

Adaptive Neural-Fuzzy Inference System for Classification of Rail Quality Data with Bootstrapping-Based Over-Sampling

Y. Y. Yang, M. Mahfouf, G. Panoutsos, Q. Zhang
Dept. of Automatic Control and Systems Eng.
IMMPETUS, The University of Sheffield
Mappin Street, Sheffield S1 3JD, United Kingdom
y.y.yang@sheffield.ac.uk

S. Thornton
Teesside Technology Centre
Tata Steel Europe
Eston Rd., Middlesbrough, TS6 6US, UK
Steve.Thornton@Tatasteel.com

Abstract— An iterative bootstrapping-based data over-sampling strategy is presented in this paper together with an adaptive neural-fuzzy inference system (ANFIS) to deal with a severely imbalanced data modelling problem. As real industrial data are often very large, containing hundreds of process variables and a huge number of data records, the selection of a compact set of input variables becomes critical for any successful modelling and analysis operations. Significant efforts have been devoted to identifying the most relevant input variables through correlation analysis and neural network based forward input selection. An optimal majority to minority class data ratio, which controls the level of data imbalance for model training, is then determined through the iterative bootstrapping process such that the combined sensitivity and specificity performance is optimised. The iterative bootstrapping ANFIS modelling strategy is then applied to a real industrial case study for rail quality classification, with the original data being provided by Tata Steel Europe. Preliminary results show a good overall performance through the iterative bootstrapping data over-sampling ANFIS modelling.

Keywords: *Imbalance data; bootstrapping, data resampling; fuzzy c-means clustering, ANFIS, steel manufacturing, rail.*

I. INTRODUCTION

Data-driven modelling has gathered pace and gained much popularity due the rapid growth in computing power and the wide availability of data/information. Modelling tools and algorithms have also evolved into more complex structures capable of dealing with highly non-linear and multi-dimensional input/output mappings. However, in many real-world modelling scenarios, the available data are often highly imbalanced, such as product quality data and patient-disease data, where the data ratio of the majority class (the ‘good’ data) towards the minority class (the ‘bad’ data) can be very high. The data imbalance problem is prevalent in many applications, including fraud detection, risk management, quality classification, and medical diagnosis. Due to the overwhelming

dominance of the majority class, the resulting model is often heavily biased to the majority cases while ignoring the minority class if the imbalance is not adequately dealt with [1]. Research in imbalanced data modelling has grown rapidly [2-4] as more researchers realized that imbalance entails more often than not sub-optimal modelling. Research methods for imbalanced data modelling can be classified into two major categories: using data resampling so that the resampled data are more balanced [5]; and modifying the learning algorithm or the cost criteria so that the classifier is biased towards the minority class [6, 7].

In this paper, a modelling strategy of integrated bootstrapping [8] data resampling and adaptive neural-fuzzy inference system (ANFIS) is proposed. The core of the ANFIS model is developed using fuzzy C-means clustering [9] on the resampled data. This modelling strategy is then applied to a case study for a large scale industrial problem provided by Tata Steel Europe, relating through process manufacturing data to finished steel rail ultrasonic test results. The remainder of the paper is organized as follows. Section 2 provides an overview of the data which are extracted from the sequential steelmaking, casting, rolling and finishing processes of the large multi-product manufacturing complex. Section 3 is devoted to data pre-processing, which deals with data consolidation, input selection, and feature extraction. As the data set is very large, containing over two hundred process variables in more than sixty-five thousand data records, the data pre-processing task becomes essential and critical toward any successful modelling practices. In Section 4, an iterative bootstrapping data resampling algorithm is developed, together with an ANFIS based classification scheme. It is found that the majority to minority class data ratio, which controls the level of data imbalance in the bootstrapping resampled data, plays a pivotal role in optimising the overall classification performance over the two different categories. Finally, concluding remarks and discussions are outlined in Section 5.

II. OVERVIEW OF THE RAIL PRODUCTION DATA

A. Brief Description of the Rail Production Route

The rail manufacturing data used for this study were collected from an integrated steel manufacturing complex belonging to Tata Steel Europe. The production line which manufactures a range of steel products including rail, consists of steelmaking, continuous casting, rolling, and finishing sub-processes, as sketched in Fig. 1. A basic oxygen furnace takes molten iron from upstream blast furnaces as well as additional scraps and refines them into steel by blowing high purity oxygen through the lance onto the surface of the hot metal. The oxidation process generates heat and removes excessive carbon and other chemical impurities (a batch process lasting approximately 30 minutes). The liquid steel is then tapped into a ladle, for further secondary steelmaking, going through a ladle metallurgical furnace where the steel chemical compositions are adjusted through alloy addition and desulphurization, and a vacuum de-gasser where harmful hydrogen and other gas inclusions are removed to improve steel cleanliness. The rail grade liquid steel is then transferred to the multi-strand continuous casting machine where steel blooms of about 8-tonne are produced. These steel blooms are subsequently heated in a reheating furnace and processed to rail via multi-pass hot rolling and straightening operations to produce rails upto 120 metres in length. The final stage of manufacture entails comprehensive Non-Destructive Testing (NDT) of every rail to ensure adherence to strict standards for internal and surface quality, as well as for dimensional accuracy, before dispatch to customers. This study focuses on the small number of rails which are rejected due to internal flaws and analytical approaches to find root causes and thus by introduction of appropriate control measures, to improve product yields.

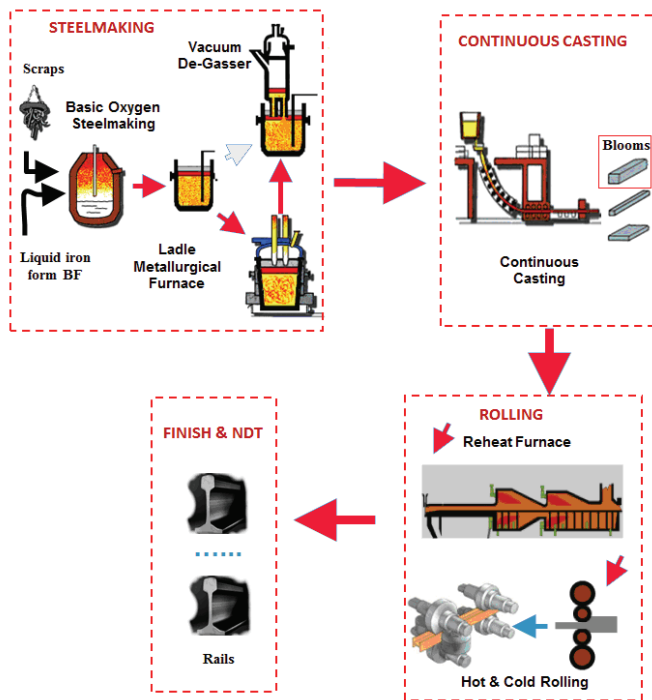


Figure 1. Overview of the rail production route

B. Overview of the Rail Manufacturing Data

In a modern rail production line, there exists extensive instrumentation for online process monitoring and control, as well as offline production management and administration. An integrated data infrastructure, which includes multiple online data servers for collecting real-time variables, process parameters, management information, key unit production time histories, and quality inspection data, is installed through the production line. These original data, which are stored in various databases, are further processed and aggregated before being stored in a master data mining server where higher level data analysis and mining tasks can be conducted. The data can also be easily accessed in this environment for other advanced modelling and optimisation tasks.

The data collected from the rail production route described above cover a two-year production period, including over 65000 data records with up to 200 variables. Preliminary investigations of the rail data revealed many challenges for modelling and analyses. Due to the large scale of the data, even simply processing the data can prove to be resource intensive, and algorithms and procedures need to be planned with great care. Basic tasks such as plotting, copying and paste are not straightforward, as they often break commonly used software specifications and norms. Often, responses become prohibitive sluggish, hence an adequate choice of analysis tools becomes a necessity as a single computing tool cannot best serve the diversified requirements demanded in the data modelling cycle. Hence, an integrated software environment has been established to deal with the large scale rail data modelling, as shown in Fig. 2.

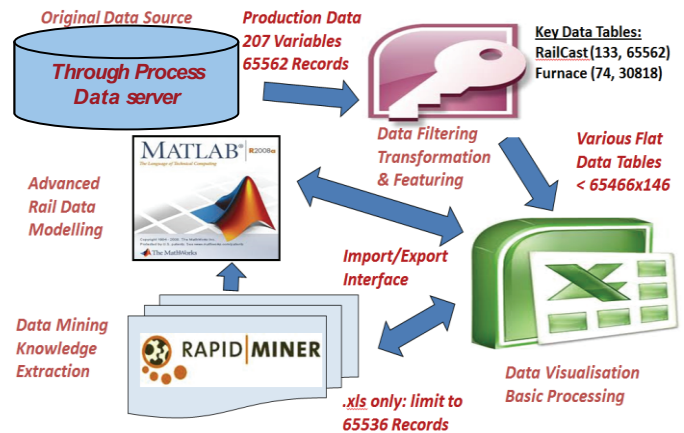


Figure 2. An integrated database and software environment

An initial inspection of the data revealed that some variables were redundant, and others contained conflicting information. A few records included also a significant amount of missing values and could not hence be used for modelling. Furthermore, many of the variables are primarily relevant to production management, hence should be eliminated for the rail quality analysis.

Based on this initial perspective, a data pre-processing strategy has been carefully designed in order to form a parsimonious model data set from the original rail production route data. As pointed out in [10-11], pre-processing is an

important step in any complicated data modelling process, and its function normally includes data cleaning, normalization, transformation, feature selection, etc. In this study, the focus of the data pre-processing operation is directed mainly towards identifying data outliers, correcting wrong data entries, checking data conflicts, missing value decision, and identifying relevant inputs. A process expert from Tata Steel was closely involved in the data pre-processing task providing essential knowledge input to explain and advise about inter variable correlations, apparent redundancy and relevance of data items. After the initial data pre-processing stage, the original rail data are reduced in both dimension and length by eliminating non-relevant variables and deleting data records with too many missing values or conflict data items. The resulting rail data set, after the pre-processing stage, consists of 140 process variables and 39687 records.

This pre-processed data set remained very large and the data dimension was still too high for any meaningful feature classification. The input dimension needed to be reduced further in order for a manageable data classification model to be elicited. Hence, the reduction of the input space dimension and the selection of the most suitable input variables will be tackled in the next section.

III. MODEL INPUT SELECTION

Input variable selection is a vital step in complex system modelling where the input-output relationship is often not too well understood. An unnecessarily large number of inputs can be detrimental to effective modelling, not only that the high dimensional space will cause the difficulty due to ‘the curse of dimensionality’, but also the irrelevant and redundant inputs tend to confuse and distort the true input-output outcomes. It is a well-known fact that the data examples needed to tune a model grow exponentially with the number of inputs, and the resources required for an increasing large data set soon become prohibitive. The central task in input selection is to identify and remove variables that have little influence on the output, so that the main emphasis can be put on the important input variables. As pointed out in [12], the objective of input variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data. Many different methods have been proposed to deal with the input selection problem, including fuzzy logic based approaches [13-14], statistical test or correlation analysis [15-16], and direct objective optimisation based approaches [17].

A. Correlation Analysis

The correlation coefficient analysis can prove sufficient enough to help identify the input/output dependency if the process under investigation is linear [18]. Even though the rail production line is complex and certainly nonlinear, it is believed that some preliminary correlation analysis will be worthwhile. Correlation coefficient analysis can help to reach a better understanding of the data to be modelled. The Pearson’s correlation coefficient can be calculated via the following equation:

$$\rho_{ij} = \frac{\sum_{k=1}^N (x_i(k) - \bar{x}_i)(x_j(k) - \bar{x}_j)}{\sqrt{\sum_{k=1}^N (x_i(k) - \bar{x}_i)^2 \sum_{k=1}^N (x_j(k) - \bar{x}_j)^2}} \quad (1)$$

Where $\rho_{ij} \in [-1, 1]$ is the correlation coefficient between variable x_i and x_j , $x_i(k)$ is the value of the k th x_i sample, N is the total number of samples, and $\bar{x}_i$ is the sample mean. When one of the variables is the output, ρ_{ij} represents the linear dependency between the output and the given input, otherwise it indicates the co-linearity between the two given inputs.

When a significant number of variables are involved in the data set, there will be a very large number of correlation coefficients. A compact form to summarize all the correlation coefficients is a correlation matrix $\mathbf{R}$, defined as follows:

$$\mathbf{R} = [\rho_{ij}]_{n \times n} = \begin{pmatrix} \rho_{11} & \cdots & \rho_{1n} \\ \vdots & \ddots & \vdots \\ \rho_{n1} & \cdots & \rho_{nn} \end{pmatrix} \quad (2)$$

Where ρ_{ij} is the correlation coefficient defined in (1), n is the total number of data variables in the data set. $\mathbf{R}$ is a symmetric and with $\rho_{ii} = 1$, $i = \{1, 2, \dots, n\}$.

For this study, data variables are limited to steelmaking and continuous casting sub-processes (named as rail-cast data, with process variables $x_1 \sim x_{70}$), as it is known that these two sub-processes have strong influences on the internal rail quality. Correlation analysis of the rail-cast data revealed that there is little linear correlation between inputs and the output (the number of rejected rails), with the top-10 correlation coefficients being listed in Table I. The correlation coefficient values are rather small, with the maximum being 0.144 related to the 9th input. The results are not constructive enough to help identify the relevant inputs for data modelling. However, the resulting correlation matrix revealed significant correlations among various potential inputs, as many correlation coefficients exceeded 0.95, as outlined in the last two columns of Table 1. Highly correlated inputs indicate significant redundant information contained in the corresponding variables, which need to be eliminated for parsimonious data modelling.

TABLE I. TOP 10 SIGNIFICANT CORRELATION COEFFICIENTS

RANK	OUTPUT CORR.	VAR	INPUT CORR.	VARS (I, J)
1	0.144	x_9	0.982	x_{50}, x_{53}
2	0.133	x_{10}	0.981	x_{24}, x_{25}
3	0.111	x_8	0.979	x_{11}, x_{40}
4	0.106	x_{13}	0.979	x_{12}, x_{40}
5	-0.097	x_7	0.978	x_{24}, x_{26}
6	-0.074	x_6	0.975	x_{52}, x_{53}
7	-0.071	x_{30}	0.967	x_{51}, x_{52}
8	0.060	x_3	0.966	x_{67}, x_{68}
9	-0.050	x_{37}	0.965	x_{68}, x_{69}
10	-0.047	x_{54}	0.953	x_{66}, x_{67}

B. Input Selection using a Neural Network Model

As is clear from the previous correlation analysis that the input-output relation is highly nonlinear, a neural network (NN) based forward selection procedure is proposed here to identify the most relevant input variables. A simple 3-layer feed-forward neural network possesses very powerful input-output mapping capability and can approximate any continuous non-linear function with arbitrary accuracy provided that enough hidden neurons have been used [19]. The output of such an NN can be expressed as follows [20]:

$$\hat{y}(k) = \sum_{j=1}^M w_j \phi_j \left(\sum_{i=1}^n w_{ij} x_i(k) + w_{j0} \right) + w_0 \quad (3)$$

where $\hat{y}(k)$ is the NN output prediction for the k th sample, w_j and w_{ji} are the relevant NN weighting coefficients at the output layer and hidden layer, respectively, w_0 and w_{j0} are the corresponding bias, ϕ_j is the nonlinear activate function of the j th hidden neuron, and M is the total number of hidden neurons.

A forward input selection procedure is proposed with the above NN model as a performance evaluator. The input selection is implemented through an iterative process, starting from selecting a single input for the NN modelling. All the potential inputs are sampled as the single input candidate, and the neural network model with the minimum root mean squared error (RMSE) is identified as NN*. The inputs corresponding to the ones used in NN* are ranked as the most important inputs identified so far. In the next round of input selection, the inputs of the NN are increased by one from the identified input set, with the newly increased input selected from all remaining potential input candidates. The above forward input selection procedure was repeated until either a sufficient number of inputs identified or no performance improvement achieved by adding more inputs. The NN training procedure used here is adapted from the one used in [22], with critical training parameters fixed via trial-and-error tuning. The number of hidden neurons is determined by the number of NN model inputs by using empirical rules, as optimisation in this scenario will be computationally too expensive. The iterative forward input selection algorithm is summarized as follows:

Step i: Initialisation of the forward input selection: $i = 1$, $\mathbf{x}^0 = \{\}$, put all the variables into the potential input candidate set $\mathbf{X}^0 = [x_1, x_2, \dots, x_n]$

Step ii: Start a new round of input selection. For each of the potential input candidate $x_j \in \mathbf{X}^0$, perform neural network modelling with extended input $\mathbf{x}^0 = [\mathbf{x}^0, x_j]$. Computing the corresponding model performance $J_j, j = \{1, 2, \dots, \dim(\mathbf{X}^0)\}$.

Step iii: Identify the best neural network model NN_{j^*} using the corresponding RMSE (Root mean square error) performance, and electing x_{j^*} as the most important input in the current round. Expanding the fixed input set $\mathbf{x}^0 = [\mathbf{x}^0, x_{j^*}]$, and eliminate it from $\mathbf{X}^0$ ($\mathbf{X}^0 = \mathbf{X}^0 \setminus x_{j^*}$).

Step iv: If the best NN performance in the current round does not improve over the previous one, or if $\dim(\mathbf{x}^0) \geq n_{\max}$, or if $\mathbf{X}^0$ is empty, then go to Step v, otherwise return to Step ii.

Step v: The elements in $\mathbf{x}^0$ are the most important input variables identified by the NN forward selection procedure. The importance ranking is the order of inclusion of the elements.

Table II summarises the results obtained from the aforementioned forward input selection procedure. As the neural network model is very powerful for input-output mapping, the nonlinear behaviour in the data set can be better dealt with. However, the computation involved in this NN based forward input ranking is quite heavy, especially when a large data set is involved. For the current input ranking task with 70 potential inputs, a maximum number of 2450 neural networks need to be trained via some kind of error back-propagation algorithm, and each neural network need to be trained over more than 20000 samples. Extra in-loop optimisations, for the hidden neuron number or learning parameters, are not computationally feasible. Hence, key training parameters and network structure parameters are fixed based on the modelling experience.

TABLE II. INPUT RANKING USING THE NN FORWARD SELECTION

No	Var	No	Var	No	Var	No	Var
1	x_8	11	x_{24}	21	x_{51}	31	x_{23}
2	x_2	12	x_{46}	22	x_{63}	32	x_{20}
3	x_9	13	x_{13}	23	x_{54}	33	x_{48}
4	x_{53}	14	x_{34}	24	x_{58}	34	x_{59}
5	x_{16}	15	x_{67}	25	x_{12}	35	x_{61}
6	x_{47}	16	x_{22}	26	x_{25}	36	x_{50}
7	x_3	17	x_{28}	27	x_{37}	37	x_{15}
8	x_{68}	18	x_{19}	28	x_{62}	38	x_{55}
9	x_{11}	19	x_{41}	29	x_1	39	x_7
10	x_{65}	20	x_6	30	x_{37}	40	x_{35}

IV. BOOTSTRAPPING RESAMPLING AND DATA CLASSIFICATION

The number of model inputs has been reduced to 39 by selecting the first 39 variables listed in Table 2 obtained by using the NN forward input ranking scheme described in Section 3. The original output, $y \in \{0, 1, 2, 3\}$, is chosen as the feature variable originating from the NDT ultrasonic testing, with $y=0$ representing a good rail and $y = \{1, 2, 3\}$ indicating a rejected rail. The non-zero discrete value of y indicates the presence of single or multiple flaws. An adaptive neural-fuzzy modelling approach was developed based on the ANFIS framework [21], however the success rate for the rejected rail classification is very poor with less than 10% correctly classified, when the imbalanced data is used for training the model. Further investigations revealed that the main cause of such poor classification is the severe data imbalance in the training data set.

A. Bootstrapping for Data Resampling

Since samples belonging to the minority class are fewer than those belonging to the majority class in the current modelling scenario, a direct approach for data resampling is adopted here in order to change the class distributions. A more balanced data can be obtained through different resampling

strategies, such as under-sampling, over sampling, or the combination of both, as is briefly discussed in [22]. In order to avoid unnecessary information loss, the resampling scheme used here is over-sampling of the minority class data, with the resampling implemented via bootstrapping [23]. In order to carry out the over-sampling, the original rail-cast data are first segregated into two data subsets, each containing data samples from a single class. The data subset corresponding to the minority class is then fed into the bootstrapping resampling algorithm, which produces a multiple randomly resampled subsets, each with the same size as the original minority subset. Those resampled subsets are then combined into an over-sampled minority class data, and then mixed with the majority class data to form the final resampled training data. The level of imbalance for the resampled training data can be controlled via a design parameter R_{MM} , defined as the ratio of the number of samples belong to the majority class to that belong to the minority class. As a price paid to avoid information loss during the resampling, the size of the resampled training data set is increased. Fig. 3 shows the effect of the R_{MM} on the size of the resampled training data.

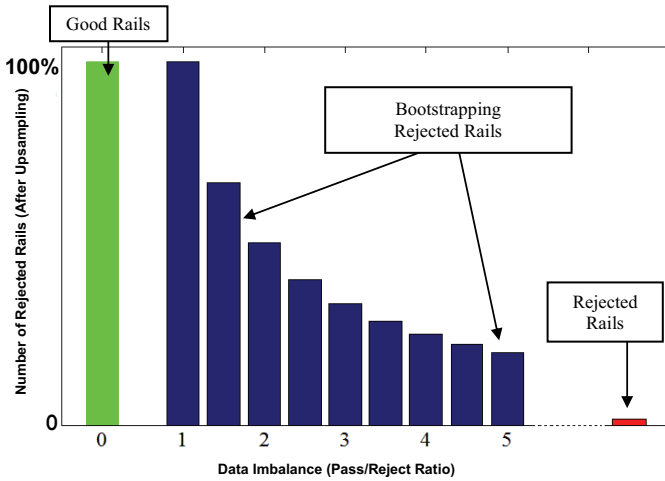


Figure 3. Effect of R_{MM} to the size of resampled training data

B. Fuzzy C-means Clustering based Classification

An adaptive neural-fuzzy inference system can be viewed as a fuzzy logic system whose rules are generated through fuzzy neural network training from a carefully assembled training data set [24-25]. A generic fuzzy inference model can be presented as a collection of fuzzy rules in the following form:

$$R_i: \text{If } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \dots \text{ and } x_n \text{ is } A_{in} \text{ then } y = z_i(\mathbf{x})$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n] \in U$ and $y \in V$ are linguistic variables, A_{ij} are fuzzy sets defined on the universe U , and $z_i(\mathbf{x})$ is a function of $\mathbf{x}$. Fig. 4 shows a simplified ANFIS architecture with two rules, where Layer 1 is for fuzzy membership mapping of the input, Layer 2 implements the fuzzy ‘and’ operation, Layer 3 normalises the firing strength of each fuzzy rule, Layer 4 performs the then part function $y = z_i(\mathbf{x})$ of the corresponding fuzzy rule, and Layer 5 computes the overall output of the fuzzy system for the given input.

In this study, the fuzzy membership takes a form of a Gaussian function, and the fuzzy rule is trained using fuzzy c-means clustering through a Matlab built-in fuzzy logic toolbox routine *genfis3* [21]. Fuzzy c-means clustering (FCM) is a well-recognized paradigm to generate fuzzy models, due to its efficiency and simplicity. After the number of clusters p (equivalent to the number fuzzy rules) is chosen, the FCM algorithm partitions all the data samples $\{\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(N)\}$ into p fuzzy clusters such that the following objective function is minimized:

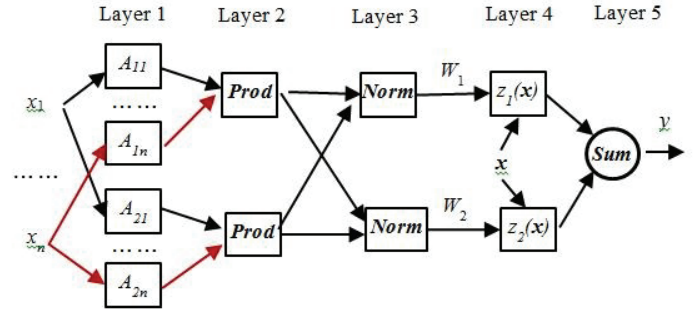


Figure 4. A simplified ANFIS architecture with 2 rules [21]

$$J_m = \sum_{k=1}^N \sum_{i=1}^p \mu_{ik}^\alpha \|\mathbf{x}(k) - \mathbf{c}_i\|^2 \quad (4)$$

where $\mathbf{c}_i$ is the centre of the i th cluster generated by fuzzy clustering, μ_{ik} is the membership degree of the k th data belonging to the i th cluster, and α is the exponent of μ_{ik} . The fuzzy membership μ_{ik} normally satisfies the following constraints:

$$\sum_{i=1}^p \mu_{ik} = 1, \quad \forall k \in \{1, N\}; \quad 0 \leq \mu_{ik} \leq 1, \quad \forall k, i \in \{1, p\} \quad (5)$$

In this study, a hybrid learning algorithm is used to optimise the parameters of the fuzzy inference systems. It applies a combination of the ‘least-squares’ method and the error back-propagation gradient descent method for training the membership function and other ANFIS parameters using the inputs-outputs in the training data set.

The number of fuzzy rules p and the majority-to-minority ratio R_{MM} are two critical parameters which will influence the ultimate modelling performance. We have developed an iterative optimisation procedure in order to identify the ‘best’ combination of these two critical design parameters under the paradigm of bootstrapping over-sampling and adaptive neural-fuzzy classification. Fig. 5 shows the diagram for the iterative optimisation scheme.

C. Preliminary Results

The pre-processed rail-cast data need to be divided into two data sets, one used for training and another for testing and with the training/testing data ratio set to 70:30. 39 inputs are used for modelling, as specified by the forward NN input selection scheme discussed in Section 3. The training data set will be used for bootstrapping resampling in order to generate a balanced training set for data modelling. The range of majority to minority case data ratio R_{MM} is selected as $[R_{min}, R_{max}] = [1.0, 5.0]$ with an increment step length of 0.5. The range for p , the number of fuzzy rules, is chosen as $[p_{min}, p_{max}] = [2, 10]$ with

incremental step of 1. The maximum training epoch is 100. Some preliminary modelling results are shown in Fig. 6.

In a classification problem, the performance of misclassification gives a better measure of the quality of the developed model than the RMSE based performance. In order to calculate such misclassification performance, the continuous outputs of the optimised ANFIS model are transformed into binary outputs, with 1 indicating the minority class ('Reject') and 0 for the majority class ('Good'). Specificity, sensitivity and accuracy performances originated in medical research [26] are adopted in this paper as the binary classification performances. 'Specificity' measures the proportion of 'negative' samples (majority class) which are correctly identified, while 'sensitivity' measures the proportion of 'positive' samples (minority class) which are correctly identified. In other words, sensitivity represents the true 'negative' rate, while specificity is the true 'positive' rate. 'Accuracy' is the overall percentage that both categories are correctly classified.

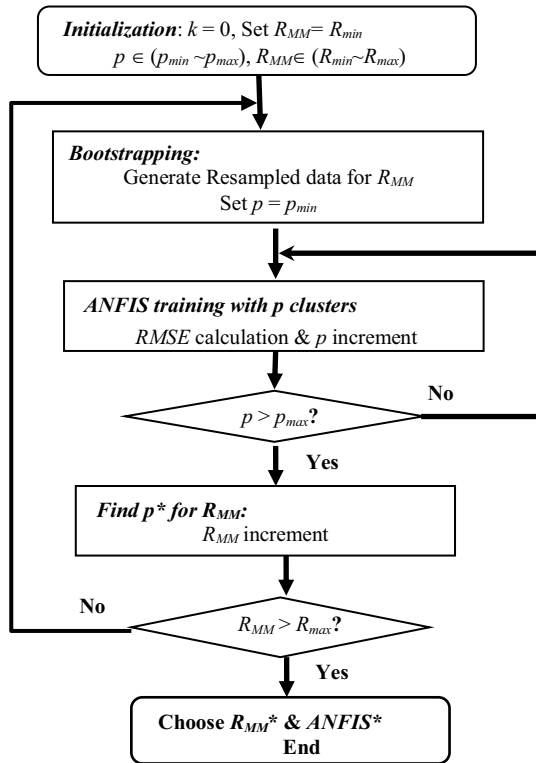
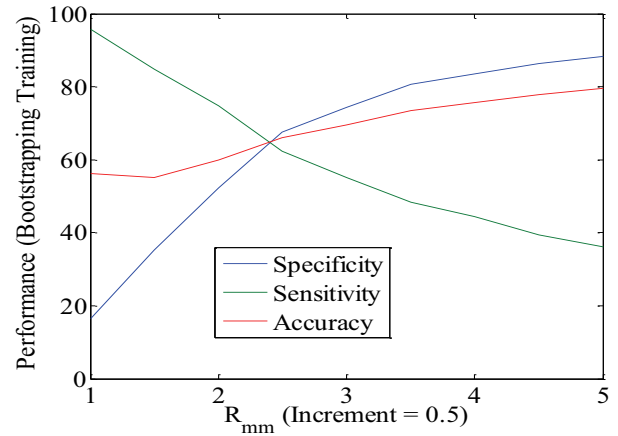


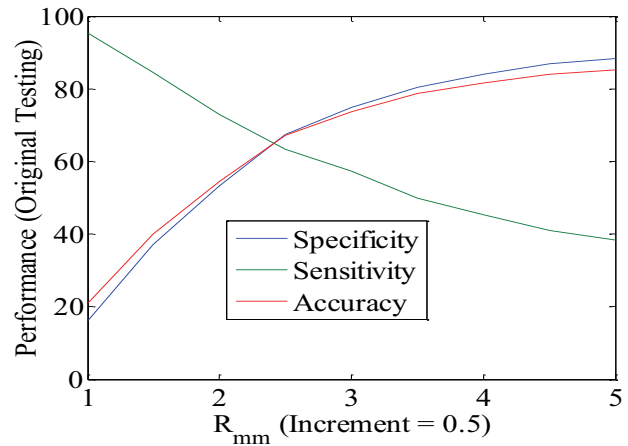
Figure 5. Iterative optimisation for p & R_{MM} in bootstrapping-ANFIS scheme

Fig. 6 shows a clear trend of the three different performances against the variation of the R_{MM} data ratio, which is used to control the data imbalance level in bootstrapping. The data over-sampling is carried out on the original training data alone, so the whole testing data remain to be 'unseen' for the model to be developed using the resampled training data. A large R_{MM} value indicates more dominance for the majority class in the resampled training data, and hence is beneficial to the performance related to the majority class, i.e., the specificity. In contrast, a small R_{MM} value indicates a reduced dominance of the majority class, and hence tends to benefit the

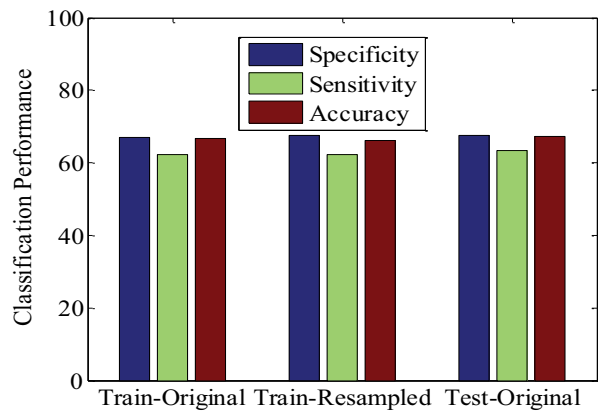
performance for the minority class, i.e., sensitivity. There exists an optimal R_{MM} such that the overall classification performance reaches its optimum. Simulation results show that, the optima R_{MM}^* value is approximately 2.5 for the current study.



(a). Measured by the over-sampled training data (Train-Original)



(b). Measured by the original testing data (Test-Original)



(c). Classification performance for $R_{MM}^* = 2.5$

Figure 6. Performance of bootstrapping ANFIS scheme

The iterative bootstrapping ANFIS model produced a good balance between the data fitting and generalisation performance, as exemplified by the results shown in Fig. 6. In

the original data modelling without resampling applied (see Fig. 7), the success rate for rejected rail classification (sensitivity) was very poor at less than 10% for the testing data set and about 16% for the training data set. Although in that case, the specificity performance was very good (over 96% for both the training and testing data sets), the model was of little practical value as the concern in the production route is to identify robust relationships that support identification and elimination of root causes for rail rejection. The sensitivity of the optimal model (discovered through the iterative procedure presented in Fig. 5) is much better at 65% for the resampled training data (Fig. 6a), and when this optimal model is applied to the original training data set (without resampling) similar results are observed. Furthermore, when the model is applied for the previously unused testing portion from the original data set the same performance results are observed (Fig. 6b) indicating that the matched performance for familiar data and for ‘unseen’ data are consistent. Fig. 6c shows the sensitivity, specificity and accuracy performances for the ANFIS model developed under the optimal resampled training data set corresponding to $R_{MM}^* = 2.5$.

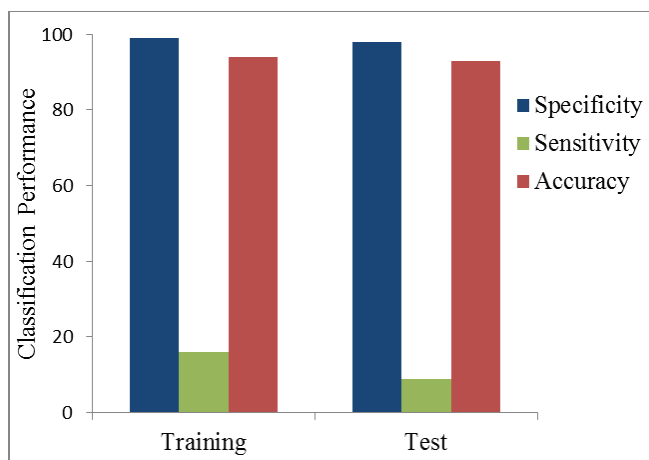


Figure 7. Classification performance using the rail data without resampling

V. CONCLUSIONS

In this paper an integrated bootstrapping-ANFIS modelling scheme has been developed to deal with severely imbalanced data classification. The fuzzy classification is implemented through an iterative process of generating oversampling training data and learning the ANFIS model parameters using the resampled training data through fuzzy c-means clustering. Since the case study involves large scale industrial data, data pre-processing plays a significant role for effective data modelling. One of the key objectives of the data pre-processing is to form a compact data set by selecting the most relevant optimal input variables, thus minimising redundancy and eliminating data conflicts. Correlation analysis and forward NN-based input ranking have been incorporated in the data pre-processing stage. Preliminary results show that the integrated bootstrapping-ANFIS modelling scheme developed is very promising, with significant performance improvement especially in terms of the sensitivity. Moreover, the developed

classification model achieved a good balance between data fitting and data generalisation.

One disadvantage of using bootstrapping-based oversampling is that the resampled training data set includes a significant level of redundancy related to the minority class data. The increase in the overall training data set size may also pose difficulties for large scale data modelling and classification, due to the relatively big increase in computing resources. We will investigate new methodologies to overcome these shortcomings in future research.

ACKNOWLEDGMENT

Financial support from the UK-EPSCRC is acknowledged under Grant EP/F023464/1. The authors wish to thank Tata Steel Europe for providing the data used in this research as well as expert knowledge to guide the data consolidation.

REFERENCES

- [1] N. V. Chawla, N. Japkowicz, and A. Kolcz, “Editorial: Special issue on learning from imbalanced data sets”, *ACM SIGKDD Explorations*, vol. 6, no. 1, pp. 1–6, 2004.
- [2] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, “Handling imbalanced datasets: A review”, *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 25–36, 2006.
- [3] G. Weiss, “Mining with rarity: A unifying framework”, *SIGKDD Explorations*, vol. 6, no. 1, pp. 7–19, 2004.
- [4] N. V. Chawla, N. Japkowicz, and A. Kotcz, Eds., *ICML’2003 Workshop on Learning from Imbalanced Data Sets*, Washington DC, 21 August 2003.
- [5] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, “SMOTE: Synthetic minority oversampling technique”, *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [6] P. Domingos, “MetaCost: A general method for making classifiers cost-sensitive”, *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pp. 155–164, 1999.
- [7] K. Huang, H. Yang, I. King, and M. R. Lyu, “Learning classifiers from imbalanced data based on biased minimax probability machine”, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, pp. 558–563, 2004.
- [8] D. S. Moore, G.P. McCabe, W.M. Duckworth, and S.L. Sclove, “Bootstrap methods and permutation tests”, *The Practice of Business Statistics*, Ch18, pp. 1–73, W. H. Freeman Eds., 2003. http://bcs.whfreeman.com/pbs/cat_140/chap18.pdf
- [9] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*, John Wiley & Sons, Inc., 1999.
- [10] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, “Data preprocessing for supervised learning”, *International Journal of Computer Science*, vol. 1, no. 2, pp. 111–117, 2006.
- [11] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, London, 2006.
- [12] I. Guyon, and A. Elisseeff, “An introduction to variable and feature selection”, *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [13] Y. Hou, J. M. Zurada, W. Karwowski, W. S. Marras, and K. Davis, “Identification of key variables using fuzzy average with fuzzy cluster distribution”, *IEEE Transactions On Fuzzy Systems*, vol. 15, no. 4, pp. 673–685, 2007.
- [14] R. Sindelár, and R. Babuska, “Input selection for nonlinear regression models”, *IEEE Transactions On Fuzzy Systems*, vol. 12, no. 5, pp. 688–696, 2004.
- [15] A. Miller, *Subset Selection in Regression*, 2nd ed., Chapman & Hall/CRC, 2002.

- [16] L. Yu, and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution", Proceedings of the 20th International Conference on Machine Learning (ICML-2003), Washington DC, 2003.
- [17] S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast incremental feature selection by gradient descent in function space", Journal of Machine Learning Research, vol. 3, pp. 1333–1356, 2003.
- [18] M. P. C. Couto, "Review of input determination techniques for neural network models based on mutual information and genetic algorithms", Neural Comput & Application, vol. 18: pp. 891–901, 2009.
- [19] K. Hornik, "Approximation capabilities of multilayer feedforward networks", Neural Networks; vol. 4, no. 2, pp. 251-257. 1991.
- [20] M. Mahfouf, and Y. Y. Yang, "A GA-optimised ensemble neural network model for charpy impact energy predictions", IFAC MMM2010, The 13th Symposium on Automation in Mining, Mineral and Metal Industry, 2-4 August 2010, Cape Town, South Africa.
- [21] The MathWorks Inc., Fuzzy Logic Toolbox 2 User's Guide, http://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf, 2010.
- [22] X. Guo, Y. Yin , C. Dong, G. Yang, and G. Zhou, "On the class imbalance problem", Fourth International Conference on Natural Computation (ICNC'08), pp. 192-201, 25-27 August 2008, Jinan, China.
- [23] M. R. Chernick, Bootstrap Methods: A Guide for Practitioners and Researchers. 2nd ed., Wiley Series in Probability and Statistics, New Jersey, 2008.
- [24] J.-S. Roger Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference Systems", IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 03, pp. 665-685, 1993.
- [25] M. Y. Chen, D. A. Linkens, D. J. Howarth, and J. H. Beynon, "Fuzzy model-based charpy impact toughness assessment for ship steels", ISIJ International, vol. 44, no. 6, pp. 1108–1113. 2004.
- [26] A. J. Saah, and D. R. Hoover. "'Sensitivity' and 'Specificity' Reconsidered: The Meaning of These Terms in Analytical and Diagnostic Settings", Ann Intern Med , vol. 126, pp. 91-94.