

DEFECT PREDICTION IN HOT STRIP ROLLING USING ANN AND SVM

Manu HIETANIEMI ^a, Ulla ELSILÄ ^b, Perttu LAURINEN ^b and Juha RÖNING ^b

^a *manu@ee.oulu.fi*

^b *University of Oulu, Department of Electrical Engineering
Computer Engineering Laboratory, Intelligent Systems Group
PO BOX 4500, FIN-90401 Oulu, Finland*

Abstract.

One of the largest factors affecting the loss for steel manufacturing are defects in the steel strips produced. Therefore the prediction of these defects beforehand would be very important. In this study we used classifiers - feedforward neural networks and a support vector machine - to solve this problem. We also used different kinds of feature selection methods such as a preprocessing step for the classifiers. As a result, these two classifiers confirmed the same grade of classification error in this study.

Keywords. Hot steel rolling, feature selection, classification, neural networks, support vector machine

Introduction

Due to the complexity and the deficiencies in existing physical models of the hot strip rolling process, defects are quite common when steel strips are produced. The defects are typically a result of improper temperatures or dimension measures. The thickest strips that are produced in the hot strip mill of this study (height from 6.50 mm to 20 mm) were chosen in the classification, because the defects are most common in these. Defects occur in approximately one out of four steel strips and this leads to further inspection by an inspector and additional operations, leading to a considerable financial loss for the production. Thus, there is a need for finding these failures beforehand.

The amount of data and measurements is huge; the relation can potentially be non-linear and also involves a lot of uncertainty, whereupon the traditional physical estimators are almost useless. Therefore the prediction of these failures by a proper classifier, e.g. artificial neural network (ANN) or support vector machine (SVM) methods, was the main goal of this study. There are a lot of existing studies on hot strip rolling using classifiers, but they are concentrated on modelling optimized rolling process control parameters *i.e.* temperatures [1], rolling force [2]. However, we had the opposite approach in the sense that the parameters of the rolling process were predetermined and the defect prediction had to be made on these parameters.

ANNs and SVMs are useful tools for comparison because they have a lot of different characteristics. While ANNs can suffer from multiple local extrema, the solution

of a SVM is global and unique. The model selecting for ANN, *i.e.* hidden layers, the number of neurons and the functions that are utilized, are predetermined, whereas the supervised SVM training determines its parameters based on the available data, although you need to adjust the hyperparameters properly. Also, the ANN uses only empirical risk minimization whereas the SVM uses structural risk minimization [3]. The structural risk minimization principle finds the best solution in terms of empirical risk but also the simplest in terms of the model complexity. The large number of features on the dataset and their significance for the classification was also an important issue in the study. Thus the complexity of the dataset led to different kinds of approaches in the preprocessing stage.

1. Hot Strip Rolling Process

The data used was gathered from the hot strip mill of Ruukki in Raahe, Finland. The mill with its pyrometers is illustrated in Figure 1. Slab dimensions, the chemical composition of the slabs as well as target values of the rolling process are collected from the production planning computer of the process control system. Before rolling, the slabs are heated in reheating furnaces. At the time of the collection of the used datasets, there was one walking beam furnace and three pusher type furnaces. After discharging, the slab is rolled at the reversing roughing mill, typically in seven passes from the thickness of some 210 millimeters to a target thickness of *e.g.* 30 millimeters. After roughing, the rolling stock is called a transfer bar. Next, the transfer bar goes through the finishing mill. In the case of the most recent datasets, a coil box had been added on the line immediately before the finishing mill. The coil box makes rolling of bigger coils possible and facilitates more uniform temperature control of the transfer bar. After finishing, the rolling stock, which is now called the hot strip, enters the cooling area where the targeted temperature profile of the strip is achieved. Finally, the strip is coiled at the down-coiler.

During the rolling process, many measurements are made and some of the meter positions are shown in Figure 1. Temperatures are measured at several places from the furnace up to the coiler; the width at the rougher and after finishing; and the thickness, profile, flatness, and speed of the strip are measured after the finishing line. Also, time stamps and durations of the various process stages are recorded. From these measurements, the process control system calculates properties of the rolling stock: mean values, variations, segmented values, deviations from the targets, classification, which are subsequently stored in the database.

Since the actual rolling process after the heating is fast, with a duration of a mere few minutes, the process is very sensitive to various errors. The maintenance of the right and uniform temperature during the process is a particularly challenging task. At any moment during the rolling, the automation system or the rolling mill operator may store a strip-specific retention code in the database. Each product with this type of code has to be checked manually and sometimes reshaped before it can be approved for selling. In this study, these products are collectively termed retained samples.

2. Dataset

On the basis of a previous study "Defect prediction in hot strip rolling" [4], the original dataset was reduced by 83% from over 200 to 37 features with basic statistical analysis

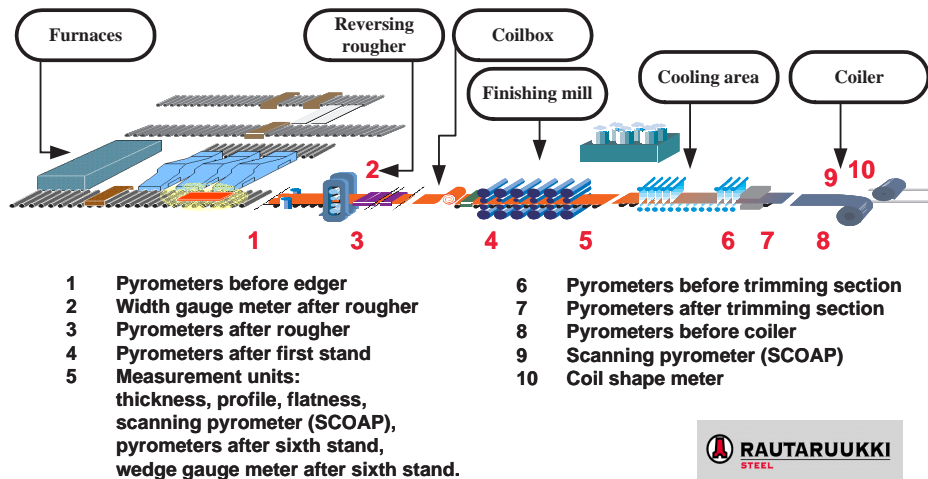


Figure 1. Illustrated hot strip rolling mill in Rautaruukki, Raaha, Finland

and linear correlation, and according to advice received from the rolling experts. These remaining 37 features included different kinds of material properties, temperature measurements, dimensional measurements and rolling properties of the steel slab. The continuous features were scaled from 0 to 1. The class features, which had discrete values, were also separated into components, e.g. if a feature had four possible values, it was separated into four different binary features with a value of 0 or 1. The dataset eventually consisted of 77 features and 6004 measurements. 25.5 % of these examples included the retention code.

3. Classification Methods

3.1. Artificial Neural Network

A feedforward multilayer perceptron (MLP) neural network model was used with a back-propagation algorithm to minimize the errors at the output [5]. The dataset (with 6004 samples) was divided as follows: half for training, one quarter for testing and the last quarter for validation.

Training was executed with (i) the Levenberg-Marquardt algorithm, which appears to be the fastest method for training moderate-sized feedforward neural networks [6,7], and (ii) the Scale conjugate gradient algorithm [5]. The selected transfer function was a log-sigmoid function, which is commonly used in back propagation networks partly because it is differentiable. It takes the input, which can have any value between plus and minus infinity, and compresses the output into the range of 0 to 1.

The decision regarding how many hidden layers should be used was limited to one or two. It has been proven that a three-layer feedforward network (i.e. input layer, one hidden layer and output layer) with $(N - 1)$ hidden neurons can give any N input-target relations precisely [6]. The same size input in a four-layer-network i.e. two hidden layers, can give the target relations with a negligibly small error using only $(N / 2) + 3$ hidden neurons. So using only one or two hidden layers was the obvious choice. The number of

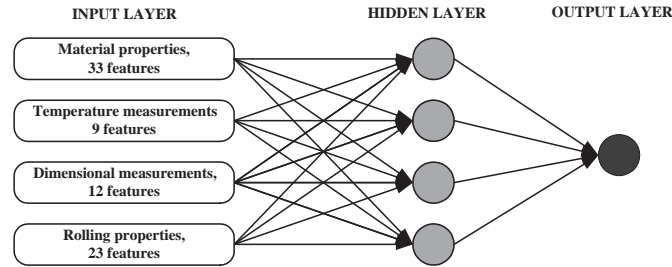


Figure 2. Illustration of the feedforward neural network with the available features in input, a single hidden layer and an output layer.

neurons was optimized case-specifically depending on the number of features, which was verified by the validation dataset.

3.2. Support Vector Machine

A support vector machine with a Gaussian Radial Basis Function (RBF) kernel was used. The most important part of a SVM is the selection of the hyperparameters. But the optimal spread of the RBF and the corresponding penalty parameter were attained by pinpointing a large interval of possibilities and verifying the results with cross-validation. In optimization a "LibSVM" tool was used [8]. However, some manual adjustment on the bias and the threshold of the predictions were done to get more accurate results for the test set.

With the use of a non-linear RBF kernel in an SVM, we can map any non-linear input function into high dimensional feature spaces that are illustrated in Figure 3. This leads to a convex linear optimization problem, which is relatively easy to calculate. This way we can obtain a global optimum and overfitting is not problem in the sense that it is in an ANN.

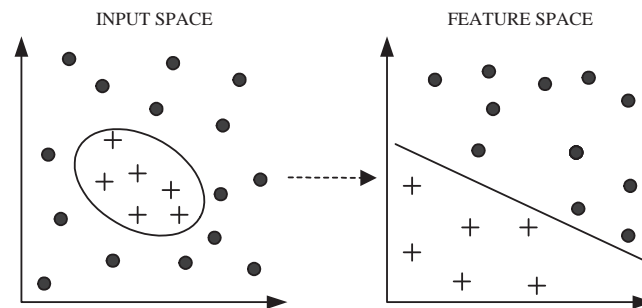


Figure 3. Illustrates the two-class classification problem, where the input is mapped on a higher dimensional feature space, which leads to a linear classification problem.

4. Feature Selection

4.1. Forward/Backward Feature Selection Algorithm

The main method for pre-processing was the so-called "Forward Backward Feature Selecting" algorithm (FBFS) which has also been used in *e.g.* adaptive context selection [9]. The purpose of this technique was to select only the most suitable features for the dataset. The features were selected one by one into the dataset as long as the result got better, and along the way it was possible to drop one out if it became worth it. But this technique was only necessary for the ANN because the large number of features in the dataset creates huge computational demands. However, the computational complexity of SVM does not depend on the dimensionality of the input space.

4.2. Self-organizing Maps and K-Means Clustering

On the basis of a previous study "Defect prediction in hot strip rolling" [4], which was done for the same dataset, there was a lot of useful information on the features and their segmentation in proportion to the retentions. The study was utilized using the self-organizing maps, originally developed by Kohonen [10], which are an unsupervised learning method that organizes n-dimensional input space into a two dimensional line of knots, where each knot includes similar samples of each other measured by Euclidian distance. By the help of SOMs we can analyse the significance of the input features in relation to the other features. The SOMs were done in many stages where insignificant features were excluded from the dataset. After the final stage 16 most meaningful features were left in the dataset. These were clustered by using k-means clustering algorithm. This 16 feature dataset was also used in the results in comparison to other feature selection methods.

4.3. Grouping Features

With the information obtained by the aforementioned clustering and analytic evaluation of the most significant and advantageous features gave us three features that we used in grouping the dataset. From the boundaries of these features we attained 9 different more homogenous groups.

Classification within these groups did not improve the results, however, but we found an interesting group that had a significantly low retention percentage, which we isolated from the classification input. This group had only 2.1% of retentions, and it contained 145 examples, 2.4% of all the examples. Hence, it could be left out of classification due to the small percentage of retentions, which could not be achieved by prediction.

5. Results

In the classification problem of two classes, *i.e.* 0 and 1, the confidence value for the predictions given by the classifier on each sample are given from 0 to 1. In this case we used a boundary point of 0.5 *i.e.* the samples predicted below 0.5 are rounded downwards to 0 and samples with a prediction of 0.5 or greater are rounded up to 1. Results obtained

CLASSIFIER	FEATURES	TOTAL ERROR
ANN(LM)	FBFS (12-feat.)	17.5%
ANN(LM)	Clustering (16-feat.)	17.6%
ANN(LM)	All features	17.7%
ANN(SCG)	FBFS (12-feat.)	17.8%
ANN(SCG)	Clustering (16-feat.)	18.1%
ANN(SCG)	All features	17.6%
SVM	FBFS(12-feat.)	19.8%
SVM	Clustering (16-feat.)	18.7%
SVM	All features	18.1%

Table 1. Comparison of results obtained by all classifiers

using this binary type of classification are shown in Table 1. As we can see, the ANN works most effectively with the FBFS algorithm and the SVM is most effective when all the features are included in the dataset. Although these results in Table 1 are comparable, they are useless in the sense that the prediction error for retained samples are on average 45% and 5% for the successful samples.

This led to an approach that gave a stronger weight to the retained samples that were predicted correctly. It was obtained with using supervised training and then manually adjusting the threshold and bias in terms of the training data and verifying these results with a test set. Due to this adjusting the total loss of classification increased while the prediction error for retained samples decreased, which is illustrated in Table 2 for the ANN and in Table 3 for the SVM, where the predictions are in the columns and the true values in the rows.

Due to these unsatisfactory results obtained by only using this classical binary -type of classification there was a need to examine where the errors were located. This procedure would enable us to find intervals on the distribution of the classification, where the predictions would be comparatively reliable. In Figures 4 and 5 we have examined the distribution of the classification errors on the confidence values with respect to one decimal. On the x -axis we have the confidence value given by the classifier *e.g.* if all samples would be classified correctly then the figures would contain only gray colored successful samples on the left from 0.0 to 0.5 and the black colored samples would lie on the right from 0.5 to 1.0. With this technique we obtain reliability levels for the classifications given by the classifier *i.e.* if the NN gives a confidence value from 0.9 to 1.0, we can tell that it is a sample with a retention code of 96% certainty. The largest percentage of the incorrectly predicted samples lies close to probability 0.5 on both classifiers, which is the most problematic area. This problem area is slightly wider for the ANN (from 0.4 to 0.7) than it is for the SVM (from 0.4 to 0.6), where the prediction error is greater than 25%. If these problem areas would be removed from the classification set, we could predict more than three quarters of the samples with about 90% accuracy.

6. Discussion and Conclusions

The study was made at the steel plant of Ruukki due its high financial significance. This kind of approach could work, but in our point of view the dataset is missing some valuable information. This information could be attained by adding some pyrometer in the

ANN	Successful	Retention	True codes
Successful	1556	610	2166
Retention	206	557	763
Predicted codes	1762	1167	2929

Table 2. Error percentage of samples with retention code 27.0% and 28.0% respectively for the successful samples. 27.7% in total. The rows indicate the true codes and the columns the predicted codes for the samples.

SVM	Successful	Retention	True codes
Successful	1620	546	2166
Retention	222	541	763
Predicted codes	1842	1087	2929

Table 3. Error percentage of samples with retention code 29.1% and 25.2% respectively for the successful samples. 26.2% in total. The rows indicate the true codes and the columns the predicted codes for the samples.

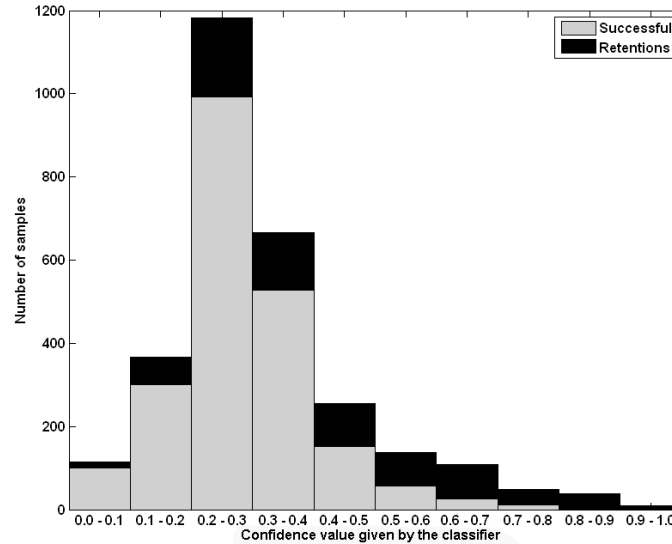


Figure 4. Predictions made by SVM

production line or improving the existing physical models. Also these physical models are related in large part to the parameters that are estimated beforehand and used in classification. Because of this, many parameters may have more or less random error, which makes the classification more unreliable. We have nevertheless shown that the classifiers performs quite similarly in this subject with regard to the classification error, although the distributions are segmented quite differently.

With respect to the distribution of the confidence values and the reliability levels given by the classifiers there could be some closer inspection, *i.e.* clustering, on the boundaries of confidence values. This would make it possible to find homogenous groups in the predictions, which could be classified correctly to a large degree.

A different kind of approach could be the combination of the online rolling process parameter optimization and the defect classifier. In this case the rolling parameters could be optimized for instance by a genetic algorithm and the defect classifier would verify

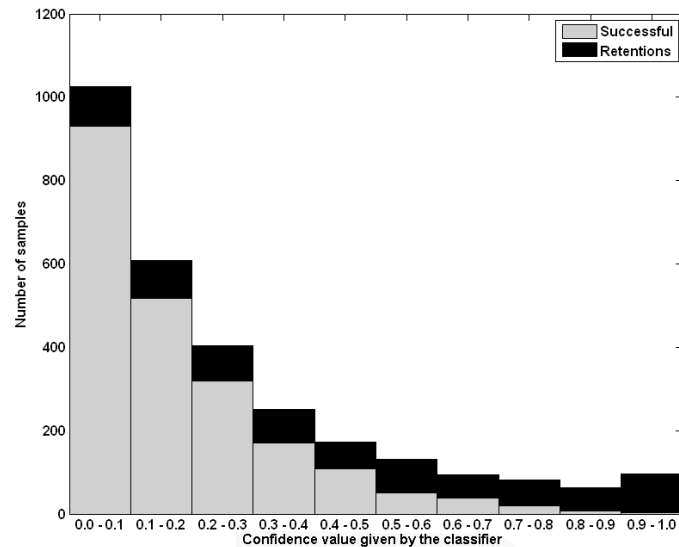


Figure 5. Predictions made by NN(SCG)

that no defects occur, and if some defects would appear, then the genetic algorithm could find the second best solution, in terms of the optimal, that does not cause any defects.

References

- [1] P. Laurinen and J. Roning, An adaptive neural network model for predicting the post roughing mill temperature of steel slabs in the reheating furnace, *Journal of Materials Processing Technology* **168** (2005), 423–430.
- [2] T. Fechner, D. Neumerkel and I. Keller, Adaptive neural network filter for steel rolling, *IEEE International Conference on Neural Networks* **6** (1994), 3915–3920.
- [3] V. N. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer-Verlag, (1995).
- [4] U. Elsila and J. Roning, Defect prediction in hot strip rolling, *Ironmaking & Steelmaking* **3** (1993), 241–248.
- [5] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, (1995).
- [6] M. T. Hagan and H. B. Demuth, *Neural Network Design*, MA: PWS Publishing, (1996).
- [7] W. H. Chen and J. Y. Shih, Comparison of support vector machines and back propagation neural networks in forecasting the six major stock markets, *Int. J. Electronic Finance* **1**, (2006), 49–67.
- [8] C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, (2001), Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [9] W. Jiang and A. Ortega, Forward/backward adaptive context selection with applications to motion vector field encoding, *International Conference on Image Processing*, **2**, (1997), 168–171.
- [10] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, (1995)