# Optimum model for predicting temperature settings on hot dip galvanising line

**F. J. Martínez-de-Pisón*[1], A. V. Pernía[1], A. González[1], L. M. López-Ochoa[2] and J. B. Ordieres[3]**

Controlling the annealing cycle in a hot dip galvanising line (HDGL) is vital if each coil treated is to be properly galvanised and the steel is to have the right properties. Current HDGL furnace control models usually take into account the dimensions of the coil to be dipped and, in some cases, the type of steel. This paper presents a new model for monitoring furnace temperature settings, which considers not just the coil dimensions but also the chemical composition of the steel. This enables the model to be adjusted more suitably to each type of steel to be dipped, so that the HDGL annealing cycle is optimised and rendered more efficient in dealing with new products. The ultimate aim is to find a model that is equally efficient for new types of steel coil that have not been processed before and whose dimensions and chemical compositions are different from coils processed previously. To find the best model, this paper compares various new and classical algorithms for developing a precise and efficient prediction model capable of determining the three temperature settings for heating on an HDGL located in Avilés (Spain) on the basis of the physical and chemical characteristics of the coils to be processed and the preset process conditions.

**Keywords:** Hot dip galvanising line, Annealing furnace, Data mining, Artificial intelligence, Modelling industrial processes

## List of symbols

| | |
|---|---|
| Al, Cu, Ni, Cr, Nb | chemical composition of steel, wt-% |
| C, Mn, Si, S, P | chemical composition of steel, wt-% |
| THC1 | zone 1 set point temperature (initial heating zone), °C |
| THC3 | zone 3 set point temperature (intermediate heating zone), °C |
| THC5 | zone 5 set point temperature (final eating zone), °C |
| THICKCOIL | strip thickness at the furnace entrance, mm |
| TMPP2 | strip temperature at the heating zone exit, °C |
| TMPP2CNG | strip set point temperature at the heating zone exit, °C |
| TMPP1 | strip temperature at the heating zone entrance, °C |
| V, Ti, B, N | chemical composition of steel, wt-% |
| VELMED | strip velocity inside the furnace, m $min^{-1}$ |
| WIDTHCOIL | strip width at the furnace entrance, mm |
| $\varepsilon$ | emissivity |

## Introduction

The incorporation of new products into industrial processing plants frequently necessitates changes in the ways in which those processes are controlled or manually adjusted. Adjustment usually takes a long time, can result in errors and defects in products and puts production engineers under additional stress.

There is increasing demand in industry for a process model capable of responding correctly to the requirements not just of product types already processed, but also of new types.

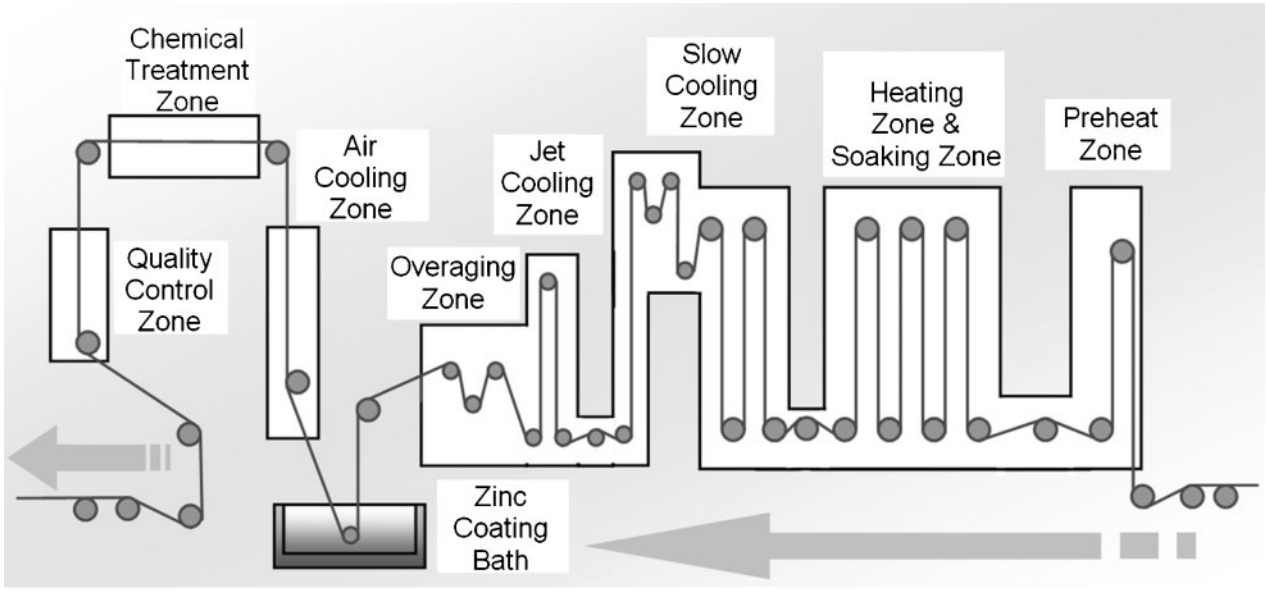
The new and classical techniques of data mining (DM) and artificial intelligence (AI) enable past data to be used to develop efficient models capable of improving on previous results. The challenge is to develop overall models that can learn from the past but can still be efficient when faced with new operating conditions in the future.

This article compares numerous new and classical DM and AI techniques and their practical application to the development of an overall model capable of precisely predicting furnace heating temperatures on a hot dip galvanising line (HDGL). These comparisons are used

[1]EDMANS Group, Departamento de Ingeniería Mecánica, Universidad de La Rioja, Logroño, Spain
[2]GI-TENECO Group, Departamento de Ingeniería Mecánica, Universidad de La Rioja, Logroño, Spain
[3]PMQ Group, Departamento de Ingeniería de Organización, ETSII, Universidad Politécnica de Madrid, Madrid, Spain

*Corresponding author, email fjmartin@unirioja.es


Published by Maney on behalf of the Institute
Received 21 August 2009; accepted 1 November 2009
DOI 10.1179/030192309X12573371383604




**1 Basic schematic drawing of HDGL**

to determine which technique is best suited to generating the best models.

The ultimate goal is to produce a model capable of predicting furnace temperature settings not just for types of steel coil that have already been processed but also for new types not previously encountered. In this work, the new model is a substantial development on earlier models: it is better able to react to new product types, which it does by adjusting to the characteristics of the steel to be processed.

The model is developed in three stages:

(i) first, a database is created that contains historical data on the process. A stratified sample is developed that enables existing cases to be homogenised so as to increase the reliability of the models created

(ii) then it is checked using a number of different DM and AI techniques to determine which is best suited to developing models capable of predicting the temperature settings needed on the furnace depending on the physical and chemical characteristics of the coils that make up the strip and the preset process conditions

(iii) finally, the models set-up are tested on new types of steel coil to determine the extent to which they can be generalised.

## Description of problem

A continuous HDGL is composed of several stages (Fig. 1). The initial material is the steel coil from the cold rolling process with the required thickness. The steel is unwound and run through a series of vertical loops within the furnace. The temperature and cooling rates are controlled to obtain the desired mechanical properties for each steel type.

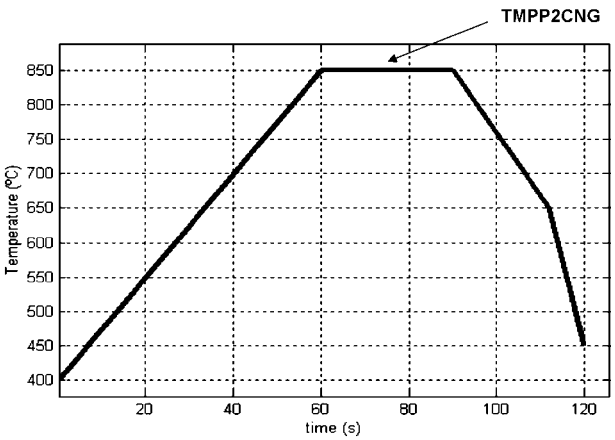
Figure 2 represents one example of the annealing treatment that each steel coil has to undergo in the annealing furnace. An efficient control of this annealing treatment is fundamental for the process of coating, for improving the properties of the coil's steel and for reducing energy costs. This can be carried out by regulating the temperature in each area of the furnace and controlling the strip velocity within an optimal range of values.

Over the last few decades, different control methods have been developed, which make use of mathematical models based on differential equations to help explain the phenomena of heat transmission by way of radiation and convection.[1–4]

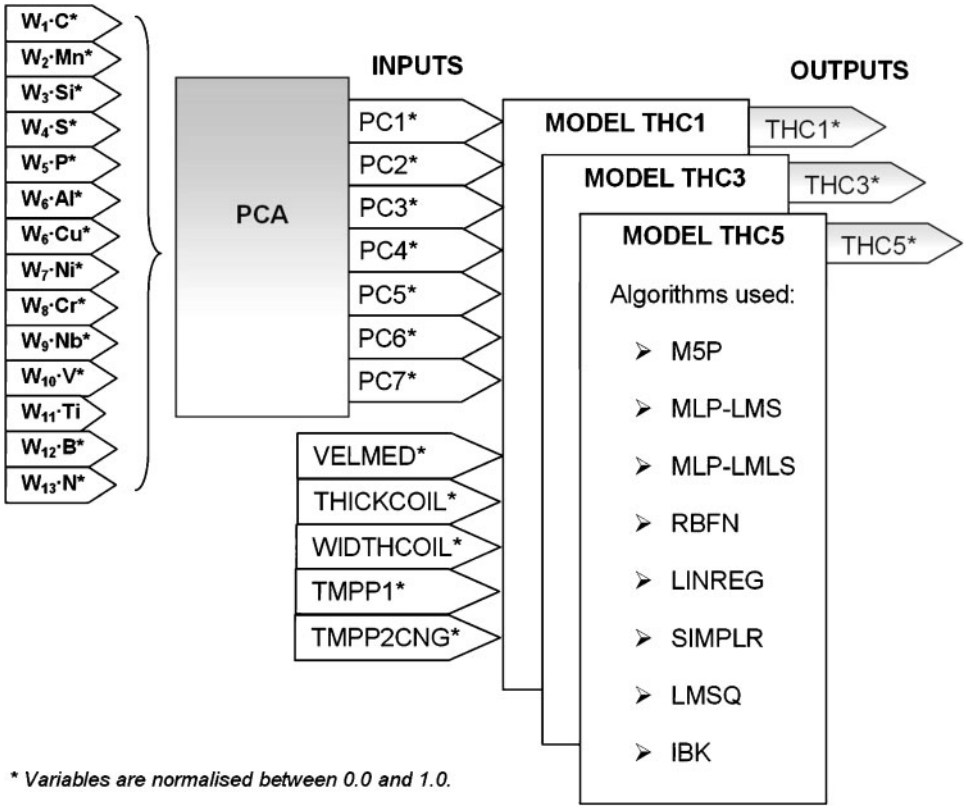
But Bloch *et al.*[5] conclude that metallurgical knowledge alone does not suffice for determining precise furnace temperature settings for each product type because there are complex, non-linear relationships between all the variables involved in the process. This justifies the use of non-linear learning techniques capable of learning from the adjustments made by operators over the years.

In recent years, research has been directed more towards the use of neural networks to control modelling, optimisation and steel manufacturing processes.[6–9] This is due primarily to the fact that these processes and subprocesses are repetitive, highly automated and have a large number of well known defining variables.

Various authors have presented papers on the use of neural networks to improve process controls in the iron and steel industry, in general, and in the galvanising process, in particular.[10–17]




**2 Example of annealing treatment curve**




**3 Models for THC1, THC3 and THC5 using principal component analysis (PCA) to reduce input dimension and dependences**

Most prediction models developed to date are designed for specific groups of steel. For these models to be implemented in an industrial plant, a different model must be developed for each type of steel in the database. It takes a great deal of time and effort to generate and validate the models required for all the products handled by a firm. Moreover, coils whose chemical composition differs slightly from the rest may be processed incorrectly by the control model.

It would therefore be much more useful to develop an overall model capable of working out the furnace settings not only for products already recorded in the historical database but also for coils with dimensions and steel types that have not been processed before. Such a model must be capable of using not only the coil dimensions and process conditions preset but also the chemical composition of the steel in each coil.

## Methodology

### Data acquisition, preprocessing and selection of variables

Data are acquired from the computer processing area based on the historical data generated continuously during the galvanising process. The variables are selected according to their relevance to the furnace heating zone.

Finally, the database is composed of 48 017 instances with input variables: the width of each coil (WIDTHCOIL), its thickness (THICKCOIL), the temperature of the strip on entering the furnace (TMPP1), the speed of the strip (VELMED), the requested output temperature of the strip (TMPP2CNG) and the chemical composition of the steel (C, Mn, Si, S, P, Al, Cu, Ni, Cr, Nb, V, Ti, B and N), and output variables: the temperature of three zones of the furnace (THC1= initial zone, THC3=intermediate zone and THC5=final zone).

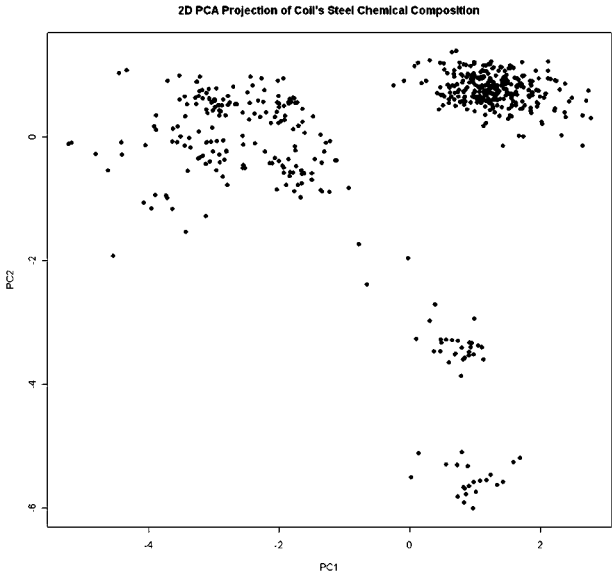
All variables are measured every 100 m along the strip. The strip velocity is measured in the centre of the furnace, and it is reasonable to assume that the strip maintains the same velocity throughout the heating zone.

The cases selected are those in which the input and output temperatures are in a steady state, and the absolute errors of the final temperature of the strip (difference between the target and actual temperature of the strip at the exit of the furnace) into the heating zone are less than 10°C. In this way, the model learns to predict the correct temperature settings of the strip furnace for coils with different thicknesses and widths, both when the settings are controlled manually and when the system runs automatically. In other words, it learns both from the experience of the human operator when the process is carried out correctly by hand and from the mathematical model when the process is correctly controlled automatically.

### Design of regression model

There are 19 input variables in the model (14 of them for the chemical composition of the steel), so principal component analysis (PCA) is used to reduce dimensions and eliminate the high level of dependence between these last variables.

Figure 3 shows the final design of the model. Each variable for the chemical composition of the steel is first multiplied by a weight $w_i$, which determines its approximate estimated level of influence into the non-linear relations with heat transfer coefficients and thermal emissivity coefficient of the steel $\varepsilon$.




**4 Principal component analysis (PCA) projection of coils according to chemical composition, using two principal components**

Figure 4 shows the PCA projection for the coils using the first two principal components obtained from the 14 normalised and weighted values for their chemical composition. It can be seen that there is one large group of coils of a particular steel type and several scattered groups of coils with other chemical compositions. From the PCA obtained, the first seven principle axes are selected. Between them, they account for 87·44% of the original variance. The 14 input variables are thus reduced to seven variables, which are independent of one another. Finally, each furnace temperature setting prediction model comprises 12 input variables and one output variable (Fig. 3).

## Final preprocessing

To improve the prediction capability of models and prevent them from learning more from the most numerous coil types and less from the least numerous, a preliminary stratified sample is taken and repeated to homogenise the number of cases in the database. To that end, a hierarchical clustering is drawn up (Fig. 5) on the basis of the 12 input variables in the database. One thousand clusters are obtained. Finally, a sample of 55 data instances is taken from each cluster to make up a database of 55 000 items.

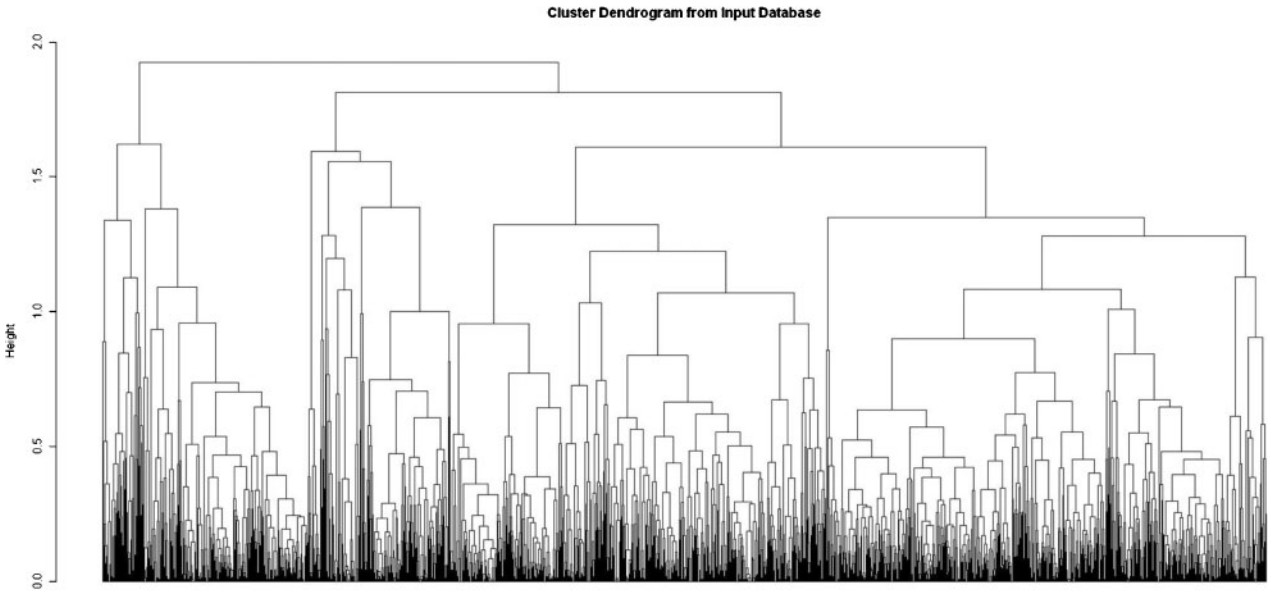
## Selecting best data mining techniques

To find models that generate low prediction errors, a battery of algorithms are used:

(i) M5P algorithm (M5P): this implements base routines for generating M5Model trees. A decision list for regression problems is generated using separate and conquer. In each iteration, it builds a model tree using M5 and makes the 'best' leaf into a rule. Quinlan's M5P[18] can learn such piecewise linear models. M5P also generates a decision tree that indicates when to use which linear model

(ii) MultilayerPerceptron (MLP):[19] a classifier and predictor that uses back propagation to classify instances. All nodes in this network are sigmoid, except when the class is numeric. In the latter case, the output nodes become unthresholded linear units. Two types of MLP are used: least mean square error (LMS) and least mean log squares error (LMLS[20]) criteria. The second type allows neural networks to be obtained that are less affected by noise at network outlets. Training is performed with networks that have between three and 45 neurons in the hidden layer

(iii) RBFNetwork (RBFN): this implements a normalised Gaussian radial basis function network.[19] It uses the $k$ means clustering algorithm to provide the basis functions and learns either a logistic regression (discrete class problems) or a linear regression (numeric class problems). In addition, a symmetric multivariate Gaussian is fitted to the data from each cluster. If the class is nominal, it uses the given number of clusters per class. It standardises all




**5 Cluster dendrogram for obtaining homogeneous training cases**

numeric attributes on a zero mean and unit variance

(iv) LinearRegression (LINREG):[21] a class for using linear regression for prediction. It uses the 'Akaike' criterion for variable selection and is able to deal with weighted instances

(v) SimpleLinearRegression (SIMPLR): this uses only the best attribute to obtain the model. It is useful for running comparisons with other algorithms

(vi) LeastMedSq (LMSQ):[22] this implements a least median squared linear regression to make predictions. Least squared regression functions are generated from random subsamples of the data. The least squared regression that has the lowest median squared error is chosen as the final model

(vii) IBk (IBk):[23] this is a version of the $k$ nearest neighbour algorithm. $K$ is the number of neighbours to be used. Also, it enables distance weighting to be used. As it is a lazy algorithm, there is no training time.

The purpose of this work is to determine the algorithm or group of algorithms that provide the best prediction or, in other words, the algorithm that yields the lowest root mean squared error (RMSE) and mean absolute error (MAE) for different coils other than those used for model construction. These errors are

$$\mathrm{RMSE}=\left\{\frac{1}{n}\sum_{k=1}^{n}[y(k)-\hat{y}(k)]^2\right\}^{1/2} \quad (1)$$

and

$$\mathrm{MAE}=\frac{1}{n}\sum_{k=1}^{n}|y(k)-\hat{y}(k)| \quad (2)$$

where y and $\hat{y}$ are the measured and predicted outputs respectively, and n is the number of points of the database used to validate the models.

For this purpose, 10 models of the configuration of each type of algorithm are trained with 70% of the data, and the remaining data (30%) are used for validating each model.

The WEKA[24] suite and AMORE[25] library from R software[26] are used to develop the different models.

## Results and discussion

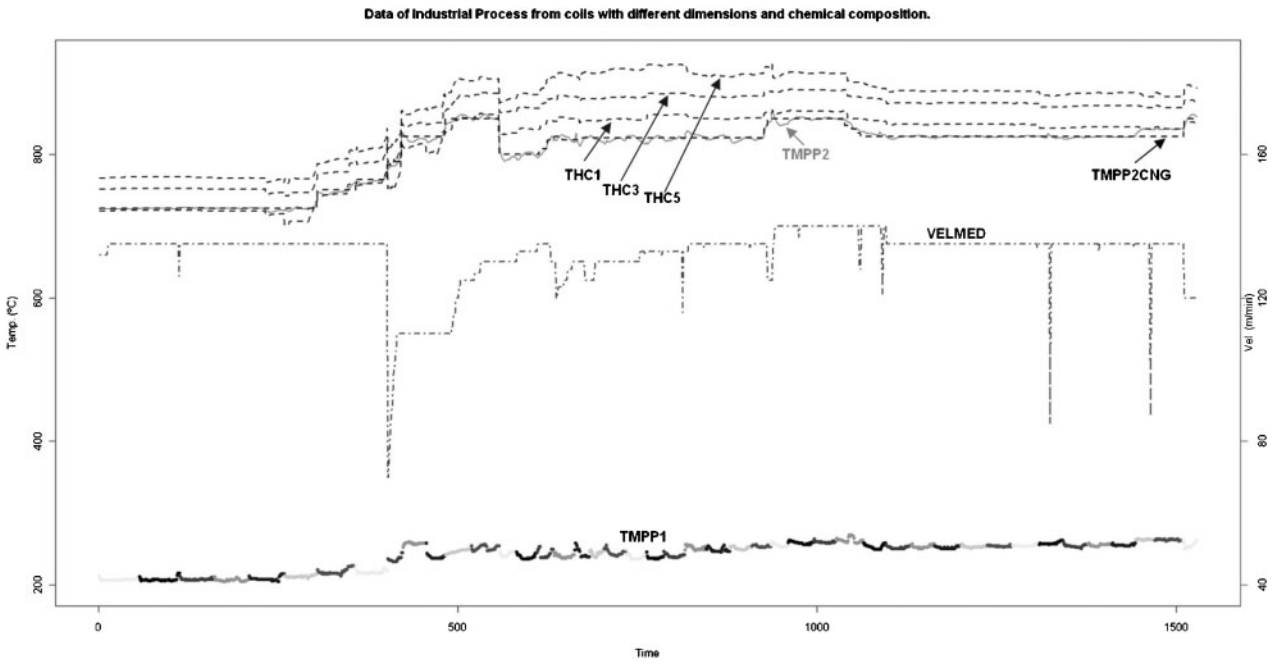
The results of the training process for THC1 are given in Table 1. This table presents the mean (MEAN), maximum (MAX), minimum (MIN) and standard deviation (SD) of RMSE and MAE validation errors for 10 models of each type of configuration of the algorithm.

The last column shows the time required to set up 10 models and obtain the relevant validation errors. Clearly, the longest training time is required by MLP networks with large numbers of neurons in the hidden layer and by LMLS networks.

As can be seen from these tables, the average RMSE for validation, drawn up with the 30% of data items not used to set up the models, is less than 1·4% for models based on $k$ neighbours (IBk). Similarly, models based on

**Table 1 Results of modelling process for THC1: validation errors for each model configuration (order by mean of RMSE)**

| Algorithm | RMSEMEAN | RMSEMAX | RMSEMIN | RMSESD | MAEMEAN | MAEMAX | MAEMIN | MAESD | Time, s |
|---|---|---|---|---|---|---|---|---|---|
| IBk($K$=1) | 0·0121 | 0·0127 | 0·0111 | 0·0005 | 0·0040 | 0·0041 | 0·0038 | 0·0001 | 0·08 |
| IBk($K$=2) | 0·0131 | 0·0136 | 0·0124 | 0·0004 | 0·0045 | 0·0046 | 0·0044 | 0·0001 | 0·08 |
| IBk($K$=3) | 0·0138 | 0·0147 | 0·0133 | 0·0004 | 0·0049 | 0·0050 | 0·0048 | 0·0001 | 0·08 |
| M5P | 0·0180 | 0·0219 | 0·0163 | 0·0017 | 0·0089 | 0·0091 | 0·0087 | 0·0001 | 137·67 |
| MLP LMLS 25 | 0·0285 | 0·0316 | 0·0264 | 0·0017 | 0·0191 | 0·0220 | 0·0179 | 0·0014 | 13 148·00 |
| MLP LMLS 35 | 0·0292 | 0·0316 | 0·0273 | 0·0015 | 0·0192 | 0·0215 | 0·0174 | 0·0014 | 16 496·00 |
| MLP LMLS 15 | 0·0293 | 0·0310 | 0·0279 | 0·0009 | 0·0196 | 0·0217 | 0·0189 | 0·0009 | 6876·00 |
| MLP LMS 30 | 0·0294 | 0·0319 | 0·0268 | 0·0019 | 0·0200 | 0·0229 | 0·0175 | 0·0018 | 11 035·00 |
| MLP LMLS 45 | 0·0296 | 0·0334 | 0·0268 | 0·0021 | 0·0202 | 0·0243 | 0·0179 | 0·0020 | 13 594·00 |
| MLP LMS 15 | 0·0296 | 0·0319 | 0·0280 | 0·0011 | 0·0197 | 0·0220 | 0·0187 | 0·0010 | 6412·00 |
| MLP LMLS 20 | 0·0296 | 0·0339 | 0·0272 | 0·0021 | 0·0200 | 0·0245 | 0·0181 | 0·0020 | 8014·00 |
| MLP LMS 20 | 0·0298 | 0·0353 | 0·0277 | 0·0022 | 0·0200 | 0·0252 | 0·0181 | 0·0021 | 9195·00 |
| MLP LMS 25 | 0·0298 | 0·0337 | 0·0278 | 0·0017 | 0·0202 | 0·0236 | 0·0179 | 0·0017 | 8377·00 |
| MLP LMLS 30 | 0·0300 | 0·0331 | 0·0274 | 0·0019 | 0·0202 | 0·0228 | 0·0179 | 0·0015 | 9724·00 |
| MLP LMLS 40 | 0·0302 | 0·0355 | 0·0269 | 0·0029 | 0·0206 | 0·0255 | 0·0177 | 0·0029 | 13 534·00 |
| MLP LMS 45 | 0·0303 | 0·0352 | 0·0272 | 0·0026 | 0·0207 | 0·0259 | 0·0175 | 0·0028 | 17003·00 |
| MLP LMS 40 | 0·0305 | 0·0354 | 0·0284 | 0·0020 | 0·0211 | 0·0261 | 0·0184 | 0·0022 | 12 843·00 |
| MLP LMLS 10 | 0·0306 | 0·0316 | 0·0295 | 0·0005 | 0·0208 | 0·0211 | 0·0202 | 0·0003 | 2834·00 |
| MLP LMS 10 | 0·0309 | 0·0325 | 0·0292 | 0·0009 | 0·0210 | 0·0226 | 0·0199 | 0·0008 | 2996·00 |
| MLP LMS 35 | 0·0311 | 0·0396 | 0·0270 | 0·0033 | 0·0214 | 0·0299 | 0·0178 | 0·0032 | 9231·00 |
| MLP LMS 07 | 0·0317 | 0·0328 | 0·0307 | 0·0007 | 0·0217 | 0·0229 | 0·0209 | 0·0006 | 1771·00 |
| MLP LMLS 07 | 0·0317 | 0·0329 | 0·0305 | 0·0007 | 0·0217 | 0·0222 | 0·0210 | 0·0004 | 1930·00 |
| MLP LMS 05 | 0·0328 | 0·0337 | 0·0314 | 0·0007 | 0·0226 | 0·0230 | 0·0221 | 0·0003 | 1463·00 |
| MLP LMLS 05 | 0·0332 | 0·0349 | 0·0323 | 0·0008 | 0·0231 | 0·0251 | 0·0220 | 0·0009 | 1448·00 |
| MLP LMLS 03 | 0·0353 | 0·0360 | 0·0342 | 0·0005 | 0·0248 | 0·0252 | 0·0244 | 0·0003 | 606·00 |
| MLP LMS 03 | 0·0356 | 0·0366 | 0·0351 | 0·0005 | 0·0252 | 0·0262 | 0·0247 | 0·0004 | 575·00 |
| LINREG | 0·0475 | 0·0479 | 0·0471 | 0·0003 | 0·0350 | 0·0352 | 0·0348 | 0·0001 | 2·30 |
| LMSQ | 0·0533 | 0·0687 | 0·0476 | 0·0075 | 0·0377 | 0·0450 | 0·0353 | 0·0034 | 917·16 |
| RBFN(30) | 0·0742 | 0·0776 | 0·0687 | 0·0031 | 0·0533 | 0·0558 | 0·0498 | 0·0021 | 298·27 |
| RBFN(20) | 0·0831 | 0·0869 | 0·0785 | 0·0028 | 0·0615 | 0·0675 | 0·0556 | 0·0031 | 193·02 |
| RBFN(15) | 0·0869 | 0·0896 | 0·0843 | 0·0016 | 0·0651 | 0·0679 | 0·0607 | 0·0019 | 127·47 |
| SIMPLR | 0·0913 | 0·0922 | 0·0906 | 0·0005 | 0·0689 | 0·0695 | 0·0683 | 0·0004 | 1·00 |
| RBFN(10) | 0·0935 | 0·1004 | 0·0866 | 0·0046 | 0·0719 | 0·0805 | 0·0657 | 0·0046 | 111·40 |
| RBFN(05) | 0·1036 | 0·1134 | 0·0959 | 0·0042 | 0·0815 | 0·0909 | 0·0758 | 0·0038 | 66·85 |
| RBFN(03) | 0·1215 | 0·1270 | 0·1126 | 0·0049 | 0·0981 | 0·1028 | 0·0912 | 0·0038 | 48·55 |




**6 Industrial process data from coils with different dimensions and chemical composition used to test models**

M5P regression trees have an average RMSE of less than 1·9%.

The next best models are MLP networks (LMLS and LMS) with large numbers of neurons in the hidden layer (15, 20, 25, 30 or 40), though the standard deviation is lower in MLPs with fewer neurons in their hidden layer. This indicates that these last models are much more stable and are consistent in their errors.

The average RMSE for linear models (LINREQ and LMSQ) is ~1% worse than that of the worst MLP network.

Finally, radial basis function networks perform worst.

The results of the training processes for the models generated for THC3 and THC5 are very similar to those for THC1.

For new cases to be predicted, IBk based models select the $K$ nearest cases according to a distance function and give the average of those cases as their result. Since the data involved are measurements in industrial processes, there will most likely be several cases in the training database (70%) which repeat each new case in the validation database (30%). Algorithms of this type give very good results in the training phase when there are many repeated cases in the databases.

Regression trees are highly complex models (each tree comprises over 780 leaves), so they fit each example in the database exactly.

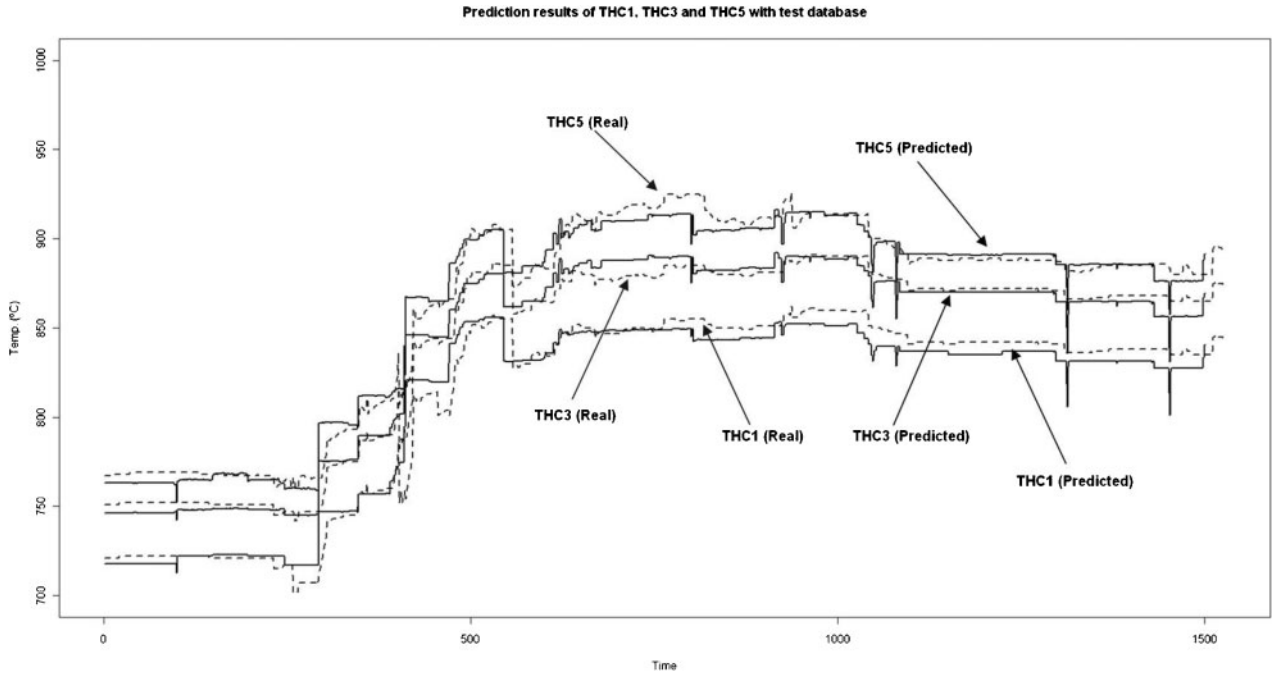
But what happens if the authors introduce a new testing database with coils made from different types of steel or with dimensions not found in the initial database, i.e. cases different from those in the training database?

Figure 6 shows the process data graphs used for testing on coils with different chemical compositions and dimensions from those used in the previous phase. The coil entry temperature (TMPP1) shows each coil processed in a different colour. The average speed, the furnace temperatures and the actual and target temperatures of the strip can be seen at the top. Table 2 shows the test errors obtained from this database for THC1.

When the models are compared with data from new coils, it emerges that $k$ neighbour and M5P based algorithms perform far worse than MLP neural networks with small numbers of neurons (three, five or

**Table 2 Test errors of model for THC1 from new database with new coils with different dimensions and chemical steel composition**

| Algorithm | RMSETEST | MEANTEST |
|---|---|---|
| MLP LMLS 03 | 0·0309 | 0·0241 |
| MLP LMS 03 | 0·0312 | 0·0244 |
| MLP LMLS 05 | 0·0315 | 0·0227 |
| LMSQ | 0·0324 | 0·0252 |
| MLP LMS 05 | 0·0326 | 0·0234 |
| LINREG | 0·0333 | 0·0257 |
| MLP LMS 07 | 0·0337 | 0·0237 |
| MLP LMLS 07 | 0·0341 | 0·0244 |
| MLP LMLS 25 | 0·0346 | 0·0252 |
| MLP LMS 15 | 0·0351 | 0·0255 |
| MLP LMS 20 | 0·0357 | 0·0260 |
| MLP LMS 25 | 0·0362 | 0·0266 |
| MLP LMLS 30 | 0·0365 | 0·0262 |
| MLP LMLS 10 | 0·0366 | 0·0265 |
| MLP LMS 30 | 0·0367 | 0·0269 |
| MLP LMLS 35 | 0·0369 | 0·0268 |
| MLP LMS 40 | 0·0370 | 0·0271 |
| MLP LMLS 40 | 0·0371 | 0·0272 |
| MLP LMS 10 | 0·0373 | 0·0267 |
| MLP LMLS 20 | 0·0374 | 0·0268 |
| MLP LMLS 45 | 0·0375 | 0·0269 |
| MLP LMLS 15 | 0·0375 | 0·0271 |
| MLP LMS 45 | 0·0384 | 0·0284 |
| MLP LMS 35 | 0·0392 | 0·0289 |
| M5P | 0·0482 | 0·0333 |
| IBk($K$=1) | 0·0501 | 0·0356 |
| IBk($K$=2) | 0·0510 | 0·0358 |
| IBk($K$=3) | 0·0513 | 0·0361 |
| SIMPLR | 0·0747 | 0·0571 |
| RBFN(30) | 0·0809 | 0·0646 |
| RBFN(10) | 0·0846 | 0·0720 |
| RBFN(15) | 0·1119 | 0·0874 |
| RBFN(20) | 0·1205 | 0·0895 |
| RBFN(03) | 0·1536 | 0·1302 |
| RBFN(05) | 0·1644 | 0·1412 |




**7 Prediction results of THC1, THC3 and THC5 with test data**

seven). LMLS networks outperform LMS in some cases, though in general, they perform very similarly.

It can be clearly seen that MLP networks produce overall models that are better at predicting for any type of steel, including types not previously entered in the training database. Figure 7 shows a detail of the actual temperature settings and the predicted settings for the best model in each case.

Also, it is interesting to note that linear models (LINREG) give a fairly low RMSE test error, close to that of the best MLP networks. They are less efficient than MLPs with few neurons in their hidden layer, but they can be used to identify approximately what level of influence each input variable may have on the models created (Table 3).

Finally, the best predictive models from THC1, THC3 and THC5 are selected and included in the furnace control model.

For the purposes of determining the extent of the improvement achieved, a coil is considered as processed correctly when the difference (ERROR) between the actual temperature of the strip when it exits the heating zone of the furnace (TMPP2) and the temperature setting given by the annealing curve for that same point (TMPP2C) is no more than 30°C.

Using a new database of 59 coils with 25 types of steel and different dimensions (Table 4), the heating process for each coil is simulated using both the old and new control models. This simulation is carried out using an advanced model robust MLP–LMLS network for steel strip developed initially by Martínez-de-Pisón *et al.*[12] and substantially improved in recent years.

The results demonstrate that with the old model, 8·5% of coils (five coils) have ERROR levels of more than 30°C, while for the new model, the figure is just 1·7% (one coil). In other words, the new model decreases the percentage of coils not properly treated by 6·8%.

**Table 3 Influence of input variables in linear models of THC1, THC3 and THC5**

| | THC1, % | THC3, % | THC5, % |
|---|---|---|---|
| THICKCOIL | 51·7 | 50·4 | 39·4 |
| WIDTHCOIL | −3·5 | −2·7 | −1·9 |
| TMPP1 | 11·9 | 10·1 | −3·1 |
| TMPP2CNG | 52·3 | 52·1 | 45·8 |
| VELMED | 36·2 | 36·2 | 32·5 |
| PC1 | −2·6 | −2·9 | 0·3 |
| PC2 | 3·5 | 3·9 | 4·3 |
| PC3 | −5·1 | −4·5 | −5·3 |
| PC4 | −5·0 | −2·6 | −4·5 |
| PC5 | −7·9 | −4·9 | −1·1 |
| PC6 | 2·4 | 0·4 | 7·4 |
| PC7 | −1·2 | 0·0 | −7·0 |

## Conclusions

The use of simple or cross-validation techniques to determine the best models based on historical databases from industrial processes may lead to the selection of models that fit closely for products already processed, but are less efficient when dealing with potential new products.

An analysis of the models obtained with new cases different from all those processed previously reveals that although there are other DM techniques that can develop closer fitting models in the process of training and validation, neural networks continue to be one of the most promising techniques for developing new, overall predictive models.

In efforts to create an overall prediction model capable of determining the temperature settings in the

**Table 4 New database used to test new control model**

| Description | Value |
|---|---|
| Number of coils of database | 59 |
| Different type of steels into database | 25 |
| THICKCOIL (range), mm | 0·601–0·775 |
| WIDTHCOIL (range), mm | 805–1180 |

heating area of an HDGL annealing furnace, the models, which proved best able to predict settings for new types of steel coil with dimensions and chemical compositions not previously encountered, were robust MLP networks (LMLS) with three, five or seven neurons in their hidden layer.

The authors' models improve prediction because they introduce the chemical composition of the steel directly, making the models better able to adjust because it can predict potential deviations from one coil to another made of steel from the same family. More complex models usually use neural networks, but the authors conduct a comparative analysis of modelling techniques, some of them new, to determine which technique is most suitable and to find its optimum configuration.

Checking with a new database shows that the new models adjust better to coils with new types of steel or different dimensions and improve the prediction of temperature settings in the annealing furnace, reducing the gap between TMPP2 and TMPP2C and thus enhancing the quality of the annealing process.

## Acknowledgements

The authors thank the 'Dirección General de Investigación' of the Spanish Ministry of Science and Innovation for the financial support of the projects DPI2006-03060, DPI2006-14784, DPI-2006-02454 and DPI2007-61090, and the European Union for the project RFS-PR-06035. Finally, the authors also thank the Autonomous Government of La Rioja for its support through the 3rd Plan Riojano de I+D+i.

## References

1. M. M. Prieto, F. J. Fernández and J. L. Rendueles: *Ironmaking Steelmaking*, 2005, **32**, 165–170.
2. M. M. Prieto, F. J. Fernández and J. L. Rendueles: *Ironmaking Steelmaking*, 2005, **32**, 171–176.
3. S. S. Sahay and K. Krishnan*: Ironmaking Steelmaking*, 2007, **34**, 89–94.
4. W. F. Wu, F. Yu, X. X. Zhang and Y. Zuo: *J. Therm. Sci.*, 2002, **11**, (2), 134–137.
5. G. Bloch, F. Sirou, V. Eustache and P. Fatrez: *IEEE Trans. Neural Networks*, 1997, **8**, (4), 910–918.
6. Y. Lu and S. W. Markward: *IEEE Trans. Neural Networks,* 1997, **6**, (8), 1328–1337.
7. J. Tenner, D. A. Linkens, P. F. Morris and T. J. Bailey: *Ironmaking Steelmaking*, 2001, **28**, (1), 15–22.
8. A. Gorni: 'The modelling of hot rolling processes using neural networks: a bibliographical review', 1998, available at: http://www.gorni.eng.br/e/neural_1998.html
9. M. Schlang, B. Lang, T. Poppe, T. Runkler and K. Weinzierl: *Control Eng. Pract.*, 2001, **9**, 975–986.
10. A. González-Marcos, J. B. Ordieres-Meré, A. V. Pernía-Espinoza and V. Torre-Suárez: *Rev. Metal. Madrid*, 2008, **44**, (1), 29–38.
11. F. J. Martínez-de-Pisón, J. Ordieres, A. Pernía, F. Alba and V. Torre: *Rev. Metal. Madrid*, 2007, **43**, (5), 325–336.
12. F. J. Martínez-de-Pisón, F. Alba, M. Castejón and J. A. González: *Ironmaking Steelmaking*, 2006, **33**, (4), 344–352.
13. V. Pernía-Espinoza, M. Castejón-Limas, A. González-Marcos and V. Lobato-Rubio: *Ironmaking Steelmaking*, 2005, **32**, (3), 1–9.
14. J. B. Ordieres-Meré, A. González-Marcos, J. A. González and V. Rubio: *Ironmaking Steelmaking*, 2004, **31**, (1), 43–50.
15. F. J. Martínez-de-Pisón: 'Optimización mediante técnicas de minería de datos del ciclo de recocido de una línea de galvanizado', PhD thesis, Universidad de La Rioja, Logroño, Spain, 2003.
16. C. Schiefer, H. P. Jörgl, F. X. Rubenzucker and H. Aberl: Proc. 14th IFAC World Cong., Beijing, China, July 1999, IFAC, 61–66.
17. Y. I. Kim, K. C. Moon, B. S. Kang, C. Han and K. S. Chang: *Control Eng. Pract.*, 1998, **6**, 1009–1014.
18. J. R. Quinlan: Proc. Australian Joint Conf. on 'Artificial intelligence', 343–348; 1992, Singapore, World Scientific.
19. S. Haykin: 'Neural networks, a comprehensive foundation', 2nd edn; 1999, Upper Saddle River, NJ, Prentice Hall.
20. V. Pernía-Espinoza, J. B. Ordieres-Meré, F. J. Martínez-de-Pisón and A. González-Marcos: *Neural Networks*, 2005, **18**, 191–204.
21. G. N. Wilkinson and C. E. Rogers: *Appl. Stat.*, 1973, **22**, 392–399.
22. S. Portnoy and R. Koenke: *Stat. Sci.*, 1997, **12**, 279–300.
23. D. Aha and D. Kibler: *Mach. Learn.*, 1991, **6**, 37–66.
24. H. Witten and E. Frank: 'Data mining: practical machine learning tools and techniques', 2nd edn; 2005, San Francisco, CA, Morgan Kaufmann.
25. M. Castejón-Limas, J. B. Ordieres Meré, E. P. Vergara, F. J. Martínez-de-Pisón, A. V. Pernía and F. Alba: 'The AMORE package: a MORE flexible neural network package', CRAN Repository, 2009, available at: http://wiki.r-project.org/rwiki/doku.php?id=packages:cran:amore
26. R Development Core Team: 'R: a language and environment for statistical computing'. R Foundation for Statistical Computing, Vienna, Austria, 2008, available at: http://www.R-project.org