Combination of Machine-Learning Algorithms for Fault Prediction in High-Precision Foundries.

Javier Nieves, Igor Santos, and Pablo G. Bringas

S^{3}Lab, DeustoTech Computing
University of Deusto
Bilbao, Spain
{jnieves,isantos,pablo.garcia.bringas}@deusto.es

Abstract. Foundry is one of the activities that has contributed to evolve the society, however, the manufacturing process is carried out in the same manner as it was many years ago. Therefore, several defects may appear in castings when the production process is already finished. One of the most difficult defect to detect is the *microshrinkage*: tiny porosities that appear inside the casting. Another important aspect that foundries have to control are the attributes that measure the faculty of the casting to withstand several loads and tensions, also called *mechanical properties*. Both cases need specialised staff and expensive machines to test the castings and, in the second one, also, destructive inspections that render the casting invalid. The solution is to model the foundry process to apply machine learning techniques to foresee what is the state of the casting before its production. In this paper we extend our previous research and we propose a general method to foresee all the defects via building a meta-classifier combining different methods and without the need for selecting the best algorithm for each defect or available data. Finally, we compare the obtained results showing that the new approach allows us to obtain better results, in terms of accuracy and error rates, for foretelling microshrinkages and the value of mechanical properties.

Key words: fault prediction, machine learning, meta-classification, process optimization.

1 Introduction

The manufacturing process is an important part of the current society. Thanks to it, consumers can have different products and services. Within the manufacturing process, the casting production or the foundry process is considered as one of the main factors that influences the development of the world economy. Thousands of castings are created in foundries composing complex systems. In fact, the actual capacity of the casting production of the world, which is higher than 60 million metric tones per year, is strongly diversified.

Due to current trends, it is really easy to produce castings and suddenly discover that every single one is faulty. The techniques for the assurance of failure-free foundry processes are exhaustive production control and diverse simulation

techniques [1] but they are extremely expensive and only achieve good results in an *a posteriori* fashion. These methods are also still incapable of preventing two of the most difficult targets to detect in ductile iron castings, i.e., *the microshrinkage* and *the mechanical properties*. The first one, also called secondary contraction, consists in tiny porosities that appear inside the casting when it is cooling down. For the second one: mechanical properties, we have selected the *ultimate tensile strength* that is the force, which a casting can withstand until it breaks, in other words, it is the maximum stress any material can withstand when subjected to tension.

The problem of foreseeing the apparition of both flaws is very difficult to solve [2–5] 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. One way to solve this problem is the employment of machine learning methods.

Currently, *machine-learning* classifiers have been applied in domains alike with outstanding results, for instance, for fault diagnosis [6], malware detection [7] or for cancer diagnosis [8]. Machine learning is being used increasingly in the field of metallurgy in several aspects such us classifying foundry pieces [9], optimising casting parameters [10], detecting causes of casting defects [11] amongst other related problems [12]. We have also applied these ideas and, we tested several machine-learning classifiers [2, 4, 5, 13–15] to identify which is the best classifier to predict microshrinkages and the ultimate tensile strength.

These classifiers, used as a stand-alone solution, are capable to predict several defects. But this process has some shortcomings such as: (i) we cannot be completely sure that the selected classifier is the best one to generalise the manufacturing process, (ii) the learning algorithms employed for creating some of the machine learning classifiers only find a local maximum and, hence, the final result is not optimal and (iii) by using a single classifier, we should generate a classifier close to the process nature (linear or non-linear). Combination of different classifiers can solve these problems. Firstly, it is more safe if we use all the classifiers instead of selecting one. Secondly, by combining different sub-optimal classifiers, we can approximate their behaviour to the optimal one. Finally, in this combination process we are able to select several classifiers building a linear meta-classifier (all of them are linear classifiers), non-linear (all of them are non-linear classifier) or hybrid (classifiers belong to both classes).

Against this background, we present here the first approach that employs a meta-classification technique, specifically, methods that allow us to combine several machine learning classifiers for categorising castings and to foresee microshrinkages and the ultimate tensile strength. These methods are able to learn from labelled data to build accurate classifiers that are going to share its knowledge under some rules. We propose the adoption of this method for the detection of microshrinkages and the ultimate tensile strength using features extracted from the foundry production parameters as we did before [2, 4, 5, 13–15].

2 High Precision Foundry

2.1 Foundry Process

A foundry is a factory where metal is melted and poured into containers specially shaped to produce objects such as wheels and bars. In other words, the factory in which metals are melt. Despite the fact that the process seems to be simply, the whole process become complex due to the hard conditions in which is developed. In this research we focus on foundries which produce castings that are close to the final product shape, i.e., 'near-net shape' components. To obtain the final casting, metals, in our case iron metals, have to pass through several stages in which raw materials are transformed. The most important stages are the following [16]:

- **Pattern making.** In this step, moulds (exteriors) or cores (interiors) are produced in wood, metal or resin in order to be used to create the sand moulds in which the castings are 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, 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 onto the sand mould. It can be done using various types of ladles or, in high volume foundries that generate small castings, automated pouring furnaces. Later, the metal begins to cool. This step is one of the most 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.

As aforementioned, to detect faulty castings and in order to know the behaviour of the casting to withstand several forces and loads, several tests are done when the casting is finished. The complexity of carrying out this process before doing it, i.e., using *ex-ante* methods, stems from the huge amount of variables to monitor along the whole foundry process and, therefore, the way in which these variables influence the final design of a casting. Consequently, we have simplified the manufacturing and the main variables to control in order to

foresee the faulty castings, and also features of the casting, can be classified into the following categories: (i)*metal-related* and (ii)*mould-related*.

- **Metal-related variables**
  - *Composition:* Type of treatment, inoculation and charges [17].
  - *Nucleation potential and melt quality:* Obtained by means of a thermal analysis program [18].
  - *Pouring:* Duration of the pouring process and temperature.
- **Mould-related variables**
  - *Sand:* Type of additives used, sand-specific features and carrying out of previous test or not.
  - *Moulding:* Machine used and moulding parameters.

Generally, 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. In addition, we took into account other parameters regarding the configuration of each machine working in the manufacturing process [19]. Finally, we can represent the castings with 24 different variables [2].

2.2 Microshrinkages

An irregularity in the casting is called a casting defect. When a defect appears, the casting must be corrected or, in the worst case, rejected. There are several defects that might arise along the foundry process and affect the metal [16].

Michroshrinkages is a kind of defect that usually appears during the cooling phase of the metal but it cannot be noticed until the production is finished. Particularly, 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. The reason of its apparition is that metals are less dense as a liquid than as a solid. And during the solidification process, the density of the metal increases while the volume decreases in parallel. In this way, diminutive, microscopically undetectable interdendritic voids may appear leading to a reduction of the castings hardness and, in the cases of high precision foundries (where the casting is a part of a very sensitive piece), this defect renders the piece useless [20].

The existing tests to detect microshrinkages use non-destructive inspections. The most widely techniques are the analysis via 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 supervised machine-learning-based approach predicting those imperfections [2, 13, 14], these approaches require to test several classifiers and identify which classifier fits to the foundry process. Moreover, if the research is not developed in an exhaustive manner or the selected learning methods only detect local maximums,

we may select a non-optimal classifier to foresee microshrinkages. In addition, as we show in our previous research, we do not select always the same classifier with the same configuration to predict every defect.

2.3 Mechanical Properties

When the foundry process is accomplished, the final casting is a part of a more complex system that will be subject to several forces (loads). During the design step, engineers calculate these forces and how the material deforms or breaks as a function of applied load, time or other conditions. And later, after the whole process, they select some specimens to test their actual behaviour. Therefore, it is important to recognise how mechanical properties influence iron castings [18]. Specifically, the most important mechanical properties of foundry materials are the following ones [21]: strength (there are many kinds of strength such as ultimate strength and ultimate tensile strength), hardness, toughness, resilience, elasticity, plasticity, brittleness, ductility and malleability.

To assure the performance of castings, there are common or standard procedures for measuring the mechanical properties of the materials in a laboratory. Unfortunately, the only way to know how the castings withstand the forces and loads and take measurements of the behaviour is employing destructive inspections. In addition, this complex process, like in microshrinkages tests, requires suitable devices, specialised staff and quite a long time to analyse the materials.

Regarding the ultimate tensile strength, on which we focus here on, its checking method is performed as follows. First, a scientist prepares a testing specimen from the original casting. Second, the specimen is placed on the tensile testing machine. Finally, the machine pulls the sample from both ends and measures the force required to pull the specimen apart and how much the sample stretches before breaking.

Moreover, the main variables to control in order to predict the mechanical properties of metals are the composition [17], the size of the casting, the cooling speed and thermal treatment [18]. In this way, the system should take into account all these variables to issue a prediction on those mechanical properties. Hence, our machine-learning models are composed of about 25 variables.

We developed several researches applying machine-learning-based classifiers with the aim of predicting these features [4, 5, 13, 15]. By carrying out this approach, foundries can reduce the cost of their quality tests because the destruction of the casting is no longer required. In our research, we obtained significant results that prove the plausibility of this technique. Nevertheless, as happened with microshrinkages, we cannot assure that the optimal classifier is included in the tested classifiers.

3 Combining Machine-Learning Classifiers

Classifiers by themselves are able to obtain good results, but we cannot ensure that a specific classifier is perfectly suitable for the prediction of every defect in

the foundry process. To solve this problem, several studies have been developed to combine classifiers [22]. These techniques seek to obtain a better classification decision despite of incorporating a higher degree of complexity to the process.

From a statistical point of view [23], assuming a labelled data set $\mathbf{Z}$ and the n number of different classifiers with relatively good performance making predictions for $\mathbf{Z}$, we can select one of them to solve classification problems, but there is a risk of not choosing the proper one. Therefore, the safest option is to use all of them and make an 'average' of their outputs. The resulting classifier is not necessarily better but will decrease or eliminate the risk induced because of the use of non appropriate classifiers.

From a computational point of view [22], some supervised machine-learning algorithms, in their learning phase, generate models based on local maximum solutions. Thus, an aggregation of classifiers is much closer to the optimal classifier than only one of them.

Similarly, the foundry process itself can be categorised into linear or nonlinear. By using these combination methods, we are capable of designing a collective intelligence system for classification which incorporates both linear and nonlinear classifiers.

The combination methods we used to develop the experiments are detailed below.

3.1 By Vote

The democracy for classifying elements is one of the oldest strategies for decision making. Extending the electoral theory, other methods can allow the combination of classifiers [24]:

- **Majority Voting Rule.** Assuming that the labelled outputs of classifiers are given as c-dimensional binary vectors $[d_{i,1}, ..., d_{i,c}]^T \in \{0,1\}^c, i = 1, ..., L$ where $d_{i,j} = 1$ if the classifier D_i categorises $\mathbf{x}$ in ω_j, or 0 otherwise. The plurality of the votes results in a set of classification for the class ω_k such as $\sum_{i=1}^{L} d_{i,k} = \max_{j=1}^{c} \sum_{i=1}^{L} d_{i,j}$. Regarding the problem of ties, these are solved arbitrarily.
- **Product Rule.** This second method takes into account the probabilities [24]. Thus, for the Product Rule, $p(x_1, ..., x_R|\omega_k)$ represents the joint probability distribution of the measurements taken from the classifiers. We assume that these representations are statistically independent. By including the Bayesian decision theory[24], the method assigns $Z \rightarrow \omega_j$ if $P^{-(R-1)}(\omega_j) \prod_{i=1}^{R} P(\omega_j|x_i) = \max_{k=1}^{m} P^{-(R-1)}(\omega_k) \prod_{i=1}^{R} P(\omega_k|x_i)$. The decision rule quantifies the probability of a hypothesis by combining the *a posteriori* probabilities generated by the classifiers. Indeed, this fusion rule is really hard because it may inhibit one of the outputs when the probability is close to 0.
- **Average Rule.** To obtain the Average Rule, we must start generating the Sum Rule to subsequently make a division employing the number of base

classifiers, R, as denominator [24]. Assuming that the *a posteriori* probabilities computed for each classifier are not derived from *a priori* probabilities, we obtain the Sum Rule in which we assign $Z \rightarrow \omega_j$ if $(1-R)P(\omega_j) + \sum_{i=1}^{R} P(\omega_j|x_i) = \max_{k=1}^{m}[(1-R)P(\omega_k) + \sum_{i=1}^{R} P(\omega_k|x_i)]$

- **Max Rule.** We start with the Sum Rule and obviate the product of *a posteriori* probabilities and assuming prior equalities, the method assigns $Z \rightarrow \omega_j$ if $\max_{i=1}^{R} P(\omega_k|x_i) = \max_{k=1}^{m} \max_{i=1}^{R} P(\omega_k|x_i)$
- **Min Rule.** For the Min Rule, starting with the Product Rule and obviating the product of *a posteriori* probabilities and assuming prior equalities, we will assign $Z \rightarrow \omega_j$ if $med_{i=1}^{R}\ P(\omega_j|x_i) = \max_{k=1}^{m} med_{i=1}^{R}\ P(\omega_k|x_i)$

3.2 Grading

The base classifiers are all the classifiers that we want to combine through the *grading* method [25] and these have been evaluated using k-fold cross-validation [26] ensuring that each of the instances has been employed for the learning phase of each classifier.

Formally, let p_{ikl} as the calculated class probability for each base classifier k for the class l and the instance i. To simplify the equations, we write P_{ikl} to refer to the vector $(p_{IK1}, p_{IK2}, ..., p_{ikn_l})$ of all probabilities for the instance i and the classifier k. In addition, the prediction of the base classifier k for i is the class L, p_{ikL}, is calculated by the maximum likelihood, in other words, $c_{ik} = argmax_l\{p_{ikl}\}$.

Moreover, *grading* builds n_c training datasets, one for each base classifier k, adding the predictions g_{ik} to the original data set as the new class. $prMeta_{ik}$ is the probability calculated by the meta-classifier of k that the base classifier k is going to correctly foresee the instance i. Regarding this information, the final estimated probability for the class l and the instance i, if there is, at least, one meta-classifier which indicates that its classifier is going to foresee the result in a correct manner (i.e., $prMeta_{ik} > 0.5$), is calculated as $prGrading_{il} = \sum\{prMeta_{ik}|c_{ik} = l \wedge prMeta_{ik} > 0.5\}$.

Therefore, the classification step is as follows [25]. First, each base classifier makes a prediction for the instance you want to foresee. Second, meta-classifiers qualify the result obtained by the base classifiers for the instance we are trying to classify. And, finally, the classification is derived using only the positive results. Conflicts (i.e., multiple classifiers with different predictions have got a correct result) can be solved using the *by vote* method or employing the estimated confidence for the base classifier.

3.3 Stacking

The *stacking* method [27] is another manner of combining classifiers that tries to improve the union based on cross-validation method.

Hence, we use several classifiers or generalisers. To learn these classifiers, we select a set of r partitions, each one divides θ (the training set) into two sets,

usually disjoint. We label the set of partitions as θ_{ij}, where $1 \leq i \leq r$ and $j \in \{1, 2\}$. Then, we define the space, in which these classifiers are, as the level 0 space. The classifiers use the original data set θ for the learning step.

Then, for each r_i partition of θ, $\{\theta_{i1}, \theta_{i2}\}$, we generate a set of k numbers. Typically, this k numbers can be: (i) the assumptions made by the original classifier or generaliser, (ii) the input component θ_{i2} or (iii) the vector in the input space which connects the component θ_{i2} to its θ_{i1} nearest neighbour. Subsequently, we take each group of k numbers as input component in a second space, *level 1 space*. Due to we have r partitions of θ, there are r points in the space of level 1. These points are known as the *reduced* or *level 1* training set for the level 1 classifiers.

For the classification process, firstly, we carry out a question to the classifiers in level 0 (original classifiers). Secondly, once we get the answer from all of them, we apply the transformations of k numbers that produce the input data set for the level 1 (this is the results transformation step). Thirdly, level 1 classifiers will derive the solution. And finally, the response is transformed back into the level 0 space to provide the final result. The whole process is known as *stacked generalisation* and can be more complex adding multiple stacking levels.

To apply the *stacking* method, we miss a set of rules such as (i) which classifiers should be selected at level 0, (ii) which ones at level 1 and which k numbers ought to be employed to generate the level 1 space [27].

3.4 Multischeme

This method is a meta-classification method implemented by Weka [28] which allows the combination of classifiers in a simple manner. This method employs a combination rule based on the results obtained by the cross-validation and the error rate measured as the mean square error from several classifiers.

The cross-validation is a simple way of mapping a classifier G with a set of training data θ and estimating the error rate of G when θ is generalised. More rigorously, multischeme method is performed by calculating the mean for each instance i and the error rate of G achieved predicting the output target related to the input data set θ_{i2} and when the learning phase is performed using the rest of θ, θ_{i1}. The error estimation is made through cross-validation as follows: $CV(G, \theta) \equiv (\sum_i [G(\theta_{i1}, \text{input of } \theta_{i2})(\text{output of } \theta_{i2})]^2)/m$

By using this measure, multischeme method is able to determine which classifier has to be taken into account to make the most precise classification.

4 Experimental results

To prove our hypothesis, we have collected data from a real 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. These experiments are focused exclusively on the prediction of the aforementioned targets: (i) microshrinkages and (ii) the ultimate tensile strength. Note that, as we have

already mentioned, the only way to examine both objectives is after the production is done.

Moreover, pieces flawed with a microshrinkage or an invalid ultimate tensile strength must be rejected because of the very restrictive quality standards (which is an imposed practice by the automotive industry). Therefore, regarding the acceptance/rejection criterion of the studied models, we defined several risk levels that resembles the one applied by the final requirements of the customer.

For microshrinkages, we defined the following 4 levels of risks: *Risk 0* (there is no microshrinkages), *Risk 1* (the probability of being microshrinkages is really low), *Risk 2* (there are some possibilities that the casting is flawed with a miscroshrinkage) and *Risk 3* (It is sure that the casting has a microshrinkage). In these 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 castings (note that each reference may involve several castings or pieces) performed in beforehand.

For the ultimate tensile strength, we have defined two risk levels: Risk 0 (more than 370 MPa[1]) and Risk 1 (less than 370 MPa). In these experiments, the machine-learning models have been built with the aforementioned 24 variables. We have worked with 11 different references and, in order to test the accuracy of the predictions, we have used as input data the results of the destructive inspection from 889 castings performed in beforehand. In spite of the fact that in our previous research we have examined this dataset with diverse sizes [4, 15], currently, we are interested in the accuracy level with the full original dataset since the foundry, we are collaborating with, always works with the whole dataset.

Specifically, we have conducted the next methodology in order to evaluate properly the combination of classifiers:

- **Cross-validation:** We have performed a *k-fold cross-validation* [26] with $k = 10$. In this way, our dataset is 10 times split into 10 different sets of learning (90% of the total dataset) and testing (10% of the total data).
- **Learning the model:** We have made the learning phase of each algorithm with each training dataset, applying different parameters or learning algorithms depending on the model. More accurately, we have use the same set of models that in our previous work [2, 4, 5, 13–15]: *Bayesian networks* (with K2, Hill Climber, Tree Augmented Naïve (TAN) as structural learning algorithms and Naïve Bayes), *K-Nearest Neighbour* (with values for k between 1 and 5), *Support Vector Machines* (with polynomial, normalised polynomial, radial basis function (RBF) and Pearson VII function-based kernels), Decision Trees (using random forests with different amount of trees (n),s $n = 50$, $n = 100$, $n = 150$, $n = 200$, $n = 250$, $n = 300$ and $n = 350$, and a J48 tree) and *Artificial Neural Networks* (specifically a MultiLayer Perceptron).

[1] MegaPascal, unit of pressure.

- **Learning the combination of the classifiers:** Once the machine-learning classifiers were built, we teach the different combination methods using the aforementioned models. More accurately, we tested the following combination methods:
  - *By vote:* There are several ways to combine the results of the classifiers by vote. For these experiments we have used *the majority vote rule* [22], *the product rule* [24], *the average rule* [24], *the max rule* [24] and *the min rule* [24].
  - *Grading:* This method needs a first level classifiers that have to assure that the predictions achieved by the original classifiers are correct. In this way, we have performed our experiments using the following first level classifiers: *a Naïve Bayes*, *Tree Augmented Naïve* and a *K-Nearest Neighbour* where $1 \leq k \leq 5$.
  - *Stacking:* For combining the original classifiers, *stacking* creates two separate spaces, in the first space, there are the original classifiers, and in the second one, there are several classifiers that derive the final result accordingly to the results achieved by the previous ones. To create the second space we have tested the following classifiers: *a Naïve Bayes*, *Tree Augmented Naïve*, a *K-Nearest Neighbour* with $k = 1$, $k = 2$, $k = 3$, $k = 4$ and $k = 5$; and a J48 decision tree.
  - *Multischeme:* This method combines the results using the cross-validation outputs and the error rates from the original classifiers. Thus, in this research we have used *multicheme* as it is.
- **Testing the model:** For each combination method, we have evaluated the percentage of correctly classified instances and the area under the Receiver Operating Characteristic (ROC) area that establishes the relation between false negatives and false positives [29]. We have decided to use the ROC area due to the realization that simple classification accuracy is often a poor metric for measuring the performance [30].

After applying the aforementioned methodology, we have obtained the following results. In order to facilitate the readability, we have divided the results by the classification target.

4.1 Microshrinkage

As we mentioned before, we have evaluated the meta-classifiers in terms of prediction accuracy and the area under the ROC curve. In this way, Table 1 illustrates the obtained results in terms of prediction accuracy. Using the full original dataset of 951 evidences, we can achieve a 94.47% of accuracy level. *Stacking built through a Tree Augmented Naïve* outperformed the rest of combination methods. On the one hand, the *stacking* method seems to be the best combination method due to the first three meta-classifiers was built using this technique. In addition, these three meta-classifiers achieved a similar result, more than a 94% of accuracy. The deviation between the majority of the meta-classifiers is really small (only 2.29 units). On the other hand, there are three meta-classifiers

that obtained an accuracy under the 75%. Those classifiers are based on *By Vote Rule*. Except for Majority Voting and Average Rule, this method is not adequate for the foundry process.

Table 1. Results in terms of accuracy and AUC predicting microshrinkages

Combination Method	**Accuracy** (%s)	**AUC**
Stacking (TAN)	94.47	0.9873
Stacking (Naïve Bayes)	94.12	0.9659
Stacking (KNN k=5)	94.12	0.9657
Grading (KNN k=5)	93.94	0.9077
MultiScheme	93.94	0.9820
Grading (KNN k=4)	93.93	0.9073
Grading (KNN k=3)	93.87	0.9062
Grading (J48)	93.86	0.9059
By Vote (Majority Voting Rule)	93.85	0.9071
Grading (Naïve Bayes)	93.81	0.9053
Grading (TAN)	93.78	0.9049
Stacking (KNN k=3)	93.72	0.9528
Stacking (KNN k=4)	93.66	0.9600
Grading (KNN k=2)	93.49	0.9017
Grading (KNN k=1)	93.48	0.9013
By Vote (Average Rule)	93.33	0.9820
Stacking (J48)	93.26	0.9145
Stacking (KNN k=2)	92.46	0.9420
Stacking (KNN k=1)	92.18	0.9021
By Vote (Max Rule)	73.29	0.9538
By Vote (Product Rule)	68.17	0.9076
By Vote (Min Rule)	68.17	0.9076

Notwithstanding, despite some Stacking meta-classifiers have achieved better accuracy levels than the Grading, Grading-based classifiers could achieve good results. Surprisingly, *MultiSheme*, one of the simplest method, is ranked in the fifth position. Hence, using this method (with an accuracy of 93.94%) or Majority Voting Rule (with an accuracy of 93.85%), other really simply method for combining classifiers, we can reduce the computational complexity while we maintain good results.

Table 1 also shows the area under the ROC curve (AUC). In this way, the obtained results in terms of AUC are similar to the ones of prediction accuracy. The *Stacking built through a Tree Augmented Naïve* also outperformed the rest of algorithms. More accurately, ROC analysis provides tools to select possible optimal models and to discard the suboptimal ones [29]. Therefore, if the results are closer to a value of 1 than to a value of 0, the classifier achieves a better performance because it obtains less amount of false positives and false negatives. In summary, although all of them accomplish acceptable values (they exceed the line of no-discrimination, in other words, more than a 0.90), Stacking based on a TAN classifier outshine the other classifiers achieving a 0.9873.

As it is shown in [23], these methods approximate the results achieved in our previous research using single classifiers [2, 13, 14]. Actually, there is one of them that overtakes the single classifiers. Therefore, we can conclude that the combination of classifiers can obtain (i) a good generalisation of the process

improving the accuracy and reducing the error rates; and (ii) reduction of the problem for selecting one single classifier.

4.2 Mechanical Properties

We have measured the same parameters for testing the combination methods in the prediction of the ultimate tensile strength, more particularly, the accuracy and the area under the ROC curve. Thus, table 2 shows the achieved results. In this case, the meta-classifier with the best performance was the *Grading method (using a Tree Augmented Naïve)*. In terms of accuracy, the classifier obtained an 86.63%. As we can deduct, the combination of classifiers also depends on the defect to be foreseen. For microshrinkages, the *stacking* method built through a TAN was the better, however, for the ultimate tensile strength, it is the worst.

Table 2. Results in terms of accuracy and AUC predicting ultimate tensile strength

Combination Method	**Accuracy** (%s)	**AUC**
Grading (TAN)	86.63	0.8101
MultiScheme	86.37	0.9171
Grading (J48)	86.34	0.8077
By Vote (Majority Voting Rule)	85.95	0.8005
Grading (Naïve Bayes)	85.91	0.7990
By Vote (Average Rule)	85.86	0.9146
Stacking (TAN)	85.73	0.9098
Grading (KNN k=5)	85.71	0.7977
Stacking (Naïve Bayes)	85.48	0.9094
Stacking (KNN k=5)	85.33	0.8778
Grading (KNN k=4)	85.23	0.7929
Grading (KNN k=3)	85.15	0.7922
Grading (KNN k=1)	85.05	0.7940
By Vote (Max Rule)	84.92	0.8968
Stacking (J48)	84.79	0.7996
Grading (KNN k=2)	84.77	0.7884
Stacking (KNN k=3)	84.17	0.8527
Stacking (KNN k=4)	83.69	0.8665
Stacking (KNN k=1)	81.43	0.7676
Stacking (KNN k=2)	79.57	0.8258
By Vote (Product Rule)	69.80	0.5994
By Vote (Min Rule)	69.80	0.5994

The *MultiScheme* and the *Grading* methods follow closely the best meta-classifier. For the second one, its first level classifiers are J48 decision trees. In terms of accuracy, the difference between the three methods does not exceed 0.29 units. Surprisingly, as in the microshrinkages, two very simple methods, such as MultiScheme and the Majority Voting Rule, are among the best meta-classifiers. Similarly, results illustrate us that the Product Rule and Min Rule are not able to make good predictions of the ultimate tensile strength. Moreover, although the classifiers are not ranked in the same position that in the microshrinkages, the overall behaviour is the same. All of them got very similar results, while three of them departed from the general behaviour.

Regarding the error rates, all of the classifiers obtained very similar results, but the best classifier is not the best dealing with false positives. Nevertheless, the

second one, the MultiSchema, is the best one. Thus, and because of the fact that the success rate is approximately the same, this could be the choice for predicting this feature. Another interesting aspect that we saw is that the best-performing classifier for microshrinkages problem is one of the best classifiers in terms of area under the ROC curve. Notwithstanding, the difference in accuracy with the best classifier is 0.9 units, hence, this method could be used for foreseeing both defects.

In this second experiment, the behaviour of a single classifier could not be improved. In our previous work [4, 5, 13, 15] we reach an accuracy of 86.84% using *Random Forests with 250 trees*. However, the difference is pretty small, hence, we can confirm that combining methods can approximate the results of the best classifier [23]. In addition, we do not care if the process is lineal or non-linear due to this method allow us to create a model that includes both type of classifiers. And finally, although it does not exceed the single classifiers, this method reduces the error rates, hence, we consider it as the best way for predicting the ultimate tensile strength.

5 Conclusions

On the one hand, microshrinkages are tiny porosities that appear when the casting is cooling down. On the other hand, ultimate tensile strength is the capacity of a metal to resist deformation when subject to a certain load. The prediction of the apparition of microshrinkages and the value of ultimate tensile strength renders as the hardest issues in foundry production, due to many different circumstances and variables that are involved in the casting process and determine it.

Our previous research [2, 4, 5, 13–15] pioneers the application of Artificial Intelligence to the prediction of these two features. Specifically in this paper, we have extended that model to the prediction via the combination of standalone classifiers. The majority of them behave well, but stacking built with a *Tree Augmented Naïve* for microshrinkages and grading also built with a TAN outperform the rest of the meta-classifiers. Moreover, the achieved results, in the case of microshrinkages, improve the classification done with a single classifier. On the other hand, as it is shown in [22], for the ultimate tensile strength, the meta-classifier approximates the previous results and reduces the error rates.

In addition, as we noticed in our previous work [4, 15, 2], 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, future work will be focused on 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 plan to integrate this meta-classifier, which will work as a black box combining all partial results to predict any defect, into a Model Predictive Control system in order to allow an hybrid prediction model. Third, we plan to employ some techniques (e.g., Bayesian

compression) to give more relevance to the newer evidences than to the older ones. The main objective is to develop a new method to quickly adapt the machine learning classifiers included in this meta-classifier. And, finally, we plan to test a preprocessing step to reduce the irregularities in the data.

References

1. Sertucha, J., Loizaga, A., Suárez, R.: Improvement opportunities for simulation tools. In: Proceedings of the 16th European Conference and Exhibition on Digital Simulation for Virtual Engineering. (2006) invited talk.
2. Santos, I., Nieves, J., Penya, Y.K., Bringas, P.G.: 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) 553–560
3. Santos, I., Nieves, J., , Penya, Y.K., Bringas, P.G.: Towards noise and error reduction on foundry data gathering processes. In: Proceedings of the International Symposium on Industrial Electronics (ISIE). (2010) in press.
4. Nieves, J., Santos, I., Penya, Y.K., Rojas, S., Salazar, M., Bringas, P.G.: Mechanical properties prediction in high-precision foundry production. In: Proceedings of the 7^{th} IEEE International Conference on Industrial Informatics (INDIN 09). (2009) 31–36
5. Nieves, J., Santos, I., Penya, Y.K., Brezo, F., Bringas, P.G.: 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) 213–220
6. Yang, J., Zhanga, Y., Zhu, Y.: Intelligent fault diagnosis of rolling element bearing based on svms and fractal dimension. Mechanical Systems and Signal Processing **1** (2007) 2012–2024
7. Santos, I., Penya, Y.K., Devesa, J., Bringas, P.G.: N-grams-based file signatures for malware detection. In: Proceedings of the 11^{th} International Conference on Enterprise Information Systems (ICEIS), Volume AIDSS. (2009) 317–320
8. Lisboa, P., Taktak, A.: The use of artificial neural networks in decision support in cancer: a systematic review. Neural Networks **19**(4) (2006) 408–415
9. Lazaro, A., Serrano, I., Oria, J., de Miguel, C.: Ultrasonic sensing classification of foundry pieces applying neuralnetworks. In: 5^th International Workshop on Advanced Motion Control. (1998) 653–658
10. Zhang, P., Xu, Z., Du, F.: 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) 545–548
11. M., P., A., K.: Detection of causes of casting defects assisted by artificial neural networks. In: Proceedings of the I MECH E Part B Journal of Engineering Manufacture. Volume 217. (2003)
12. Sourmail, T., Bhadeshia, H., MacKay, D.: Neural network model of creep strength of austenitic stainless steels. Materials Science and Technology **18**(6) (2002) 655–663
13. Santos, I., Nieves, J., Bringas, P.G., Penya, Y.K.: Machine-learning-based defect prediction in highprecision foundry production. In Becker, L.M., ed.: Structural

Steel and Castings: Shapes and Standards, Properties and Applications. Nova Publishers (2010) 259–276
14. Santos, I., Nieves, J., Bringas, P.G.: Enhancing fault prediction on automatic foundry processes. In: World Automation Congress (WAC), 2010, IEEE (2010) 1–6
15. Santos, I., Nieves, J., Penya, Y.K., Bringas, P.G.: Machine-learning-based mechanical properties prediction in foundry production. In: In Proceedings of ICROS-SICE International Joint Conference (ICCAS-SICE). (2009) 4536–4541
16. Kalpakjian, S., Schmid, S.: Manufacturing engineering and technology. Pearson Pentice Hall (2005)
17. F., C.J., Ríos, R.: A fracture mechanics study of nodular iron. Revista de Metalurgía **35**(5) (1999) 279–291
18. Gonzaga-Cinco, R., Fernández-Carrasquilla, J.: Mecanical properties dependency on chemical composition of spheroidal graphite cast iron. Revista de Metalurgia **42** (March–April 2006) 91–102
19. Sertucha, J., Suárez, R., Legazpi, J., Gacetabeitia, P.: Influence of moulding conditions and mould characteristics on the contraction defects appearance in ductile iron castings. Revista de Metalurgia **43**(2) (2007) 188–195
20. Margaria, T.: Inoculation alloy against micro-shrinkage cracking for treating cast iron castings. (November 13 2003) WO Patent WO/2003/093,514.
21. Lung, C.W., H. March, N.: Mechanical Properties of Metals: Atomistic and Fractal Continuum Approaches. World Scientific Pub Co Inc (July 1992)
22. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
23. Dietterich, T.: Ensemble methods in machine learning. Multiple classifier systems (2000) 1–15
24. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. Pattern Analysis and Machine Intelligence, IEEE Transactions on **20**(3) (1998) 226–239
25. Seewald, A., Fürnkranz, J.: An evaluation of grading classifiers. Advances in Intelligent Data Analysis (2001) 115–124
26. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence. Volume 14. (1995) 1137–1145
27. Wolpert, D.: Stacked generalization. Neural networks **5**(2) (1992) 241–259
28. Garner, S.: Weka: The Waikato environment for knowledge analysis. In: Proceedings of the New Zealand Computer Science Research Students Conference. (1995) 57–64
29. Singh, Y., Kaur, A., Malhotra, R.: Comparative analysis of regression and machine learning methods for predicting fault proneness models. International Journal of Computer Applications in Technology **35**(2) (2009) 183–193
30. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proceedings of the third international conference on knowledge discovery and data mining, Amer Assn for Artificial (1997) 43–48