International Journal of Production Research,
Vol. 45, No. 3, 1 February 2007, 699–717

Exploring the non-linearity in empirical modelling of a steel system using statistical and neural network models

PRASUN DAS*† and SHUBHABRATA DATTA‡

†SQC & OR Unit, Indian Statistical Institute, New Academic Building,
6th Floor, 203 B.T. Road, Kolkata – 700 108, India
‡School of Materials Science & Engineering, Bengal Engineering & Science University,
Shibpur, Howrah – 711 103, India

(*Revision received March 2006*)

The relationship between the physical properties of metal is often very complex in nature with its chemistry and several other rolling parameters in operation. Non-linear regression models play a very important role in modelling the underlying mechanism, provided it is known. Artificial neural networks provide a wide class of general-purpose and flexible non-linear regression models. The most commonly used neural networks, called multi-layered perceptrons, can vary the complexity of the model from a simple parametric model to a highly flexible nonparametric model. In this particular work, an industry-based data set is used for learning and optimizing the neural network architecture using some well-known algorithms for prediction under neural-net systems. The outcome of the analysis is compared with the results achieved through empirical statistical modelling from its prediction error level and the knowledge of materials science.

Keywords: Artificial neural network; Steel; Regression models; Learning; Backpropagation algorithm

1. Introduction

An artificial neural network (ANN) is an information processing paradigm that is inspired by the way that biological nervous systems, such as the brain, process the information. It is composed of a large number of highly interconnected processing elements, known as neurons, working in unison to solve specific problems. ANN is configured for a specific problem such as pattern recognition, data classification or prediction through a learning process. Learning in biological systems involves adjustment to the synaptic connections that exist between the neurons. All neural networks have some set of processing units that receive inputs from the outside world, which can be referred to appropriately as the 'input units' or 'input nodes'. It does have one or more layers of 'hidden' processing units that receive inputs only from the other processing units. The set of processing units that represent the final result of the neural network computation is designed as the 'output units'. The process, which the ANN uses for learning, has a lot of similarities with the process of statistical estimation. Neural networks are nothing more than function approximation tools which learn the relationship between independent and dependent variables, much like regression analysis or other traditional approaches.

Concrete applications of neural networks to real problems started from the end of the 1980s when the backpropagation algorithm was introduced for the computation of the network parameters. In the last decade many statisticians have investigated the properties of neural networks and it appears that there exists a considerable overlap between statistical and neural network modelling. Neural network can be seen as a general way to parameterize data through arbitrary non-linear functions from the space of explanatory (casual, factor) variables to the space of explained (response, dependent) variables. The most commonly used artificial neural networks, called multilayer perceptrons, are nothing more than the nonlinear regression and discriminate analysis (Sarle 1994). Along with some relevant statistical techniques, ANN provides better treatments to some common problems of modelling and inference (Cheng and Titterington 1994). There are similarities of terminologies between statistics and neural network like (variables, features), (estimation, learning), (estimates, weights), (interpolation, generalization); (observations, patterns) and many others (Sarle 1994). Some neural network models are similar or almost identical to statistical techniques. For example, feed-forward nets with no hidden layers are the generalized linear model, probabilistic neural nets are identical to kernel discriminate analysis, Kohonen learning for adoptive vector quantization is very similar to K-means cluster analysis, etc. The principal difference between neural networks and statistical approaches is that neural network makes no assumptions about the statistical distribution or properties of the data, and therefore tends to be more useful in business and practical situations.

Neural network can also be applied very effectively and efficiently to those areas where developing a physical model is very difficult. Several attempts have already been made in semiconductor processing, but those were mostly confined to the plasma etching process of the wafers (Himmel and May 1993, Kim and May 1994). Jeong *et al.* (2005) obtained NNPLS-based prediction model and optimized the relationship between critical process parameters of photo-etching process and the defect quality of shadow mask using a genetic algorithm. Jiao *et al.* (2005) modeled the dimensional error in turning operations using a fuzzy adaptive neural network considering the complexity of machine tool structure and the cutting process. In another work, a multi-layered fuzzy neural network is used to improve the performance of conventional automatic landing systems (Juang and Chio 2005). Pacella *et al.* (2004) compared the performance of adaptive resonance theory-based neural network and the conventional control charts to detect the abnormal patterns observed during control of manufacturing processes. There are several such applications in the steel-processing sector, like steel plate rolling process where predicting the properties of steel products such as yield, tensile strength and elongation are very important. Gorni (1997) described the performance of neural networks developed under different process conditions at COSIPA industrial rolling mills, Brazil. The models show better precision than the already developed intensive statistical counterparts. Singh *et al.* (1998) developed neural network models for estimating the yield and tensile strength as a function of steel composition and rolling parameters. The predictions are reasonable in the context of metallurgical principles and other data published in the literature. Datta and Banerjee (2004) optimized a NN model with 4 layers and 48 hidden nodes for studying the input–output relationship of high-strength low-alloy (HSLA) steels using the hyperbolic tangent transfer function.

This paper includes a comparative study of statistical process of prediction and that of a neural network for a highly complicated system in steel. The data used in the context are generated in the metallurgical laboratory. Dual phase (DP) steel, one of the prospective members of the family of HSLA steels, has been considered for this study. This steel possesses composite microstructure comprising hard martensite particles dispersed in the soft ferrite matrix. C, Mn, Si, Cu, Ti and B, cold deformation, cooling rate, aging time and aging temperature have been taken as input parameters, whereas hardness is designated as the output variable.

The paper contains the general discussion on the basis of prediction tools like linear regression, stepwise regression, generalized linear model (GLM), and projection pursuit regression (PPR) along with the backpropagation algorithm used by ANN for learning, generalized regression neural network (GRNN) are discussed in section 2. The implementations and results are given in section 3 followed by comparison of results. A brief discussion on the results and conclusions are stated in sections 4 and 5, respectively.

2. Materials and methods

2.1 *Linear regression analysis*

Regression analysis is a statistical technique for investing and modelling the relationship between variables. The simple linear regression model can be written as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon \tag{1}$$

where y is the response variable, $x_1, x_2, \ldots, x_p$ are predictors and error is defined by ε. The term linear signifies that (1) is a linear function of the unknown parameters $\beta_0, \ldots, \beta_p$. These are known as regression coefficients. The parameter β_j indicates the unit change in the response y per unit change in the x_j when all other predictors, x_i $(i \neq j)$, are held constant. In some situations we may get a more complex relation with quadratic or higher-order polynomials.

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \sum_{i=1}^{p} \beta_{ii} x_i^2 + \sum_{j>i} \beta_{ij} x_i x_j \tag{2}$$

The linear regression model is based on the basic assumption that the errors are random and independently normally distributed with mean $E(\varepsilon_i) = 0$ and variance $Var(\varepsilon_i) = \sigma^2$. If the assumption holds true then the parameters β_j can be estimated by the least-squares method.

One of the important parameters for regression analysis is residual mean square error (MSE). The lower the value of MSE, the higher is the adequacy of the model. Instead of MSE, there are two major parameters, by which we can assess the overall adequacy of the model—R^2 and *adjusted* R^2. These give an idea about the proportion of the overall variability in response as expressed by the model (Montgomery *et al.* 2001).

2.2 *Stepwise regression*

Evaluating all possible regressions can be burdensome computationally; various methods have been developed for evaluating only a small number of subset regression models by either adding or deleting regressors one at a time. These methods are generally referred to as stepwise type procedures. These are forward selection, backward elimination and stepwise regression. The last one is a popular combination of procedures of the first two. It is a process in which at each step all regressors, entered into the model previously, are reassessed via their partial F-statistic and considered as current members in the model (Montgomery *et al.* 2001).

2.3 *Generalised linear model (GLM)*

A generalized linear model provides a way to estimate a function (called the link function) of the mean response as linear function of the values of some set of predictors. This is written as

$$g(E(Y/X) = g(\mu) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i = \eta(X) \tag{3}$$

where g is the link function. The linear function of the predictors, $\eta(X)$, is the linear predictor. For GLM, the variance of Y may be a function of the mean response μ, i.e. $Var(Y) = \varphi\, Var(\mu)$.

A very fundamental idea is that there are two components to a GLM: the response distribution (also called the error distribution) and the link function. Another concept underlying GLM is the exponential family of distributions, which includes the Normal, Poisson, Binomial, Exponential and Gamma distributions as members. Since, the normal error in linear model is just a special case of GLM, it can, therefore, be thought of as a unifying approach to many aspects to empirical modelling and data analysis.

The estimates of the regression parameter in GLM are maximum likelihood estimates, produced by iteratively reweighted least squares (IRLS) (Myers *et al.* 2002).

2.4 *Projection pursuit regression (PPR)*

Let Y_i (response) be the observation corresponding to $X_i = (X_{i1}, \ldots, X_{ip})^T$ $\forall i = 1, \ldots, n$. Let, $a_1, a_2, \ldots,$ denote p-dimensional 'unit vectors' known as 'direction' vectors.

To find M_0, direction vector $a_1, a_2, \ldots, a_{M_0}$ and the nonlinear transformations $\phi_1, \phi_2, \ldots, \phi_{M_0}$ such that

$$Y \approx \bar{Y} + \sum_{m=1}^{M_0} \beta_m \phi_m(a_m^T X) \tag{4}$$

provide a 'good' model for the data (Y_i, X_i), $\forall\, i = 1, \ldots, n$. The 'projection' part of the PPR is that the carrier vector X is projected onto the direction vectors $a_1, a_2, \ldots, a_{M_0}$ to get the lengths $a_m^T X$, $\forall i = 1, \ldots, n$ of the projections, and the 'pursuit' part indicates that an optimization technique is used to find 'good' direction vectors $a_1, a_2, \ldots, a_{M_0}$.

More formally, Y and X are presumed to satisfy the conditional expectation model

$$E[Y/X_1, \ldots, X_p] = \mu_Y + \sum_{m=1}^{M_0} \beta_m \phi_m(a_m^T X) \tag{5}$$

where $\mu_Y = E(Y)$, and φ_m have been standardized to have mean zero and unity variance:

$$E\big[\varphi_m(a_m^T X)\big] = 0, \quad E\big[\varphi_m^2(a_m^T X)\big] = 0, \quad m = 1, \ldots, M_0$$

The true model parameters $\beta_m, \phi_m, a_m, m = 1, \ldots, M$ in equation (5) minimize the mean square error

$$E\left[Y - \mu_Y - \sum_{m=1}^{M_0} \beta_m \phi_m(a_m^T X) \right]^2 \tag{6}$$

over all possible values of β_m, ϕ_m, *and* a_m.

In order to determine a 'good' model for PPR, the value of M_0 should be optimized. First, it has to be started with $M_{\min} = 1$ and set M at a value large enough for the data analysis problem in hand. For a relatively small number of variables p, say $p \leq 4$, one can choose $M \geq p$. But for large p, it is preferred to choose $M < p$. For each order m, $1 \leq m \geq M$, PPR will evaluate the fraction of unexplained variance

$$e^2(m) = \frac{SSR(m)}{\sum_{i=1}^{n} w_i \big[Y_i - \bar{Y}\big]^2} \tag{7}$$

A plot of $e^2(m)$ versus m which is decreasing in m may suggest a good choice of $m = M_0$. Often $e^2(m)$ will decrease more rapidly when m is smaller than a good model of order M_0 and then tend to flatten out and decrease more slowly for m larger than M_0 (Friedman and Stuetzle 1981).

2.5 *Human brain and artificial neural network*

In the human brain, a typical neuron collects a signal from others through a host of fine structures called 'dendrites' (figure 1). Then it sends the spikes of electrical activity through an 'axon' into thousands of branches. At the end of each branch, the 'synapse (link)' converts these activities into excitation of other connected neurons.

The magnitude of the signal received by a neuron depends on the efficiency of the synaptic transmission. A neuron will fire (i.e. send an output impulse) if its net excitation exceeds its inhabitation by a critical amount (threshold). Firing is followed by refractory period during which the neuron will stay inactive.

2.5.1 Backpropagation. Generalizing the Widrow–Hoff learning rule to multiple-layer networks and non-linear differentiable transfer functions created backpropagation. Input vectors and the corresponding target vectors are used to train

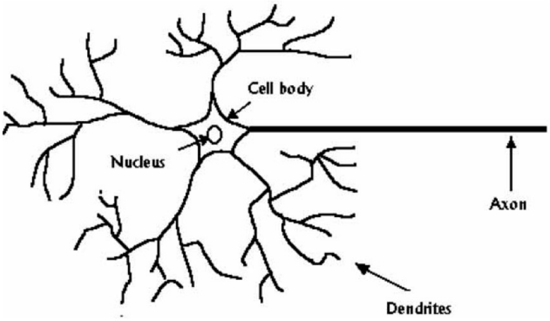

Figure 1. Human nervous system.

a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by users. Standard backpropagation is a gradient descent algorithm in which the network weights are moved along the negative of the gradient of the performance function. The term backpropagation refers to the manner in which the gradient is computed for non-linear multilayer networks. There are a number of variations on the basic algorithms that are based on other standard optimization techniques, but scale conjugate gradient and Levenberg–Merquardt algorithms are the most effective and appreciable.

2.5.1.1 Scaled conjugate gradient (SCG). The basic backpropagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient). In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. In the case of quadratic functions, exact answers are obtainable without calculating second-order derivatives.

Given a symmetric matrix $\mathbf{Q}$, two vectors $\mathbf{d_1}$ and $\mathbf{d_2}$ are said to be *conjugate* with respect to $\mathbf{Q}$ if $\mathbf{d_1^T Q d_2} = \mathbf{0}$. An important result is that when the matrix $\mathbf{Q}$ is positive-definite, a set of non-zero conjugate vectors is also linearly independent. The conjugate gradient algorithm for a quadratic problem is defined as follows:

1. Let $d_0 = -\nabla f\,(x_0) = b - Qx_0$, where $x_0 \in R^n$ is an arbitrary starting point.
2. For $k = 0, 1, \ldots, (n-1)$, define $\nabla f\,(x_k) = Qx_k - b$, and do

$$\text{(a)} \quad x_{k+1} = x_k + a_k d_k, \quad \text{where } a_k = -\frac{(\nabla f(x_k))^T d_k}{d_k^T Q d_k} \tag{8}$$

$$\text{(b)} \quad d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k, \beta_k = \frac{(\nabla f(x_k))^T Q d_k}{d_k^T Q d_k} \tag{9}$$

Commonly used stopping criteria are:

$$\left| f(x_{k+1}) - f(x_k) \right| < \varepsilon \quad \text{and} \quad \left| x_{k+1} - x_k \right| < \delta \tag{10}$$

2.5.1.2 Levenberg–Marquardt (LM). The Levenberg–Marquardt algorithm was designed to approach second-order training speed without having to compute the

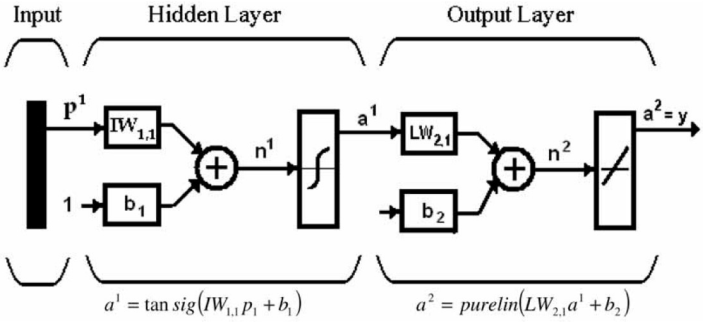
Input
Hidden Layer
Output Layer
p¹
IW₁,₁
1
b₁
n¹
a¹
LW₂,₁
b₂
n²
a² = y
$a^1 = \tan sig(IW_{1,1} p_1 + b_1)$
$a^2 = purelin(LW_{2,1} a^1 + b_2)$

Figure 2. Feed forward architecture (single hidden layer).

Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as $\mathrm{M} = J^T J$ and the gradient can be computed as $g = J^T e$, where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix. The LM algorithm uses this approximation to the Hessian matrix in the following update:

$$x_{k+1} = x_k - \left[J^T J + \mu \mathrm{I}\right]^{-1} J^T e \tag{11}$$

μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm.

The network architecture that is most commonly used with the backpropagation algorithm is multilayer feed-forward network (Rumelhart *et al.* 1986). Figure 2 shows a fully connected feed-forward network architecture.

Here

- the hidden nodes are arranged in a series of layers;
- the only permissible connection between nodes is in consecutive layers; and
- weights are specified for all connections. Biases and transfer functions are proposed for each of the hidden and output nodes.

2.5.2 Generalized regression neural network. A generalized regression neural network (GRNN) is often used for function approximation. It has a radial basis layer and a special linear layer. The architecture for the GRNN contains a hidden layer with radial basis neurons, and a special linear layer for the output layer.

The input to the radial basis transfer function is the vector distance between its weight vector W and the input vector P, multiplied by the bias b. The transfer function for a radial basis neuron is $radbas(n) = e^{-n^2}$. The plot of the *radbas* transfer function looks like a Gaussian distribution. So, it is clear from the plot that as the distance between W and P decreases, the output increases. The bias b allows the sensitivity of *radbas* neuron to be adjusted. The network will tend to respond with the target vector associated with the nearest design input vector. As spread gets larger the radial basis function's slope gets smoother and several neurons may respond to an input vector (Specht 1991).

3. Implementation and results

3.1 *Data background*

Development of high-strength steels combined with adequate ductility, formability and fracture toughness has drawn the interest of several metallurgists. The demand of such steel is mainly in the sectors like automobile, defense and naval applications. Dual phase (DP) steel provides the option for achieving judicious balance among the desired mechanical properties. The microstructural constitution of DP offers the opportunity of configuring the same in a flexible manner by varying the volume fraction, morphology and distribution of the constituent phases.

In this regard, copper (Cu), being an element known to strengthen through solid solution in ferrite as well as through precipitation hardening, a Ti-B micro alloyed steel has been alloyed with Cu to utilize the individual effect of Ti, B, and Cu as well as their synergistic combined effect. The purpose is to improve the hardenability of austenite which in turn is expected to give rise a DP microstructure depending on the finish rolling condition and cooling rate.

The alloy chemistry (C, Mn, Si, Cu, Ti and B), cold deformation, cooling rate, aging time and aging temperature have been taken as input parameters, whereas hardness is designated as the output variable. The data used for the present exercise have been generated in the laboratories. The chemical analyses are done in atomic spectrometer. Rolling has been carried in a laboratory scale two-high rolling mill. The hardness testing has been carried out in Vicker's hardness testing machine.

The ranges of the variables used in the present work are listed in table 1. Each variable is normalized within the range of 0 to 1 for ANN modelling by the operation given below and used in the same form for other statistical techniques as well.

$$X_N = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{12}$$

Table 1. The minimum and maximum limits of the parameters.

Parameter	Maximum	Minimum
C (wt%)	0.055	0.035
Mn (wt%)	1.72	1.47
Si (wt%)	0.569	0.336
Ti (wt%)	0.047	0
B (wt%)	0.0025	0
Cu (wt%)	2.17	0
Cooling rate (CoolRt) (°C/s)	90	3
Cold deformation (ColdDf) (%)	70	0
Aging time (AgTime) (mins)	600	200
Aging temperature (AgTmp) (°C)	600	15
Hardness (VHN)	405	132

where X_N is the normalized value of the variable X, and $X_{\max}$ and $X_{\min}$ are the maximum and the minimum values of X, respectively.

3.2 *Results*

The results of modelling as found using different statistical techniques along with neural network models are presented below.

3.2.1 Linear model. The simple linear regression model for the data considered is:

$$\begin{aligned}\mathit{Hardness} = {} & 0.1381 - 0.1221(\mathrm{C}) + 0.1248(\mathrm{Mn}) - 0.1972(\mathrm{Si}) + 0.1544(\mathrm{Ti}) - 0.0227(\mathrm{B}) \\ & + 0.4271(\mathrm{Cu}) + 0.1769(\mathit{CoolRt}) + 0.359(\mathit{ColdDf}) - 0.1236(\mathit{AgTmp}) \\ & - 0.0355(\mathit{AgTime}) \qquad (13)\end{aligned}$$

The ANOVA table for the above model in given in table 2. The MSE and Multiple-R^2 found for this model were 0.00472 and 89.95%, respectively.

3.2.2 Stepwise regression. The best-fitted second-order linear model found through stepwise regression analysis is

$$\begin{aligned}\mathit{Hardness} = {} & -1.2146 + 4.3838(\mathrm{C}) + 6.9557(\mathrm{Mn}) - 4.2404(\mathrm{Si}) - 0.1474(\mathrm{Ti}) + 1.6710(\mathrm{B}) \\ & + 0.9606(\mathrm{Cu}) + 0.3151(\mathit{CoolRt}) + 0.6309(\mathit{ColdDf}) + 0.2289(\mathit{AgTmp}) \\ & - 0.0772(\mathit{AgTime}) - 5.1646(\mathrm{C}^2) - 3.3686(\mathrm{Mn}^2) - 0.1292(\mathit{ColdDf}^2) \\ & - 0.2453(\mathit{AgTmp}^2) - 0.0564(\mathrm{Mn})(\mathit{ColdDf}) \\ & - 0.0708(\mathrm{Si})(\mathit{CoolRt}) - 0.1402(\mathit{CoolRt})(\mathit{ColdDf}) - 0.0974(\mathit{CoolRt})(\mathit{AgTmp}) \\ & - 0.2114(\mathit{ColdDf})(\mathit{AgTmp}) + 0.5662(\mathit{ColdDf})(\mathit{AgTime}) \qquad (14)\end{aligned}$$

The ANOVA table for the model is shown in table 3. The MSE and Multiple-R^2 found for this model were 0.00349 and 92.89%, respectively.

Table 2. The ANOVA table for simple linear model.

Source	df	Sum of squares	Mean sqr	*F*-value	*P*-value
C	1	0.0370	0.03709	7.8572	0.0054
Mn	1	0.0096	0.00961	2.0358	0.1549
Si	1	1.4389	1.43891	304.79	0
Ti	1	0.8742	0.87426	185.19	0
B	1	0.0548	0.05487	11.621	0.0007
Cu	1	2.3189	2.31894	491.20	0
CoolRt	1	1.9215	1.92153	407.02	0
Cold.Df.	1	2.9005	2.90055	614.41	0
AgTmp.	1	0.1940	0.19403	41.100	0
AgTime	1	0.0127	0.01279	2.7092	0.1011
Residuals	231	1.0905	0.00472		

Table 3. The ANOVA table for the stepwise regression model.

Source	df	Sum of squares	Mean square	*F*-value	*P*-value
C	1	0.037	0.037	10.62	0.001
Mn	1	0.009	0.009	2.75	0.09
Si	1	1.438	1.438	412.3	0
Ti	1	0.874	0.874	250.5	0
B	1	0.054	0.054	15.72	9.9E−05
Cu	1	2.318	2.31	664.5	0
CR	1	1.921	1.921	550.6	0
CD	1	2.9	2.9	831.1	0
AgTemp	1	0.194	0.194	55.59	0
AgTime	1	0.012	0.012	3.66	0.056
C^2	1	0.0002	0.0002	0.072	0.787
Mn^2	1	0.079	0.079	22.85	3.2E−06
CD^2	1	0.015	0.015	4.27	0.039
$AgTemp^2$	1	0.009	0.009	2.66	0.103
Mn:CD	1	0.006	0.006	1.84	0.175
Si:CR	1	0.019	0.019	5.67	0.018
CR:CD	1	0.076	0.076	21.7	5.5E−06
CR:AgTemp	1	0.031	0.031	9.15	0.002
CD:AgTemp	1	0.042	0.042	12.22	0.0005
CD:AgTime	1	0.038	0.038	11.03	0.001
Residuals	221	0.771	0.003		

3.2.3 Generalized linear model. The generalized linear model (considering the second-order polynomial) is given below:

$$\begin{aligned}
Hardness = {} & -1.179 + 4.262(\mathrm{C}) + 6.706(\mathrm{Mn}) - 4.105(\mathrm{Si}) - 0.155(\mathrm{Ti}) + 1.622(\mathrm{B}) \\
& + 0.969(\mathrm{Cu}) + 0.34(CoolRt) + 0.628(ColdDf) + 0.207(AgTmp) \\
& - 0.056(AgTime) - 4.981(\mathrm{C}^2) - 3.259(Mn^2) - 0.128(ColdDf^2) \\
& - 0.240(AgTmp^2) + 0.011(AgTime^2) - 0.066(\mathrm{C})(CoolRt) \\
& - 0.033(\mathrm{C})(AgTime) + 0.016(\mathrm{Mn})(CoolRt) - 0.048(\mathrm{Mn})(ColdDf) \\
& + 0.021(\mathrm{Mn})(AgTmp) - 0.022(\mathrm{Mn})(AgTime) - 0.080(\mathrm{Si})(CoolRt) \\
& + 0.112(\mathrm{Si})(AgTime) + 0.032(\mathrm{Ti})(CoolRt) - 0.012(\mathrm{Ti})(AgTime) \\
& - 0.013(\mathrm{B})(CoolRt) - 0.008(\mathrm{B})(AgTme) - 0.021(\mathrm{Cu})(CoolRt) \\
& - 0.102(\mathrm{Cu})(AgTime) - 0.154(CoolRt)(ColdDf) - 0.095(CoolRt)(AgTmp) \\
& - 0.051(CoolRt)(AgTime) - 0.204(ColdDf)(AgTmp) \\
& - 0.589(ColdDf)(AgTime) + 0.0197(AgTmp)(AgTime) \qquad (15)
\end{aligned}$$

The MSE for this model was found as 0.00362.

3.2.4 Projection pursuit regression. The value of μ_y was found as 0.5182. Also, the value of m was selected as $m = 5$ from the plot of $esq(m)$ versus m. The estimates for β_m, φ_m and a_m were obtained. The MSE for this model was found as 0.00122.

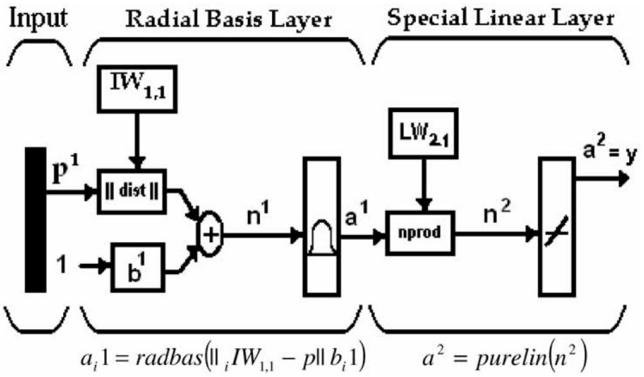

Figure 3. GRNN architecture.

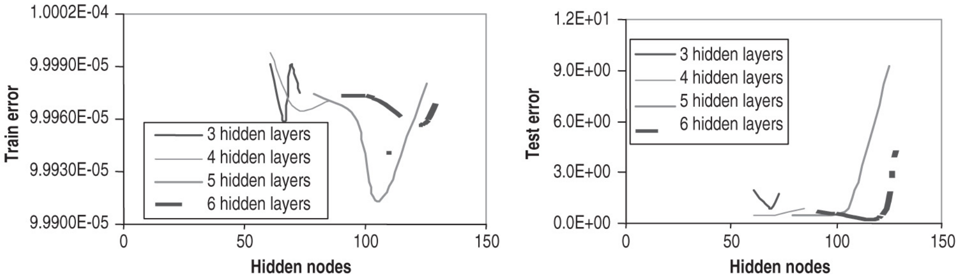

Figure 4. Training and test error vs. hidden nodes using SCG.

3.2.5 Backpropagation. The ANN used in the present case happens to be a supervised multilayered feed-forward network, trained with standard gradient descent backpropagation algorithms. The input variables, namely, six compositional and four process variables are defined as input nodes and one property variable is described as output variable. A total of 242 observations were considered for analysis of which train and test data were divided randomly in the 70:30 ratio.

Similar operations are repeated for varying numbers of hidden layers and nodes within layers in order to find out suitable network architecture. In the process of learning, the error of the calculated or predicted output in relation to the actual output is backpropagated to adjust all the weights and bias values, using both SCG and LM algorithms. For choosing the type of transfer function to be used, it has been described by the previous workers that owing to the attainable flexibility, hyperbolic tangent function is most suitable for modelling metallurgical problems.

From figure 4, the decision regarding the number of hidden layers and hidden units to be used can be taken. It shows that the training error is minimum when five hidden layers are used, however the test error is minimum for a six-layered network. But there is not much deviation between the minimum test errors estimated for five- and six-layered networks.

But in the case of the LM algorithm it is very clear from figure 5 that the minimum values of both train and test errors are obtained when four hidden layers

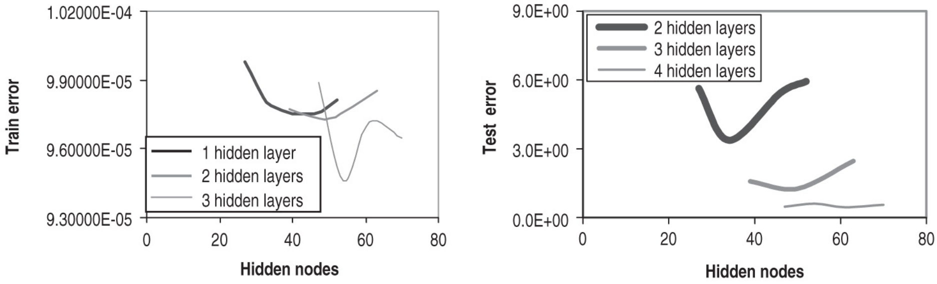

Figure 5. Training and test error vs. hidden nodes using LM.

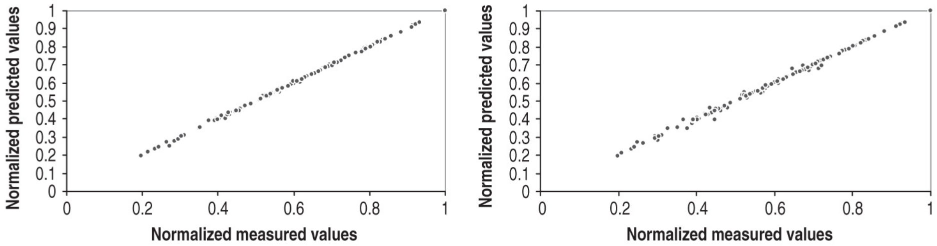

Figure 6. Predicted vs. measured values after training with SCG and LM.

are used. Also the learning algorithm LM has produced much better predictions than that of the SCG, but still it is not convenient to choose LM as the better-optimized learning algorithm. It was further observed that training through LM takes much smaller number of epochs to reach the goal; however, computational time is also very important where LM takes much higher time per epoch because of its complicated algorithm than with SCG. The following figure shows the scatter diagram for predicted versus measured values under normalized condition for the networks trained with SCG and LM.

Now, it is clear from figure 6 that both the algorithms have reasonably good capability to train the network for this specific problem, but the network trained by SCG has slightly better prediction ability. Finally, the two optimized architecture were selected as (10-35-28-21-14-8-1) and (10-20-16-11-17-1) when SCG and LM algorithms, respectively, were used for training.

The individual effects of different alloying elements and processing parameters as predicted by the trained ANN with optimized architecture are shown in figure 7. The training was done using the SCG algorithm. During prediction, when one parameter is varied the others are fixed in the mean value of the range of variables used for training.

3.2.6 Generalized regression neural network. It has already been discussed that in GRNN, the spread of the radial basis function is the most important criteria of optimization. As the spread changes, the test and training error changes significantly. In this study, this was observed as below.

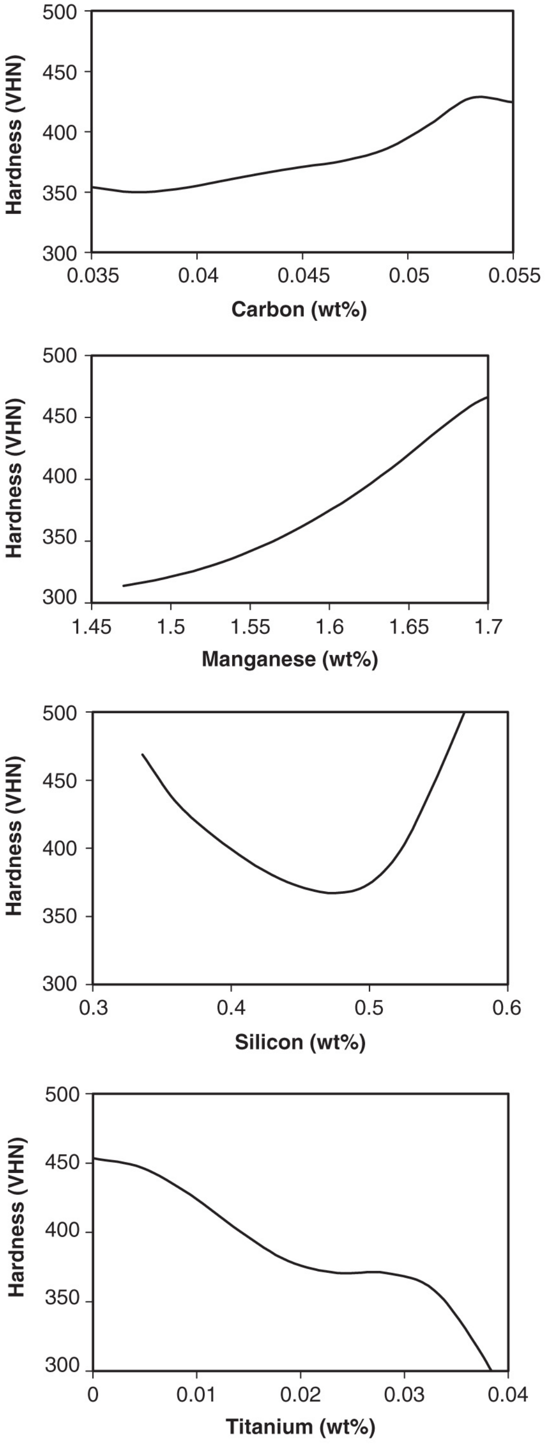

Figure 7. Individual effect of different variables on the hardness of the steel as learnt by the ANN (learning with SCG algorithm).

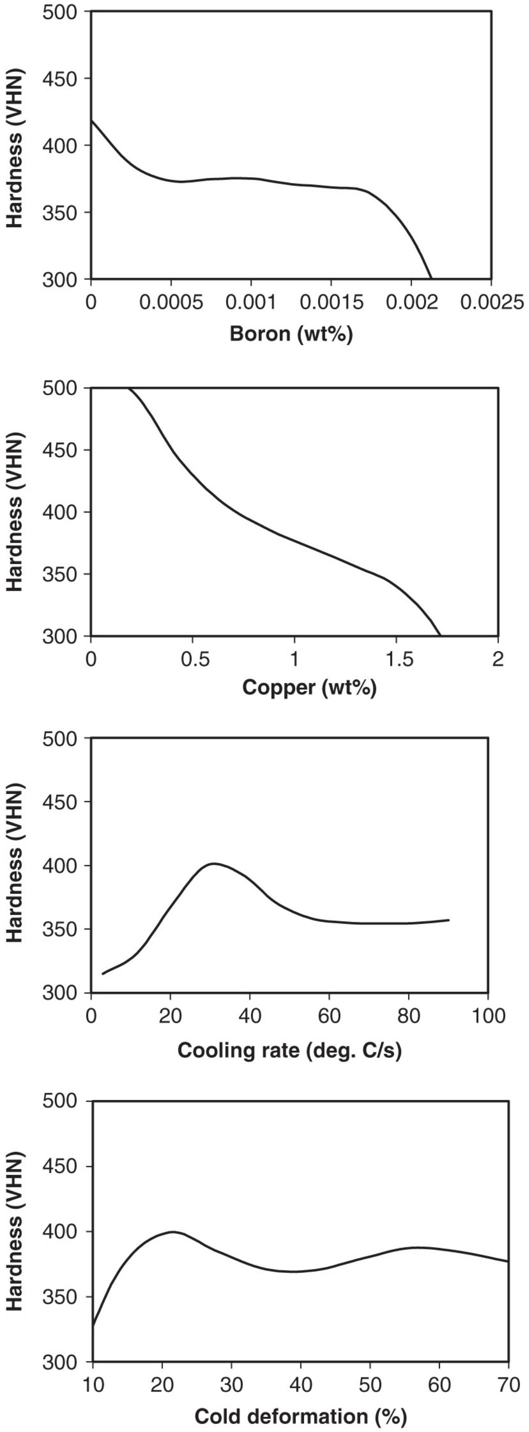

Figure 7. Continued.

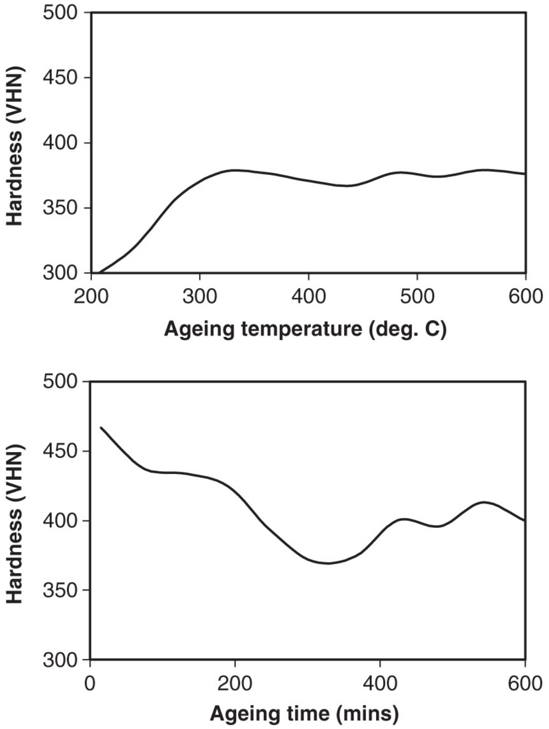

Figure 7. Continued.

It can be seen that the train error increases as the spread increases, but it is just the opposite for test error, which initially falls very rapidly, stabilizes and then starts increasing. The optimized network is chosen with spread 0.11. The magnitude of training error and testing error were 0.0012 and 0.0396, respectively.

4. Discussion

The previous section shows the approaches adopted by different statistical tools and neural network models to function approximation problem and their results considering the given data set. The comparative study reveals that the neural network with backpropagation algorithm (specifically SCG) exhibits the better prediction capability. The comparative performance of the techniques used is done based on the commonly used characteristic, i.e. mean squared error (MSE). Other means of comparison could be with respect to mean absolute percentage error (MAPE), root mean square deviation (RMSD), etc.

It is very clear from the overall discussion that in industry, where fitting a physical model is very difficult, ANN can be applied efficiently with significant prediction capability. ANN does not make any assumption on the distribution and the properties of the data and therefore tends to be more useful in practical scenarios, not only in the manufacturing industry, but also in the nonmanufacturing business sector.

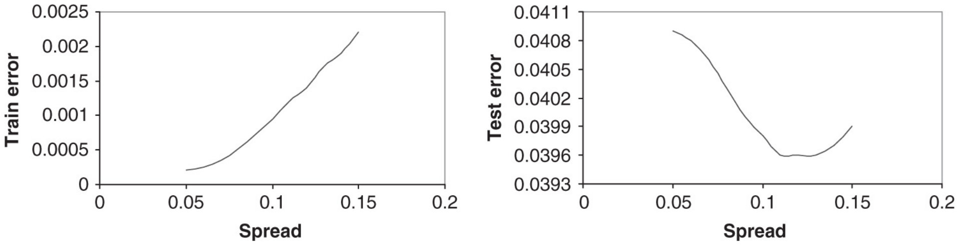

Figure 8. Train error and test error versus spread.

Table 4. Comparative study of prediction errors.

Techniques used	Mean squared error
Linear regression	0.00472
Stepwise regression	0.00349
Generalised linear model	0.00362
Projection pursuit regression	0.00122
Backpropagation	9.97985E−05
GRNN	0.0012

Although one aspect of a satisfactory model is to reach a sufficiently low prediction error level, another aspect of the model that can draw some interest to the workers is the validation of the model from the point of prior knowledge of the system being modeled. In this case the effects of the different parameters on the final output could be analysed from statistical as well as ANN models. In the case of the linear regression model, it shows that carbon has a negative effect on the hardness of the final product, which is against the prior understanding of any kind of steel. Similarly, both aging time and aging temperature have a negative effect on the hardness, which is also difficult to agree in copper-added steel. As copper precipitates in the steel during aging, the hardness of the steel increases. The *P*-value of manganese in the ANOVA table suggests lesser significance of this element in the hardness, which is absolutely not in any steel. It is quite apparent from the above observations that a linear regression model does not represent the complexity and non-linearity of the system, which is quite common for any modern steel. On the other hand, in the case of stepwise regression analysis, effects of individual variables are described in a fashion more in conformance with the metallurgical understanding, if higher-order terms are not considered. In case of interaction between the variables, interaction effect of the amount of cold deformation and the aging parameters are well recognized in physical metallurgy, as prior cold deformation increases precipitation sites, and the aging effect is enhanced considerably. In the stepwise regression model the presence of these terms are quite significant. But the other terms of interaction between the variables shown in the model cannot be explained from metallurgical concepts, particularly the interaction between manganese and cold deformation, although the significance of this term seem to be quite low as depicted in the ANOVA table. The above factors are almost the same in the case of generalized regression analysis and similar comments can be made for this model also. When trained ANN is used to study the individual effect of different variables, carbon and manganese have almost linear relationships with hardness, and the influence of carbon is higher than that of manganese in increasing the hardness of steel, which is generally true from the materials science point of view. It can be noted here that stepwise and generalized regression model show a reverse role for these elements. In the case of silicon, all the regression models demonstrate that it has a detrimental effect on the hardness of the steel. But in the neural network model the hardness value initially decreases with increase in silicon content, but after it reaches a value near to 0.5 wt% the hardness increases again. This trend may be explained from the fact that silicon has the role of hardening the ferrite as well as suppressing the formation of pearlite in steel. But from the hardness point of view the first phenomenon increases the hardness and the second produces the opposite result. In this particular system the effect of initial softening of the steel due to addition of silicon may be due to suppression of pearlite formation, which is overcome by the ferrite strengthening at higher level of copper addition. This non-linearity in the behavior of silicon could not be incorporated in the regression model. Lower amounts of titanium decrease the hardness of the steel by depleting carbon from the ferrite matrix through formation of titanium carbides. But at higher titanium content, precipitation hardening of titanium carbide are significant. In the ANN prediction of the effect of titanium in the steel under investigation, it is seen that though the formation of precipitates at higher titanium content could not increase the hardness, but at least it has stopped the softening trend. Similar phenomenon could be observed in case of boron. The effect of copper, as described by the ANN, cannot be justified totally. The only justification is that copper has an austenite hardening effect, which delays the formation of low temperature transformation products, which has decreased the overall hardness of the steel. Copper is also a ferrite strengthener, but that effect is not seen in this case. At higher copper content the strengthening effect should be more pronounced due to the precipitation of copper. In our case only the rate of softening has decreased at that level of copper addition. An increase in cooling rate after hot rolling increases the hardness due to the formation of low temperature transformation products such as bainite and/or martensite. In the present case ANN prediction shows the trend up to a certain extent of cooling rate, but the trend of softening at a much higher cooling rate is beyond explanation from the materials point of view. Cold deformation percentage and aging temperature have increasing effect on the hardness, but to a certain extent. In the present steel it is seen from the ANN prediction that all the hardening effects due to both of the variables are completed in the initial stages, and then the hardness reaches a saturation level. In the present steel the age hardening effect is purely due to precipitation of copper, which is known to precipitate in a short aging time at elevated temperature. The ANN prediction seems to suggest the insignificance of this variable in the final output.

From the above discussion it is quite evident that ANN has the superiority in reflecting the prior knowledge of the system in its prediction than the statistical methods in case of a highly complicated steel system used in this case. Although the comparisons were made based on only one system, still it seems that ANN can accommodate the complexity and non-linearity of a system better than the regression models, keeping the peculiarity of the system used in this work in mind. Hence from the above experience it can be predicted here that ANN models could produce better results compared to the statistical methods when used in other systems.

5. Conclusion

Among the statistical methods, the projection pursuit regression model is found to have better capacity for predicting the property of the steel modeled in the present case. Among the neural networks, the network using the SCG algorithm for error optimization was found to be superior to the other most common (LM) algorithm. It was also found that the neural network with backpropagation algorithm exhibits much better prediction capability than the generalized regression neural network. As the present work is carried out on a highly complex steel system it can be concluded, in general, that the neural network is able to explain such systems in a better way than using statistical methods. The ANN models can also be validated from the system's point of view in a superior manner.

References

Cheng, B. and Titterington, D.M., Neural networks: a review from a statistical perspective. *Statist. Sci.*, 1994, **9**(1), 2–54.

Datta, S. and Banerjee, M.K., Optimizing parameters of supervised learning techniques (ANN) for precise mapping input output relationship in TMCP steels. *Scand. J. Metall.*, 2004, **34**, 1–6.

Friedman, J.H. and Stuetzle, W., Projection pursuit regression. *J. Am. Stat. Assoc.*, 1981, **76**(376), 817–823.

Gorni, A.A., The application of neural networks in the modeling of plate rolling process. *JOM-e*, 1997, **49**(4) (http://www.tms.org/pubs/journals/JOM/9704/Gorni/).

Himmel, C.D. and May, G.S., Advantages of plasma etch modeling using neural networks over statistical techniques. *IEEE Trans. Semiconductor Manuf.*, 1993, **6**(2), 103–111.

Jeong, B., Lee, J. and Cho, H., Efficient optimization of process parameters in shadow mask manufacturing using NNPLS and genetic algorithm. *Int. J. Prod. Res.*, 2005, **43**(15), 3209–3230.

Jiao, Y., Pei, Z.J., Lei, S., Lee, E.S. and Fisher, G.R., Fuzzy adaptive networks in machining process modelling: dimensional error prediction for turning operations. *Int. J. Prod. Res.*, 2005, **43**(14), 2931–2948.

Juang, J.-G. and Chio, J.-Z., Fuzzy modelling control for aircraft automatic landing system. *Int. J. Syst. Sci.*, 2005, **36**(2), 77–87.

Kim, B. and May, G.S., An optimal neural network process model for plasma etching. *IEEE Trans. Semiconductor Manufact.*, 1994, **7**(1), 12–21.

Montgomery, D.C., Peck, E.A. and Vining, G.G., *Introduction to Linear Regression Analysis*, 2001 (John Wiley & Sons, Inc.: New York).

Myers, R.H., Montgomery, D.C. and Vining, G.G., *Generalized Linear Models*, 2002 (John Wiley & Sons, Inc.: New York).

Pacella, M., Semeraro, Q. and Anglani, A., Adaptive resonance theory-based neural algorithms for manufacturing process quality control. *Int. J. Prod. Res.*, 2004, **42**(21), 4581–4607.

Rumelhart, D.E., Geoffrey, E.H. and Ronald, J.W., Learning representation by back-propagating errors. *Nature*, 1986, **323**, 533–536.

Sarle, W.S., Neural networks and statistical methods, in *Proceeding of the Nineteenth Annual SAS Users Group International Conference*, 1994.

Singh, S.B., Bhadeshia, H.K.D.H., Mackay, D.J.C., Carey, H. and Martin, I., Neural network analysis of steel plate processing. *Iron Making Steel Making*, 1998, **25**(5), 355–365.

Specht, D.F, A general regression neural network. *IEEE Trans. Neural Netw.*, 1991, **3**(6), 568–576.