World Automation Congress © 2010 TSI Press.

ENHANCING FAULT PREDICTION ON AUTOMATIC FOUNDRY PROCESSES

IGOR SANTOS, JAVIER NIEVES AND PABLO G. BRINGAS
S^3lab,University of Deusto
Bilbao, Spain
{isantos, javier.nieves, pablo.garcia.bringas}@deusto.es

ABSTRACT— Microshrinkages are known as probably the most difficult defects to avoid in high-precision foundry. This failure renders the casting invalid, with the subsequent cost increment. Modelling the foundry process as an expert knowledge cloud allows machine learning algorithms to foresee the value of a certain variable, in this case, the probability that a microshrinkage appears within a casting. In this paper, we extend previous research on foundry production control by adapting and testing support vector machines and decision trees for the prediction in beforehand of microshrinkages. Finally, we compare the obtained results and show that decision trees are more suitable than the rest of the counterparts for the prediction of microshrinkages.

Key Words: fault prediction, machine-learning, industrial processes optimisation

1. INTRODUCTION

Foundry has evolved from the ancient-magic-surrounded activity it was once, to a backbone that maintains the whole industrial world. Hence, the foundry process is subject to very strict safety controls in order to assure the quality of the manufactured castings since, as one may think, the tiniest defect may become fatal. Commonly, the techniques for the assurance of failure free foundry processes are exhaustive production control and several simulation techniques [1]. Unfortunately, these methods are not still capable of preventing what is known to be the most difficult flaw in ductile iron castings, namely the microshrinkage. Specifically, this imperfection, also called secondary contraction, consists of tiny porosities that appear when the casting is cooling down. In this way, almost all process parameters interact on its apparition making it very difficult to avoid so far.

Further, the problem of the microshrinkage apparition is very difficult to solve due to the following reasons [2]. First, many variables have an effect in the creation of the secondary contraction. Second, the data-acquisition systems gather much information but it is not prioritised in any way. Third, it is very hard to establish cause-effect relationships between the variables of the system. Finally, the human knowledge used in this task tends to be subjective, incomplete and not subjected to any empirical test [3].

As shown in [2, 4], a machine-learning-based tool could help in this goal. After a training period, the machine-learning classifiers learned the behaviour of the model and thereafter, they were able to foresee its outcome (i.e. the formation or not of the microshrinkage).

Still, similar machine-learning classifiers have been applied in domains alike with outstanding results, for instance, artificial neural networks [5] or the k-nearest neighbour algorithm [6]. In this way, successful applications of artificial neural networks include for instance spam filtering [7], decision support systems for cancer [8] or industrial fault diagnosis [9]. Similarly, k-nearest neighbour algorithm has been applied for instance to visual category recognition [10], automated transporter prediction [11], weather forecasting [12], protein function prediction [13], malware detection [14] or image retrieval [15].

These good results boosted us to test another machine learning models. In those experiments, [4, 16], we discovered that we do not use the same classifier for all defects and thus, we decided to get the best classifier for each fault.

Finally, more machine learning models have been used, for instance, support vector machines [17] and decision trees [18]. To this extent, some examples of success applying support vector machines and decision trees are the following: fault prediction using early lifecycle data [19], identification of gas turbine faults [20], fault diagnosis [21, 22] and prediction of time-series [23].

Against this background, this paper advances the state of the art in two main ways. First, we address here a methodology to adapt machine learning classifiers, specifically, support vector machines and decision trees, to the prediction of microshrinkages and we describe the method for training them. Second, we evaluate the classifiers with an historical dataset from a real foundry process in order to compare the accuracy and suitability of each method.

2. RELATED WORK

Nowadays, machine-learning models are being used increasingly more in fault prediction issues. There has been a hectic activity around the applications of Artificial Neural Networks (ANN) to solve other problems of foundry process, for instance, on the prediction of the ferrite number in stainless steel arc welds [24]. ANNs have also been used in classifying foundry pieces [25], optimising casting parameters [26] and detection of causes of casting defects [27].

Similarly, successful experiments involving k-nearest neighbour (KNN) algorithm include fault detection of semiconductor manufacturing processes [28]. In addition, KNN algorithm and ANN have been applied for enhancing quality of steel [29], achieving an overall root mean square error of 0:38 (a comparison between their results and ours is showed in section 5). Likewise, Bayesian networks have been applied as previous methods in Bayesian neural networks methodology (i.e. forecasting the ferrite number in stainless steel [30]) and as the axis to establish the level of microshrinkage in iron castings [2]. The good results obtained by these researches encouraged us to tailor these approaches into our concrete problem domain. Therefore, in our previous work we predicted the mechanical properties, more accurately, being able to foresee the ultimate tensile strength [3, 15].

Remarkably, there are several experiments that try to predict castings properties using machine learning methods [31]. Despite these experiments, to our knowledge there is no single published research on the prediction of microshrinkages within iron castings, specially using support vector machines and decision trees.

3. MICROSHRINKAGES AND FOUNDRY PRODUCTION

Microshrinkages appear during the cooling phase of the metal but they cannot be noticed until the production is accomplished. More accurately, this flaw consists of minuscule internal porosities or cavities. Since metals are less dense as a liquid than as a solid, the density of the metal increases while it solidifies and the volume decreases in parallel. In this process, diminutive, microscopically undetectable interdendritic voids may appear leading to a reduction of the castings hardness and, in the cases faced here where the casting is a part of a very sensitive piece, rendering the piece useless [32].

Commonly, the way to examine finished parts is the usage of non-destructive inspections that are performed one the final piece is obtained. In this way, the most common techniques are X-ray and ultrasound emissions but, unfortunately, both require suitable devices, specialised staff and quite a long time to analyse all the produced parts. Therefore, postproduction inspection is not an economical alternative to the preproduction detection of microshrinkages.

Specifically, the main variables to control in order to predict the apparition of microshrinkages can be classified into metal-related and mould-related categories. On one hand, metal-related variables are divided into the following categories:

- **Composition***:* Type of treatment, inoculation and charges.
- **Nucleation potential and melt quality:** Obtained by means of a thermal analysis program [33].
- **Pouring:** Duration of the pouring process and temperature.

On the other hand, mould-related variables can be split into the following categories:

- **Sand:** Type of additives used, sand-specific features and carrying out of previous test or not.
- **Moulding:** Machine used and moulding parameters.

Furthermore, there are some variables that may influence the apparition of secondary contraction during the foundry process, like the composition, the size of the casting, the cooling speed and thermal treatment [2, 3, 4]. The system must take into account all of them in order to achieve a prediction of that defect. In this way, the machine-learning classifiers are the same we used in our previous experiments, and they were composed of about 24 variables [4].

4. MACHINE LEARNING CLASSIFIERS

4.1 Support Vector Machines (SVM)

Support vector machines [17] consist on finding an hyperplane or a set of hyperplanes that divides the n-dimensional space of the data in two regions (although there are enhancements for multi-class classification). This hyperplane is the one that maximises the distance between the examples of the two classes. This separation between those regions is called functional margin. In general, the larger the functional margin, the lower the generalization error of the classifier.

Supporting vectors are the instances that are situated near the hyperplane. Since sometimes the space cannot be divided with an hyperplane, a kernel function K is used. This function studies the relations within the data and creates complex divisions in the space. Although the resulting algorithm is similar, every dot product is replaced by a non-linear kernel function.

4.2 Decision Trees

These classifiers [18] constitute a decision support tool represented as a tree-like graph. They are also used to model the decisions and their possible consequences. To this extent, decision trees are composed by three kinds of nodes: decision nodes, chance nodes and end nodes.

Moreover, there are several training algorithms that are typically used for learning the graph structure of these trees using a labelled dataset. In this work, we used random forest, which is an ensemble (i.e. combination of weak classifiers) of different randomly-built decision trees [34]. Further, we also used J48 (the Weka [35] implementation of the C4.5 algorithm developed by Ross Quinlan [36]). Since generated decision tree can be used for classification, it is often referred as a statistical classifier.

C. Bayesian Networks

Bayesian networks are probabilistic models for multivariate analysis. Formally, they are directed acyclic graphs associated to a probability distribution function [37]. Nodes in the graph represent variables (i.e. they can be either a premise or a conclusion), and the arcs represent conditional dependencies between such variables.

Further, the probability function illustrates the strength of these relationships (i.e. arcs) in the graph [37]. To our needs, the most important ability of Bayesian networks is their capability of inferring the probability that a certain hypothesis becomes true (i.e. the probability of a microshrinkage to appear).

5. EXPERIMENTAL RESULTS

In order to validate our method, we have acquired data from a foundry specialised in safety and precision components for the automotive industry, principally in disk-brake support with a production over 45000 tons a year. Note that, as aforementioned, microshrinkages have internal presence, hence, the evaluation must be done according to non-destructive X-ray, first, and ultrasound testing techniques, thenceforth, to ensure that even the smallest microshrinkages are found [3].

In the experiments, the machine-learning classifiers have been built with the aforementioned 24 variables. We have worked with two different references (i.e. type of pieces) and, in order to test the accuracy of the predictions, with the results of the non-destructive X-ray and ultrasound inspections from 951 production stocks (note that each production stocks may involve several castings or pieces) performed in beforehand. To this extent, we have defined risk levels as follows: Risk 0 (no microshrinkages seen) and Risk 1 (more than a microshrinkage seen for a production stock). Specifically, we have followed the next configuration for the performed experiment:

- **Cross validation:** Despite the small dataset, we have to use as much of the available information in order to obtain a proper representation of the data. To this extent, K-fold cross validation is usually used in machine learning experiments [38]. In our experiments, we have performed a K-fold cross validation with k=10. In this way, our dataset is 10 times split into 10 different sets of learning (66 % of the total dataset) and testing (34 % of the total data).
- **Learning the model:** For each fold, we have performed the learning phase of each algorithm with the corresponding training dataset, applying different parameters or learning algorithms depending on the model. More accurately, we have use the following models:
  - *Support Vector Machines:* In order to train support vector machines we have used different kernels: a polynomial kernel [39], a normalised polynomial kernel [40], a Pearson VII function-based universal kernel [41] and a radial basis function (RBF) based kernel [42].
  - *Decision Trees:* We have performed experiments with random forest [42], an ensemble of randomly constructed decision trees using different amount of trees (n): n = 10, n = 50, n = 100, n = 150, n = 200, n = 250 and n = 300. And we have also used J48 (the Weka [38] implementation of the C4.5 algorithm).

  - *Bayesian Network:* We have used Tree Augmented Naïve (TAN) [37] as structural learning algorithm.

- **Testing the model:** For each fold, we evaluated the accuracy and the error rate between the predicted value set X and the real value set Y (both with the size of the testing dataset m) with mean absolute error (MAE): $MAE(X,Y)=\sum_{i=1}^{m}\frac{|X_i-Y_i|}{m}$. Likewise, we have also used root mean square error (RMSE) that is defined: $RMSE(X,Y)=\frac{1}{m}\sum_{i=1}^{m}(X_i-Y_i)^2$.

As we mentioned before, we have evaluated the classifiers in terms of prediction accuracy and error (i.e. MAE and RSME). In this way, Table I illustrates the obtained results. Regarding the prediction accuracy, using the full original dataset we are able to achieve a 94.24% of accuracy level. *Random forest with 200* trees outperformed the rest of classifiers. On one hand, each an every of the random forest are better classifiers than the J48.

Machine-learning Model	**Accuracy (%)**	**MAE**	**RMSE**
Bayesian Network with TAN	91.52	0.11	0.25
SVM with Polynomial Kernel	91.86	0.08	0.28
SVM with Normalised Polynomial Kernel	92.29	0.08	0.27
SVM with Pearson VII universal kernel	73.29	0.27	0.52
SVM with Radial Basis Function Kernel	90.98	0.09	0.30
Decision Tree: RandomForest with 10 trees	92.46	0.15	0.25
Decision Tree: RandomForest with 50 trees	94.02	0.16	0.23
Decision Tree: RandomForest with 100 trees	94.06	0.16	0.23
Decision Tree: RandomForest with 150 trees	94.15	0.16	0.23
Decision Tree: RandomForest with 200 trees	94.24	0.16	0.23
Decision Tree: RandomForest with 250 trees	94.16	0.16	0.23
Decision Tree: RandomForest with 300 trees	94.25	0.16	0.23
Decision Tree: J48	91.91	0.12	0.27

Table I. Results in terms of accuracy and error rates

With that specific configuration, random forest achieves the best accuracy level and before and after that point the accuracy is lower. Similarly, SVM with polynomial-based kernel were also able to overcome the Bayesian network with an accuracy of 92,29 %. Besides, J48 performed a little better than the Bayesian network.

Furthermore, regarding the mean absolute error and the root mean square error, the results are similar to the ones of prediction accuracy and the random forest with 200 trees also outperformed the rest of algorithms in terms of root square error distribution whilst SVM with Radial-basis-function-based kernel outperformed in terms of mean absolute error.

Summarizing, the system has achieved overall good results (more than 94%). In this way, we can reduce in a significant manner the cost and the duration of the actual testing methods. Remarkably, the outstanding results achieved by the *random forest with 200 trees* show that it can be used in a similar way as we have used the Bayesian networks or artificial neural networks in previous works. Still, the Sensitive Module (SM) [26] we used for microshrinkage apparition provided an interesting decision support system for the operators in the foundry rely on cause-effect relationships. Specifically, SM studies the different values that each variable adopts in order to trace the influence of such values in the apparition of microshrinkages.

6. CONCLUSIONS

Predicting the apparition of microshrinkages in ductile iron castings is one of the most hard challenges in foundry-related research. Our work in [2] pioneers the application of Artificial Intelligence to the prediction of microshrinkages.

Specifically, in this research, we have included and adapted to our particular problem domain two classifiers that have been used widely in similar issues. All of them behave well, but random forests outperform the rest of the classifiers. In addition, as we noticed in previous works [2, 4, 16], there are some irregularities in the data that may alter the outcome rendering it not as effective as it should. More accurately, these inconsistencies appear because the data acquisition is performed in a manual fashion.

Accordingly, the future development of this predictive tool will be oriented in four main directions. First, we plan to extend our analysis to the prediction of other defects in order to develop a global system of incident analysis. Second, we will compare more supervised and semi-supervised machine learning algorithms in order to prove their effectiveness to predict foundry defects. Third, we plan to integrate the best classifiers in a meta-classifier which will work as a black box combining all partial results to predict any defect. And, finally, we plan to test a preprocessing step to reduce the irregularities in the data.

REFERENCES

1. J. Sertucha, A. Loizaga, and R. Suárez, “Improvement opportunities for simulation tools,” in Proceedings of the 16th European Conference and Exhibition on Digital Simulation for Virtual Engineering, 2006, invited talk.
2. Y. Penya, P. García Bringas, and A. Zabala, “Advanced fault prediction in high-precision foundry production,” in Proceedings of the 6th IEEE International Conference on Industrial Informatics, 2008, pp. 1673– 1677.
3. A. Zabala, R. Suárez, and J. Izaga, “Iron castings, advanced prediction tools, foundry process control and knowledge management,” in Proceedings of the 68th WFC - World Foundry Congress, 2008, pp. 355–360.
4. I. Santos, J. Nieves, Y. K. Penya, and P. G. Bringas, “Optimising machine-learning-based fault prediction in foundry production.” in Proceedings of the 2nd International Symposium on Distributed Computing and Artificial Intelligence (DCAI),. S. Omatu et al. (Eds.): IWANN 2009, Part II, LNCS 5518, Springer-Verlag Berlin Heidelberg, 2009, pp. 553–560.
5. C. M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press, 1995.
6. E. Fix and J. L. Hodges, “Discriminatory analysis: Nonparametric discrimination: Small sample performance,” Technical Report Project 21-49-004, Report Number 11, 1952.
7. S. Elfayoumy, Y. Yang, and S. Ahuja, “Anti-spam filtering using neural networks,” in Proceedings of the International Conference on Artificial Intelligence, vol. 4. IC-AI, pp. 984–989.
8. P. Lisboa and A. Taktak, “The use of artificial neural networks in decision support in cancer: a systematic review,” Neural Networks, vol. 19, no. 4, pp. 408–415, 2006.
9. S. Simani and C. Fantuzzi, “Neural networks for fault diagnosis and identification of industrial processes,” in Proceedings of 10th Eurorean Symposium on Artificial Neural Networks (ESANN), 2002, pp. 489–494.
10. H. Zhang, A. Berg, M. Maire, and J. Malik, “Svm-knn: Discriminative nearest neighbor classification for visual category recognition,” in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2006, pp. 2126–2136.
11. H. Li, X. Dai, and X. Zhao, “A nearest neighbor approach for automated transporter prediction and categorization from protein sequences,” Bioinformatics, vol. 24, no. 9, p. 1129, 2008.
12. M. Sharif and D. Burn, “Improved k-nearest neighbor weather generating model,” Journal of Hydrologic Engineering, vol. 12, p. 42, 2007.
13. S. Kharsikar, D. Mugler, D. Sheffer, F. Moore, and Z. Duan, “A weighted k-nearest neighbor method for gene ontology based protein function prediction,” Computer and Computational Sciences, 2007. IMSCCS 2007. Second International Multi-Symposiums on, pp. 25–31, 2007.
14. I. Santos, Y. K. Penya, J. Devesa, and P. G. Bringas, “N-grams-based file signatures for malware detection,” in Proceedings of the 11th International Conference on Enterprise Information Systems (ICEIS), Volume AIDSS, 2009, pp. 317–320.
15. Z. Lu and H. Burkhardt, “Fast Image Retrieval Based on Equal-average Equal-variance K-Nearest Neighbour Search,” in Proceedings of 18th International Conference on Pattern Recognition, vol. 2, p. 853.
16. I. Santos, J. Nieves, Y. K. Penya, and P. G. Bringas, “Machine-learning-based mechanical properties prediction in foundry production,” in Proceedings of ICROS-SICE International Joint Conference (ICCAS-SICE), 2009, pp. 4536–4541.
17. V. Vapnik, The nature of statistical learning theory. Springer, 2000.
18. J. Quinlan, “Induction of decision trees,” Machine learning, vol. 1, no. 1, pp. 81–106, 1986.

19. Y. Jiang, B. Cukic, and T. Menzies, “Fault prediction using early lifecycle data,” in ISSRE ’07: Proceedings of the The 18th IEEE International Symposium on Software Reliability. Washington, DC, USA: IEEE Computer Society, 2007, pp. 237–246.
20. M. Maragoudakis, E. Loukis, and P.-P. Pantelides, “Random forests identification of gas turbine faults,” in ICSENG ’08: Proceedings of the 2008 19th International Conference on Systems Engineering. Washington, DC, USA: IEEE Computer Society, 2008, pp. 127–132.
21. J. Y. amd Youyun Zhanga and Y. Zhu, “Intelligent fault diagnosis of rolling element bearing based on svms and fractal dimension,” Mechanical Systems and Signal Processing, vol. 1, pp. 2012–2024, 2007.
22. S. Yoon and S. Kim, “Mutual information-based svm-rfe for diagnostic classification of digitized mammograms,” Pattern Recognition Letters, vol. 30, no. 16, pp. 1489–1495, December 2009.
23. U. Thissena, R. van Brakela, A. P. de Weijerb, W. J. Melssena, and L. M. C. Buydens, “Using support vector machines for time series prediction,” Chemometrics and Intelligent Laboratory Systems, vol. 69, pp. 35–49, 2003.
24. J. Vitek, S. David, and C. Hinman, “Improved ferrite number prediction model that accounts for cooling rate effects part 1 model development,” Welding Journal, vol. 82, no. 10, pp. 43–50, 2003.
25. A. Lazaro, I. Serrano, J. Oria, and C. de Miguel, “Ultrasonic sensing classification of foundry pieces applying neuralnetworks,” in 5th International Workshop on Advanced Motion Control, 1998, pp. 653–658.
26. P. Zhang, Z. Xu, and F. Du, “Optimizing casting parameters of ingot based on neural network and genetic algorithm,” in ICNC ’08: Proceedings of the 2008 Fourth International Conference on Natural Computation. Washington, DC, USA: IEEE Computer Society, 2008, pp. 545–548.
27. P. M. and K. A., “Detection of causes of casting defects assisted by artificial neural networks,” in Proceedings of the I MECH E Part B Journal of Engineering Manufacture, vol. 217, no. 9, 2003.
28. H. Peter and J. Wang, “Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes,” IEEE Transactions on Semiconductor Manufacturing, vol. 20, no. 4, pp. 345–35, 2007.
29. K. Schnelle and R. Mah, “Product quality management using a real-time expert system,” ISIJ International, vol. 34, no. 10, pp. 815–821, 1994.
30. M. Vasudevan, M. Muruganath, and A. K. Bhaduri, “Application of bayesian neural network for modelling and prediction of ferrite number in austenitic stainless steel welds,” ser. Mathematical Modelling of Weld Phenomena - VI. London: Institute of Materials, pp. 1079–1100, 2002.
31. J. Vorácek, “Prediction of mechanical properties of cast irons,” Appl. Soft Comput., vol. 1, no. 2, pp. 119–125, 2001.
32. Inoculation alloy against microshrinkage cracking for treating cast iron castings. Patent US 2005/0180876 A 1
33. P. Larrañaga, J. Sertucha and, R., Suárez, “Análisis del proceso de solidificación en fundiciones grafíticas esferoidales”, Revista de Metalurgia, vo. 42, no 4 , 2006, pp 244–255.
34. L. Breiman, “Random forests,” Machine learning, vol. 45, no. 1, pp.5–32, 2001.
35. S. Garner, “Weka: The Waikato environment for knowledge analysis,” in Proceedings of the New Zealand Computer Science Research Students Conference, 1995, pp. 57–64.
36. J. Quinlan, C4. 5 programs for machine learning. Morgan Kaufmann Publishers, 1993
37. E. Castillo, J. M. Gutiérrez, and A. S. Hadi, *Expert Systems and Probabilistic Network Models*. Springer-Verlag, 1997
38. R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in International Joint Conference on Artificial Intelligence, vol. 14, 1995, pp. 1137–1145.
39. S. Amari and S. Wu, “Improving support vector machine classifiers by modifying kernel functions,” Neural Networks, vol. 12, no. 6, pp. 783– 789, 1999
40. S. Maji, A. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” in Proc. CVPR, vol. 1, no. 2, 2008, p. 4.
41. B. Ustun, W. Melssen, and L. Buydens, “Visualisation and interpretation of support vector regression models,” Analytica chimica acta, vol. 595, no. 1-2, pp. 299–309, 2007.
42. B. Cho, H. Yu, J. Lee, Y. Chee, I. Kim, and S. Kim, “Nonlinear support vector machine visualization for risk factor analysis using nomograms and localized radial basis function kernels,” IEEE Transactions on Information Technology in Biomedicine, vol. 12, no. 2, p. 247, 2008.