Pre-publication draft of paper accepted for publication in the Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA'02) pp 143-159
Publisher: ACTA Press

Predicting the rolling force in hot steel rolling mill using an ensemble model

Y. Frayman[1], B. F. Rolfe[1], P. D. Hodgson[2], G. I. Webb[1]
[1]School of Computing and Mathematics
Deakin University
Geelong, Vic. 3217, Australia
[2]School of Engineering and Technology
Deakin University
Geelong, Vic. 3217, Australia
email: {yfraym,brolfe,phodgson,webb}@deakin.edu.au

Abstract

Accurate prediction of the roll separating force is critical to assuring the quality of the final product in steel manufacturing. This paper presents an ensemble model that addresses these concerns. A stacked generalisation approach to ensemble modeling is used with two sets of the ensemble model members, the first set being learnt from the current input–output data of the hot rolling finishing mill, while another uses the available information on the previous coil in addition to the current information. Both sets of ensemble members include linear regression, multilayer perceptron, and k–nearest neighbour algorithms. A competitive selection model (multilayer perceptron) is then used to select the output from one of the ensemble members to be the final output of the ensemble model. The ensemble model created by such a stacked generalization is able to achieve extremely high accuracy in predicting the roll separation force with the average relative accuracy being within 1% of the the actual measured roll force.

Keywords: *hot steel rolling, ensemble modeling, regression, stacked generalization, competitive combination*

1 Introduction

Maintaining product consistency and quality in the manufacturing process has become a widespread concern as a result of increasing competition in the world markets.

Increasing demands on the quality of rolling mill products have led to great efforts to improve the control and automation systems of the rolling process. Hot steel rolling is one of the most important steel manufacturing processes. Hot rolling is the first metal shaping process after the slab has been cast, in flat products such as plate, strip and sheet. The final shaping stage of hot rolling steel strip is normally performed on a tandem mill known as a finishing mill consisting typically of two to six stands. Here the final thickness, flatness and profile of the workpiece are determined. It is important to have a sound understanding of the behaviour of the roll gaps in the finishing mill for design, scheduling and control purposes. In particular, accurate predictions of the roll separating force are necessary to meet the current and the future quality standards of final product dimensions and flatness.

This paper focuses on the development of an ensemble model to address these concerns specifically within the steel industry. The motivation for the application of inductive learning–based methodologies lies in the fact that they do not require the expert development of phenomenological models. This technology could provide a powerful tool for accurate prediction of the roll separating force, thereby ensuring that the products manufactured conform to target specifications and thus contribute to enhanced business benefits.

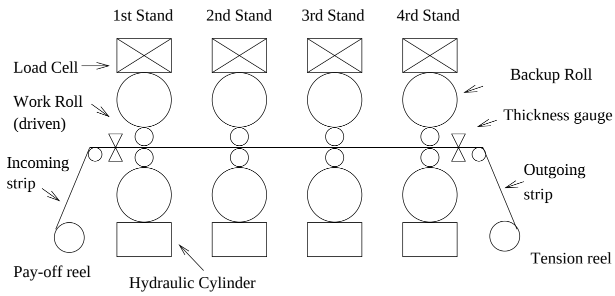

Figure 1. Finishing hot rolling mill

As seen in Fig. 1, a hot strip tandem finishing mill takes a bar of metal (at roughly 1100 degrees C for mild steel) and puts it through a series of rolling stands typically reducing its average thickness by a factor of 10. The outgoing strip should have a uniform thickness with typical dimensions being 600–1800 mm in width, and 1.5–6 mm in thickness. As the width does not change much there is a corresponding increase of length by an order of 10.

The mill settings are determined from the physical models based on expert metallurgical and mechanical knowledge. The set–up of these models is crucial since it determines, to a large extent, the thickness of the final product. In practice, it is sometimes observed that the roll–gap settings produced by set–up models are not as accurate as those required by increasing consumer product–quality demands. Although small errors can usually be compensated for by the mill controllers, larger errors lead to quality degradation and potentially out–of–specification product.

The mill set–up errors arise since the set–up model only uses factors whose exact physical relationships are understood. Unfortunately, the rolling process involves many additional factors that affect the elastic/plastic material deformation in the roll gap, particularly due to the stochastic nature of the rolling process. In this sense, the physical model is far from perfect.

The main objective of this work was to assess the the empirical model approach to the prediction of the roll separating force. An ensemble model consisting of multilayer perceptrons (neural networks), linear regressions, and k–nearest neighbours was used, as it was found that the none of the single models achieved the desired predictive accuracy.

2 Ensemble modeling

The motivation for combining models in ensembles is to improve their generalisation ability. This idea has a long history, for example, in the area of forecasting, where better results can be achieved by combining forecasts (model mixing) than by choosing the best single forecast [1].

The effectiveness of an ensemble can be measured by the extent to which the members are error–independent [11], that is show different patterns of generalisation). The ideal would be a set of models where each of the models generalised well, and when they did make errors on the unseen data, these errors were not shared with any other models.

The important point is that each of the models in an ensemble has to generalise well in order for the ensemble to be effective. Some authors, for example [4] suggest to over–fit the models in an ensemble to make them disagree. While such an approach will ensure that the performance of the ensemble is better than of any of the ensemble members, it is quite likely that it could be worse than that of a single model which is trained to generalise well. The aim of the ensemble learning is not just to improve the performance of the ensemble over that of the ensemble members, but to achieve the best performance possible. This point, while quite obvious, is commonly overlooked.

2.1 Selection of ensemble members

The aim then is to find models that generalise differently. There are several possible ways to achieve that which include the following:

Sampling data: Each model in the ensemble is trained on a different subsample of the training data. Resampling methods are particularly useful where there is a shortage of data.

Boosting and adaptive re–sampling: In boosting, a series of weak learners could be converted to a strong learner as a result of training the members of an ensemble on patterns that have been filtered by previously trained members. The AdaBoost algorithm [6] uses training sets that are adaptively resampled, such that the weights in the resampling are increased for those cases that are most often misclassified. While boosting is generally designed for classification tasks, it has also been recently applied for regression problems {citefriedman01. The gradient boosting machine {citefriedman01 is closely related to artificial neural networks and optimization in general, which opens a new link between optimization and computational learning [10].

Varying the learning method employed: The learning method used to train the models can be varied while holding the data constant. For example, an ensemble might be constructed from models generated by a combination of learning techniques such as various statistical methods, linear regression, neural networks, k–nearest neighbours, decision trees, and Markov chains. While this approach is rarely used, in our opinion, it is a very promising approach to ensemble learning, as the use of different learning methods for ensemble members are more likely to result in different patterns of the generalization, than by varying the data. Further, there is a possibility to combine varying both the learning method and the data, for example, with adaptive re–sampling of the data. This paper pursues the approach of varying the learning method.

2.2 Combining ensemble members

The next step in ensemble learning is to find an effective way of combining model outputs. Methods of combining the models in ensemble learning include the following:

Averaging and weighted averaging: Linear combination of the outputs of the ensemble members are one of the most popular aggregation methods. A single output can be created from a set of model outputs via simple averaging, or by means of a weighted average that takes account of the relative accuracy of the models to be combined.

Stacked generalisation: Under stacked generalisation [14] a nonlinear model learns how to combine the ensemble members with weights that vary over the feature space. The outputs from a set of lower level generalisers are used as the input to a higher level generaliser that is trained to produce the appropriate output. The same idea has been adapted to regression tasks, where it is termed stacked regression [3].

Appropriate determination of the stacking weights is essential for good modeling performance. A simple approach is to take equal weights (mean) for the individual models. Alternatively, the weights can be calculated using multiple linear regression. However, stacking weights using this technique, has been shown not to give good results, owing to the highly correlated nature of the individual model predictors.

Another possibility is to consider a competitive combination of ensemble members [12] for instance by using a non–linear model (for example a neural network) that is trained to select the appropriate output of the ensemble member as a final output of the ensemble model based on the performance of ensemble members on a particular data tuple. The aim here is basically to create a global model (ensemble model) where each of the ensemble members is acting as a local predictor in the area of its best performance. This is the approach used in current paper. In such a rule–based switching [2], [13] the control is switched between the ensemble members depending on the output of one of the members.

The goal of combining models is to create effective ensembles that perform better than the best single model. An ensemble combined by means of averaging will not necessarily result in better performance than choosing the best model as the effectiveness of an ensemble depends on the extent to which its members make different errors, or are error–independent [11].

It has been argued [8] that the presence of harmful collinearity or correlation between the errors made by the component models in an ensemble will reduce the effectiveness of the ensemble itself. For example, although varying the data might be an effective way of producing models which generalise differently, this will not necessarily result in low error correlations due to the concept of training set representativeness [5]. A representative training set is one that leads to good generalisation where a function being inferred is similar to that which generated the test set. However, two representative training sets could lead to very similar functions being inferred, so their pattern of errors on the unseen data will be very similar.

On the other hand, if a set of models were trained using unrepresentative training sets, the resulting generalisation performance would be poor. The models might each infer quite different functions, and show different patterns of generalisation to the test set, but as the number of errors increases so does the probability that the errors that they make on the test set will overlap.

It is possible to select ensemble models by applying selection procedures to a set of models which have been created through the use of methods designed to promote diversity, and by continuing the process of generation and selection, for example using genetic algorithms to actively search for ensemble members which generalise well, but which disagree as much as possible [9].

Table 1. Process variables nomenclature

Variable Name	Unit
setup exit thickness	mm
aim width	mm
setup reduction (stands 1-4)	%
setup force (stands 1-4)	t
setup roll speed (stands 1-4)	m/min
setup looper tension (stands 1-4)	kN
setup bending force at (stands 1-4) (per side)	MN
setup strip temperature (stands 1-4)	deg C
setup predicted forward slip (stands 1-4)	
chemical composition of the slab grades	
work roll diameter (stands 1-4)	mm
measured roll force (stands 1-4)	t

3 Experimental Results

Measurements of 48 process variables (Table 1) were recorded for each of 4961 coils rolled during routine production at the BHP Billiton Coated Steel Australia, Western Port Work Hot Strip Mill in Hastings, VIC, Australia.

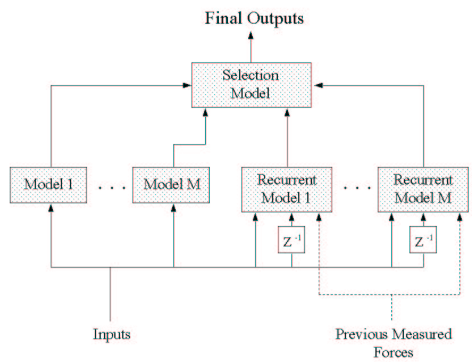

Figure 2.

All of the rolling parameters in the Table 1 were used as inputs to the ensemble members, except the measured roll force at stands 1–4, which is the prediction (output) variable. The data was split into two parts, first 3282 coils were used to develop the ensemble model, and the other 1679 coils formed a test data on which the proposed technique was evaluated.

As ensemble members we used a standard linear regression, a feedforward multilayer perceptron with backpropagation learning algorithm, and a k–nearest neighbor.

The linear regression used is a standard multivariate linear regression algorithm that generates a linear weighted sum of the inputs plus a constant bias for each output. The coefficients (weights) of a linear regression minimize the least–mean–square error between the desired outputs and linear regression outputs.

The feedforward multilayer perceptron (MLP) used is a neural network algorithm that generates input–output mappings based on computations of interconnected nodes arranged in layers. The output of each node is a nonlinear function of the weighted sum of inputs from the nodes in the preceding layer. The MLP with one hidden layer was used. The learning algorithm used was backpropagation with momentum. The activation function used was a hyperbolic tangent. Updating the weights was done using the pattern (on–line) adaptation. The output node were using a linear finction. The input and output data was normalized using a zero mean–unit variance normalization. As a stopping condition a minimal root-mean-squared-error (RMSE) between the predicted output values from the model and the actual output values was used.

The optimal parameters (learning rate and momentum) and the topology (number of nodes in a hidden layer) of the MLP were selected based on a combination of a genetic algorithm search through different sets of network structures and parameters to limit the search space and the an exhaustive search to fine tune the network structure and the parameters found by the genetic search. The parameters which resulted in smallest RMSE between the predicted output values and the actual output values on a test data, were used.

K–nearest neighbor (KNN) generates an output based on extracting the k nearest patterns to the input in the training set. For estimation, the output value is the average of the outputs of the k nearest patterns. The Euclidean distance metric is used to compute nearness.

The first set of the ensemble members were created using the current rolling mill inputs, while another set of the ensemble members (being in both cases the linear regression, the MLP, and the KNN) were created using in addition to the current inputs, the inputs plus the output (measured force) at the previous time step (previous coil). The second set of ensemble members therefore becomes a set of recurrent (or time–delay) models.

As a selection model we used another MLP with the same setting as the ensemble members, but with optimal parameters and topology selected to achieve the best performance for the selection of the appropriate ensemble members to produce the final prediction of the ensemble model.

The competitive selection model was used, as it was discussed previously. Its aim here is to select the best predicted output of the six ensemble members to act as a final output of the model. In such a way each of the ensemble members is effectively acting a local predictor in the area of its best performance.

It is interesting that while the performance of the KNN models are much worse than that of the other two algorithms employed (which are quite similar) it contributes to the final performance of the ensemble model to a large degree, as it is the only method of the ones used that can give an exact prediction in cases where it is accurate. In general, the addition of the KNN models to the ensemble improves the overall ensemble performance to a greater extent than the combination of just the linear regression and the MLP models. This suggests that the linear regression and nonlinear MLP models, while different in form, still exhibit quite similar generalisation patterns. In contrast, the KNN models, while poor predictors by themselves, work well in combination with other models. There is still much to be done, in our opinion, in order to produce reliable guidance as how to select and combine the ensemble members effectively. We hope the the general discussion above on the ensemble learning can be helpful in this direction.

The performance results are shown in Figs. 3–10 and Table 2. The average relative error in Table 2 was calculated as the absolute difference between the predicted and measured value of the force divided by the measured force.

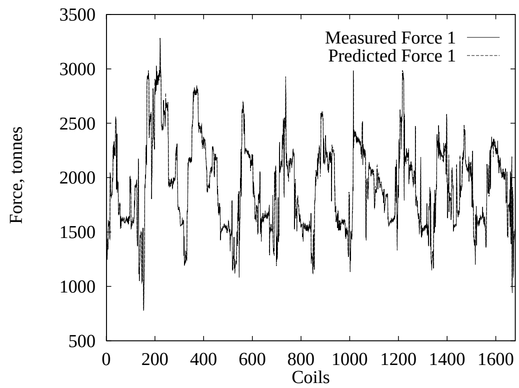

Figure 3. Predicted Force 1 vs Measured Force 1

While the general performance of the ensemble

Table 2. Performance Results

Learning Method	Average Relative Error				Standard Deviation			
	Force 1	Force 2	Force 3	Force 4	Force 1	Force 2	Force 3	Force 4
Best single model	0.015233	0.018881	0.021528	0.025546	0.017799	0.020697	0.022469	0.026775
Ensemble	0.008267	0.010127	0.011016	0.012503	0.010772	0.012598	0.013657	0.015290

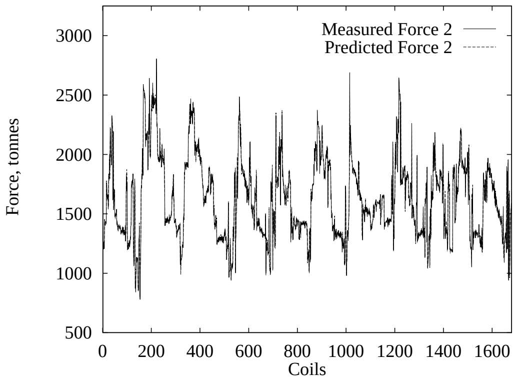

Figure 4. Predicted Force 2 vs Measured Force 2

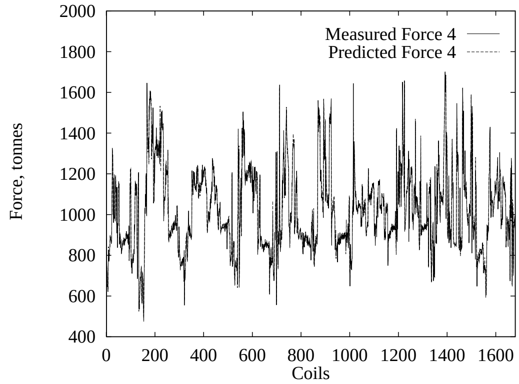

Figure 6. Predicted Force 4 vs Measured Force 4

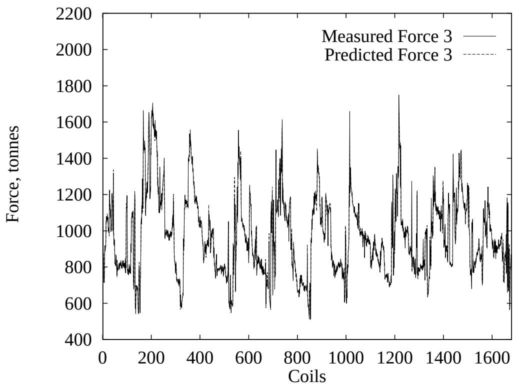

Figure 5. Predicted Force 3 vs Measured Force 3

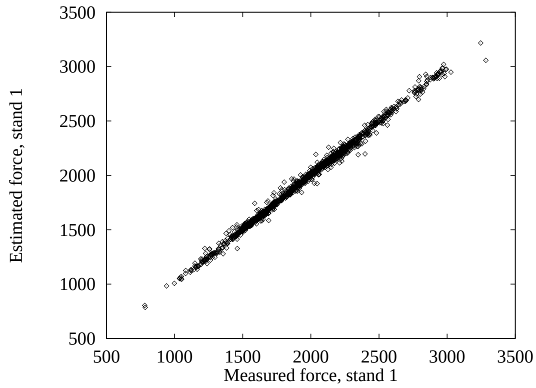

Figure 7. Scatter Plot of Predicted Force 1 vs Measured Force 1

model is extremely good with predicted rolling force being within 1% of the actual measured force, the ensemble model sometimes gives errors greater than two standard deviations from the mean. In our case there were significant errors, 34 out of 1679 on the test data set. While this is a very good result, current efforts are directed to investigating and correcting these errors as are they are obviously going to affect the quality of the final product. One of the possible directions currently under investigation, is adaptive re–sampling in combination with stacked generalisation.

4 Conclusion

This paper focuses on developing an ensemble model to address the need for accurate predictions of the roll separating force in order to meet the quality standards of the final product in steel manufacturing. A stacked generalisation approach to ensemble modeling is used with two sets of ensemble model members, the first set being learnt from the current input–output data of the hot rolling finishing mill, while another augments the information about the current coil with information from the previous coil. Both sets of ensemble members include linear regression, multilayer perceptron, and k-nearest neighbour algorithms. A competitive selection model (multilayer perceptron) is then used to select the output of the ensemble member that has the smallest error to act as the final output of the ensemble model. The ensemble model created by such stacked generalization is able to achieve extremely high accu-

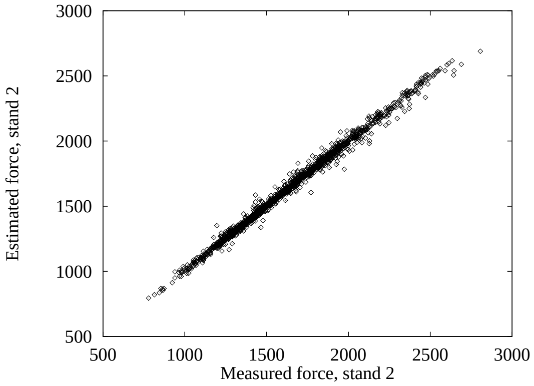

Figure 8. Scatter Plot of Predicted Force 2 vs Measured Force 2

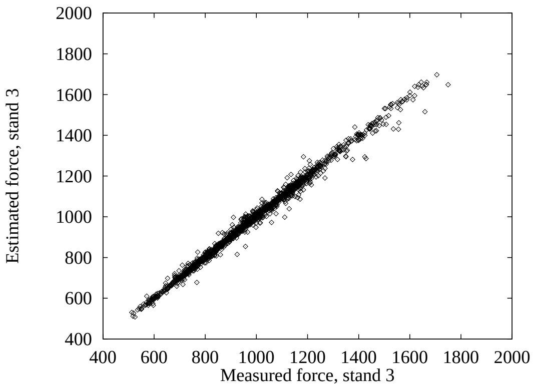

Figure 9. Scatter Plot of Predicted Force 2 vs Measured Force 2

racy in predicting the roll separation force with the average relative accuracy being within 1% of the actual measured roll force.

5 Acknowledgements

The authors wish to thank BHP Billiton Coated Steel Australia, especially Ron E. Gloss for providing the rolling mill data and the permission to publish this work.

References

[1] Bates, J. M. and C. W. J. Granger. "The combination of forecasts". *Operations Research Quaterly*, 20:451–468, 1969.

[2] Baxt, W. G. "Improving the accuracy of an artificial neural network using multiple differently trained networks". *Neural Computation*, 4:772–780, 1992.

[3] Breiman, L. "Stacked regression". Technical Report 367, Statistics Department, University of California, Berkeley, 1993.

[4] Breiman, L. "Bagging predictors". *Machine Learning*, 26(2):123–140, 1996.

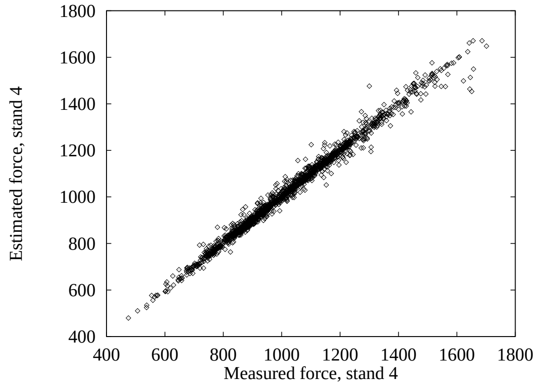

Figure 10. Scatter Plot of Predicted Force 4 vs Measured Force 4

[5] Denker, J., D.Schwartz, B.Wittner, S.Solla, R.Howard, L.Jackel, and J.Hopfield. "Large automatic learning, rule extraction and generalisation". *Complex Systems*, 1:877–922, 1987.

[6] Freund, Y. and R.Schapire. "Experiments with a new boosting algorithm". *Proceedings of the Thirteenth International Conference on Machine Learning*, 149–156. 1996.

[7] Friedman, J. "Greedy function approximation: a gradient boosting machine". *Annals of Statistics*, 29(4). 2001.

[8] Hashem, S. "Optimal linear combinations of neural networks". *Neural Networks*, 10(4):599–614, 1997.

[9] Optiz, D.W. and J.W. Shavlik. "Actively searching for an effective neural network ensemble". *Connection Science*, 8(3–4):337–354, 1996.

[10] Ridgeway, G. "The state of boosting". *Computing Science and Statistics*, 31:172–7181, 1999.

[11] Rogova, G. "Combining the results of several neural network classifiers". *Neural Networks*, 7(5):777–781, 1994.

[12] Sharkey, A.J.C. (Ed.) *Combining artificial neural nets: ensemble and modular multi-net systems,* Springer-Verlag, 1999.

[13] Ting, K. M. "The characterisation of predictive accuracy and decision combination". *Proceedings of the 13th International Conference on Machine Learning*, pp. 498–506, 1996.

[14] Wolpert, D.H. "Stacked generalization". *Neural Networks*, 5:241–259, 1992.