Collective Prediction of Ultimate Tensile Strength in High-precision Foundries

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

***Abstract*—Mechanical properties are the features that measure the ability of a metal to withstand several loads and tensions. Specifically, ultimate tensile strength is the force a material can resist until it finally breaks. This property is one of the variables controlled during the foundry process. The only way to examine this feature is to apply destructive inspections that make the casting invalid 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 of a certain value of ultimate tensile strength for 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

Foundry production manufactures a huge number of castings to be part of more complex systems e.g., the brake component of a car, the propeller of a boat, components of the wings of an aircraft and the trigger in a weapon and, therefore, foundry can be considered as one of the axis of current economy. Indeed, if one of the pieces is faulty, it can be detrimental to both individuals and businesses activities. The casting production or the foundry process is considered as one of the main factors influencing the development of the world economy. The 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.

The foundry process is subject to very strict safety controls in order to be sure of the quality of the manufactured castings because, as one may think, the tiniest defect may become fatal. Exhaustive production control and diverse simulation techniques [2] are applied for the assurance of failure-free foundry processes. Many of the techniques used can only be used once the casting production is finished. Hence, when a defective casting is detected, it must be remelted, which can be translated into a cost increment. In this paper, we focus on the so-called Ultimate Tensile Strength (UTS). This mechanical property is defined as the force a casting can resist until it breaks; i.e., the maximum stress any material can withstand when subjected to tension. Manufactured iron castings must assure certain value (or threshold) of UTS to pass the tests. Unfortunately, the only available approach to examine the UTS is destructive inspection which breaks the piece.

Machine-learning methods are being increasingly used to solve fault prediction problems. Artificial Neural Networks (ANN) have been widely applied in order to solve other problems of foundry process e.g., on the prediction of the ferrite number in stainless steel arc welds [3]. ANNs have also been used for classifying foundry castings [4], optimising casting parameters [5] and detection of causes of casting defects [6]. In a similar vein, there are successful experiments involving the k-nearest neighbour algorithm include fault detection of semiconductor manufacturing processes [7]. K-nearest neighbour algorithm and artificial neural networks have been applied for enhancing quality of steel [8]. Bayesian networks have been applied as previous methods in Bayesian neural networks methodology for forecasting the ferrite number in stainless steel [9] and as the base to establish the level of microshrinkage in iron castings [10].

These results encouraged us to tailor these approaches into our concrete problem domain. Therefore, in our previous work [11], [12], [13], we have tested the ability of several machine-learning classifiers for the prediction of mechanical properties. We used Bayesian networks, support vector machines, decision trees, artificial neural networks and support vector machines, to identify the best overall machine-learning classifier capable of predicting the value of UTS and to reduce the noise in the manual data-gathering process [14].

However, machine-learning classifiers (or supervised learning methods) require a high number of labelled castings for each of the classes (i.e., faulty and not-faulty castings) to train the different models. However, it is quite difficult to acquire this amount of labelled data for a real-world problem such as production control. To generate these data, a time-consuming process of analysis is mandatory that renders in a cost increment during the process.

Semi-supervised learning is a type of machine-learning technique specially useful when a limited amount of labelled data exists for each class [15]. In particular, collective classification [16] 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 [16], malware detection [17] or spam filtering [18].

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 the value of ultimate tensile strength. 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 invalid values of ultimate tensile strength using features extracted from the foundry production parameters as we did in previous work [11], [14].

Summarising, our main contributions in this paper are: (i) we describe how to adopt collective classification for detection of invalid values of ultimate tensile strength, (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 UTS and several other related mechanical properties. 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 Processes and Mechanical Properties

Several factors, for instance the extreme conditions in which it is performed, make the foundry process very complex. Starting from the raw material to the manufactured item, this procedure involves numerous stages, several of which may be performed in parallel. When it comes to iron ductile castings, this process presents the following phases:

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 [19]:

- **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.

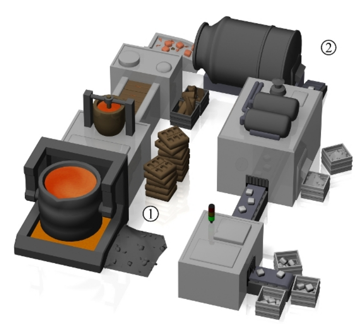


Fig. 1. Foundry process work flow showing the different phases castings have to pass through. More accurately, in 1 it is performed the metal melting step, and in 2 it is performed the casting preparation and separation step.

Once these phases finish, foundry materials are subject to forces (loads). Engineers calculate these forces and how the material deforms or breaks as a function of applied load, time or other conditions. It is important to know how mechanical properties affect to iron castings [20], because they directly affect the final quality of the manufactured casting. The most important mechanical properties of foundry materials are the following ones [21]: strength, hardness, resilience, elasticity, plasticity, brittleness, ductility and malleability. In this work, we focus on Ultimate Tensile Strength (UTS) that is a type of strength, which is the property that enables a metal to resist deformation under load. The testing method of UTS is conducted as follows (shown in Figure 2). First, a scientist prepares a testing specimen from the original casting. Second, the specimen is placed on the tensile testing machine. Finally, this machine pulls the sample from both ends and measures the force required to break the specimen apart and how much the sample stretches before breaking.

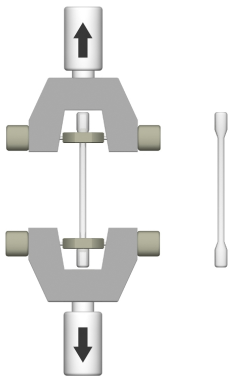


Fig. 2. Ultimate Tensile Strength Test.

The complexity of UTS prediction of the resulting castings arises mainly from the large number of variables involved in the production process and, therefore, this variables influence the final design of castings. The total number of variables we focus on has been reduced to 24, and more specifically, the control variables can be divided into metal-related variables and variables related to the mould.

- **Metal-related**
  - *Composition:* type of treatment, inoculation and quantities.
  - *Thermal:* Nucleation potential and quality of the mixture, obtained by thermal analysis [22].
  - *Pouring:* Pouring duration and temperature.
- **Mould-related**
  - Sand: types of additives used for sand, the specific characteristics of the sand.
  - Mould: mould and machine parameters used.

Generally, the size and geometry of the casting play a very important and, therefore, we also included several variables to monitor these features. Similarly, the system takes into account the parameters related to the configuration of each machine working in the manufacturing process. Also, we added other variables such as cooling rate and heat treatment applied to the piece.

Although we have already obtained overall good results using a machine-learning-based approach for predicting imperfections and mechanical properties [23], [11], [13], [24], [14], [25], [26], [27], these approaches require a manual labour to label every instance of the training dataset. This process can be specially time-consuming and, also, means a cost increment.

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 [28]. 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) [29] 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.

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

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

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

In order to evaluate our faulty casting detector, we collected a dataset from a foundry, which is specialised in safety and precisions 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 UTS prediction. Note that, as we have already mentioned, the only way to examine the mechanical properties is the employment of destructive inspections and, therefore, the evaluation must be performed after the production is done.

The acceptance/rejection criterion of the studied models resembles the one applied by the final requirements of the customer. Pieces flawed with an invalid UTS must be rejected due to the very restrictive quality standards (which is an imposed practice by the automotive industry). To this extent, we have defined two risk levels: *Valid* (more than 370 *MPa*) and *Invalid* (less than 370 *MPa*).

We worked with two different references, in other words, type of pieces and, in order to test the proposed method, with the results of the destructive inspections of the 889 production stocks performed in beforehand. More accurately, the dataset comprises 645 correct castings and 244 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 [30] (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 [29]. 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 an invalid value of UTS 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:

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

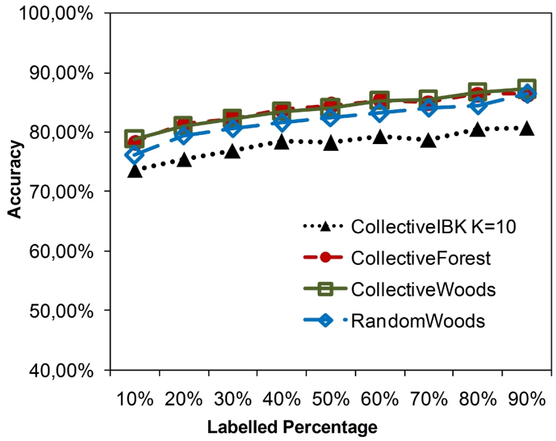
  where FP is the number of not faulty castings incorrectly detected as faulty and TN is the number of correct castings correctly classified.

  Furthermore, we measured *accuracy*, i.e., the total number of hits of the classifiers divided by the number of instances in the whole dataset:

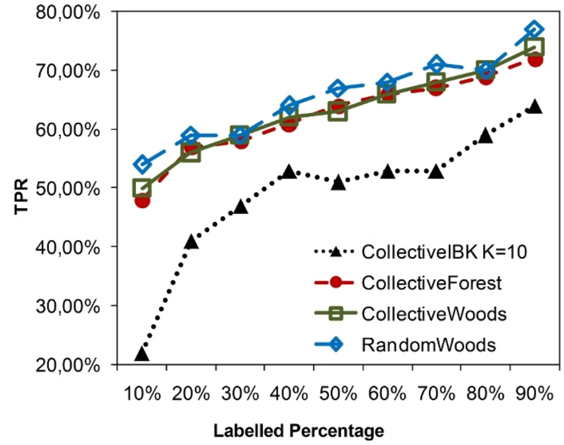
  $$Accuracy(\%) = \frac{TP + TN}{TP + FP + TP + TN} \tag{3}$$

  Besides, we measured the *Area Under the ROC Curve* (AUC), which establishes the relation between false negatives and false positives [31]. The ROC curve is obtained by plotting the TPR against the FPR. All these measures refer to the test instances.

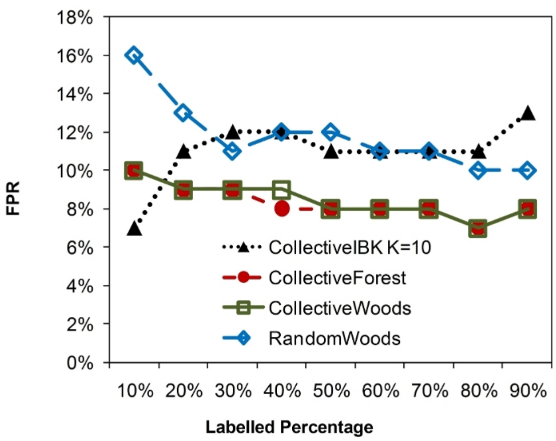
Fig. 3 shows the obtained results in terms of accuracy, TPR, FPR and AUC. Our results outline that, obviously, the higher the number of labelled castings in the dataset the better results achieved. However, by using only the 60% of the available data, with the exception of CollectiveIBK and RandomWoods,



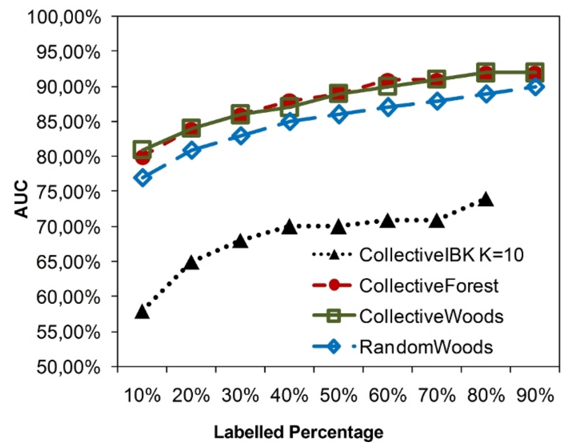
(a) **Accuracy results.** The accuracy axis (Y axis) has been scaled from 40% to 100% in order to appreciate better the evolution of the classifier. The classifiers were sensitive to the increase of the training dataset. With the exception of CollectiveIBK, the rest of the classifiers obtained accuracies higher than 85% using only a 60% of labelled instances.



(b) **TPR results.** CollectiveIBK was the worst classifier with a maximum TPR of 64% using the 90% of the dataset. The remainder of the classifiers obtained a detection rate of more than 65% with only a 60% of training size.



(c) **FPR results.** The FPR is scaled from 0.00% to 16% 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 15%.



(d) **AUC results.** The AUC axis (Y axis) has been scaled from 50% to 100% in order to appreciate better the evolution of the classifiers. As it happened with accuracy, CollectiveIBK was the worst classifier. Anyhow, the rest of the classifiers obtained AUC values higher than 80% using only a 20% of labelled instances.

Fig. 3. Results of our collective-classification-based UTS prediction method. Collective Woods was the overall classifier with the highest accuracy, TPR and AUC.

the collective classifiers CollectiveWoods and CollectiveForest were able to achieve TPRs higher than 65% and FPRs lower than 10%. In particular, Collective Forest trained with the 60% of the data obtained 85,26% of accuracy, 66,00% of TPR, 8.00% of FPR and 91% of AUC. Fig. 3(a) shows the accuracy results of our proposed method. All the tested classifiers, with the exception of CollectiveIBK, achieved accuracy results higher than 85% with some labelling percentage. In particular, CollectiveForest was overall the best, achieving an accuracy of 81,48% using only a 20% of the instances for training and 86,38%% with the 90% of the whole dataset. Fig. 3(b) shows the obtained results in terms of correctly classified faulty castings. In this way, Collective Woods was the best detecting from 50% to 74% of the faulty castings in different labelled percentage configurations. Fig. 3(c) shows the FPR results. Every classifier obtained results lower than 14%. In particular, the lowest FPR achieved was of 8%, achieved by CollectiveIBK with the 10% of dataset as well as Collective Forest and Collective Woods with the 80% of the dataset Finally, regarding AUC, shown in Fig. 3(d), Collective Forest was again the best, with results higher than 90% with a 60% of labelled data or higher.

V. Conclusions

Foreseeing the value of UTS in ductile iron castings is one of the most hard challenges in foundry-related research. Our work in [11], [13] pioneered the application of artificial intelligence methods to the prediction of the value of UTS.

In this paper, our main contribution is the collective-classification-based approach employed for UTS prediction 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 a little bit lower than the ones reported in our previous work using supervised learning [23], [25], which renders collective classification as the best learning procedure for UTS prediction, if we take into account the labelling reduction.

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] K. Schnelle and R. Mah, "Product quality management using a real-time expert system," *ISIJ International*, vol. 34, no. 10, pp. 815–821, 1994.

[9] 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.

[10] Y. Penya, P. García Bringas, and A. , Zabala, "Advanced fault prediction in high-precision foundry production," in *Proceedings of the 6^{th} IEEE International Conference on Industrial Informatics*, 2008, pp. 1673–1677.

[11] 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.

[12] J. Nieves, I. Santos, Y. K. Penya, F. Brezo, and P. G. Bringas, "Enhanced foundry production control," in *Proceedings of the 21^{st} International Conference on Database and Expert Systems Applications (DEXA)*, 2010, pp. 213–220.

[13] 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.

[14] 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.

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

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

[17] 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.

[18] 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.

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

[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] C. W. Lung and N. H. March, *Mechanical Properties of Metals: Atomistic and Fractal Continuum Approaches*. World Scientific Pub Co Inc, 1992.

[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] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

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

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

[31] 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.