Collective Classification for the Prediction of Microshrinkages in Foundry Production

Igor Santos, Javier Nieves, Carlos Laorden, Borja Sanz and Pablo Garcia Bringas
S^3Lab, DeustoTech - Computing
University of Deusto
Bilbao, Spain
Email: {isantos, jnieves, claorden, borja.sanz, pablo.garcia.bringas}@deusto.es

***Abstract*—Microshrinkages are known as probably the most difficult defects to avoid in high-precision foundry. This failure is not corrigible, with the subsequent cost increment. Modelling the foundry process using machine learning allows algorithms to foresee the value of a certain variable, in this case, the probability that a microshrinkage appears within a foundry casting. However, this approach needs to label every instance to generate the model that will classify the castings. In this paper, we present a new approach for detecting faulty castings through collective classification to reduce the labelling requirements of completely supervised approaches. Collective classification is a type of semi-supervised learning that optimises the classification of partially-labelled data. We perform an empirical validation demonstrating that the system maintains a high accuracy rate while the labelling efforts are lower than when using supervised learning.**

I. Introduction

The casting production or the foundry process is considered as one of the main factors influencing the development of the world economy. The actual capacity of the world's casting production, which is higher than 60 million metric tones per year, is strongly diversified [1]. The last decade brought significant changes worldwide for the greatest casting producers. Currently, the biggest producer is China, closely followed by Europe. Producers supply key pieces to many other industries, such as automotive, naval, weapon and aeronautic. Therefore, the foundry process is subject to very strict safety controls to assure the quality of the manufactured castings because, as one may think, the tiniest defect may become fatal.

The techniques for the assurance of failure-free foundry processes are exhaustive production control and diverse simulation techniques [2]. Many of the techniques used can only be applied when the casting is done. Thus, when a faulty casting is detected, it must be remelted, which can be translated into a cost increment.

Unfortunately, these methods are still incapable of preventing what is known to be the most difficult flaw in ductile iron castings, namely the microshrinkage. More specifically, this imperfection, also called secondary contraction, consists in tiny porosities that appear inside the casting when it is cooling down. The difficulty of its detection is due to the fact that almost all the parameters of the foundry process influence the apparition of microshrinkages.

Indeed, the problem of the microshrinkage apparition is very difficult to solve [3], [4], [5] due to the following reasons: (i) A huge amount of data, not prioritised or categorised in any way, is required to be managed, (ii) it is very hard to find cause-effect relationships between the variables of the system, and (iii) the human knowledge used in this task usually tends to be subjective, incomplete and not subjected to any test.

Currently, machine learning is being used increasingly in the field of metallurgy to solve the aforementioned problems. One of the most widely used methods is the application of neural networks in several aspects such us classifying foundry pieces [6], optimising casting parameters [7], detecting causes of casting defects [8] amongst other related problems [9], [10]. Similarly, other experiments involving the K-nearest neighbour algorithm include fault detection of semiconductor manufacturing processes [11]. Bayesian networks are also used as previous methods in Bayesian neural networks methodology (e.g., predicting the ferrite number in stainless steel [12]).

In our previous work, we tested several machine-learning classifiers [3], [13] (e.g., Bayesian networks, support vector machines, decision trees, artificial neural networks among others) to identify which is the best classifier to predict microshrinkages and, also, to reduce the noise in the data gathering process produced by the foundry workers [4].

However, these supervised machine-learning classifiers require a high number of labelled castings for each class. It is quite difficult to obtain this amount of labelled data for a real-world problem such as foundry defect prediction. To gather these data, a time-consuming process of analysis is mandatory, and in the process, some changes in the production may appear.

Semi-supervised learning is a type of machine-learning technique specially useful when a limited amount of labelled data exists for each class [14]. In particular, collective classification [15] is an approach that uses the relational structure of the combined labelled and unlabelled data-sets to enhance the classification accuracy. With these relational approaches, the predicted label of an example will often be influenced by the labels of related samples. Collective classification has been used with success in text classification [15], malware detection [16] or spam filtering [17].

The idea underlying collective classification is that the predicted labels of a test sample should also be influenced by the predictions made for related test samples. Sometimes, we can determine the topic of not just a single evidence but to infer it for a collection of unlabelled evidences. Collective classification tries to collectively optimise the problem taking into account the connections present among the instances. In summary, collective classification is a semi-supervised technique, i.e., uses both labelled and unlabelled data — typically a small amount of labelled data and a large amount of unlabelled data —, that reduces the labelling work.

Given this background, we present here the first approach that employs collective classification techniques for classifying castings and to foresee microshrinkages. These methods are able to learn from both labelled and unlabelled data to build accurate classifiers. We propose the adoption of collective learning for the detection of microshrinkages using features extracted from the foundry production parameters as we did in previous work [3], [13].

Summarising, our main contributions in this paper are: (i) we describe how to adopt collective classification for microshrinkage detection, (ii) we empirically determine the optimal number of labelled instances and we evaluate how this parameter affects the accuracy of the model, () and (iii) we demonstrate that labelling efforts can be reduced in the fault prediction problem, while still maintaining a high accuracy rate.

The remainder of this paper is organised as follows. Section II details the casting production process and presents the most difficult defect to avoid, the microshrinkage. Section III describes different collective classification methods and how they can be adopted for fault prediction. Section IV describes the experiments and presents results. Finally, Section V concludes the paper and outlines avenues for future work.

II. Foundry Process and Microshrinkages

The foundry process can be considered as one of the axes of our society. However, a task that seems simple becomes complex due to the hard conditions in which it is carried out. Besides the casting process, the foundry workers produce castings that are close to the final product shape, i.e., 'near-net shape' components. To this end, the production has to pass through several stages in which the castings are transformed to obtain the final casting.

Although all of the foundry processes are not equal, the work flow performed in foundries is very similar to the work flow shown in Fig. 1. The most important stages are the following [18]:

- **Pattern making:** In this step, moulds (exteriors) or cores (interiors) are produced in wood, metal or resin for being used to create the sand moulds in which the castings will be made.
- **Sand mould and core making:** The sand mould is the most widely extended method for ferrous castings. Sand is mixed with clay and water or other chemical binders. Next, the specialised machines create the two halves of the mould and join them together to provide a container in which the metals are poured into.
- **Metal melting:** In this process (see 1 in Fig. 1), raw materials are melt and mixed. Molten metal is prepared in a furnace and depending on the choice of the furnace, the quality, the quantity and the throughput of the melt change.
- **Casting and separation:** Once the mixture is made, the molten material is poured into the sand mould. It can be done using various types of ladles or, in high volume foundries, automated pouring furnaces. Later, the metal begins to cool. This step (see 2 in Fig. 1) is really important because the majority of the defects can appear during this phase. Finally, when the casting has been cooled enough to maintain the shape, the casting is separated from the sand. The removed sand is recovered for further uses.
- **Removal of runners and risers:** Some parts of the casting that had been used to help in the previous processes are then removed. They can be detached by knocking off, sawing or cutting.
- **Finishing:** To finish the whole process some actions are usually performed, e.g., cleaning the residual sand, heat treatment and rectification of defects by welding.

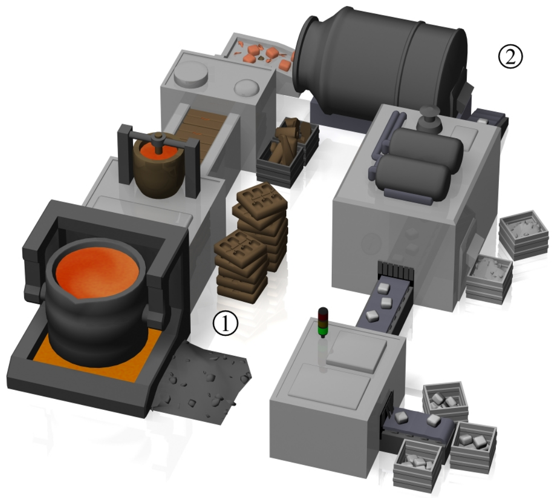


Fig. 1. Foundry process work flow showing the different phases castings have to pass through. More accurately, in 1 it is performed the metal melting step, and in 2 it is performed the casting preparation and separation step.

The complexity of detecting faulty castings using an *ex-ante* method arises principally from the high number of variables that participate in the production process and, therefore, may influence on the final design of a casting.

In consequence, the foundry process is simplified to solve the aforementioned problem. In our case, the main variables to control in order to predict the faulty castings can be classified into metal-related and mould-related categories. Metal-related variables are divided into the following categories:

- **Composition:** Type of treatment, inoculation and charges [19].
- **Nucleation potential and melt quality:** Obtained by means of a thermal analysis program [20], [21], [22].

- **Pouring:** Duration of the pouring process and temperature.

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.

The dimension and geometry of the casting also play a very important role in this practice and, thus, we included several variables to control these two features. We also took into account other parameters regarding the configuration of each machine working in the manufacturing process [23]. In this way, we represent the castings with 24 variables [3].

A casting defect is an irregularity in the casting. Defects are defined as conditions that make a casting to be corrected or rejected. There are several defects that affect metal castings such as, shrinkages, gas porosities or pouring metal defects [18]. In this paper, we deal with microshrinkages. This kind of defect usually appears during the cooling phase of the metal but it cannot be noticed until the production is accomplished. This flaw consists of a form of filamentary shrinkage in which the cavities are very small but large in number and can be distributed over a significant area of the casting, i.e., a minuscule internal porosities or cavities. It is produced because the metals are less dense as a liquid than as a solid. The density of the metal increases and it solidifies while the volume decreases in parallel. During 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 [24].

The way to examine castings is the usage of non-destructive inspections. The most common techniques are X-ray and ultrasound emissions. Unfortunately, both require suitable devices, specialised staff and quite a long time to analyse all the parts. Moreover, every test has to be done once the casting is done. Therefore, post-production inspection is not an economical alternative to the pre-production detection of microshrinkages.

Although we have already obtained overall significant results through a machine-learning-based approach predicting those imperfections [3], [25], [26], [27], [28], [29], [13], [4], these approaches require a manual labour to label every casting within the dataset. This process can be specially time-consuming for several machine-learning models and hinders a subsequent cost increment. Note that in the year 2009, China, which is the biggest producer of castings in the world, produced 35.3 million tons of castings [1] and Europe, the second producer, made 12 million tons of castings [1]. Although not all the castings were labelled, the cost of the foundry workers developing labelling tasks would be too high. Therefore, if only a little piece of the production is labelled, the cost of the prediction preprocessing steps would be reduced. Therefore, we present here a collective classification approach that requires fewer castings to be labelled. Such an approach will indeed reduce the efforts of labelling castings, working with less information available in beforehand.

III. Collective Classification

Collective classification is a combinatorial optimisation problem, in which we are given a set of castings, or nodes, $\mathcal{E} = \{e_1, ..., e_n\}$ and a neighbourhood function N, where $N_i \subseteq \mathcal{E} \setminus \{\mathcal{E}_i\}$, which describes the underlying network structure [30]. Being $\mathcal{E}$ a random collection of castings, it is divided into two sets $\mathcal{X}$ and $\mathcal{Y}$, where $\mathcal{X}$ corresponds to the castings for which we know the correct values and $\mathcal{Y}$ are the castings whose values need to be determined. Therefore, the task is to label the nodes $\mathcal{Y}_i \in \mathcal{Y}$ with one of a small number of labels, $\mathcal{L} = \{l_1, ..., l_q\}$.

We use the *Waikato Environment for Knowledge Analysis* (WEKA) [31] and its Semi-Supervised Learning and Collective Classification plugin[1]. In the remainder of this section we review the collective algorithms used in the empirical evaluation.

A. CollectiveIBK

This model uses internally WEKA's classic IBK algorithm, an implementation of the *K-Nearest Neighbour* (KNN), to determine the best k instances on the training set and builds then, for all instances from the test set, a neighbourhood consisting of k instances from the pool of train and test set (either a naïve search over the complete set of instances or a k-dimensional tree is used to determine neighbours). All neighbours in such a neighbourhood are sorted according to their distance to the test instance they belong to. The neighbourhoods are sorted according to their 'rank', where 'rank' means the different occurrences of the two classes in the neighbourhood.

For every unlabelled test instance with the highest rank, the class label is determined by majority vote or, in case of a tie, by the first class. This is performed until no further test instances remain unlabelled. The classification terminates by returning the class label of the instance that is about to be classified.

B. CollectiveForest

It uses WEKA's implementation of RandomTree as base classifier to divide the test set into folds containing the same number of elements. The first iteration trains the model using the original training set and generates the distribution for all the instances in the test set. The best instances are then added to the original training set (being the number of instances chosen the same as in a fold).

The next iterations train the model with the new training set and generate then the distributions for the remaining instances in the test set.

[1]Available at: http://www.scms.waikato.ac.nz/~fracpete/projects/collectiveclassification

C. CollectiveWoods & CollectiveTree

CollectiveWoods works like CollectiveForest using CollectiveTree algorithm instead of RandomTree.

Collective tree is similar to WEKA's original RandomTree classifier. It splits the attribute at a position that divides the current subset of instances (training and test instances) into two halves. The process finishes if one of the following conditions is met: (i) only training instances are covered (the labels for these instances are already known); (ii) only test instances in the leaf, case in which distribution from the parent node is taken, and (iii) only training instances of one class, case in which all test instances are considered to have this class.

To calculate the class distribution of a complete set or a subset, the weights are summed up according to the weights in the training set, and then normalised. The nominal attribute distribution corresponds to the normalised sum of weights for each distinct value and, for the numeric attribute, distribution of the binary split based on median is calculated and then the weights are summed up for the two bins and finally normalised.

D. RandomWoods

It works like WEKA's classic RandomForest but using CollectiveBagging (classic Bagging, a machine learning ensemble meta-algorithm to improve stability and classification accuracy, extended to make it available to collective classifiers) in combination with CollectiveTree. RandomForest, in contrast, uses Bagging and RandomTree algorithms.

IV. Empirical Validation

To evaluate our semi-supervised microshrinkage detector, we collected a dataset from a foundry specialised in safety and precision components for the automotive industry, principally in disk-brake support, with a production over 45,000 tons a year. The experiments were focused exclusively on the microshrinkage prediction. 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, afterwards, to ensure that even the smallest microshrinkages are found [5]. The acceptance/rejection criterion of the studied models resembles the one applied by the final requirements of the customer (i.e., in the examined cases, the automotive industry). According to the very restrictive quality standards imposed by these clients, pieces flawed with an unacceptable microshrinkage must be rejected.

In the validation, we worked with two different references, i.e., type of pieces and, to evaluate the proposed method, with the results of the non-destructive X-ray and ultrasound inspections from 951 production stocks performed in beforehand. Thereby, the dataset comprises 690 correct castings and 261 faulty castings.

Next, we split the dataset into different percentages of training and testing instances. In other words, we changed the number of labelled instances to measure the effect of the number of previously labelled instances on the final performance of collective classification in detecting faulty castings.

By means of this dataset, we conducted the following methodology to evaluate the proposed method:

- **Training and Test Generation.** We constructed an ARFF file [32] (i.e., Attribute Relation File Format) with the resultant vector representations of the castings to build the aforementioned WEKA's classifiers.
  We did not use cross-validation because in the evaluation we did not want to test the performance of the classifier when a fixed size of training instances is used iteratively. Otherwise, we employed a variable number of training instances and tried to predict the class of the remaining ones using collective classification in order to determine which is the best training set size. In this case, the training instances are the labelled ones whereas the unlabelled ones are the ones in the test dataset.
  Therefore, we split the dataset into different percentages of training and tested instances, changing the number of labelled instances from 10% to 90% to measure the effect of the number of labelled instances on the final performance of collective classification in detecting faulty castings.
  As aforementioned, we used the collective classification implementations provided by the *Semi-Supervised Learning and Collective Classification* package for the well-known machine-learning tool WEKA [31]. All the classifiers were tested with their default parameters.
- **Testing the Models.** To test the approach, we measured the *True Positive Rate* (TPR), i.e., the number of castings affected with microshrinkage correctly detected divided by the total number of castings:

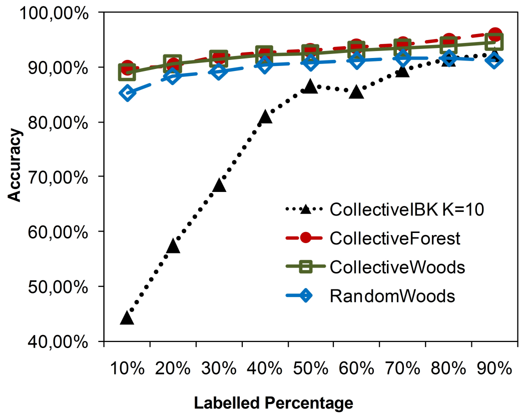
$$TPR = \frac{TP}{TP + FN} \tag{1}$$

  where TP is the number of faulty instances correctly classified (true positives) and FN is the number of faulty instances misclassified as correct castings (false negatives).
  We also measured the *False Positive Rate* (FPR), i.e., the number of not faulty castings misclassified as faulty divided by the total number of correct castings:

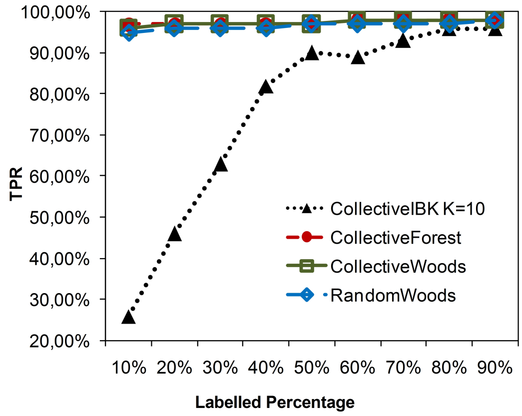
$$FPR = \frac{FP}{FP + TN} \tag{2}$$

  where FP is the number of not faulty castings incorrectly detected as faulty and TN is the number of correct castings correctly classified.
  Furthermore, we measured *accuracy*, i.e., the total number of hits of the classifiers divided by the number of instances in the whole dataset:

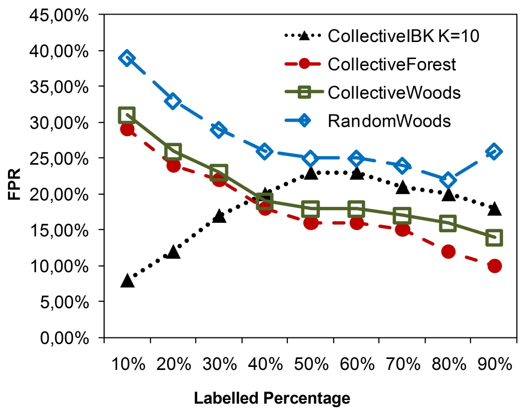
$$Accuracy(\%) = \frac{TP + TN}{TP + FP + TP + TN} \tag{3}$$



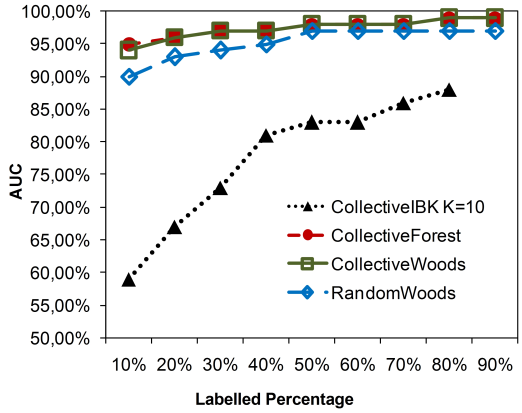
(a) **Accuracy results.** The accuracy axis (Y axis) has been scaled from 40% to 100% in order to appreciate better the evolution of CollectiveIBK. This classifier was the only one that was highly sensitive to the increase of the training dataset. The rest of the classifiers obtained accuracies higher than 85% using only a 10% of labelled instances.



(b) **TPR results.** CollectiveIBK was the only classifier that obtained low results using a low amount of labelled instances. In particular, this classifier only achieves a TPR higher than 80% using a labelled dataset size of 40% or higher. The remainder of the classifiers obtained a detection rate of more than 95% with only a 10% of training size.



(c) **FPR results.** The FPR is scaled from 0.00% to 45% in order to magnify the differences among the configurations. In general, the higher the amount of labelled instances, the lower the FPR. The FPR is always lower than 40%. Indeed, using only a 10% of the instances, we can achieve a FPR lower than 0.4% using CollectiveForest.



(d) **AUC results.** The AUC axis (Y axis) has been scaled from 50% to 100% in order to appreciate better the evolution of CollectiveIBK. As it happened with accuracy, CollectiveIBK was the only one that was highly sensitive to the increase of the training dataset. Anyhow, the rest of the classifiers obtained AUC values higher than 90% using only a 10% of labelled instances.

Fig. 2. Results of our collective-classification-based microshrinkage detection method. Collective Forest was the overall classifier with the highest accuracy, TPR and AUC.

Besides, we measured the *Area Under the ROC Curve* (AUC), which establishes the relation between false negatives and false positives [33]. The ROC curve is obtained by plotting the TPR against the FPR. All these measures refer to the test instances.

Fig. 2 shows the obtained results in terms of accuracy, TPR, FPR and AUC. Our results outline that, obviously, the higher the number of labelled castings in the dataset the better results achieved. However, by using only the 10% of the available data, with the exception of CollectiveIBK, the collective classifiers were able to achieve TPRs higher than 95% and FPRs lower than 40%. In particular, CollectiveForest trained with the 10% of the data obtained 89,87% of accuracy, 97.00% of TPR, 29.00% of FPR and 95% of AUC. Fig. 2(a) shows the accuracy results of our proposed method. All the tested classifiers, with the exception of CollectiveIBK, achieved accuracy results higher than 85%. In particular, CollectiveForest was overall the best, achieving an accuracy of 92.05% using only a 30% of the instances for training and 96,11% with the 90% of the whole dataset. Fig. 2(b) shows the obtained results in terms of correctly classified faulty castings. In this way, Collective Forest was also the best detecting the 97% or 98% of the faulty castings in different labelled percentage configurations. Fig. 2(c) shows the FPR results. Every classifier obtained results lower than 40%. In particular, the lowest FPR achieved was of 8%, achieved by CollectiveIBK with the 10% of dataset. However, in order to guarantee results of TPR higher than 80%, Collective Forest only needs to be trained with, at least, 40% of the dataset. Finally, regarding AUC, shown in Fig. 2(d), Collective Forest was again the best, with results ranging 95% and 99%.

V. Conclusions

Foreseeing the apparition of microshrinkages in ductile iron castings is one of the most hard challenges in foundry-related research. Our previous work in [3] pioneered the application of Artificial Intelligence to the prediction of microshrinkages. In this paper, our main contribution is the collective-classification-based approach employed for microshrinkage detection. This method does not require as much labelling of the castings as our previous supervised learning based approach. In our experiments the results were higher than the ones reported in our previous work using supervised learning [3], [13], which renders collective classification as the best learning procedure for microshrinkage prediction.

Future work will be focused on three main directions. First, we plan to extend our study of collective learning by applying more algorithms to this issue. Second, we will use different features for training these kinds of models. Finally, we will focus on different defects in foundry production in order to generate a global fault detector.

References

[1] "44^{th} census of world casting production," Modern Casting, Tech. Rep., 2010, monthly report, edited annually with the data concerning the number of casting houses and the world casting production in the year preceding the issue.

[2] 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.

[3] I. Santos, J. Nieves, Y. K. Penya, and P. G. Bringas, "Optimising machine-learning-based fault prediction in foundry production." in *Proceedings of the 2^{nd} 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.

[4] I. Santos, J. Nieves, , Y. K. Penya, and P. G. Bringas, "Towards noise and error reduction on foundry data gathering processes." in *Proceedings of the International Symposium on Industrial Electronics (ISIE)*, 2010, 1765—1770.

[5] A. Zabala, R. Suárez, and J. Izaga, "Iron castings, advanced prediction tools, foundry process control and knowledge management," in *Proceedings of the 68^th WFC - World Foundry Congress*, 2008, pp. 355–360.

[6] A. Lazaro, I. Serrano, J. Oria, and C. de Miguel, "Ultrasonic sensing classification of foundry pieces applying neuralnetworks," in 5^th *International Workshop on Advanced Motion Control*, 1998, pp. 653–658.

[7] 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.

[8] M. Perzyk and A. Kochanski, "Detection of causes of casting defects assisted by artificial neural networks," in *Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture*, vol. 217, no. 9, 2003, pp. 1279–1284.

[9] H. Bhadeshia, "Neural networks in materials science," *ISIJ international*, vol. 39, no. 10, pp. 966–979, 1999.

[10] T. Sourmail, H. Bhadeshia, and D. MacKay, "Neural network model of creep strength of austenitic stainless steels," *Materials Science and Technology*, vol. 18, no. 6, pp. 655–663, 2002.

[11] 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.

[12] 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.

[13] I. Santos, J. Nieves, and P. Bringas, "Enhancing fault prediction on automatic foundry processes," in *World Automation Congress (WAC), 2010*. IEEE, 2010, pp. 1–6.

[14] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*. MIT Press, 2006.

[15] J. Neville and D. Jensen, "Collective classification with relational dependency networks," in *Proceedings of the Workshop on Multi-Relational Data Mining (MRDM)*, 2003.

[16] I. Santos, C. Laorden, and P. G. Bringas, "Collective classification for unknown malware detection," in *Proceedings of the 6^{th} International Conference on Security and Cryptography (SECRYPT)*, 2011, pp. 251–256.

[17] C. Laorden, B. Sanz, I. Santos, P. Galán-García, and P. G. Bringas, "Collective classification for spam filtering," in *Proceedings of the 4^{th} International Conference on Computational Intelligence in Security for Information Systems (CISIS)*, 2011, pp. 1–8.

[18] S. Kalpakjian and S. Schmid, *Manufacturing engineering and technology*, 2005.

[19] C. J. F. and R. Ríos, "A fracture mechanics study of nodular iron," *Revista de Metalurgía*, vol. 35, no. 5, pp. 279–291, 1999.

[20] R. Gonzaga-Cinco and J. Fernández-Carrasquilla, "Mecanical properties dependency on chemical composition of spheroidal graphite cast iron," *Revista de Metalurgia*, vol. 42, pp. 91–102, March–April 2006.

[21] M. Hecht and F. Condet, "Shape of graphite and usual tensile properties of sg cast iron: Part 1," *Fonderie, Fondeur d'aujourd'hui*, vol. 212, pp. 14–28, 2002.

[22] P. Larrañaga, J. Sertucha, and R. Suárez, "Análisis del proceso de solidificación en fundiciones grafíticas esferoidales." *Revista de Metalurgia*, vol. 42, no. 4, pp. 244–255, 2006.

[23] J. Sertucha, R. Suárez, J. Legazpi, and P. Gacetabeitia, "Influence of moulding conditions and mould characteristics on the contraction defects appearance in ductile iron castings." *Revista de Metalurgia*, vol. 43, no. 2, pp. 188–195, 2007.

[24] "Inoculation alloy against microshrinkage cracking for treating cast iron castings," patent US 2005/0180876 A 1.

[25] J. Nieves, I. Santos, Y. K. Penya, S. Rojas, M. Salazar, and P. G. Bringas, "Mechanical properties prediction in high-precision foundry production." in *Proceedings of the 7^{th} IEEE International Conference on Industrial Informatics (INDIN 09)*, 2009, pp. 31–36.

[26] I. Santos, J. Nieves, Y. K. Penya, and P. G. Bringas, "Machine-learning-based mechanical properties prediction in foundry production," in *In Proceedings of ICROS-SICE International Joint Conference (ICCAS-SICE)*, 2009, pp. 4536–4541.

[27] I.Santos, J. Nieves, P. Bringas, and Y. Penya, "Machine-learning-based defect prediction in highprecision foundry production," in *Structural Steel and Castings: Shapes and Standards, Properties and Applications*, L. M. Becker, Ed. Nova Publishers, 2010, pp. 259–276.

[28] J. Nieves, I. Santos, and P. Bringas, "Overcoming data gathering errors for the prediction of mechanical properties on high precision foundries," in *World Automation Congress (WAC), 2010*. IEEE, 2010, pp. 1–6.

[29] J. Nieves, I. Santos, Y. K. Penya, F. Brezo, and P. G. Bringas, "Enhanced foundry production control," in *Proceedings of the 21st International Conference on Database and Expert Systems Applications (DEXA). Lecture Notes in Computer Science 6262, Springer-Verlag Berlin Heidelberg*, 2010, pp. 213–220.

[30] G. Namata, P. Sen, M. Bilgic, and L. Getoor, "Collective classification for text classification," *Text Mining*, pp. 51–69, 2009.

[31] S. Garner, "Weka: The Waikato environment for knowledge analysis," in *Proceedings of the New Zealand Computer Science Research Students Conference*, 1995, pp. 57–64.

[32] G. Holmes, A. Donkin, and I. H. Witten, "Weka: a machine learning workbench," August 1994, pp. 357–361.

[33] Y. Singh, A. Kaur, and R. Malhotra, "Comparative analysis of regression and machine learning methods for predicting fault proneness models," *International Journal of Computer Applications in Technology*, vol. 35, no. 2, pp. 183–193, 2009.