the same steel with identical mechanical properties, however systematic differences from one site to another mean that in practice this ideal situation may not occur. All predictions appear to be roughly the same values, however, using the SD of predictors as a reliability guide it can be seen that the sites where the least number of examples are from, i.e. Pearsons and Roundwood, have higher standard deviations of residuals confirming the data distribution. If these two results are therefore ignored, it is

noticeable that the lab result seems to be higher than the results for Whithams, West Bank and Special Steels. One possible explanation for this is that because the Lab uses smaller samples, as explained in chapter 3, they also use a shorter tempering time resulting in a harder property. Additionally, the quench effect on a smaller sample is more severe, this too will result in a higher UTS. Even though the bar size of the median analysis (the inputs from which these results were obtained from) was 180mm, which is not a lab sample, the systematic differences of heat treating with a shorter tempering time and more severe quench action appear to have been transferred.

7.4.5. Using reliability indicators for active data selection

One further experiment was to see if the error bands could be used to provide a method of determining where in the input space additional data should be selected in order to improve the model's reliability. It was anticipated that this could be assessed thoroughly by generating variable effects plots with error bands for every variable with every main steel type as a 'median' input vector within the data set (Table 7.10 details these groupings). One could then look for areas of the model that consistently showed a large deviation between predictors and then if it was possible to produce data in that area, actively search for or even specially heat treat some samples in that area of the input space to strengthen the model.

The principle of this can be demonstrated with reference to the impact transition curve of the 1%CrMo steel, which is shown in Figure 7.20. It has already been noted that there is a sparse sub-zero data region in the model, which is in turn causing a large deviation between predicted values. A narrow search was made within the BSES metallurgical database to find examples of 1%CrMo steel Charpy 2mm V impact results with sub zero impact test temperatures. In total 489 results

were found, which were mainly distributed at the West Bank and Special Steels sites with a water quench after the hardening stage. The statistics of the non-binary variables of this additional data set are shown in Table 7.13.

Variable	Max	Min	Mean	SD
Depth	146.05	12.50	30.95	21.60
Size	340.00	69.85	143.96	67.44
C	0.43	0.31	0.41	0.03
Si	0.34	0.18	0.24	0.02
Mn	0.99	0.51	0.87	0.14
S	0.03	0.00	0.01	0.01
Cr	1.20	0.95	1.06	0.04
Mo	0.24	0.16	0.23	0.01
Ni	0.29	0.07	0.19	0.04
Al	0.05	0.02	0.03	0.01
V	0.01	0.00	0.00	0.00
Hardening Temperature	900.00	810.00	866.05	6.77
Tempering Temperature	720.00	530.00	676.69	44.59
Impact Test Temperature	-3.00	-59.00	-32.71	6.48
Impact Energy	186.33	3.47	86.00	32.93

Table 7.13 Statistics of the additional data set selected to improve sub-zero impact prediction.

It can be seen that the majority of these results are at a test temperature around –30°C. These data were added to the original training set used to produce the graph in Figure 7.20, so as to 'boost' the model's sub zero performance. Ideally, the validation and test sets used to train the model should have been balanced in distribution to training data, however it was decided that on this occasion all the additional results would be used in the training set so as to maximise their effect.

The model was trained using ten ensemble members with the same architecture as previously used to develop the impact model. The statistics of the resulting model (using the same training and test sets as before) are shown in Table 7.14.

Impact data set	Training		Validation		Test	
	SD of error	RSQ	SD of error	RSQ	SD of error	RSQ
Cleaned Extra Set	13.86	0.82	19.05	0.68	19.3	0.66

Table 7.14 SD of ensemble error and r-square values for impact models trained and validated on cleaned and uncleaned data.

The results in Table 7.14 show a slight worsening in the statistical accuracy of the model with the addition of the extra data points when compared with the previous impact results in Table 6.12. This may be due to the widened input space of the model or due to a less optimal set of random initialisation weights occurring on this particular training of the ten ensemble members. The variable effect graph, shown in Figure 7.21, however, shows an improvement in shape metallurgically (an inflection

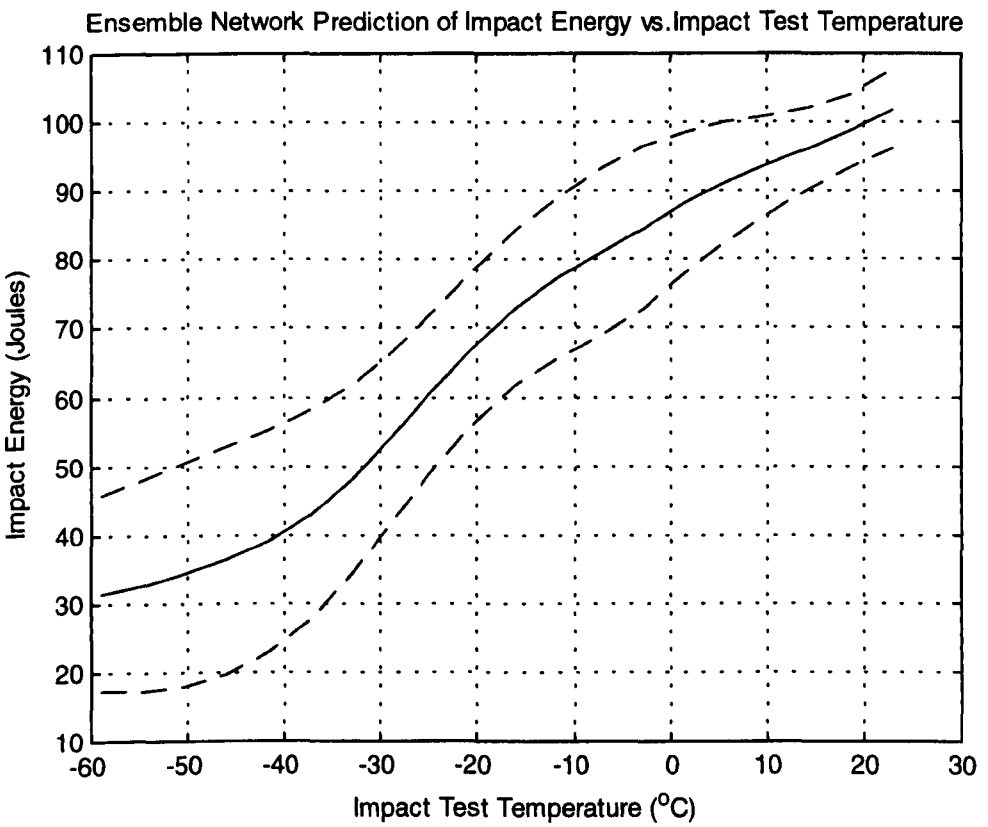


Fig.7.21 The effect of additional 1%CrMo test results to the median prediction and standard deviation between prediction members for the prediction of Impact energy against impact test Temperature

now showing around the transition region), and reduced error bands around the area of data addition.

7.5. Chapter conclusion

This chapter has investigated a variety of techniques for extracting data from the models developed in chapter6. Initially a technique to analyse the significance of variable inputs to the model was investigated, which utilised the data set used to train the model. One drawback with this method was that the calculation to determine the significance of a given input variable was affected by the distribution of the data used to train the model. Conversely, however, this might be seen as an advantage in allowing one to gain an overall impression of the most and least important variables in the model, over a range of steel types, during which the significance of key variables may change. The results of this data based sensitivity analysis method appeared to concur well with metallurgical knowledge of the process, helping to validate the function of the model.

The next method of extracting information from the model involved investigating the effects of individual variables around a given composition. For the purpose of the experiments, expert knowledge was utilised to enable a common composition and treatment pattern to be used as the fixed inputs to the model while each variable was independently varied. This method provided visual confirmation of the model function which correlated well with metallurgist's knowledge. The effect of chromium on UTS was, however, seen to be not as expected; an explanation for this later became apparent.

The variable effect was then extended to enable the generation of three-dimensional graphs, thus enabling the interaction effects between two variables on a mechanical test result prediction to be studied.

Having demonstrated two methods of extracting information from the models, other than just predicting a single mechanical test result value, a graphical user interface was developed to enable further evaluation of the model in an industrial situation. There were two important issues in the evaluation of the resulting software, the first was the ease of use and maintainability of the software and the second was the accuracy of the UTS model on trial. The GUI evaluation was made on a single UTS model, prior to the development of the ensemble technique. A number of suggestions relating to the usability of the package were raised, many of which could easily be implemented with future versions of the program. The accuracy of the neural model distributed was compared to a linear existing predictor already in use in one particular plant. In making a comparison between the neural and the existing West Bank predictor, it was found that the statistics used to compare the two techniques were important. The neural model tended to have a lower variance, whilst the linear model specific to the site had a lower bias. Having seen this, a neural model was developed which was trained only on data from the site of interest. This model had a lower bias than that of the linear site-specific model, however this was at a cost of variance. The conclusions of this real application of the model should be considered carefully when developing a model. In the case of this project, a wide-ranging model was required, and so a multiple site model was developed.

One significant point of feedback was that when the neural model was used to predict a steel grade which did not relate to any of the steels used to train the model (but which was still within the maximum and minimum range of the model) a guide to the reliability of the predictions was required. It was found both in the literature and in practice that when a neural network extrapolates or interpolates on an input vector that is in a sparse area of the training data spurious predictions may result. It

was realised through the evaluation work with BSES that serious problems may result if the user of a model was not aware when an input inquiry to the model fell in a sparse area of the training data. Experimentation therefore began to find a suitable method for providing an impression of data density and therefore reliability of model predictions. One method that was particularly suited to this project, given that an ensemble approach had already been adopted, was to look at the standard deviation of the ten ensemble members' predictions. It was shown on a two dimensional example that when in a sparse area of the model, ensemble members trained using random initialisation weights (but the same training data), deviate more from one another in areas of low data density. This technique was then applied to the predictive models developed in chapter 6, and the performance of the resulting error bands was demonstrated with key examples involving the UTS and Impact models. It was seen that the 'mysterious' plateau effect of chromium on UTS beyond a 2% addition may not have been as significant as first thought since the error bands indicated that there were little data above the 2% region and therefore the reliability of the predictions should be questioned.

Interpretation of the effects site and quench codes was also made possible through the use of the standard deviation of the individual predictions, where predictions which would appear confusing without a reliability guide became understandable once the data density was taken into account.

Finally, it was postulated that the deviation the between predictors may provide a good indicator as to where in the input space additional data for the model should be selected from (if available). An example of this was given where extra Charpy 2mm V impact results were obtained. Despite no statistical improvement in accuracy, a noticeable improvement in the shape of the impact transition curve was

seen, together with a reduction in the error bands around the sub zeros region of the model which had been 'filled' with data.

The next chapter now investigates application areas of the models developed, and demonstrates how the UTS model can be used to ascertain the effect of measurement tolerances on the resulting tensile strength of a steel. The model application area of alloy design is also considered, and some preliminary work to find an optimum composition and treatment temperature which produces a steel with user defined mechanical properties is investigated using the genetic algorithm approach.

Chapter 8

Application of the models for process optimisation

8.1. Introduction

The ultimate aim of developing the mechanical test result models is to facilitate process optimisation (as was explained in chapter1). This chapter investigates the possible application areas of the models within industry and then demonstrates several investigations of these applications. The first investigation starts to look at the UTS model's ability to generalise to a new process, involving data from steel types dissimilar to the training set it was developed from. The second investigation then uses the UTS model to see what effect measurement inaccuracies in the process have on the UTS at a given composition. Finally, a preliminary investigation into optimal alloy design is considered using UTS and ROA models as examples. This work uses the Genetic Algorithm (GA) approach which is applied to a set of input variables that will produce pre-specified mechanical test result values.

8.2. Model application areas within industry

It is anticipated that the models developed in this project could have the following applications within industry:

- Optimal design of new alloys under constraints
- Validation of process measurement requirements
- Elimination of testing requirements on common steels
- In the heat treatment works
  - Fault detection
  - Setting required treatment parameters including tempering time

The first application, to design an alloy which aims to meet one or more pre-specified requirements, such as strength, ductility or cost will be discussed and then examined in detail in section 8.5. It is worth noting at this stage that metallurgists already experiment with alloy compositions in order to meet pre-specified targets or to reduce the cost of a steel. However, given the complexity of the relationship between the variables in the process, an automated more methodical way of 'searching' for new alloy and treatment combinations would be desirable.

Exploration of the effects of measurement tolerances on the predicted UTS will be examined further in section 8.4. It is expected that, having developed a model of the process, the effects of certain scenarios such as measurement inaccuracies could be estimated via simulation of the model for a given set of input values.

It was mentioned in chapter 3 that mechanical tests are often performed on steels which are supplied to other companies (such as Aurora Forgings) in an 'as rolled' state. This means that the company who purchases the steel is planning to perform their own heat treatment of the metal and that BSES has to lab treat a sample of the steel to prove that, given a certain treatment regime, the required UTS (and possibly other test results) could be obtained. It is postulated that testing time and costs could be reduced if commonly made steel types (i.e. those which the model can predict accurately), could be supplied with predicted mechanical properties instead of tested ones. This idea obviously carries assumptions that the treatment progressed as documented in the model's inputs.

In the heat treatment Works it was also considered that the model could be used again on commonly manufactured steel grades, as the basis of a fault detection system. The composition and treatment temperatures could be entered for a particular treated steel and then, if the test results relating to that steel differed significantly,

the possibility of a process fault could be investigated. The model may also be useful in the plant situation (without automatic optimisation techniques) for determining the tempering temperature which may be required for a given set of input parameters to the model. It has already been seen how graphs of the effect of tempering temperature can be produced, which would enable the determination of a tempering temperature for a given set of variable settings. An extension of this may also be to use the tempering parameter by Holloman and Jaffe[59] to allow for the predicted results to be supplied according to varying time and temperature combinations (as explained in chapter 4).

8.3. Ability of the UTS model to generalise to 'new' processes

Ultimately in the application of the model, it may be desirable to use the model to predict UTS values on a new process or treatment site. An investigation would obviously need to be made in terms of determining the prior thermo-mechanical treatment of the steels involved in the process (such as forging), together with any other difference in the process that may exist including the distribution of the treated steels at that site. Assuming that the process was similar in terms of stages, it was decided that it would be interesting to see if the UTS model developed could predict the test results of steel specimens treated at another site. An initial investigation into the model's capability was made using results obtained from a range of steels treated at a different laboratory to that where the existing lab data set was created. The compositions of some of these steels were now very different from that of the 1%CrMo median analysis used for model investigation up until this point. In total, 5 different steel compositions treated at varying tempering temperatures were used in the investigation. The main alloying additions relating to each steel type

are shown in Table 8.1. For all steel types, a bar size of 30mm and a test depth of 15mm was used, all treatments also used an oil quench after the hardening stage.

Steel Type	C	Si	Mn	S	Cr	Mo	Ni	Al	V
37Cr4 (Ref.1)	0.37	0.16	0.57	0.02	1.1	0.05	0.21	0.046	0.005
30NiCrMo8 (Ref. 58)	0.31	0.16	0.47	0.021	2	0.42	2.01	0.031	0.005
30NiCrMo8 (Ref. 59)	0.29	0.33	1.46	0.016	1.64	0.06	0.23	0.035	0.005
30NiCrMo8 (Ref. 60)	0.32	0.26	1.3	0.017	1.65	0.14	0.21	0.037	0.005
30NiCrMo8 (Ref. 61)	0.33	0.33	1.34	0.017	1.66	0.05	0.21	0.039	0.1

Table 8.1. Compositions of the five steels treated at the 'new' lab process

Measured UTS values were obtained for the compositions shown in Table 8.1 hardened at 870°C, and tempered at a range of temperatures of 500, 550, 600 and 650°C. This enabled a plot to be made for each of the compositions tempered at the 'new site' which showed the effect of tempering temperature on UTS. The neural model was then used to generate a similar set of values, such that the fit of the model to the unseen process could be established.

The site code input of the model was varied to determine the best fit of the predicted values to those measured from the new process. It was found that the best fit on the new data (the smallest prediction error) was obtained when the binary inputs denoting 'Lab' were used as site inputs to the model. This is reasonable since the scale of the furnaces and treatment methodology used in the new process and the lab data in the training set should be similar. It was anticipated that, if a new plant was to be used as part of the prediction process in a real situation, a sample data set representative of the new process could be used to determine which site code setting would be suitable for the new plant.

The ensemble predictions for the first two compositions together with the measured values (ref. 1 and ref. 58) are shown in Figure 8.1.

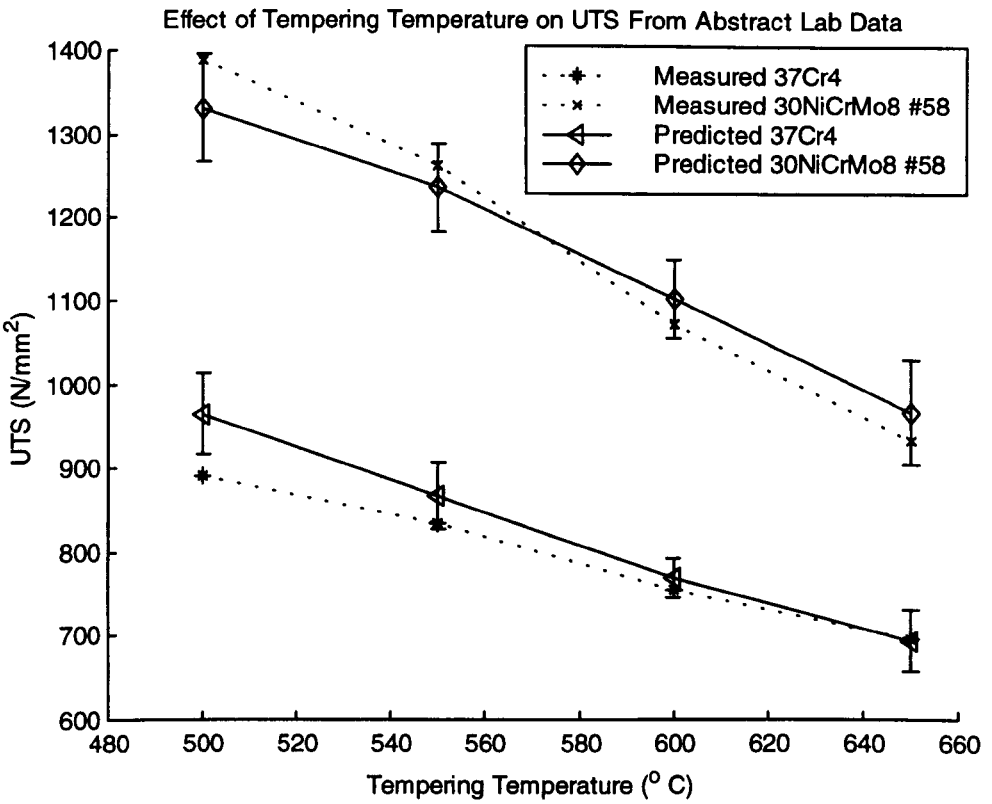


Fig 8.1 Ensemble predictions of data from a new treatment site for the 37Cr4 steel and 30NiCrMo8 steel (ref.58) tempered at varying temperatures

The two steel types for which the predictions are shown in Figure 8.1 are not 'common' steel grades in the sense of the 1%CrMo analysis. However, metallurgists have commented that similar compositions to these steels do exist with the data set. The predictions have been generated using the mean ensemble prediction, and the error bars show the SD of the individual ensemble members' predictions. It can be seen that the predicted values of the model fit quite well to this unseen process. The error bars, particularly for the 30NiCrMo8 steel, indicate that there is significant deviation between predictors which accommodate the measured values, signifying the less common nature of this steel compared to the 1%CrMo analysis previously seen. The 37Cr4 predictions above 550°C, particularly at the 600°C tempering

temperature appear to indicate a more reliable prediction than with the 30NiCrMo8 steel, however the point at 500 °C seems to be somewhat miss-predicted.

The next three steel analyses to be predicted (ref. 59, 60 and 61) are in the 'family' of the 30NiCrMo8 steel type. However, they were actually produced as part of an experiment by a metallurgist at British Steel, to investigate the possibility of using different combinations of alloy addition in order to produce similar mechanical properties at reduced cost. These three 'variants' on the original type 58 steel were not expected to appear in the data set as training examples, but do lie within the maximum and minimum ranges of the model. The predictions for these variant compositions are shown in Figures 8.2, 8.3 and 8.4.

The graphs have been shown on separate Figures for clarity, however similar axis ranges have been used to enable a comparison between the prediction accuracy

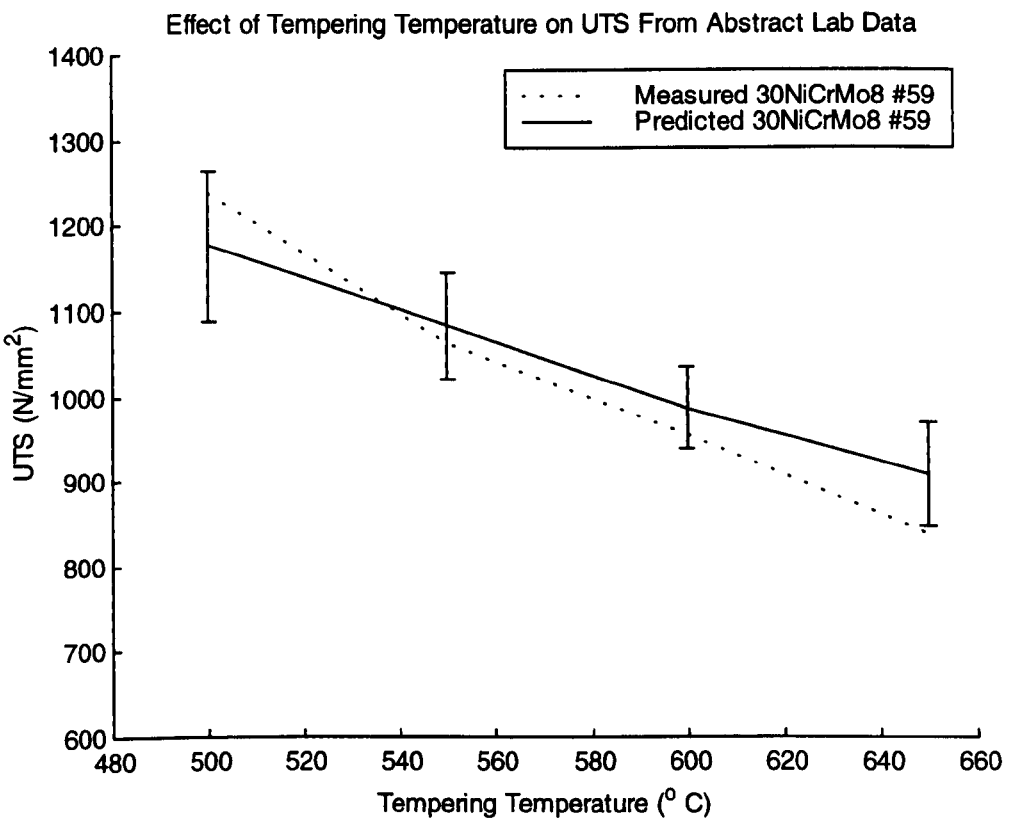


Fig. 8.2 Predicted and measured effect of tempering temperature on UTS for the type 59 variant of the 30NiCrMo8 steel treated at a new site.

in each case.

The graphs in Figures 8.2 and 8.3 for the type 59 and 60 variants of the 30NiCrMo8 analysis show that a reasonably good fit has been obtained from the predicted values. The error bars indicate that the standard deviation between the individual predictors is higher than with the more common 37Cr4 and type 58 steel, however, they appear to indicate well the uncertainty in the model's predictions. The predictions in Figure 8.4 show a somewhat worse situation for the type 61 steel. The predictions are quite poor, and the error bars are, generally, the largest of all the plots. When the composition for this steel (ref. 61) is examined it can be seen that it contains a vanadium addition. This explains why the predictions are so bad, since there are not many examples of vanadium steels in the training data set anyway, and so a novel composition with a vanadium addition produces poorer predictions. The error-bars provided by the ensemble technique are not quite big enough to

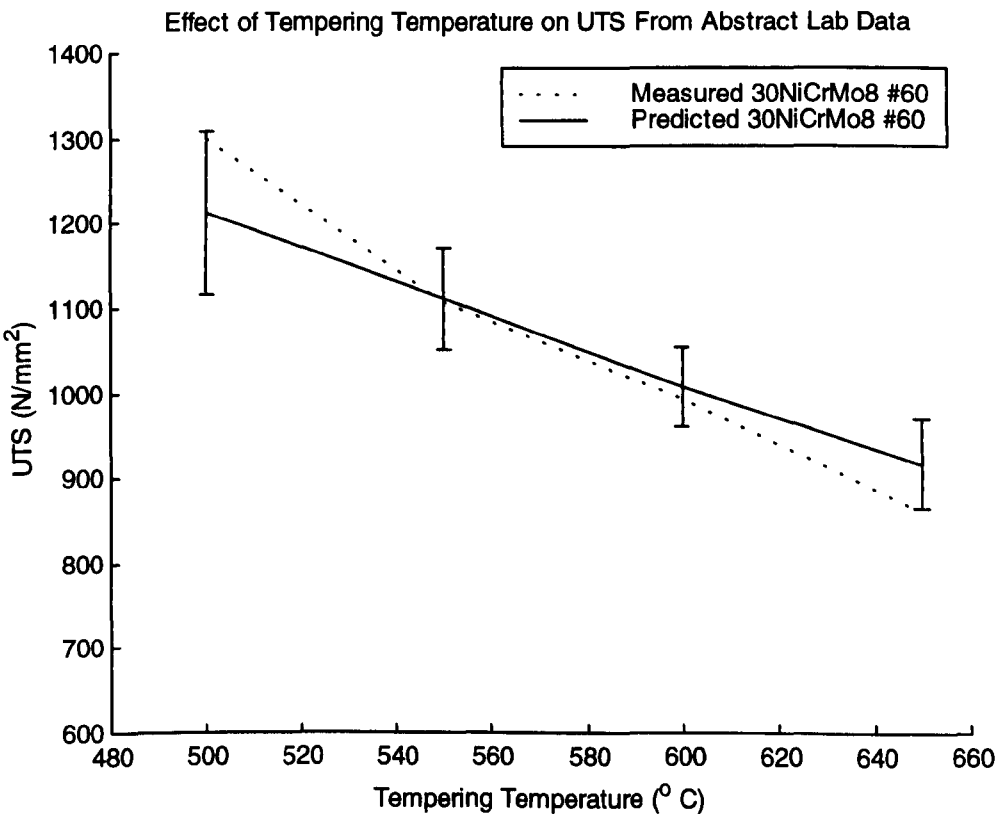


Fig. 8.3 Predicted and measured effect of tempering temperature on UTS for the type 60 variant of the 30NiCrMo8 steel treated at a new site.

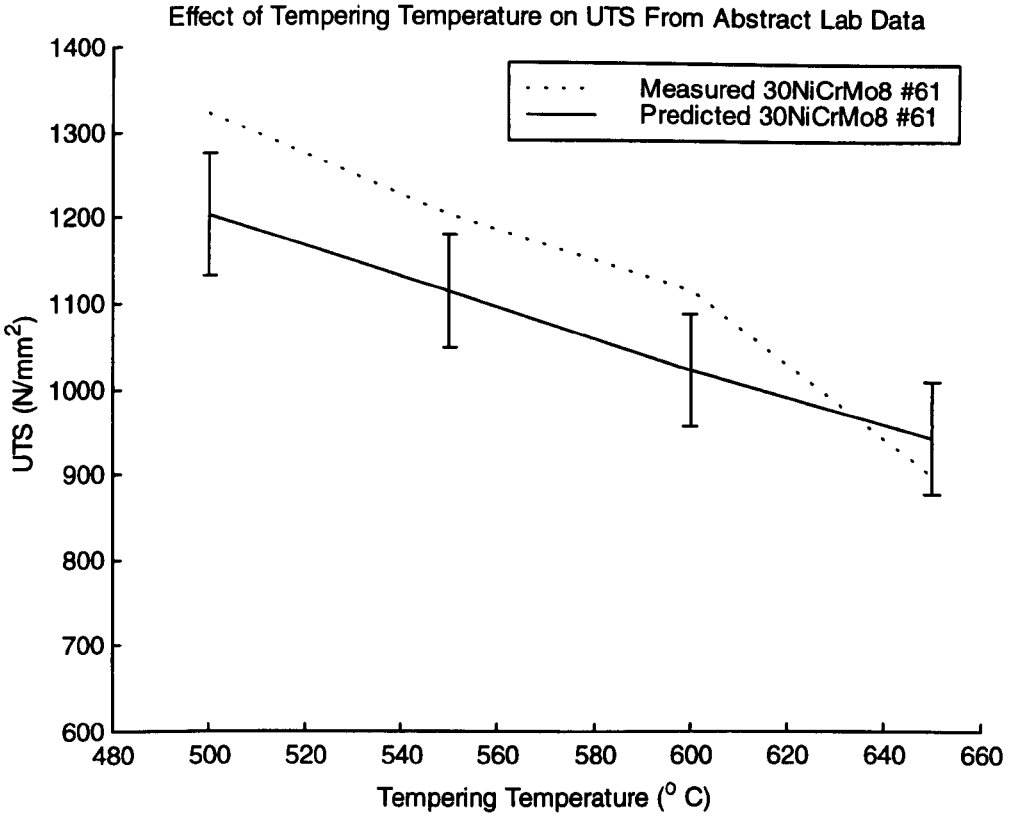


Fig. 8.4 Predicted and measured effect of tempering temperature on UTS for the type 61 variant of the 30NiCrMo8 steel treated at a new site.

accommodate the error in prediction in the case of the type 61 steel. It should be remembered, however, for all predictions that the process is different to that from which the training data came, and therefore one cannot expect that the uncertainly between the individual predictor values will definitely accommodate for the prediction error. It is also important to note that, although the SD between individual predictors in the ensemble appears to provide an accurate guide to the model prediction error, this has only been observed empirically, and is primarily used as a guide to data density.

8.4 Assessment of effects of measurement tolerances on predicted UTS

This section briefly looks at how the neural models can be used to assess the effect of measurement inaccuracies in the process, on the predicted UTS values of the model. There are three main sources of error in the predicted value, the first is

that caused by the measurement of the variables in the data set (inputs and outputs), the second is the unknown effects error (factors influencing the UTS which are un-determinable) and the third error is the fitting error of the model to the data. The following prediction accuracy assessment should however provide an estimation of the sensitivity of the model to the measurement errors in the process.

The tolerance (2σ) of each input variable's measurement was established in the process familiarisation stage, and was shown in Table 3.3. Note that the binary values carry no measurement tolerance. These values could now be used to determine the effect of the measurement tolerances on the UTS. If one is using a linear model, the effect of the variance of each input can be calculated arithmetically. However, with the neural model this is not the case since the relationships between the inputs and outputs are non-linear and contains interactions between the variables. It was therefore decided that the effect of input measurement tolerances on a certain composition could be calculated for a fixed composition by creating a distribution of errors around the input variables for that composition, geometry and treatment regime. The 1%CrMo median analysis (Table 7.6) was used as the basis for this experimentation. Measurement noise is typically normally distributed and a function within the Matlab statistical toolbox allows one to generate a random distribution of N points, which follow the normal distribution with mean μ and standard deviation σ. This function was used to generate 6000 data points with the mean of the 1%CrMo inputs and a standard deviation of one half the tolerance values shown in Table 3.3. A large number of points were generated to ensure that the data set followed the normal distribution. This was clarified by finding the mean of the 6000 generated examples and making sure it equalled the 1%CrMo inputs.

The effect of the measurement tolerances on the model inputs could then be investigated in a variety of ways using this data distribution. Firstly it was decided

that the measurement effects (size and test depth), compositional effects, and temperature effects (hardening and tempering) would be investigated separately. This would show which set of measurements had the most effect on the predicted UTS values. This was implemented by only using the normally distributed error data as inputs to the model for the variables under consideration. The remaining variables in this 'simulation' data set were kept constant at the 1%CrMo median values. For each part of the experiment, a distribution of 6000 predicted UTS values was generated from which the standard deviation of the predicted value could be calculated, which represented the effect of the input variation (measurement tolerances) on the output. The effect of the measurement tolerances on all of the relevant inputs was then also calculated. The results of this experiment, together with the effect of varying all of the affected input variables are shown in Table 8.2

Measurement variation in the simulation	**SD of predicted UTS value**
Size measurement	0.138
Compositional	17.3
Temperature	5.46
All non-binary variables	18.4

Table 8.2 Standard deviation in predicted UTS caused by individual and combined measurement inaccuracies.

The results show that the greatest effect on the predicted UTS value of the 1%CrMo analysis is caused by the inaccuracies involved in measuring the composition of the cast analysis, and that the size measurement tolerances have least effect. It has been found from experience that the measurement tolerance effect on each variable varies depending upon the composition of the steel.

8.5 Optimal design of alloys using genetic algorithms

This section details some preliminary work which has been undertaken to investigate the feasibility of using an optimisation technique such as genetic algorithms in combination with the neural network models developed in chapter 6, with the objective of finding an optimal set of process inputs in some sense.

The genetic algorithm has been selected as a suitable optimisation technique to use for this experiment because, as will be explained in the next section, it requires no information on underlying differential equations.

Examples of an objective for which an optimal set of parameters may need to be found may include simply a target tensile strength, or a combination of mechanical properties. In the case of finding a combination of mechanical properties the results from chapter 6 should be remembered, since these show that two types of mechanical test result (for example ROA and UTS), may have conflicting variable effects. The real advantage of using a genetic algorithm in such as situation as this, is that an optimal set of parameters may be found that gives a material with, say, the best ROA and UTS possible in the same steel.

Before presenting details of the work which has been undertaken it is first necessary to explain the key principles behind the genetic algorithm optimisation approach.

8.5.1 Introduction to the Genetic Algorithm approach

Genetic Algorithms (GA) are exploratory optimisation methods that stem from nature's principles of evolution and population genetics. The technique was first developed in 1973 by Holland[103], and since its first development the technique has been used for a large number of research studies, a detailed review of the technique and its applications being given by Goldberg[104]. It has already been stated

that the GA approach does not utilise differential equations to perform a gradient-based search, this makes it particularly suitable to use with the neural network approach. Instead of gradient descent the GA approach uses a 'fitness function'. The fitness function is used to assess the suitability of a particular set of values to solve the problem. For example how close is the UTS value generated by the model to that which was required if the inputs to the model are set to particular values. Initially, the GA approach generates a random population of possible candidate solutions to the problem. One can think of this as guessing a variety of different input combinations to give the desired solution out of a particular model or function. In GA, the initial population of candidates is encoded using binary values such that it is a well-defined number of chromosomes. Having generated the initial population, the candidates are

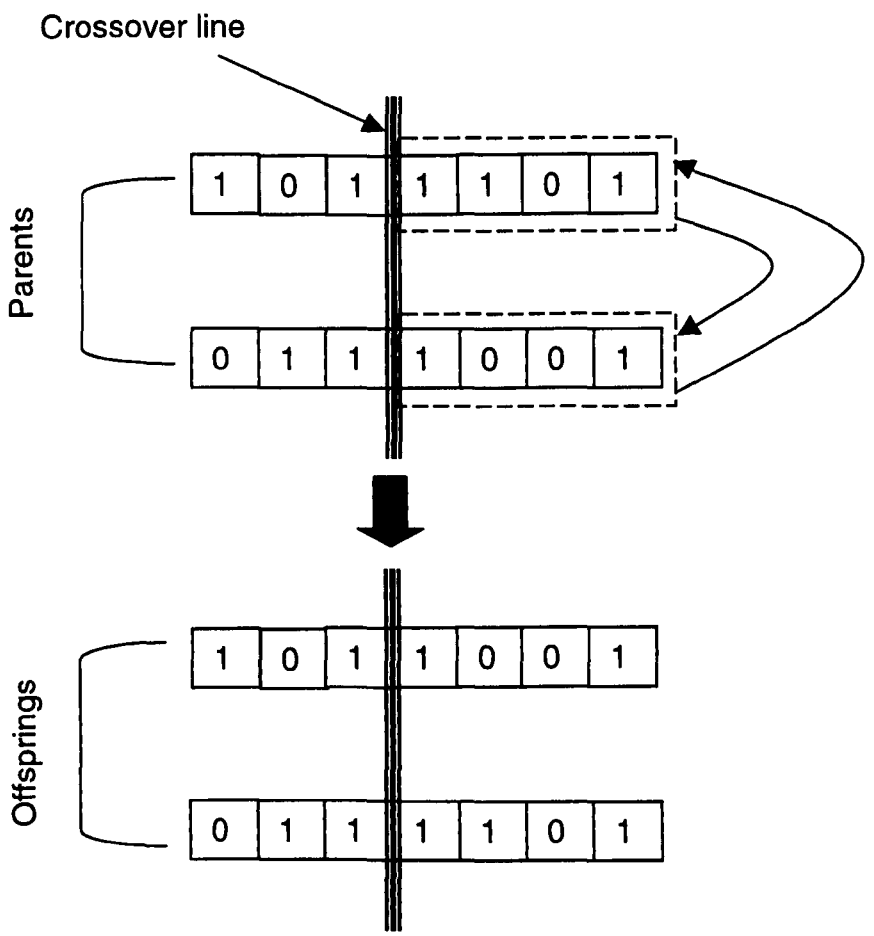


Fig. 8.5 The genetic operation of crossover

then ranked according to the particular fitness function that was devised to solve the problem. The initial population is generated at random, so all candidates will probably perform quite badly, however due to the random nature of the candidate generation, some will be better than others. The population of available candidates is then improved by a 'mating' process, where the fittest individuals in the previous generation are randomly combined in pairs to produce an 'offspring' (a new candidate which results from the two parent candidates). The candidates are combined by 'crossing over' parts of their chromosomes at a randomly chosen position of the string, as shown in Figure 8.5.

More excitation is given to the process through the use of the 'mutation' operation shown in Figure 8.6. The mutation procedure involves randomly selecting and inverting bits in the chromosome strings, this process helps to speed up convergence and provides an increased variety in generated candidates.

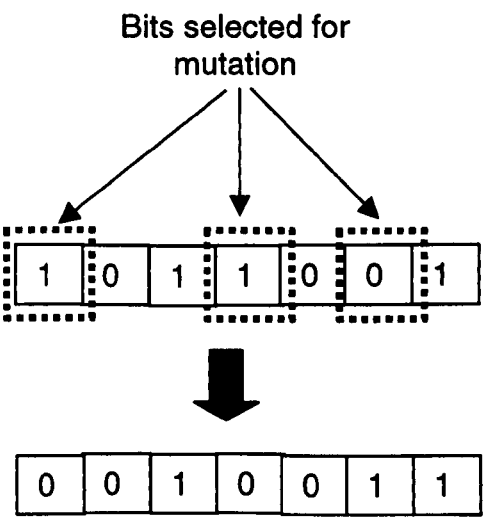


Fig. 8.6 The genetic operation of mutation

The selection of the GA parameters such as population size, the nature and rates of cross-over and mutation and the reproduction rate (the process by which parent

chromosomes are selected), all play an important part in the successful running of the GA algorithm.

The following steps summarise the function of the GA algorithm:

1 An initial population of trials, $\Pi(0)=A_m(O)$, $m=1,\ldots,M$, are generated, where M is the number of trials in the population.

2 For successive sample instances:

- The performance of each trial, $\mu(A_m(t))$, $t=0,1,\ldots,T$, is evaluated and stored.
- A number of trials are selected by taking a sample of $\Pi(t)$ using the probability distribution:

$$\rho(A_m(t)) = \frac{\mu(A_m(t))}{\sum_{i=1}^{M} \mu(Ai(t))} \qquad (8.1)$$

- One or more of the genetic operators are then applied to the selected trials to produce the offspring, $A_m^o(t)$, $m=1,\ldots,N$, where N is the number of offspring which is typically equal to the number of selected trials (parents)
- The next generation of population, $\Pi(t+1)$, is formed by selecting $A_j(t)\in \Pi(T)$, $j=1,\ldots,N$ to be replaced by the offspring, $A_j^o(t)$. A variety of criterion for selecting which trials should be replaced can be used including random selection or selection on the basis of fitness.

3 The GA algorithm is terminated after a pre-specified number of generations or according to a convergence criterion of the population.

The procedure of reproduction can replace members of the old generation. The method of choosing an individual for the production of offspring also determines its life span and the number of its offspring. Holland[103] showed that if ρ_l is the

probability that an individual $A \in \Pi$ is selected to produce offspring during a sample step and ρ_2 is the probability that it will be deleted during that sample step, then the expected number of offspring of A is given by $\frac{\rho_1}{\rho_2}$.

The most common types of reproduction techniques are Generation Replacement (GR), Steady-State(SS), Generational Gap(GG), and Selective Breeding(SB), however the SB technique was the only method used in this investigation. Selective breeding was used because it overcomes some of the shortcomings of the other methods, however a thorough evaluation of reproduction techniques was not part of this investigation.

8.5.1.1 The selective breeding reproduction technique

The selective breeding procedure is as follows:

1 An initial population $\Pi(0)$ is created in the usual manner.

2 The population is evaluated to determine the fitness of each individual.

3 For successive generations thereafter:

- An entire population $\Pi^o(0)$ is produced by selecting parents and applying the genetic operators.
- The offspring of the population are evaluated
- The next generation of population is obtained by choosing the best M individuals from both $\Pi(T)$ and $\Pi^o(T)$.

Each individual chromosome in the population is a potential solution candidate to the optimisation problem under consideration. The procedure of evaluation of these candidate solutions therefore consists of submitting each one to the simulation model

(in this case a neural model) and returning an assessment value in terms of the pre-defined fitness function.

The next section describes how the GA was used with two of the neural models developed in chapter 6.

8.5.2 Procedure for using the GA with the neural model

Implementation of the GA algorithm described previously was available within the Intelligent Systems Laboratory, and so by linking this program to the neural models developed within the Matlab environment, the feasibility of this idea could be investigated. It was decided that initially the GA would be used to find a set of input values to the neural model which gave a certain target UTS value. As has been explained throughout this thesis, the models developed contain a large number of inputs (22 in total for the UTS model), and it has been seen, both in chapter 3 and through the modelling work, that there are many factors which can influence the UTS of a steel. It is possible that the GA could operate on all inputs to the model. However, it was decided that the number of 'free' parameters of the model (i.e. those variables that the GA would be able to adjust) would be limited to five variables in order to reduce the computational burden.

Having performed sensitivity analysis of the models in chapter 6 it was decided that the variable set which the GA would adjust should include carbon, manganese, chromium, molybdenum and tempering temperature. For each variable, a binary term containing 12 bits was generated, which had a maximum and minimum value for each variable according to those defined in Table 5.7. This meant that the total chromosome comprised 60 bits, and was arranged as shown in Figure 8.7.

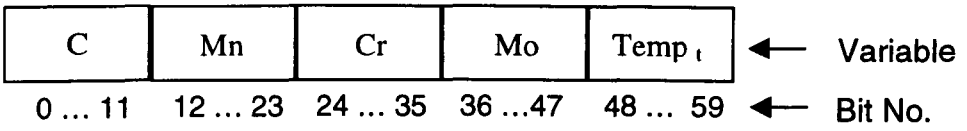


Fig. 8.7 Construction of the chromosome representing the 5 model inputs

The next decision to make was deciding how to deal with the other inputs to the model whilst the GA was adjusting the five variables in Figure 8.7. The remaining inputs, despite not being selected as the most significant variables in the optimisation process, would also affect the UTS values of the steel. For this reason it was important that the values to which the remaining variables were set so they would not prevent the GA from reaching its objective. It was decided that if the remaining input variables were set to that of the median 1%CrMo values (Table 7.6) and the target value was set to that which was found within the database as the median UTS result for the 1%CrMo analysis, then it should definitely be possible for the GA to set the free parameters in order to reach the target UTS value. There would, however, still be a variety of combinations which the free variables could take and therefore the set of values found by the GA for these free variables may not be those of the median analysis, however this would still be a valid solution according to the model.

8.5.2.1. Using GA to find a target UTS value

The first experiment performed in this investigation was to see if the GA could find the values for the free variables; carbon, manganese, chromium, molybdenum and tempering temperature, which would give a UTS result of 868N/mm^2. The remaining model inputs were kept constant at the 1%CrMo analysis.

Table 8.3 shows the probability values of the GA operators that were used for this preliminary investigation. A population size of 50 candidates was chosen.

Operator	Probability (%)
Crossover	0.95
Mutation	0.06

Table 8.3 Probability of genetic operator values used for the GA experimentation

Initially, the following objective function was used to determine the fitness of each candidate:

$$J_{UTS} = (x_t - x_c)^2 \tag{8.2}$$

where x_t is the target UTS value and x_c is the UTS value of the candidate solution. The genetic algorithm was allowed to run until the UTS target value was reached, and the values of the five free variables were stable. The trajectories of the five variables produced by the subsequent generations are shown in Figure 8.8. This shows the adjustments made by the GA with respect to objective function defined in equation (8.2). Plots of the objective function and the UTS values obtained over successive generations are shown in Figure 8.9.

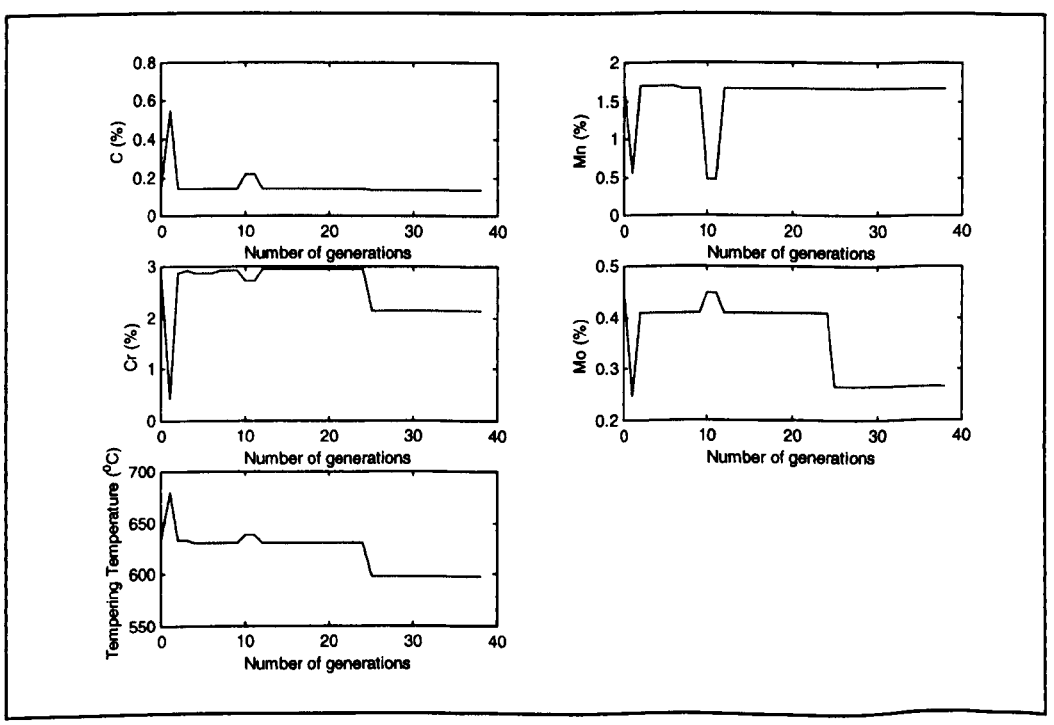


Fig. 8.8 Values of the five inputs adjusted by the GA plotted over successive generations

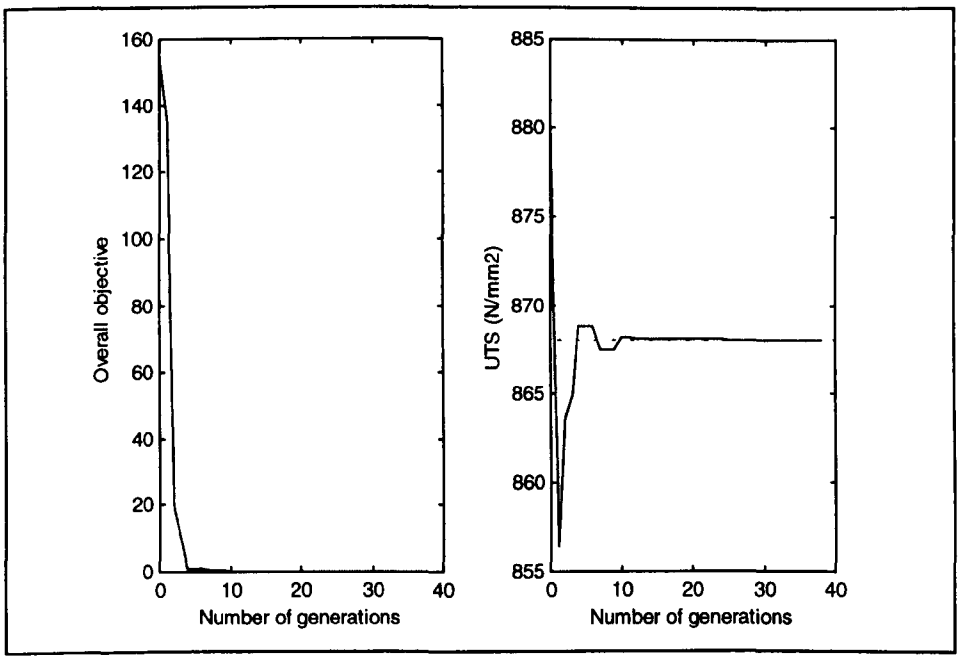


Fig. 8.9 Overall objective and resulting UTS values over successive generations of the experiment

Figure 8.9 shows that the overall obective was reached within a small number of generation cycles of the algorithm. The UTS target (marked with a dotted line), was also reached within 10 generations, however the sensitivity of this to subsequent geneation changes is more apparent than with the overall objective.

Looking at the adjustment of the variables (Figure 8.8), it can be seen that despite the overall objective and therefore target UTS being reached within 10 generations, the GA has provided a number of solutions which meet the target value. This is evident because the variable trajectories continue to move between solutions through the experiment. Having reached the overall objective the experiment was terminated after 38 generations, since a solution had been reached and without any other requirements in the objective other than UTS, no other criterion were to be fulfilled. The final composition and UTS provided by the GA's solution is shown in Table 8.4, together with that of the median analysis.

Variable	GA Adjusted Values	1%CrMoValue
Site (1-6)	3	3
Size (mm)	180	180
Test Depth (mm)	12.7	12.7
C (%)	**0.13**	0.41
Si (%)	0.27	0.27
Mn (%)	**1.66**	0.78
S (%)	0.023	0.023
Cr (%)	**2.13**	1.08
Mo (%)	**0.27**	0.22
Ni (%)	0.19	0.19
Al (%)	0.027	0.027
V (%)	0.005	0.005
Hardening Temp (°C)	860	860
Tempering Temp (°C)	**597**	630

Table 8.4 The composition found by the GA to provide a UTS of 868 N/mm^2 with the adjusted variables in bold type, compared with the median 1%CrMo analysis

Table 8.4 shows that despite the fixed variables being the same as the median analysis, the GA has provided a (non unique) solution which is quite different to that of the median analysis. The main differences compared with the median 1%CrMo analysis is that the GA has used much less carbon (which would reduce UTS), and yet dramatically increased the amount of chromium and manganese, in order to increase the UTS at an approximately similar tempering temperature. The question is, in reality would a metallurgist be able to create steel with the target UTS value given this composition. No similar examples of this steel composition were found in the database, and therefore there may well be metallurgical (let alone cost) reasons for not using this analysis. With reference to cost, carbon is regarded as a relatively cheap element and chromium a relatively expensive one. Taking this alone into account, without any of the complex metallurgical reasoning that may exist, it would not make financial sense to use less carbon and more chromium if the only mechanical test requirement was a certain UTS value. With regards to the question of whether this steel could be made, or more importantly if the neural networks prediction at this composition would be correct, one can return to the standard

deviation between the ensemble predictions as a guide. The GA composition shown in Table 8.4 gives a mean ensemble prediction of 868 N/mm^2, but this has a standard deviation between predictor members of 96.3! This would suggest that the prediction should not be relied upon and that the GA has been 'exploring' a sparse area of the model from which the solution has been found. Data sparsity has already been discussed in chapter 7, and with regard to GA interaction with a neural network it presents some severe problems. The GA should be free to search the input space for possible candidate solutions. It has been shown, however, that in sparse areas of the model, spurious predictions may result. If these spurious predictions happen to provide a good solution to the GA, it has no other way of knowing that this may not represent a realistic or accurate solution.

It was therefore decided that a more realistic and reliable solution might be obtained if the SD between the predictors was included in the cost function of the GA. In this way it was hoped that the GA would tend towards a solution which was close to that which had been found before in terms of data density, and for which the neural model should provide a more reliable prediction. In this way, it was therefore anticipated that the GA would be provided with information of the metallurgical experience that led to the data set's development.

The improved cost function can therefore be written as:

$$J_{UTS} = ((x_t - x_c)^2 + (\lambda s_e^2) \tag{8.3}$$

where S_e is the SD of the ensemble member's preidctions for a given set of input variables. From experimentation, it was found that factor of λ=5 should be used to weight the effect the SD term has in the cost function. The amount of effect the S_e^2 term has appears to be important, since if it is too small, the data density of model is not taken into account, however it if it is too large, then the target value may not be met unless it lies in a very dense area of the data.

The GA was therefore run again for the same target UTS value and fixed input variables but with the modified cost function shown in equation (8.3).

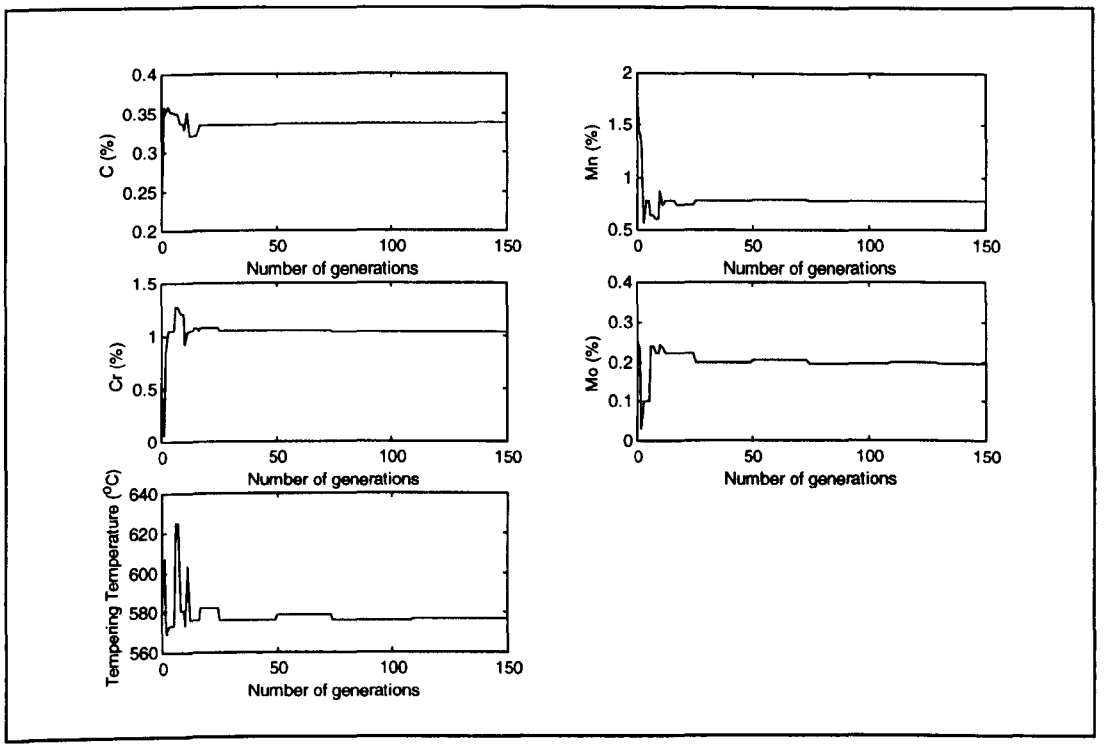


Fig. 8.10 UTS model input variable adjustment of the GA over successive generations with the modified cost function including SD of predictors.

The algorithm was allowed to run for 150 generation, by which time, the variable adjustment had been seen to be stable. The graph in Figure 8.10 demonstrates the stability of the variable adjustment, and also shows that the search area of good candidates is now limited, in other words the variable adjustment at the early stages of the algorithm shows less variance. This would indicate that the possible number of good solutions has been limited by the introduction of the SD term. Figure 8.11, shows that the overall objective was once again closely met after a small number of generations, however it can also be seen that there is a slight error between the UTS value of the solution and that of the target. The error in the UTS is only slight, a value of 869.2 was obtained instead of the target 868. Through balancing the importance of data density with the need to meet the target (i.e. though adjustment of the significance of the SD term), it may be possible to reach an even closer solution.

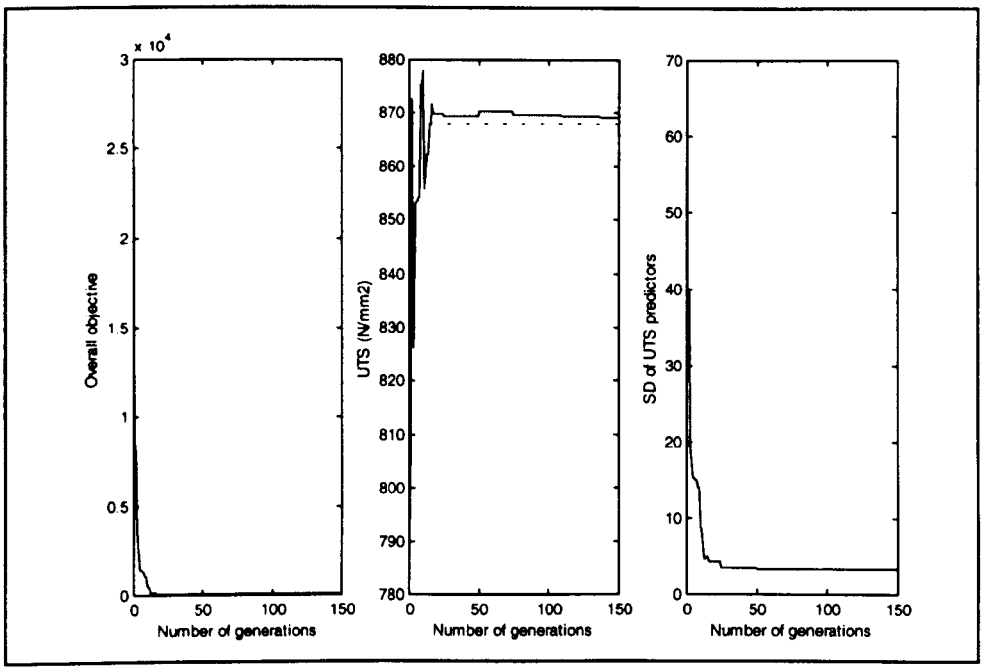


Fig. 8.11 The overall objective, UTS value and SD of predictors over subsequent generations of the algorithm

The important features of the experiment are that a set of variable values was found that gave a test result value close to that required, and that the SD of the individual predictors for this set of values was 3.44. Therefore, the solution should be within a high area of data density, and therefore should be accurate.

The values of the adjusted variables for GA solution in this experiment are shown in Table 8.5. It can be seen that the adjusted values appear to be much closer to that of the median analysis than in Table 8.4 (without the modified cost function). The tempering temperature is lower, however this is likely to be due to the reduced carbon content, however the low SD of predictors value indicates that the GA has reached a realistic solution.

Variable	GA Adjusted Values	1%CrMoValue
Site (1-6)	3	3
Size (mm)	180	180
Test Depth (mm)	12.7	12.7
C (%)	**0.33**	0.41
Si (%)	0.27	0.27
Mn (%)	**0.78**	0.78
S (%)	0.023	0.023
Cr (%)	**1.03**	1.08
Mo (%)	**0.19**	0.22
Ni (%)	0.19	0.19
Al (%)	0.027	0.027
V (%)	0.005	0.005
Hardening Temp (°C)	860	860
Tempering Temp (°C)	**577**	630

Table 8.5 The composition found by the GA to provide a UTS of 869.2 N/mm^2 with the adjusted variables in bold type, compared with the median 1%CrMo analysis when SD of predictors was taken into account

It should be remembered that only one test result is being used as a target in this experiment, whereas in a real alloy design situation, many mechanical, metallurgical and cost requirements would have to be taken into account. It is expected that the metallurgical requirements should be accommodated within the data set and therefore a solution in a high area of data density should relate well to that which a metallurgist would endorse, since steels will have been produced in this area of the input space.

The next subsection briefly looks at the possibility of using two target values in the cost function, so that a set of variables can be found which meet two mechanical test result critera.

8.5.2.2. Using GA to find a target UTS and ROA value

The next experiment with the GA technique was to investigate whether a composition could be found which would meet two target test results. It was decided

that the ROA would be a good target value to use with UTS since in chapter 7 it was seen that there is generally a trade-off between steel strength and ductility. So to validate the results easily, it was decided that the GA adjustable inputs would be kept to the same variables as in the previous sub-section, as would be the fixed model inputs. A target ROA value was then selected in the same way that the target UTS value was selected (the median ROA value for the 1%CrMo examples in the data set). As before, this would ensure that a solution existed within the freedom of the GA adjusted inputs. The cost function was then modified to take account of the target ROA value, and also the SD of the ROA ensemble model's predictors. This was important since there may be slightly different data distribution or uniformity between the predictors of the UTS and ROA models.

The modified cost function was therefore:

$$J_{UTS\&ROA} = ((x_t^{UTS} - x_c^{UTS})^2 + (\lambda s_{e(UTS)}^2) + (x_t^{ROA} - x_c^{ROA})^2 + (\lambda s_{e(ROA)}^2)) \qquad (8.4)$$

where the superscripts UTS and ROA denote the target and candidate values of each model and the bracketed subscript on the S_e term denotes the standard deviation between the predictor members of each ensemble model, and λ was set to 5.

It should be noted that, whilst there is more than one term in the objective function, the GA is still operating with a single objective function. Multi-objective GA techniques do exist, but were not investigated as part of this work, however more information on this subject can be found in Goldberg[104].

The algorithm was run with the cost function described by equation (8.4). The algorithm was stopped when it was decided that the adjusted variables had stabilised and that a reasonable solution had been obtained.

The trajectories of the GA adjusted variables are shown in Figure 8.12, which shows that the input variables found relatively stable values after about 25

generations. The trajectory of the output variables over successive generations is shown in Figure 8.13.

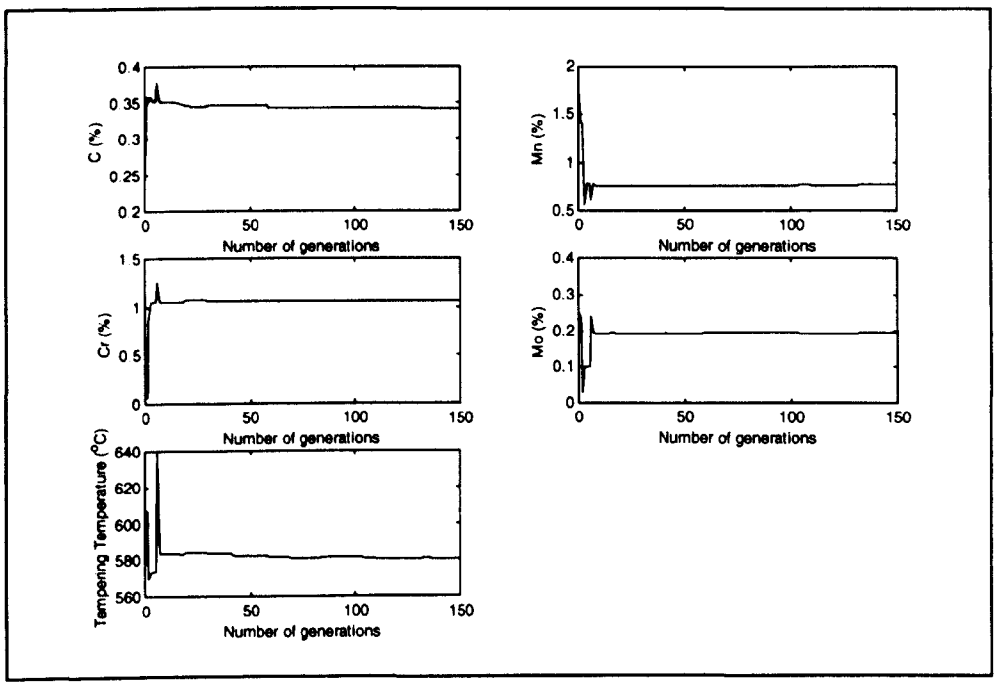


Fig. 8.12 UTS & ROA model input variable adjustment of the GA over successive generations with the modified cost function including SD between both ensembles' predictors.

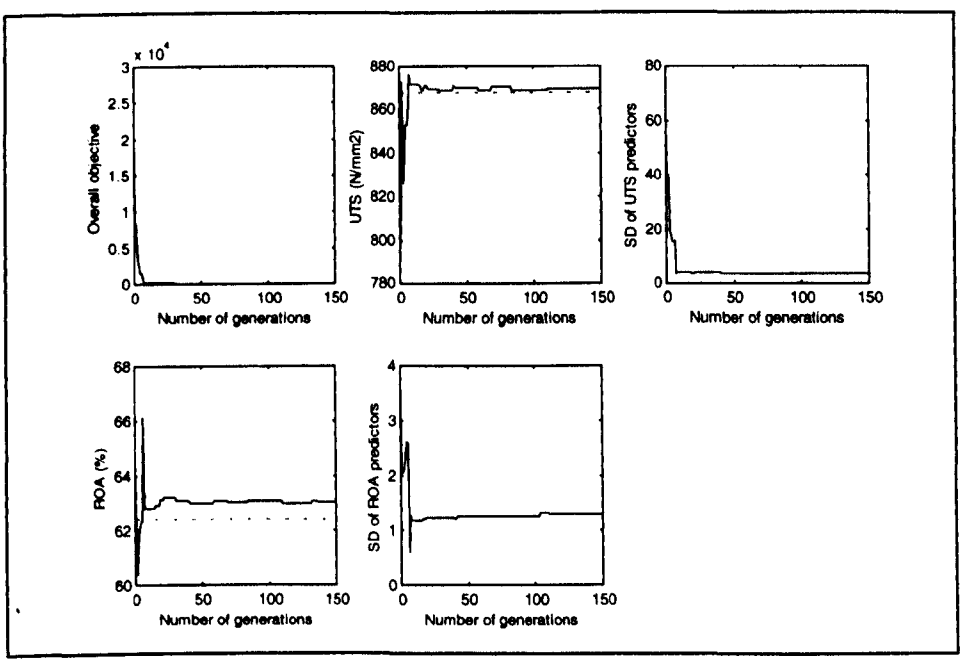


Fig. 8.13 The overall objective, UTS value ROA value and SD between both predictors over subsequent generations of the algorithm.

Figure 8.13 shows that the solution provided by the GA yields small SD values between the ensemble members for both models, and that the UTS and ROA values are close to that of their target values.

Given the target UTS and ROA values of 868 N/mm^2 and 62.4 % respectively, the GA found values of the model inputs that would give UTS values of 869.5 N/m^2 and an ROA of 63.0%. This solution is very close to that required and carries a standard deviation between ensemble member's predictions of 3.4 for the UTS model and 1.29 for the ROA model. These are low deviations between predictor values for each model, as was found from experimentation for single variable effects around the median analysis indicating areas of high data density. Table 8.6 shows the values of the input values (in bold type) which relate to this multiple target experiment, compared with that of the median analysis.

Variable	GA Adjusted Values	1%CrMoValue
Site (1-6)	3	3
Size (mm)	180	180
Test Depth (mm)	12.7	12.7
C (%)	**0.34**	0.41
Si (%)	0.27	0.27
Mn (%)	**0.76**	0.78
S (%)	0.023	0.023
Cr (%)	**1.05**	1.08
Mo (%)	**0.19**	0.22
Ni (%)	0.19	0.19
Al (%)	0.027	0.027
V (%)	0.005	0.005
Hardening Temp (°C)	860	860
Tempering Temp (°C)	**580**	630

Table 8.6 The composition found by the GA to provide a UTS of 869.2 N/mm^2 and an ROA of 62.4% with the adjusted variables in bold type, compared with the median 1%CrMo analysis when SD of both predictors was taken into account

It can be seen from Table 8.6 that as with the results in Table 8.5 a very similar solution to that of the median analysis has been obtained and therefore the solution has been obtained based on what is already within the data base, as is preferable.

There is undoubtedly a great deal more experimentation in terms of cost function structure that could be carried out. For example, one could weight the ROA and UTS terms differently to allow for differences in test result magnitude or importance in a particular optimisation. It should also be possible to find, for example, a steel composition which gives the optimum ROA and UTS combination (this could be applied to other mechanical test results). Additionally, other factors like cost could be embedded within the cost function such that steel with certain target properties could be produced at a minimal cost. Experimentation with the cost function structure is also important since it has been already found that there is a trade off between having a solution with a low SD between individual predictors and one which meets its target values. This problem was largely avoided in this experimentation because one knows that there is a solution in a highly populated area of the input space. However, if 'new' steel alloys are to be developed using this method, then a balance between a desired solution and one that is close enough to what is already known (the data set) will have to be found.

8.6 Chapter conclusion

This chapter initially investigated some potential application areas within industry for the models developed in chapter 6. A more detailed investigation into some of these ideas was then made. Initially, an investigation was made into the ability of the UTS ensemble model to predict test results for a new treatment site (in the form of a different lab) on steels of varying novelty. It was found that for the more common steel types, the prediction error and deviation between prediction members was lower. Three steel compositions were also examined which were not close to examples in the training set and it was seen that a poorer fit between measured and predicted values resulted. Overall the error bars on these predictions

generated using the standard deviation of the individual ensemble members' predictions appeared to give a good guide to the prediction error of the model, even though it was effectively predicting a new heat treatment process. It is expected that generalisation to an actual plant process may not be as good, since there may be more reasons why systematic process differences may occur in a plant environment. It was, however, also postulated that the binary codes denoting the treatment site in the model could be used to 'fit' the model to an example set of data of a new process, and indeed this was performed for the Lab data in the experiment.

The next model application investigated was that of assessing the effects of measurement tolerances on the predicted UTS values. It was seen that, rather than calculating the tolerance effect analytically, a simulation was required to assess the effect of normally distributed measurement noise. The proportional effects of size, composition and temperature inaccuracies were then investigated, as well as the variation in UTS caused by all the model inputs.

Finally, preliminary work to find optimal model inputs given one or more constraints was investigated, using the GA approach. The ability of a GA to vary five significant input parameters to meet a target UTS value was initially demonstrated. It was seen that if no constraint was placed on where in the input space the GA searched, an unreliable and often impractical solution was obtained. If, however, the SD of the ensemble's individual predictor values was used as a guide to data density and was included in the cost function, it was seen that the solution obtained by the GA technique became more rational. This was demonstrated with reference to the 1%CrMo median inputs and average UTS value as a target. The ability of the GA to find a set of model inputs which met two different and generally conflicting predicted test result values (UTS and ROA) was then also successfully demonstrated for the constrained 1%CrMo example. It is noted that more research in terms of cost

function design could be performed, having found that a 'balance' was required between the terms in the single objective function. This is anticipated to be particularly important when finding an optimal set of values whose solution may relate to a set of inputs in an area of lower data density with respect to the input space. Given this situation, one may need to decide between meeting the targets of the cost function or remaining near past experience, i.e. a higher data density.

The feasibility of using GA to optimise the neural models has been demonstrated. It is anticipated that this could have significant financial value in the optimisation of the heat treatment process, since alloy cost could also be taken into account as part of the optimisation routine. Additionally, automation of the alloy design approach should lead to reduced alloy design time and therefore reduced steel development costs.

Chapter 9

Conclusions and further work

This final chapter draws some conclusions from the work that has been undertaken in this project and then suggests possible areas for future research in the subject area.

9.1. Conclusions

This project has detailed the development of a set of neural models for the prediction of ultimate tensile strength, proof stress, impact energy, elongation and reduction of area from a set of variables relating to the heat treatment process.
At the beginning of the project a great deal of time was invested in selecting a suitable heat treatment process that would enable a large amount of historical data to be collected.

Process and data familiarisation work dominates the early chapters of this thesis, demonstrating the importance that has been placed on the need for expert process knowledge throughout the models' development. This expert information has played an important part in deciding upon the selection of training data and input variables, cleaning faulty process data, investigating possible data decomposition strategies, and validating the final models.

A structured data cleaning approach has been applied to the data sets involved in the project following the realisation that faulty data points were present in the process data. Experimentation has shown this technique to be effective, particularly in the UTS data set where the greatest improvement in modelling accuracy was seen. The investigations showed that the quality of the data used to

train and validate a neural model plays a crucial role in its successful application for prediction using unseen data sets.

Various approaches have been used to deal with the problem of an uneven data distribution. These methods include modular network approaches and the use of descriptive inputs to a single neural model. Ultimately, the aims of the project dictate that a single model covering a wide range of steels would be preferable, providing a more valuable tool for alloy design applications.

The drawback of this approach was seen when a graphical user interface was developed to enable model evaluation within industry. This interface enabled information to be extracted from the model in terms of individual variable effects and a single predicted value. It was then realised that a number of sparse regions were present in the data set, largely due to its extended range. Ensemble techniques were therefore introduced to improve prediction reliability. It was demonstrated that the reliability of the resulting predicted values could still vary significantly depending upon the quality and density of data used to train the model. It was realised, therefore, that spurious prediction may result from the neural model even when the input variables were within the maximum and minimum ranges of the training data. An extension of the ensemble approach using the deviation of the individual ensemble members' predictions, was therefore developed as a technique which was shown to provide a clear indication of data density within the input space, therefore helping the user to determine the reliability of the predicted value. This technique was also found to be useful as a guide for active data selection and experimentation.

The purpose of developing the predictive models was to optimise the heat treatment process. Using a variety of applications within industry as examples, it has been shown that this is now a possibility. The feasibility of using the genetic algorithm approach to perform automatic and optimal alloy design has also been

demonstrated for a constrained example. An important conclusion from this work was that without a guide to data density, the genetic algorithm approach did not provide reliable and practically viable solutions to a given optimisation problem. The data density indication provided by the deviation between the individual ensemble members' predictions was therefore included as part of the cost function to guide the optimal solution effectively towards design conditions which encompass sound metallurgical experimental knowledge.

It is considered that with further development, the techniques and application areas explored within this project would lead to improved product reliability and process efficiency. The following section now details further work recommended towards this goal.

9.2 Recommendations for further work

The first area of further work, only briefly tackled in this project, is to investigate the ability of the model to generalise to new processes. A small assessment of this was carried out using a 'new' lab process, however, generalisation to a new plant should present additional systematic differences. An effective technique for validating whether the steels produced performed by a new plant are compatible with a given model developed would therefore be desirable. Further exploration of the effects that the site codes within the model have on predicted test results over a range of steel compositions may help in this area.

The second area of work recommended is to establish a system of model maintenance, which is important for any predictive model in a process subject to change. The system would ideally highlight when a data set update should take place, together with which production examples should be included. It is not recommended that the system work on duration of model installation, since process changes do not

occur at a fixed rate of time. The importance of the data cleaning approach should be remembered in such an update situation, and it is felt that the techniques outlined in this thesis would enable an effective approach to this.

The method of prediction reliability indication used in the later work of this project, has been shown to be an effective and important feature for the safe and useful application of the model to process optimisation (automated or otherwise). However, this has so far only been demonstrated empirically, using knowledge of the problem domain. It would be valuable to investigate the relationship between individual ensemble members' predictions further and possibly prove analytically the relationship between data density and deviation between predictors.

Finally, the genetic algorithm approach has been introduced as a viable method of automatically performing the optimal design of new alloys. The feasibility of this approach has been demonstrated if the reliability of the model's predictions is taken into account. The final recommendation is for the further development of this technique, so as to explore the weighting effects of terms in the cost function dictating the balance between meeting a constraint and staying towards existing knowledge. Financial as well as mechanical property optimisation should ultimately also be possible using this technique and would beneficially lead to an optimised heat treatment process.

References

[1] Mardia, K., Kent, J., Bibby, J. (1980) "Multivariate Analysis" Academic Press

[2] Pierre, B., Hornik, K. (1989) "Neural Networks and Principle Component Analysis: Learning From Examples Without Local Minima" Neural Networks, **2**, pp. 53-58

[3] Oja, E. (1982) "A Simplified Neuron Model as a Principal Component Analyser" Mathematical Biology, **15**, pp. 267-273

[4] Oja, E. (1989) "Neural Networks, Principal Components, and Subspaces" International Journal of Neural Systems, **1**, 1, pp. 61-68

[5] Geladi, P., Kowalski, B.R. (1986) "Partial Least Squares Regression: A Tutorial" Analytica Chimica Acta, **185**, pp. 1-17

[6] Hiden, H.G., Willis, M.J., Tham, P., Turner, Montague, G.A. (1997) "Non-Linear Principal Components Analysis Using Genetic Programming" Genetic Algorithms in Engineering Systems, **2**, 446, pp. 302-307

[7] Qin, S.J., McAvory, T.J. (1992) "Non-linear PLS modelling using Neural Networks" Computers and Chemical Engineering, **16**, 4, pp. 379-391

[8] Wilson, D.J.H., Irwin, G.W., Lightbody, G. (1997) "Non-linear PLA using Radial Basis Functions" Trans Inst MC, **19**, 4, pp. 211-220

[9] Hartnett, M., Lightbody, G., Irwin, G.W. (1996) "Chemometric Techniques in Multivariate Statistical Modelling of Process Plant" The Analyst, **121**, pp. 749-754

[10] McCulloch, W.S., Pitts (1943) "A logical calculus of the ideas immanent in nervous activity" Bulletin of Mathematical Biophysics, **5**, pp. 115-133

[11] Minsky, M.L., Papert, S.A. (1969) "Perceptrons" Cambridge, MA: MIT Press.

[12] Bishop, C.M. (1995) "Neural Networks for Pattern Recognition" Clarendon Press, Oxford.

[13] Cybenko, G. (1989) "Approximation by Superpositions of a Sigmoidal Function" Mathematical Control, Signals and Systems, **2**, pp. 303-314

[14] Hornik, K. (1993) "Some New Results on Neural Network Approximation" Neural Networks, **6**, pp. 1069-1072

[15] Tarassenko, L. (1998) "A Guide to Neural Computing Applications" Arnold, London.

[16] van der Wolk, P., van der Zwaag, S., Bodin, A. (1996) "The Prediction of the Strip Temperature after the Last Finishing Stand Using Artificial Neural Networks" 2nd International Conference on 'Modelling of Metal Rolling Processes', pp. 304-313

[17] Vermeulen, W.G., Morris, P.F., de Weijer, A.P., van der Zwaag, S. (1996) "Prediction of Martensite Start Temperature using Artificial Neural Networks" Ironmaking and Steelmaking, **23**, 5, pp. 433-437

[18] van der Wolk, P.J., Vermeulen, S., van der Zwaag (1996) "Prediction of the Continuous Cooling Transformation Diagram of Vanadium Containing Steels Using Artificial Neural Networks" Proc. Conf. 'Modelling of Metal Rolling', pp. 378-388

[19] Vermeulen, W.G., van der Wolk, A.P., de Weijer, van der Zwaag (1996) "Prediction of Jominy Hardness Profiles of Steels Using Artificial Neural Networks" Jounrnal of Materials Engineering and Performance, **5**, pp. 57-63

[20] van der Wolk, P.J., van Helvoort, M., van Leeuwen, S., van der Zwaag (1997) "Calculating the Hardenability of Engineering Steels using Data from Multiple Sources" Proceedings of the Euromat '97 Conference, **4**, pp. 525-528

[21] Bhadeshia, H.K.D.H., MacKay, D.J.C., Svennsson, L.E. (1995) "Impact Toughness of C-Mn Steel Arc Welds - Bayesian Neural Network Analysis" Materials Science & Technology, **11**, pp. 1046-1050

[22] MacKay, D.J.C. (1995) "Bayesian Methods for Neural Networks: Theory and Applications" Neural Networks Summer School Report, Cavendish Laboratory, Cambridge, UK.

[23] MacKay, D.J.C. (1995) "Modelling Phase Transformations in Steels" University of Cambridge Programme for Industry Report, Cavendish Laboratory, Cambridge, UK.

[24] Thodberg, H.H. (1996) "A Review of Bayesian Neural Networks with an Application to Near Infrared Spectroscopy" IEEE Transactions on Neural Networks, **7**, 1, pp. 56-72

[25] van der Wolk, P.J. (1998) "Application of Artificial Neural Networks in Steel Heat Treatment Processes" Research Report, pp. 1-53, Delft University of Technology, Dept. of Materials.

[26] Cool, T., Bhadeshia, H.K.D.H., MacKay, D.J.C. (1997) "The Yield and Ultimate Tensile Strength of Steel Welds" Materials Science & Engineering, pp. 186-200

[27] Gavard, L., Bhadeshia, H.K.D.H., Macay, D.J.C., Suzuki, S. (1996) "Bayesian Neural Network Model for Austenite Formation in Steels" Materials Science and Technology, **12**, pp. 453-463

[28] Singh, S.B., Bhadeshia, H.K.D.H., MacKay, D.J.C., Carey, H., Martin, I. (1998) "Neural Network Analysis of Steel Plate Processing" Ironmaking and Steelmaking, **25**, 5, pp. 355-365

[29] Jones, J., MacKay, D.J.C. (1996) "Neural Network Modelling of the Mechanical Properties of Nickel Base Superalloys" Proceedings of the Eighth International Symposium on 'Superalloys', pp. 412-424

[30] Bulsari, A., Hocksell, E. (1995) "Neural Network Systems for Hardened Components" Steel Technology International, pp. 133-138

[31] Myllykoski, P., Larkiola, J., Nylander, J. (1996) "Development of Prediction Model for Mechanical Properties of Batch Annealed Thin Steel Strip by using Artificial Neural Network Modelling" Journal of Materials Processing Technology, **60**, 1, pp. 399-404

[32] Kudav, G.V., Dengke, S. (1996) "Artificial Neural Networks in the Prediction of Hardness Characteristics in Carbon Steels" International ICSC Symposia on Intelligent Industrial Automation and Soft Computing, pp. 80-86

[33] Datta, A., Hareesh, M., Kalra, P.K., Deo, B., Boom, R. (1994) "Adaptive Neural Net (ANN) Models for Desulphurisation of Hot Metal and Steel" Steel Research, **65**, 11, pp. 466-471

[34] Singh, H., Sridhar, N., Deo, B. (1996) "Artificial Neural Nets for Prediction of Silicon Content of Blast Furnace Hot Metal" Steel Research, **67**, 12, pp. 521-527

[35] Zadeh, L.A. (1965) "Fuzzy Sets" Information and Control, **8**, pp. 338-353

[36] Schooling, J.M., Brown, M., Reed, P.A.S. (1999) "Neurofuzzy Modelling of Fatigue Threshold Behaviour in Ni-Base Superalloys" Materials Science and Engineering, **260**, 1, pp. 222–239

[37] Wiklund, O. (1997) "Modelling of the Temper Rolling Process Using Finite Elements and Artificial Neural Networks", Research Report, MEFOS Metal Working Research Plant, Lulea, Sweden.

[38] Dumortier, C., Lehert, P., Krupa, P., Charlier, A. (1998) "Statistical Modelling of Mechanical Properties of Microalloyed Steels by Application of Artificial Neural Networks" Materials Science Forum, **284**, pp. 393-400

[39] Naylor, D.J. (1989) "Review of International Activity on Microalloyed Engineering Steels" Ironmaking and Steelmaking, **16**, 4, pp. 246-252

[40] Gryllis, R.J. (1997) "Mechanical Properties of a High-Strength Cupronickel Alloy-Bayesian Neural Network Analysis" Materials Science and Engineering, **234**, pp. 267-270

[41] Nakamura, K., Sagara, K. (1995) "Neuro-Aided Guidance System for Optimum Desulphurization Treatment in Torpedo Cars" 1995 IEEE International Conference on Neural Networks, pp. 575-579

[42] Kim, S.Y., Choi, J.D., Yu, K.B., Kim, S.J., Lee, S.Y., Lee, J.H., Moon, B.W., Baek, Y.H. (1995) "A Temperature Control of Steel Strip using Neural Network in Continuous Annealing Process" IEEE International Conference on Neural Networks, **1**, pp. 631-635

[43] Cho, S., Cho, Y., Yoon, S. (1997) "Reliable Roll Force Prediction in Cold Mill Using Multiple Neural Networks" IEEE Transactions on Neural Networks, **8**, 4, pp. 875-882

[44] Pican, N., Alexandre, F., Bresson, P. (1996) "Artificial Neural Networks for the Presetting of a Steel Temper Mill" IEEE Expert, **11**, 1, pp. 22-27

[45] Sbarbaro-Hofer, D., Neumerkel, D., Hunt, K. (1993) "Neural Control of a Steel Rolling Mill" IEEE Control Systems Magazine, **13**, 3, pp. 69-75

[46] Loney, D., Roberts, I., Watson, J. (1997) "Modelling of Hot Strip Runout Table using Artificial Neural Networks" Ironmaking and Steelmaking, **24**, 1, pp. 34-39

[47] Ogawa, M., Ohshima, M., Watanabe, F., Morinaga K., Hashimoto, I. (1996) "Quality Control System For An Industrial High Density Polyethylene Process" IFAC 13th Triennial World Congress, USA, pp. 163-168

[48] Koivo, A.J., Kim, C.W. (1989) "Robust Image Modelling for Classification of Surface Defects on Wood Boards" IEEE Transactions on Systems, Man and Cybernetics, **19**, 6, pp. 1659-1666

[49] Hentea, M., Evens, M. (1995) "Neural Network Prediction Model For Process Control and Architecture Design Issues" Proceedings of the ISCA International Conference, pp. 219-223

[50] Stelljes, T.A., Erickson, K.T. (1995) "Modelling the Quality of Steel Production with an Adaptive Logic Network" Intelligent Engineering Systems through Artificial Neural Networks, **5**, pp. 943-948

[51] Kominami, H., Kamada, N., Tanaka, T., Naitoh, S., Hamaguchi, C., Endoh, H. (1991) "Neural Network System of Breakout Prediction in Continuous Casting Process" Nippon Steel Technical Report 49, pp. 34-38

[52] Dominguez, S., Campoy, P., Aracil, R. (1994) "A Neural Network Based Quality Control System for Steel Strip Manufacturing" IFAC Conference on 'Artificial Intelligence in Real Time Control', pp. 185-190

[53] Thelning, K.E. (1984) "Steel and its Heat Treatment" Butterworth and Co., London.

[54] William, C.L. (1981) "The Physical Metallurgy of Steels" Washington Press, London.

[55] Jackson, P. (1990) "Introduction to Expert Systems" Addison-Wesley, UK

[56] Curbishley, I. (1988) "Mechanical Testing" Institute of Metals, London.

[57] Fenner, A.J. (1965) "Mechanical Testing of Materials" Newnes, London

[58] Davis, H.E., Troxell, G.E., Wiskocil, C.T. (1964) "The Testing and Inspection of Engineering Materials" McGraw-Hill.

[59] Hollomon, J.H., Jaffe, L.D. (1945) "Time-temperature Relations in Tempering Steel" Metals Technology, **1831**, pp. 1-26

[60] Martens, H., Naes, T. (1989) "Multivariate Calibration" Wiley, New York.

[61] Simoudis, E., Livezey, B., Kerber, R. (1995) "Using Recon for Data Cleaning" First International Conference on Knowledge Discovery & Data Mining, pp. 282-287

[62] Wu, X., Cinar, A. (1992) "A Knowledge-based System For Development Of Nonlinear Input-Output Models" Proceedings of the American Control Conference, pp. 1447-1448

[63] Weigend, A.S., Zimmermann, H.G., Neuneier, R. (1997) "Clearning" Proceedings of the third international conference on 'Neural Networks in Financial Engineering', pp. 511-522

[64] Forouzi, S., Meech, J.A. (1999) "An Adaptive Artificial Neural Network to Model a Cu/Pb/Zn Flotation Circuit" The Second International Conference on 'Intelligent Processing and Manufacturing of Materials', **2**, pp. 967-982

[65] Danuser, G., Stricker, M. (1998) "Parametric Model Fitting: from Inlier Characterisation to Outlier Detection" IEEE Transactions on Pattern Analysis and Machine Intelligence, **20**, 3, pp. 263-280

[66] Kosko, B. (1992) "Neural Networks and Fuzzy Systems" Prentice-Hall, New Jersey

[67] Famili, A., Shen, W., Weber, R., Simoudis, E. (1996) "Data Pre-processing and Intelligent Data Analysis" Intelligent Data Analysis, **2**, pp. 1-20

[68] Guyon, Matic, Vapic (1982) " Estimation of Dependencies Based on Empirical Data " Springer-Verlag, New York .

[69] Joe Qin, S., Rajagopal, B. (1993) "Combining Statistics and Expert Systems with Neural Networks for Empirical Process Modelling" International Conference and Exhibition on 'Advances in Instrumentation and Control', **48**, 3, pp. 1711-1720

[70] Wold, S., Esbensen, K., Geladi, P. (1987) "Principal Component Analysis" Chemometrics and Intelligent Laboratory Systems, **2**, pp. 37-52

[71] Herandez, M.A., Stolfo, S.J. (1998) "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem" Data Mining and Knowledge Discovery, **2**, 1, pp. 9-37

[72] Tenner, J., Linkens, D.A., Bailey, T.J. (1999) "Pre-processing of Industrial Process Data with Outlier Detection and Correction" The Second International Conference on 'Intelligent Processing and Manufacturing of Materials', **2**, pp. 921-926

[73] Hu, Q., Lang, E.J., Sugihara, I. (1992) "Principal Component Analysis (PCA) of Complex Spike Activity Recorded Simultaneously from Multiple Purkinje Cells" Proceedings of the IEEE-SP, **577**, pp. 231-234

[74] Sheasby, M.A., Wilson, A.J. (1996) "The Use of SPC in the Papermaking Industry" Measurement & Control, **29**, pp. 73-75

[75] Jalel, N.A., Fiacco, M., Leigh, J.R. (1994) "Monitoring Estimation And Predictive Control Based On Statistical Techniques" Proceedings of the American Control Conference, Baltimore, pp. 328-329

[76] Martin, E.B., Morris, A.J., Zhang, J. (1996) "Process Performance Monitoring Using Multivariate Statistical Process Control" Proc. IEE Control Theory Appl., **143**, 2, pp. 132-144

[77] MacGregor, J.F., Theodora, K., Nomijos, P. (1996) "Analysis, Monitoring and Fault Diagnosis of Industrial Processes Using Multivariate Statistical Projection Methods" IFAC 13th Triennial World Congress, San Francisco, USA, pp. 145-150

[78] Kourti, T., Nomikos, P., MacGregor, J.F. (1995) "Analysis, Monitoring and Fault Diagnosis of Batch Processes using Multiblock and Multiway PLS." Journal of Process Control, **5**, 4, pp. 277-284

[79] MacGregor, J.F., Kresta, J.F., Marlin, T.E. (1991) "Multivariate Statistical Monitoring of Process Operating Performance" The Canadian Journal of Chemical Engineering, **69**, pp. 35-47

[80] Thomas, C., Wada, T., Seborg, D.E. (1996) "Principal Component Analysis Applied to Process Monitoring of an Industrial Distillation Column" IFAC 13th Triennial World Congress, San Francisco, USA, pp. 61-66

[81] Nomikos, P., MacGregor, J.F. (1994) "Monitoring Batch Processes Using Multiway Principal Component Analysis" AIChE, **40**, 8, pp. 1361-1375

[82] Penny, W.D., Roberts, S.J. (1998) "Bayesian Neural Networks for Classification: How Useful is the Evidence Framework?" Pre-prints, Neural Networks Journal, from Imperial College, London.

[83] Press, W.H., Teukolsky, S.A. (1992) "Numerical Recipes in C" Cambridge University Press, Cambridge.

[84] Masters, T. (1992) "Practical Neural Network Recipes in C++" Academic Press, UK.

[85] Welstead, S.T. (1993) "Neural Network and Fuzzy Logic Applications in C++" J.Wiley and Sons

[86] Hagan, M.T., Menhaj, M. (1994) "Training Feedforward Networks with the Marquardt Algorithm" IEEE Transactions on Neural Networks, **5**, 6, pp. 989-993

[87] Xu, D.L., Martin, E.B., Morris, A.J. (1996) "Non-Linear Projection to Latent Structures" IFAC 13th Triennial World Congress, San Francisco, USA, pp. 259-264

[88] Sharkey, A.J.C. (1997) "Modularity, Combining and Artificial Neural Nets" Connection Science, **9**, 1, pp. 3-10

[89] Sharkey, A.J.C. (1996) " On Combining Artificial Neural Nets" Connection Science, **8**, 3, pp. 299-313

[90] Sollich, P., Krough, A. (1996) "Learning with Ensembles: How Overfitting can be Useful" Advances in Neural Information Processing Systems, **8**, pp. 190-196

[91] Krough, A., Vedelsby, J. (1995) "Neural Network Ensembles, Cross Validation and Active Learning" Advances in Neural Information Processing Systems, **7**, pp. 231-238

[92] Opitz, D.W., Maclin, R.F. (1997) "An Empirical Evaluation Between Bagging and Boosting" 1997 IEEE International Conference on Neural Networks, **3**, pp. 1401-1405

[93] Drucker, H., Cotrtes, A., Jackel, L.D., LeCun, Y., Vapnik, V. (1994) "Boosting and Other Ensemble Methods" Neural Computation, **1301**, 6, pp. 6-1289

[94] Rosen, B.E. (1996) "Ensemble Learning Using Decorrelated Neural Networks" Connection Science, **8**, 3, pp. 373-383

[95] Liu, Y., Yao, X. (1998) "A Co-operative Ensemble Learning System" 1998 IEEE International Conference on Neural Networks, **3**, pp. 2202-2207

[96] Opitz, D.W., Shavlik, J.W. (1996) "Actively Searching for an Effective Neural Network Ensemble" Connection Science, **8**, 3, pp. 337-353

[97] Hansen, L.K., Larsen, J. (1996) "Linear Unlearning for Cross-Validation" Advances in Computational Mathematics, **5**, pp. 269-280

[98] Sorensen, P.H., Hansen, L.K. (1996) "Cross-Validation with LULOO" Proceedings of the 1996 International Conference on Neural Information Processing, **2**, pp. 1305-1310

[99] Igelnik, B., Pao, Y.H., LeClair, S.R., Shen, C.Y. (1999) "The Ensemble Approach to Learning and Generalisation" IEEE Transactions on Neural Networks **10**, 1, pp. 19-30

[100] Nafaly, V., Intrator, N., Horn, D. (1997) "Optimal Ensemble Averaging of Neural Networks" Network Computation in Neural Systems, **8**, 3, pp. 283-296

[101] Ueda, N., Nakano, R. (1996) "Generalisation Error of Ensemble Estimators" Proceedings of the International Conference on Neural Networks, **1**, pp. 90-95

[102] Kramer, M.A. (1992) "Autoassociative Neural Networks" Computers & Chemical Engineering, **16**, 4, pp. 313-328

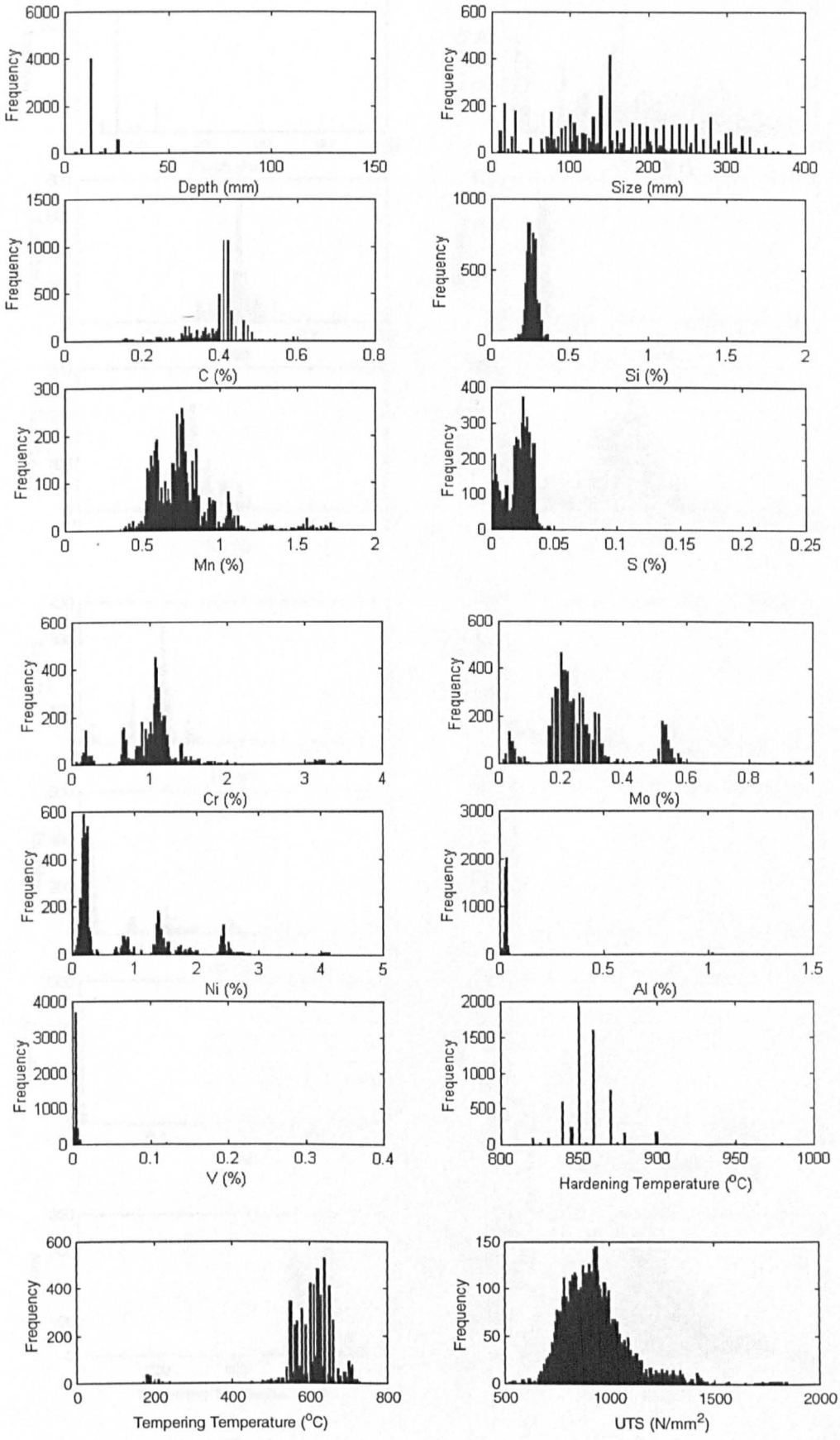
[103] Holland, J.H. (1973) "Genetic Algorithms and the Optimal Allocation of Trials" SIAM J. Computing, **2**, pp. 89-104

[104] Goldberg, D.E. (1989) "Genetic Algorithms in Search, Optimisation and Machine Learning" Addison-Wesley, Reading, Massachusetts.

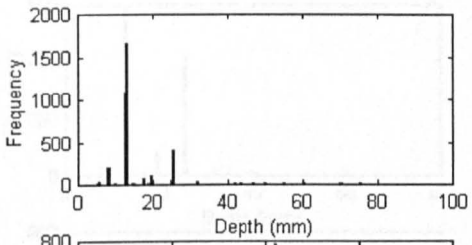
Appendix A

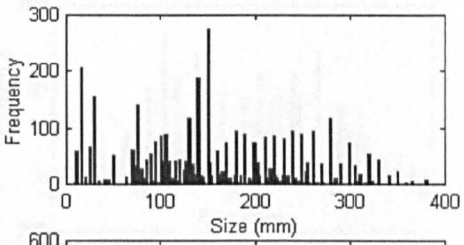
Histograms of data distribution

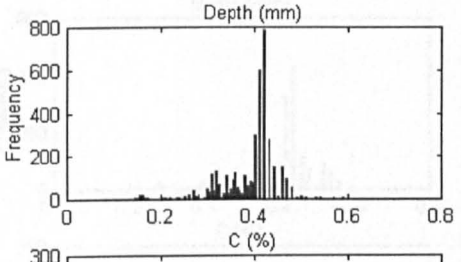
UTS data distribution

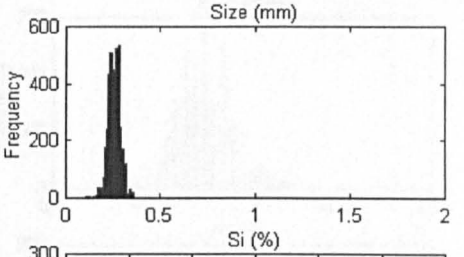


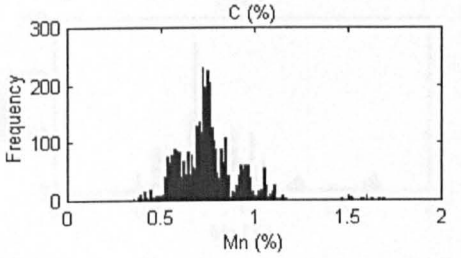
Proof Stress data distribution

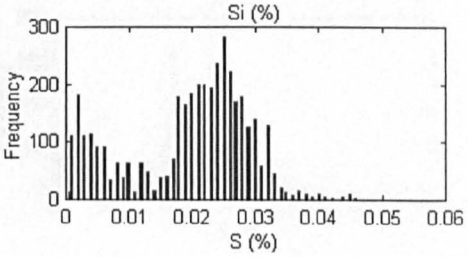


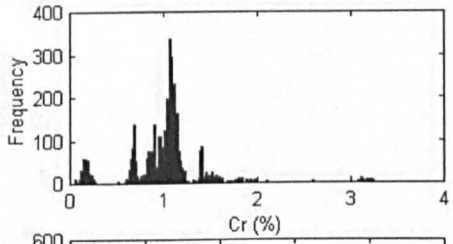


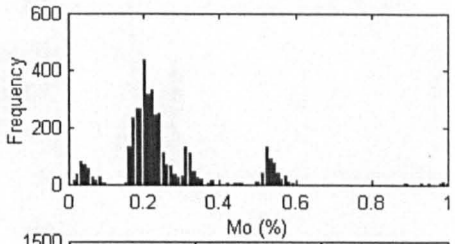


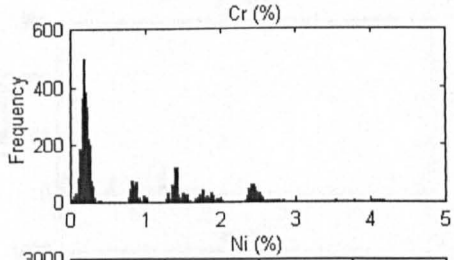


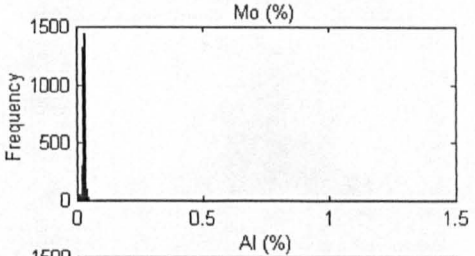


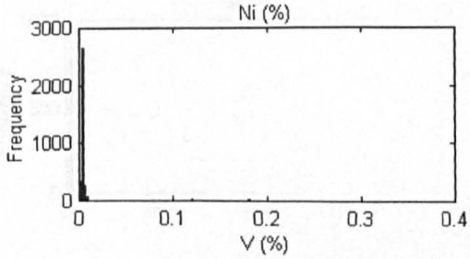


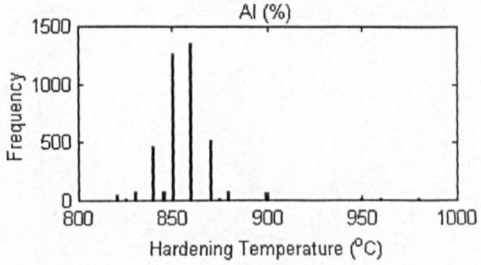


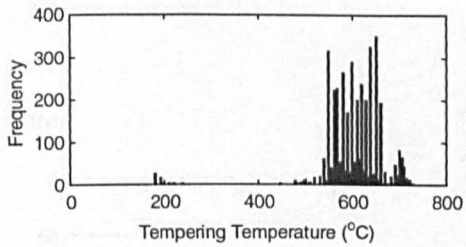


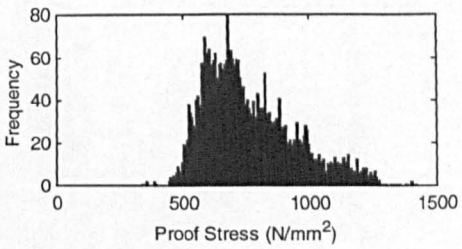




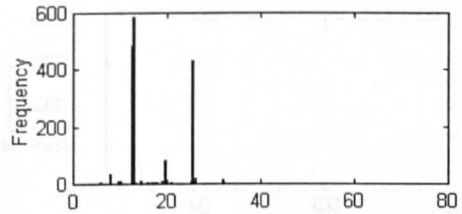


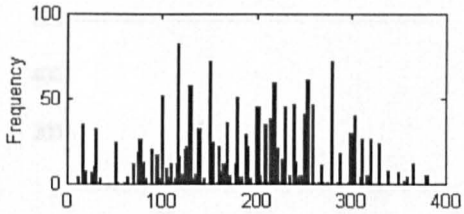


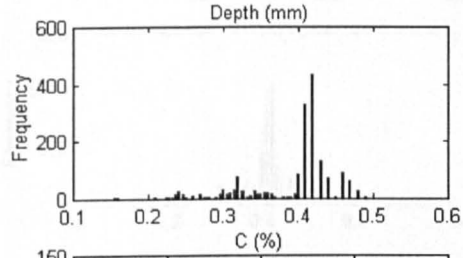


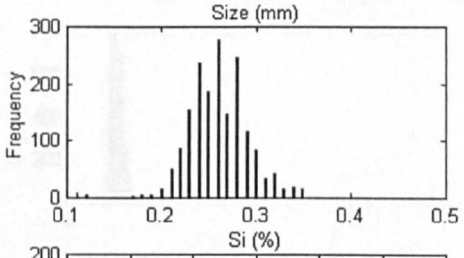


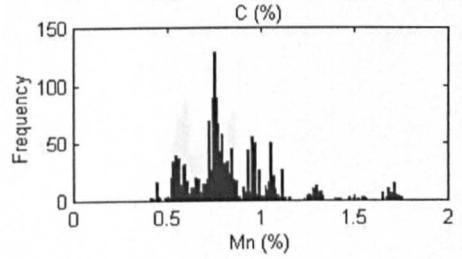
Impact Energy data distribution

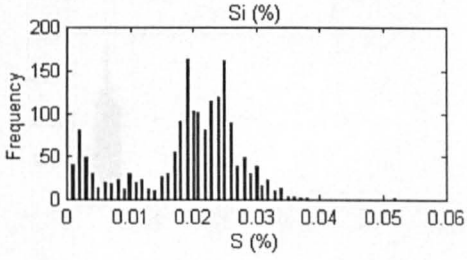


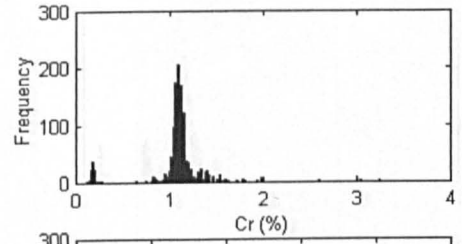


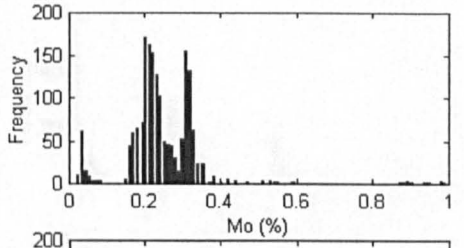


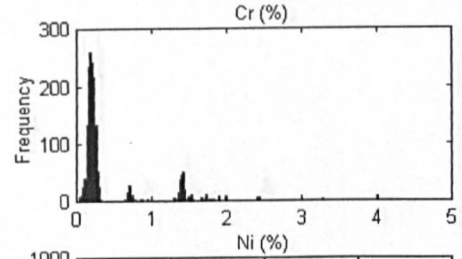


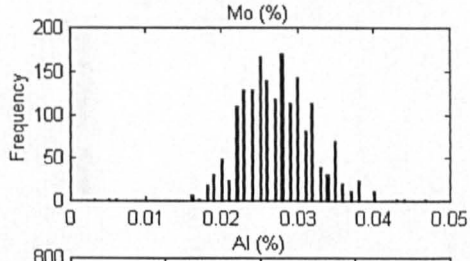


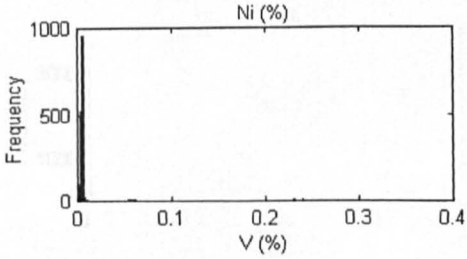


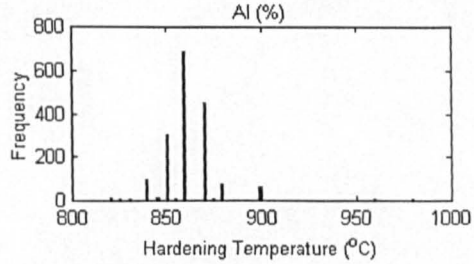


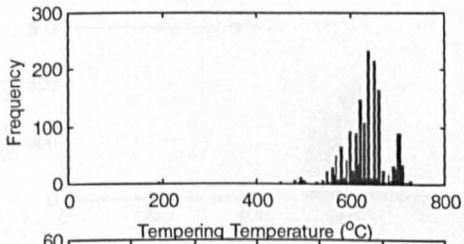


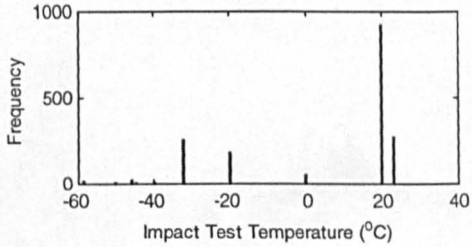


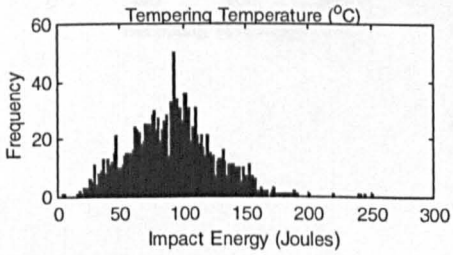




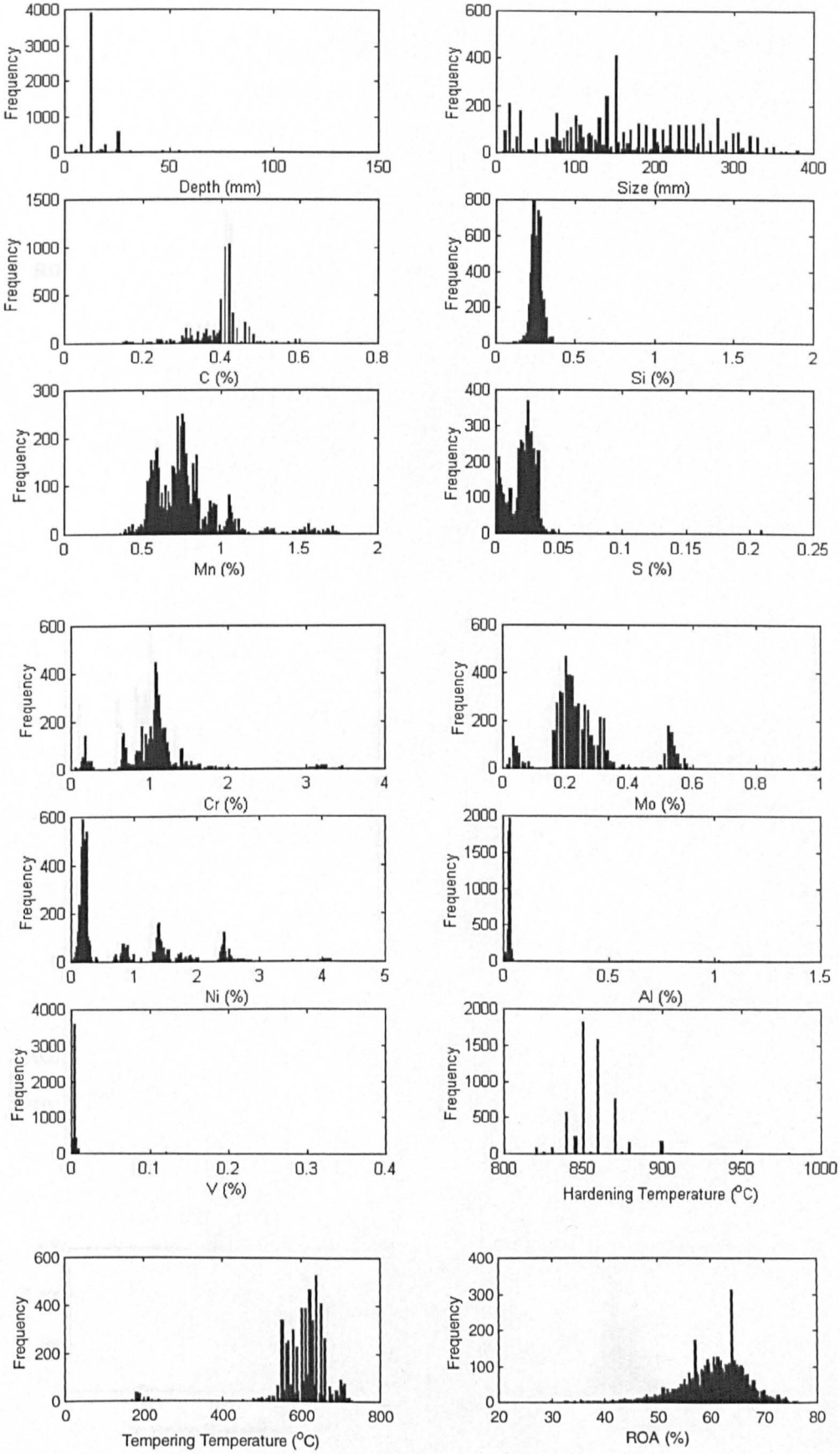




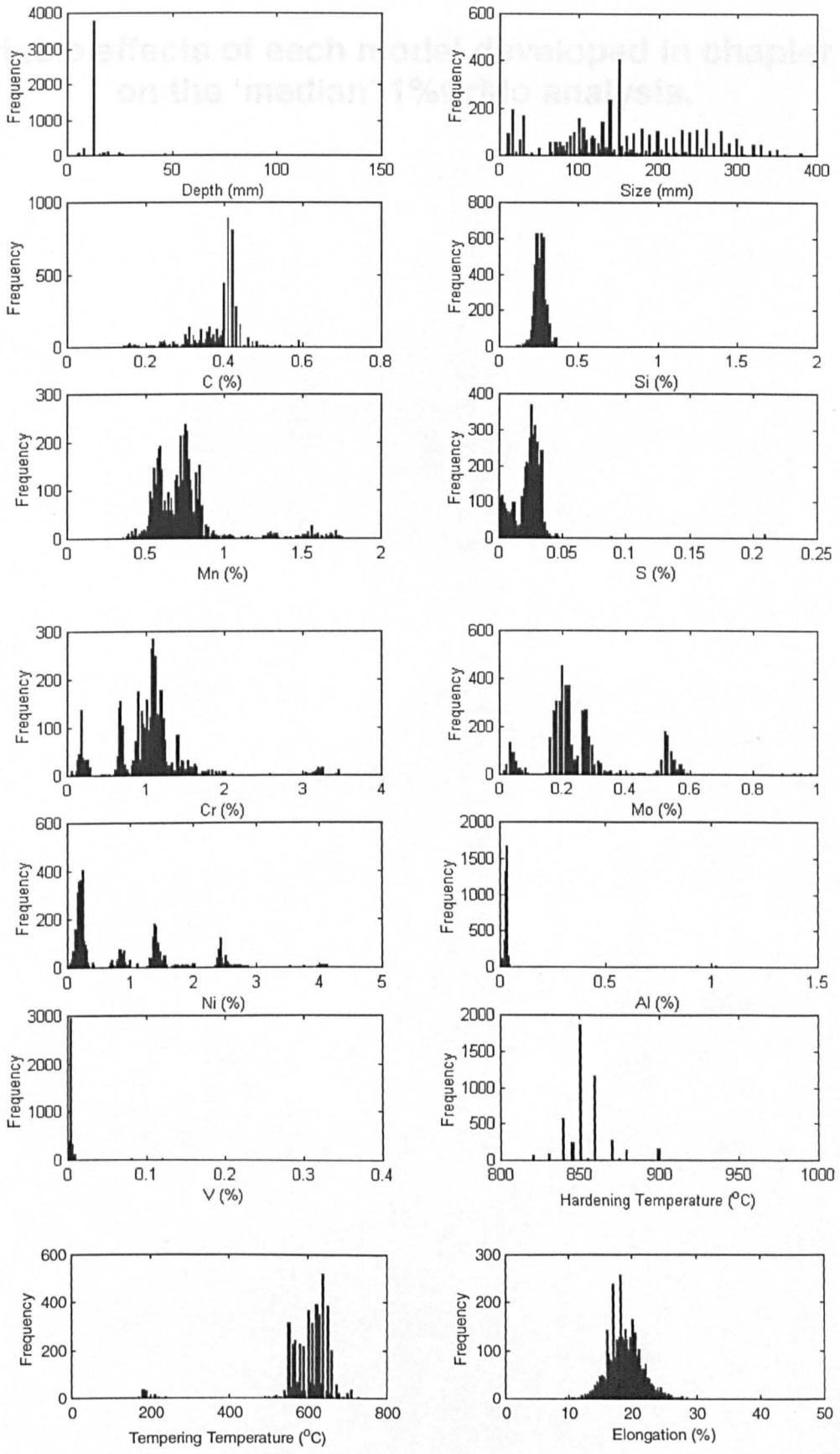




Reduction of Area data distribution



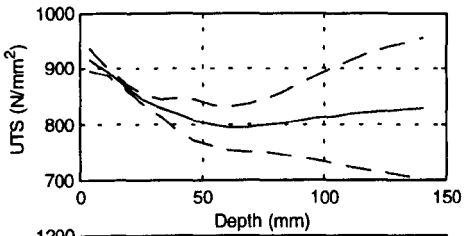
Elongation data distribution

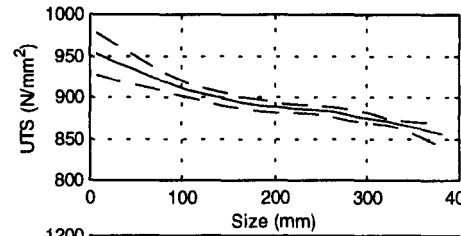


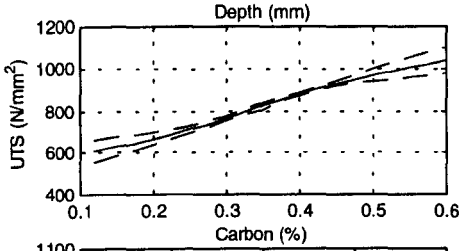
Appendix B

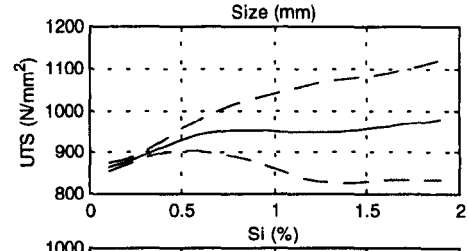
Variable effects of each model developed in chapter 6 on the 'median' 1%CrMo analysis.

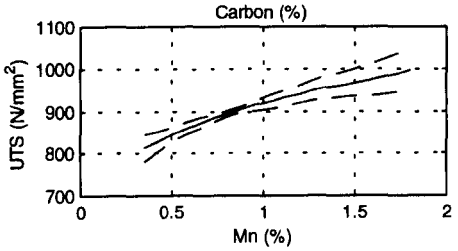
UTS model variable effects

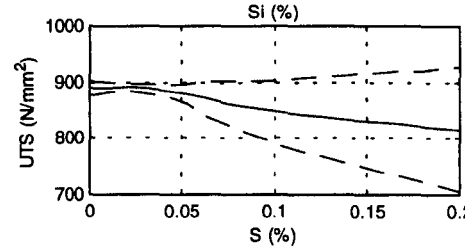


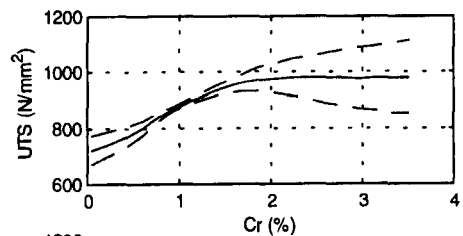


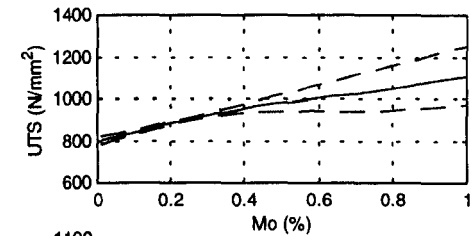


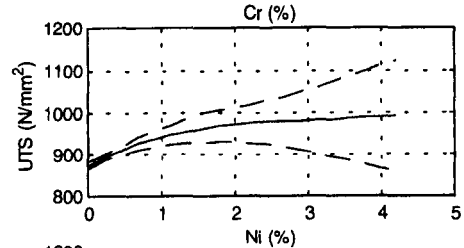


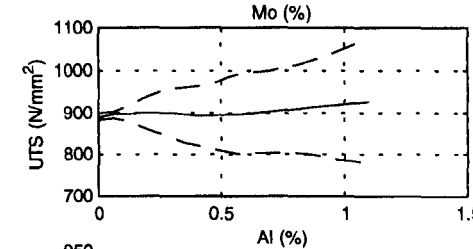


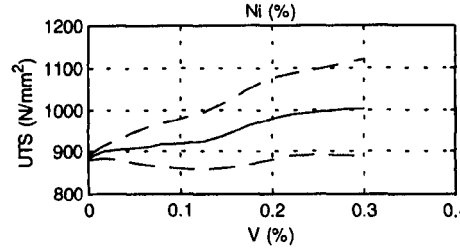


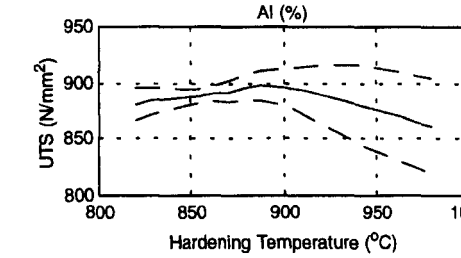


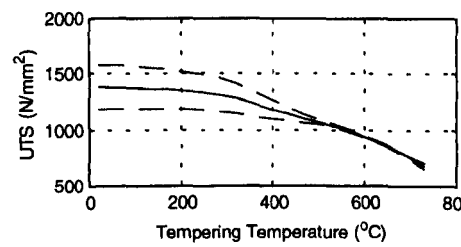




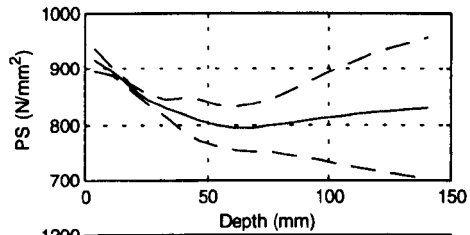


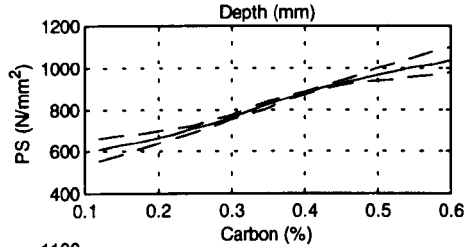


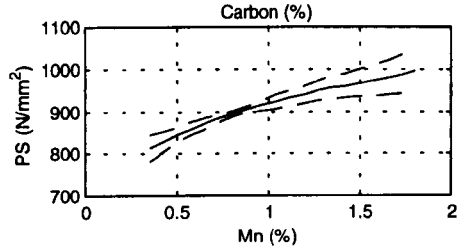


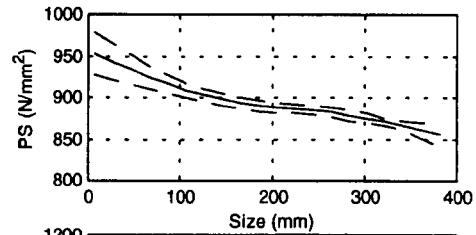


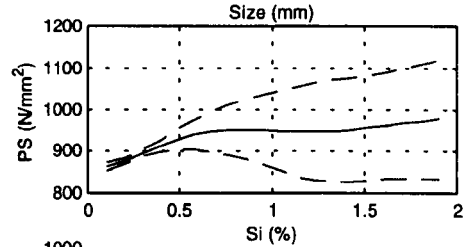
Proof Stress data distribution

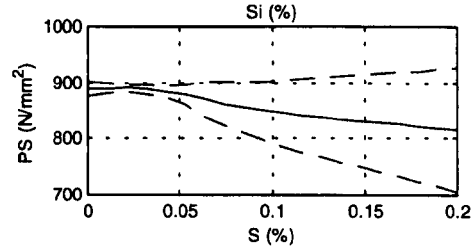


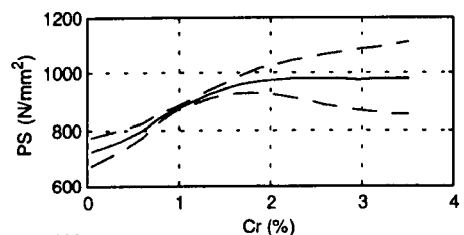


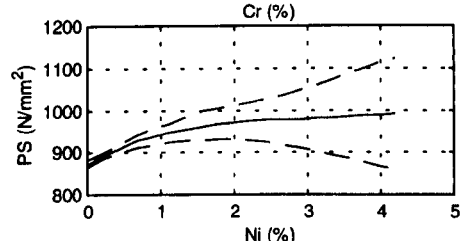


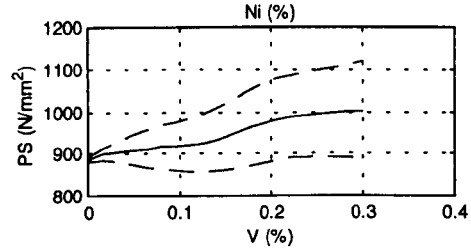


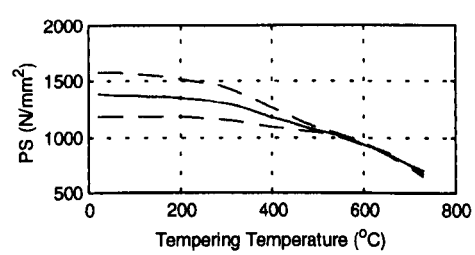


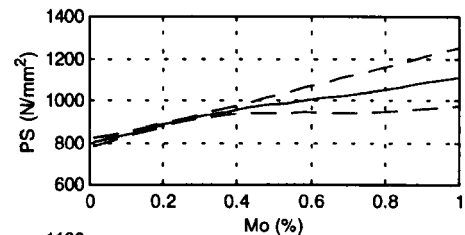


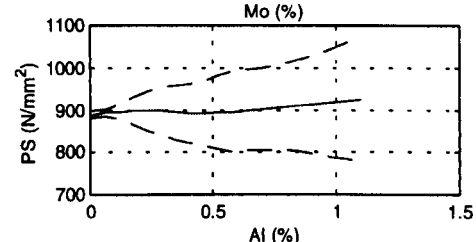


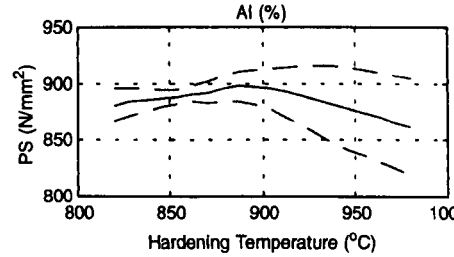




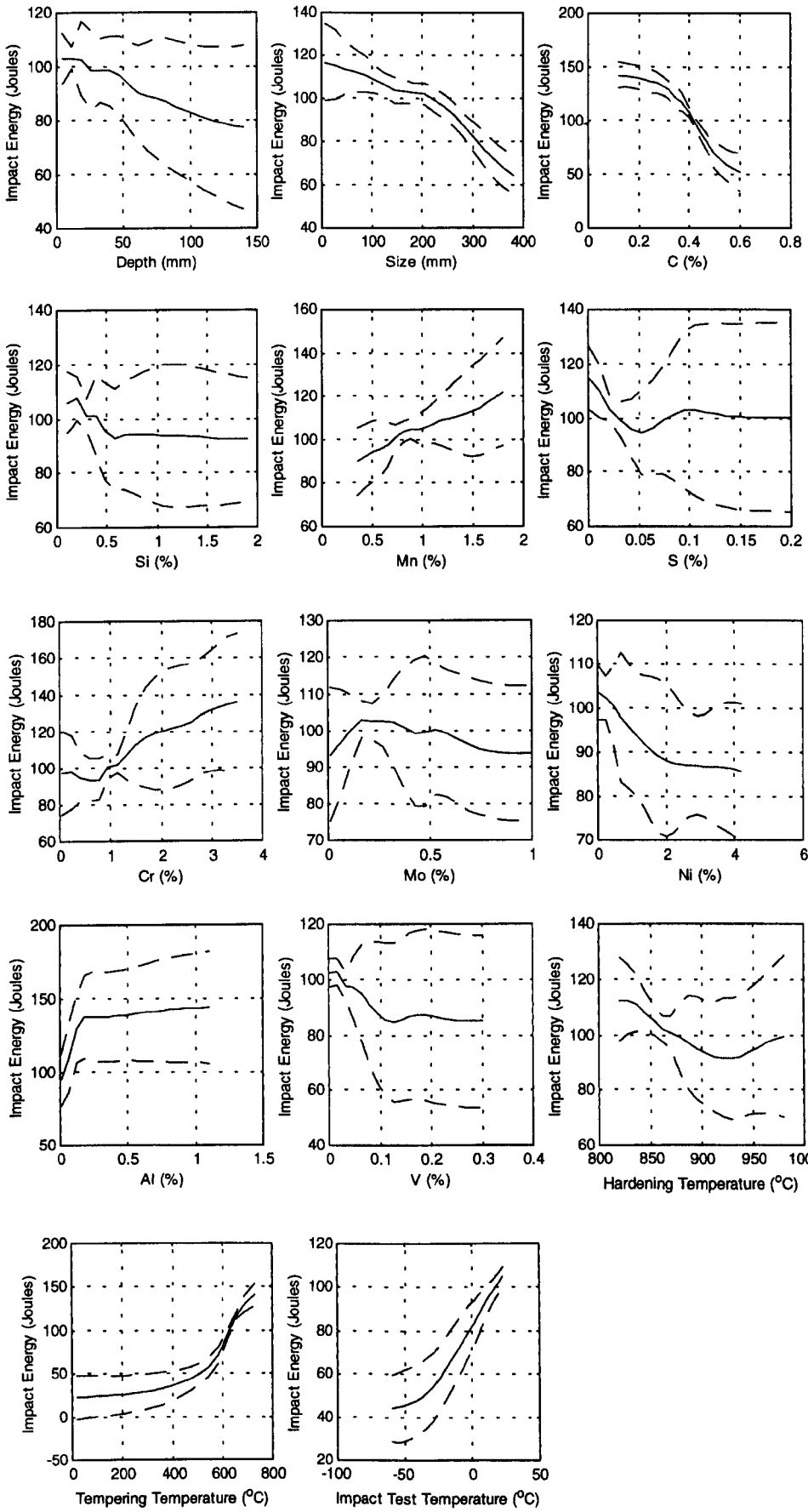




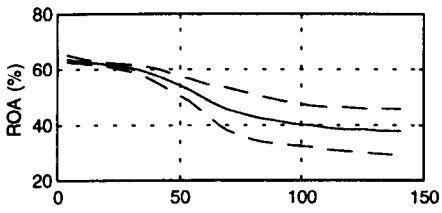


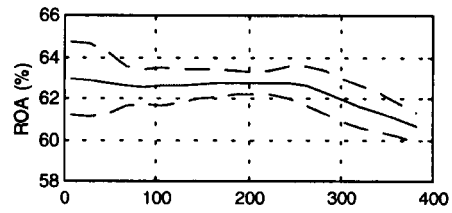


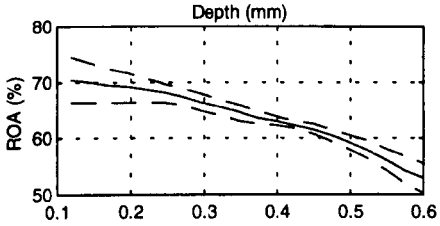
Impact model variable effects

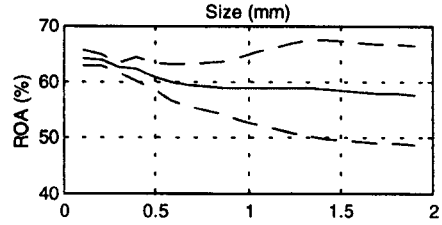


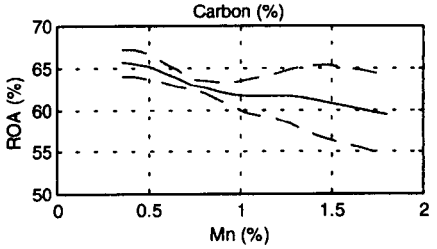
Reduction of area model variable effects

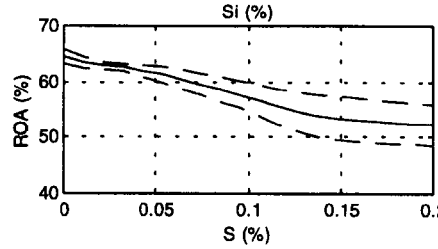


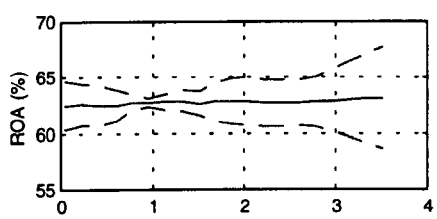


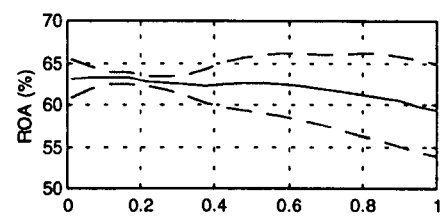


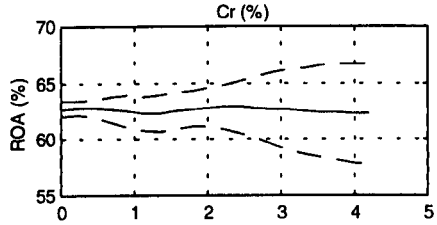


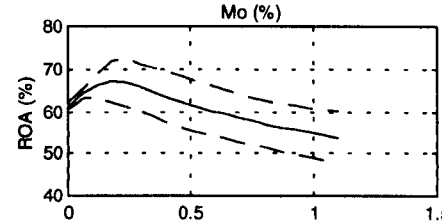


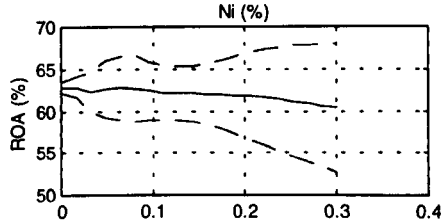


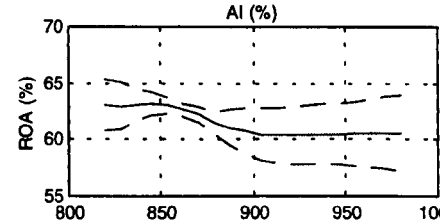


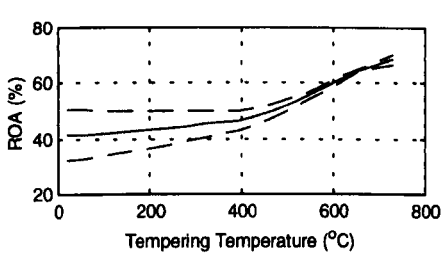












Elongation model variable effects