Hierarchical Fuzzy Support Vector Machine (SVM) for Rail Data Classification

R. Muscat*, M. Mahfouf*, A. Zughrat*, Y.Y. Yang*, S. Thornton, A.V. Khondabi* and S. Sortanos***

**Department of Automatic Control and Systems Engineering, The University of Sheffield,
Mappin St., Sheffield, S1 3JD, UK (Tel: +44 114 2225607)
(rmuscat1@sheffield.ac.uk; m.mahfouf@sheffield.ac.uk; a.zughrat@sheffield.ac.uk;
yongyyang@gmail.com; amirvase1984@gmail.com; stylianos.sortanos@avanti-conveyors.co.uk)
**Teesside Technology Centre, Tata Steel Europe, Eston Rd., Middlesbrough, TS6 6US, UK (steve.thornton@tatasteel.com)*

Abstract: This study aims at designing a modelling architecture to deal with the imbalanced data relating to the production of rails. The modelling techniques are based on Support Vector Machines (SVMs) which are sensitive to class imbalance. An internal (Biased Fuzzy SVM) and external (data under-sampling) class imbalance learning methods were applied to the data. The performance of the techniques when implemented on the latter was better, while in both cases the inclusion of a fuzzy membership improved the performance of the SVM. Fuzzy C-Means (FCM) Clustering was analysed for reducing the number of support vectors of the Fuzzy SVM model, concluding it is effective in reducing model complexity without any significant performance deterioration.

Keywords: Classification; railway steel manufacture; class imbalance; support vector machines; fuzzy support vector machines; fuzzy c-means clustering.

1 INTRODUCTION

Cost management has become a very important factor in every industry. Investments in plant, technology, research and development allow companies to reduce manufacturing costs while quality control procedures improve process output. Modelling techniques are increasingly being employed to understand the interaction and influence of input variables on the process. Tata Steel Europe is at the forefront of rail production and strives to produce the high performance products required by the rail industry.

Advances in computer processing power, together with the vast amounts of available data, have encouraged the application of machine learning techniques to different real world problems in an attempt to extract useful knowledge from the available information. Pattern classification is a supervised machine learning method in which a labelled set of data points is used to train a model which is then used to classify new test examples. Classifier performance is commonly evaluated by its accuracy. However, this metric does not correctly value the minority class in an imbalanced data set and as a result the trained model tends to be biased towards the majority class (Weiss, 2004). Many data sets from real world problems are inherently imbalanced and therefore appropriate measures need to be taken to ensure that important information due to the minority class is correctly represented by the classifier.

This study deals with the design of a modelling architecture for data related to the quality of rails produced by Tata Steel Europe. The modelling problem is not trivial as the data set is highly imbalanced with the number of good rails being much higher than the rejected rails.

The paper is organised as follows. Section 2 provides an overview of the modelling data, obtained from the rail manufacturing process, and input variable selection. Section 3 outlines the theory related to Support Vector Machines (SVMs), Fuzzy Support Vector Machines (FSVMs) and Fuzzy C-Means (FCM) Clustering, and how these techniques were implemented on the data. SVMs are not affected by local minima as they are mathematically based on the solution of a convex optimisation problem. Also, in most cases, SVM generalisation performance has been shown to be better than that of other classification methods (Burges, 1998). However, quadratic optimisation scales poorly with the number of data samples and therefore FCM was proposed for reducing training time and model complexity. The modelling results and their analysis are presented in Section 4. Concluding remarks and suggestions for future work are given in Section 5.

2 MODELLING DATA

2.1 Rail Manufacturing Data

At Tata Steel Europe, a precisely controlled rail production line, whose sub-processes are indicated in Fig. 1, produces high quality rails.

During steelmaking, the desired steel chemical composition is achieved, while maintaining the steel integrity and avoiding imperfections. Consistent steel blooms are produced though continuous casting. The blooms are rolled into rail sections and Non-Destructive Testing (NDT) systems ensure strict dimensional accuracy and detect any surface and metallurgical defects so that the delivered rail sections meet the high standards required for rail applications.

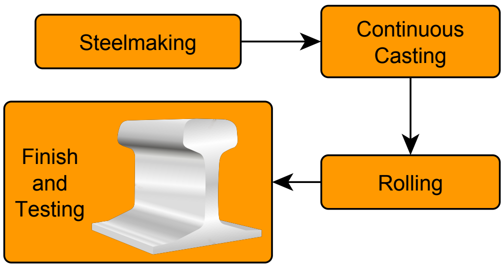

Fig. 1. Railway Production Route

Every stage of the rail manufacturing process is closely monitored and controlled to ensure the highest rail quality. Instrumentation systems and regular sampling provide data both for precise online process control and also for offline production management, planning and decision making.

The data set from the Tata Steel Europe rail production route covered a two year production period. In a previous study (Yang et al., 2011), a process expert from Tata Steel provided expert knowledge to pre-process the data which had around 200 variables. This was reduced to 70 useful process variables which could be used for modelling.

2.2 *Input Selection*

Although developments in processing power motivated data-driven modelling, data dimensionality still remains an important issue. Reducing the dimensionality improves the performance of the predictor by eliminating the curse of dimensionality and reduces the training time, resulting in faster predictors (Guyon & Elisseeff, 2003). Dimensionality reduction aims at finding a lower dimensional space which retains the input-output relationship. It can be divided into feature extraction and feature selection. On the one hand feature extraction transforms the original space, either linearly or nonlinearly, and the most representative features are used for modelling. On the other hand, for feature (or variable) selection, a number of variables are selected according to a ranking which determines their relevance with respect to the output.

Since the developed model needs to be inverted to obtain the best input values depending on specific design objectives, the appropriate dimensionality reduction technique in this case is variable selection. This ensures that the model inputs relate directly to the original variables.

A data dimension of 70 was still too high for classification as this implies that the number of training samples must be high as well leading to prohibitively long training times. Therefore, variable selection was carried out in the work by Yang et al. (2011) where the most relevant input variables were selected to reduce the dimensionality of the modelling data. Linear correlation between input variables and the output is low, implying a highly nonlinear input-output relationship. On the other hand, some high correlation coefficients between input variables imply that variables may contain redundant information. Therefore input variable ranking was required to make sure that the most relevant inputs are selected. An iterative forward input selection procedure was devised, using a three-layer multilayer perceptron neural network model as a performance evaluator. For the work being presented, the first 39 input variables were used, as obtained from the original 70 variables using this input selection procedure.

3 THEORY

3.1 *Support Vector Machines (SVMs)*

SVM is a supervised machine learning technique initially proposed by Vapnik (1979) in his pioneering work and further developed by Boser, Guyon, and Vapnik (1992). Since then, SVM application to pattern classification has increased mainly due to its attractive properties and better performance than other classifiers (Burges, 1998). This subsection reviews the theory for SVMs (Burges, 1998; Cristianini & Shawe-Taylor, 2000; Haykin, 2009; Fletcher, 2009).

Given a training data set with points in two linearly separable classes, the SVM finds the optimal hyperplane which maximises the margin of separation between the two classes.

Let the linearly separable training data be $\{\boldsymbol{x}_i, y_i\}$, $y_i \in \{-1, 1\}$, where y_i is the output class of input pattern $\boldsymbol{x}_i$. The decision surface hyperplane that separates the classes can be written as:

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0 \qquad (1)$$

where $\boldsymbol{w}$ is an adjustable weight vector representing the normal to the hyperplane, $\boldsymbol{x}$ represents the input dimensional space, b is a bias.

Suppose that the data points satisfy the following constraints:

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \geq +1 \quad \text{for} \quad y_i = +1 \qquad (2)$$
$$\boldsymbol{w}^T\boldsymbol{x}_i + b \leq -1 \quad \text{for} \quad y_i = -1 \qquad (3)$$

Support vectors are the closest points to the separating hyperplane and located on the hyperplanes defined by (2) and (3) with the equality sign. The margin, the distance between the support vectors of the two classes, can be defined as the difference between the perpendicular distances of the two hyperplanes to the origin:

$$\text{margin} = \frac{|(1-b)-(-1-b)|}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|} \qquad (4)$$

It is clear that maximising the margin is equivalent to minimising the Euclidean norm of the weight vector $\boldsymbol{w}$. The problem can be formulated as follows to benefit from the advantages of convex optimisation:

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 \geq 0 \quad \forall i \qquad (5)$$

To deal with nonseparable data, it is necessary to include slack variables, $\xi_i \geq 0$, in the constraints of (2) and (3) (Cortes & Vapnik, 1995). As ξ_i is proportional to the misclassification distance from the margin boundary, a positive penalty term, C, is included in the objective function to discourage misclassifications. The objective function can now be defined as:

$$\begin{aligned} &min\left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i \xi_i\right) \\ \text{s.t.} \quad &y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i \geq 0 \quad \forall i \end{aligned} \qquad (6)$$

To solve the problem using Lagrange multipliers, the primal form of the Lagrangian function is constructed. The primal Lagrangian is transformed into the corresponding dual objective function which is maximised with respect to the Lagrange multipliers.

The function between inputs and output in a classification problem may not be linear. The input space is mapped to a high dimensional feature space in search of a linear relationship with the output, permitting the creation of the optimal hyperplane in the high dimensional feature space. Applying the kernel trick, a nonlinear function performs the inner product $k(\boldsymbol{x}, \boldsymbol{x}_i) = \boldsymbol{\varphi}^T(\boldsymbol{x}_i)\boldsymbol{\varphi}(\boldsymbol{x})$, eliminating the need to explicitly make computations in the high dimensional space. A commonly used nonlinear kernel, and the one which will be used in this study, is the Gaussian Radial Basis Function (GRBF) kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}_i) = exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{x}_i\|^2\right) \tag{7}$$

where σ represents the spread of the function and needs to be optimised.

The dual form of the Lagrangian to be maximised can be written as:

$$\begin{aligned} L_d(\alpha) &= \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{s.t.} \quad & \textstyle\sum_i \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C \quad \forall i \end{aligned} \tag{8}$$

where α_i are the Lagrange multipliers.

Determining the optimum bias, b^*, through the Karush-Kuhn-Tucker complementary conditions, allows test points to be classified using the decision surface as follows:

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_i \alpha_i^* y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b^*\right) \tag{9}$$

where α_i^* are the optimal Lagrange multipliers.

3.2 *Fuzzy Support Vector Machines (FSVMs)*

Although SVMs are very attractive and widely applied to classification problems, the theory still suffers from some limitations. In many real world scenarios data contains noisy outlying samples. For this reason data points vary in their importance and this should be taken into consideration when training the SVM.

This can be addressed by introducing a fuzzy membership, $0 < s_i \leq 1$, corresponding to every training point $\boldsymbol{x}_i$. The fuzzy membership is regarded as a measure of belonging of a particular point towards its class.

The fuzzy membership can be included in the SVM objective function of (6) as follows:

$$\begin{aligned} & min\left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i s_i \xi_i\right) \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i \geq 0 \quad \forall i \end{aligned} \tag{10}$$

The fuzzy membership, s_i, weighs the penalty due to the misclassification error, ξ_i, of a particular training point. A high membership value assigns more weighting to the error while a low membership means that a data point is not important, thus lowering the misclassification penalty.

The derived dual form of the Lagrangian, $L_d(\alpha)$, to be maximised is the same as in (8) subject to the following constraints:

$$\textstyle\sum_i \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq s_i C \quad \forall i \tag{11}$$

As proposed by Lin and Wang (2002), the fuzzy membership can be a function of the distance between a point and its class centre. Considering that to be applied in (10), a point near the centre should have a high fuzzy membership while an outlier should have a low membership value:

$$\begin{aligned} s_i &= 1 - |\boldsymbol{x}_+ - \boldsymbol{x}_i|/(\boldsymbol{r}_+ + \delta) \ \text{ if } \ y_i = +1 \\ s_i &= 1 - |\boldsymbol{x}_- - \boldsymbol{x}_i|/(\boldsymbol{r}_- + \delta) \ \text{ if } \ y_i = -1 \end{aligned} \tag{12}$$

where $\boldsymbol{x}_+$ is the mean and $\boldsymbol{r}_+$ is the radius of class $y = +1$, $\boldsymbol{x}_-$ is the mean and $\boldsymbol{r}_-$ is the radius of class $y = -1$, $\delta > 0$ to avoid $s_i = 0$.

3.3 *The Confusion Matrix*

In the field of binary classification, a classifier's predictive performance is usually measured by the accuracy metric showing the number of correct classification predictions from both classes. However, when dealing with imbalanced data, accuracy may not be trustworthy as a performance measure. This can be appreciated using a confusion matrix shown in Table 1. The rail quality data being analysed is highly imbalanced with the number of samples representing rejected rails being the minority. It can be said that the rejected rails are the most important to be predicted correctly as it is crucial to know if a combination of input parameters will produce a rail with internal defects. In this scenario, the terms in Table 1 can be explained as follows:

- TP: Rejected rails correctly predicted as rejected rails
- TN: Good rails correctly predicted as good rails
- FP: Good rails incorrectly predicted as rejected rails
- FN: Rejected rails incorrectly predicted as good rails

		Actual Rail Quality	
		Rejected	**Good**
Predicted Rail Quality	**Rejected**	True Positive (TP)	False Positive (FP)
	Good	False Negative (FN)	True Negative (TN)

Table 1. Confusion Matrix

Using these terms, accuracy can be presented as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

It may be the case that although the number of correctly classified rejected rails (TP) is low, the accuracy of the classifier is still high because the number of correctly classified good rails (TN), which represents the majority class, is high. This is easily true with imbalanced data as the model tends to overfit the majority class in the training data, incorrectly classifying the minority class samples of the testing data.

Two useful performance measures when presented with imbalanced data for binary classification are:

$$Sensitivity = \frac{TP}{TP + FN} = \frac{TP}{Number\ of\ Positives} \qquad (14)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{TN}{Number\ of\ Negatives} \qquad (15)$$

Classifier performance is generally a compromise between these two factors in the sense that increasing sensitivity usually decreases specificity and the other way round. In this study sensitivity was given more importance since as already explained it is generally the harder to achieve.

3.4 Support Vector Machines (SVMs) for Class Imbalance

Preliminary results, which will not be presented here due to space restrictions, showed that SVMs are sensitive to class imbalance. Batuwita and Palade (2012) review methods in the literature used to alleviate this problem. These are divided into two sections, namely external imbalance learning methods such as data resampling and internal imbalance learning methods which make algorithmic modifications to the SVM learning algorithm so that it is less sensitive to data imbalance.

An internal imbalance learning method was applied by implementing a bias towards the minority class to form a Biased FSVM which is referred to as Different Error Cost (DEC) by Batuwita and Palade (2012) and was originally proposed by Veropoulos et al. (1999). This is obtained by assigning a different cost for the two classes in the misclassification penalty factor. Thus a cost C^+ is applied to the positive (minority) class while a cost C^- is applied to the negative (majority) class.

When combined with the fuzzy membership defined earlier, the following objective function is obtained:

$$min\left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C^+C\sum_{\{i|y_i=+1\}} s_i\xi_i + C^-C\sum_{\{i|y_i=-1\}} s_i\xi_i\right) \qquad (16)$$

$$\text{s.t.}\quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i \geq 0 \quad \forall i$$

The dual Lagrangian form, $L_d(\alpha)$, of this function is the same as in (8) and is maximised subject to the constraints:

$$\textstyle\sum_i \alpha_i y_i = 0, 0 \leq \alpha_i^+ \leq s_i C^+ C, 0 \leq \alpha_i^- \leq s_i C^- C \quad \forall i \qquad (17)$$

where α_i^+ and α_i^- represent the Lagrange multipliers of positive and negative samples respectively.

The ratio C^-/C^+ was set equal to the minority to majority class ratio (Akbani et al., 2004) such that the penalty for misclassifying minority examples is higher. A grid search was performed to optimise the parameter, C, and GRBF spread, σ. Considering a model with the best sensitivity (without degrading specificity), an accuracy of 68.1%, sensitivity of 53.0% and specificity of 69.2% were obtained. The ratio of the number of support vectors to the number of training points was 0.984.

Noting these results, it was subsequently decided to apply an external imbalance learning method by balancing the training data. The data set obtained by Zughrat et al. (2013) was used, where under-sampling of the majority class was performed to make the number of good rails equal to the rejected rail training examples.

3.5 Fuzzy C-Means (FCM) Clustering

As will be discussed in Section 4, the number of support vectors when applying SVMs and FSVMs on the balanced data set was still very high leading to long training times and making parameter optimisation impractical and inefficient. Therefore FCM Clustering was proposed as a way of reducing the number of support vectors, reducing training times and model complexity, and improving generalisation (Xiong et al., 2005; Cervantes et al., 2006).

The FCM Clustering algorithm is an optimisation problem whereby the coordinates of the cluster centres need to be identified. The cost function to be minimised (Bezdek, 1981) is:

$$J(X, U, V) = \sum_{i=1}^{c}\sum_{k=1}^{N} {\mu_{ik}}^m \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 \qquad (18)$$

where $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_c]$ is the vector of cluster centres, $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N]$ represents the data samples, $U = [\mu_{ik}]$ is the fuzzy partition matrix of X, m is the weighting exponent, $D_{ik}^2 = \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2$ is the squared distance norm.

The minimisation of (18) is possible if and only if:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{D_{ik}}{D_{jk}}\right)^{\frac{2}{m-1}}} \qquad 1 \leq i \leq c \quad 1 \leq k \leq N \qquad (19)$$

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N}(\mu_{ik})^m \boldsymbol{x}_k}{\sum_{k=1}^{N}(\mu_{ik})^m} \qquad 1 \leq i \leq c \qquad (20)$$

The membership degree, μ_{ik}, is inversely proportional to the squared distance from the data points to the current cluster centres. Equation (20) gives $\boldsymbol{v}_i$ as the weighted mean of the data points, where the weights are the membership degrees. The FCM algorithm iterates through (19) and (20) to optimise the fuzzy partition matrix and cluster centres.

4 RESULTS AND DISCUSSION

4.1 Support Vector Machine (SVM) Modelling

Fig. 2 shows the grid search results for the parameters C and σ for the SVM when applied on the balanced data set using a GRBF kernel. A model was chosen from the region with best sensitivity with C equal to 11776 and σ equal to 46.45. The performance on the training data was 69.22% sensitivity, 79.32% specificity and 74.42% accuracy, which indicates that the model was not overfitted. The performance on the testing data is listed in Table 2. The ratio of support vectors to the number of training points, $R_{sv/tr}$, was 0.948.

4.2 Fuzzy Support Vector Machine (FSVM) Modelling

The grid search results for the FSVM are shown in Fig. 3. A model was chosen which had the same parameters as those for the SVM model. Table 2 indicates that the FSVM provided a 9.06% improvement in terms of sensitivity. However, $R_{sv/tr}$ was still very high at 0.952.

4.3 Fuzzy C-Means Clustering, Fuzzy Support Vector Machine (FCM-FSVM)

Clustering was performed on the balanced data set which had 2877 points. Random initial cluster centres and a weighting exponent, m, of 2 were used for the FCM algorithm. Using the same values for the parameters C and σ, the average performance of 10 FSVM models was considered for every clustering level from 10% to 90% (with 10% having the least number of cluster centres resulting in the minimum number of training points) when tested on a separate unbalanced data set. Table 3 shows that the number of support vectors was reduced since clustering reduces the number of training points ($R_{sv/max}$). However, clustering grouped points with similar features and this allowed the SVM algorithm to further reduce the number of support vectors in relation to the number of training points available ($R_{sv/tr}$). Fig. 4 indicates that classifier performance is a compromise between sensitivity and specificity. Fig. 5 shows that after clustering and fuzzification, the SVM algorithm was able to build the model using 50% or less of the available training points ($R_{sv/tr}$). This highly reduced the model training time which is especially important for parameter optimisation procedures. Analysing performance, one has to keep in mind the data dimensionality and the high nonlinearities in the input-output relationship.

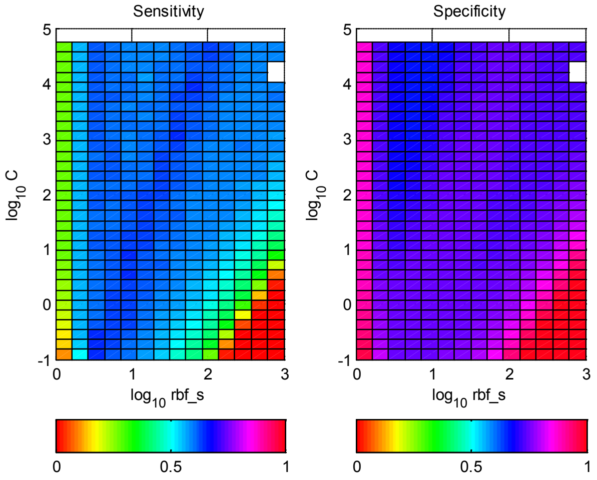

Fig. 2. Grid Search for SVM with GRBF Kernel

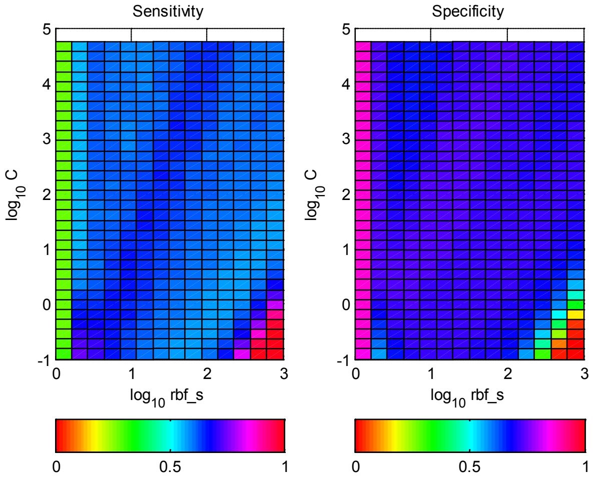

Fig. 3. Grid Search for FSVM with GRBF Kernel

	SVM	FSVM	Percentage Difference
Sensitivity [%]	61.94	67.55	+ 9.06
Specificity [%]	73.80	68.24	- 7.53
Accuracy [%]	73.06	68.20	- 6.65
$R_{sv/tr}$	0.948	0.952	+ 0.42

Table 2. Performance of SVM and FSVM Models

Clustering [% of max. # pts.]	Sens. [%]	Spec. [%]	Acc. [%]	$R_{sv/tr}$	$R_{sv/max}$
10	71.31	50.60	51.89	0.426	0.043
20	69.46	57.08	57.80	0.496	0.099
30	69.73	57.71	57.44	0.506	0.152
40	68.30	60.10	60.61	0.481	0.192
50	68.21	60.78	61.25	0.488	0.244
60	68.75	60.71	61.21	0.466	0.280
70	66.09	63.31	63.48	0.460	0.322
80	66.08	63.37	63.54	0.424	0.339
90	66.37	63.75	63.91	0.428	0.385

Table 3. FCM-FSVM with Different Clustering Levels

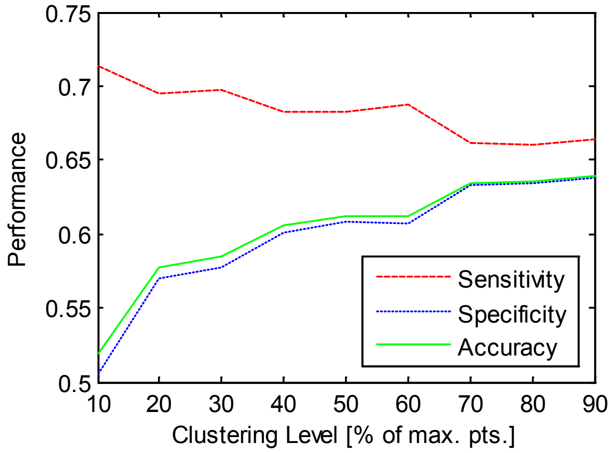

Fig. 4. Performance for Different Clustering Levels

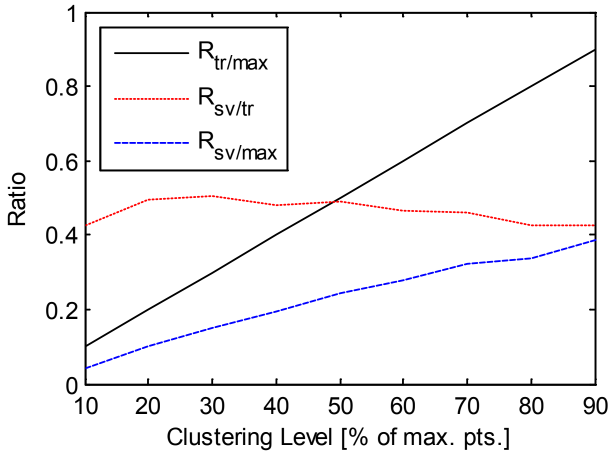

Fig. 5. Support Vector Ratios for Different Clustering Levels

5 CONCLUSIONS

In this study SVM modelling techniques were applied to classify rail data. As the original data set was highly imbalanced, internal and external imbalance learning methods were applied to improve the classifier performance. The best performance, in terms of a compromise between model performance and model training time in relation to the number of support vectors, was obtained when FCM Clustering was applied on the balanced data set before fitting a FSVM model.

Further performance improvements will help to eventually exploit the model by inverting the structure via multi-objective optimisation techniques to design specific processing routes for 'right-first-time' production of rails.

This study may be further extended by investigating the effect of the class imbalance ratio and other class imbalance learning methods on model performance. Different fuzzy membership functions and kernel functions can also be applied although these may require further parameter optimisation. Jiang et al. (2006) and Tang (2011) suggest that instead of using a fuzzy relationship in the input space, the fuzzy membership is derived as a function of the high dimensional feature space, thus taking into consideration the data nonlinearities. It would also be interesting to compare the performance of other classification methods on the same data set.

ACKNOWLEDGEMENT

The authors would like to thank Tata Steel Europe for providing us with rail production data.

REFERENCES

Akbani, R., Kwek, S. & Japkowicz, N., 2004. Applying support vector machines to imbalanced datasets. In *Proceedings of the 15th European Conference on Machine Learning.*, 2004.

Batuwita, R. & Palade, V., 2012. Class imbalance learning methods for support vector machines. In H. He & Y. Ma, eds. *Imbalanced learning: foundations, algorithms, and applications*. Hoboken, NJ: John Wiley & Sons, Inc.

Bezdek, J.C., 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press.

Boser, B.E., Guyon, I.M. & Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers. In Haussler, D., ed. *Proceedings of the 5th ACM Workshop on Computational Learning Theory*. Pittsburgh, 1992.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), pp.121-67.

Cervantes, J., Li, X. & Yu, W., 2006. Support Vector Machine Classification Based on Fuzzy Clustering for Large Data Sets. In Gelbukh, A. & Reyes-Garcia, C.A., eds. *5th Mexican International Conference on Artificial Intelligence*. Apizaco, Mexico, 2006. Springer Berlin Heidelberg.

Cortes, C. & Vapnik, V.N., 1995. Support vector networks. *Machine Learning*, 20, pp.273-97.

Cristianini, N. & Shawe-Taylor, J., 2000. *An Introduction to Support Vector Machines*. Cambridge: Cambridge University Press.

Fletcher, T., 2009. *Support vector machines explained*. [Online] Available at: http://www.tristanfletcher.co.uk/SVM%20Explained.pdf [Accessed August 2013].

Guyon, I. & Elisseeff, A., 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, pp.1157-82.

Haykin, S., 2009. *Neural Networks and Learning Machines*. 3rd ed. New Jersey: Prentice Hall.

Jiang, X.F., Yi, Z. & Lv, J.C., 2006. Fuzzy SVM with a new fuzzy membership function. *Neural Computing and Applications*, 15, pp.268-76.

Lin, C.F. & Wang, S.D., 2002. Fuzzy Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2), pp.464-71.

Tang, W.M., 2011. Fuzzy SVM with a New Fuzzy Membership Function to Solve the Two-Class Problems. *Neural Processing Letters*, 34(3), pp.209-19.

Vapnik, V.N., 1979. *Journal of Machine Learning Research*. Moscow: Nauka. (English translation: 1982, New York: Springer Verlag).

Veropoulos, K., Campbell, C. & Cristianini, N., 1999. Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on Artificial Intelligence.*, 1999.

Weiss, G.M., 2004. Mining with rarity: a unifying framework. *SIGKDD Explorations Newsletter*, 6(1), pp.7-19.

Xiong, S.W., Liu, H.B. & Niu, X.X., 2005. Fuzzy support vector machines based on FCM clustering. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*. Guangzhou, China, 2005.

Yang, Y.Y. et al., 2011. Adaptive neural-fuzzy inference system for classification of rail quality data with bootstrapping-based over-sampling. In *IEEE International Conference on Fuzzy Systems*. Taipei, 2011.

Zughrat, A., Mahfouf, M., Yang, Y.Y. & Thornton, S., 2013. Support Vector Machines for Class Imbalance Rail Data Classification with Bootstrapping-based Over-Sampling and Under-Sampling. In paper presented to: *The 19th World Congress of the International Federation of Automatic Control*. Cape Town, 2014.