

Accepted Manuscript

Support Vector Representation Machine for superalloy investment casting optimization

Carmen Del Vecchio, Gianfranco Fenu, Felice Andrea Pellegrino, Michele Di Foggia, Massimo Quatralè, Luca Benincasa, Stefania Iannuzzi, Alessandro Acernese, Pasquale Correrà, Luigi Glielmo

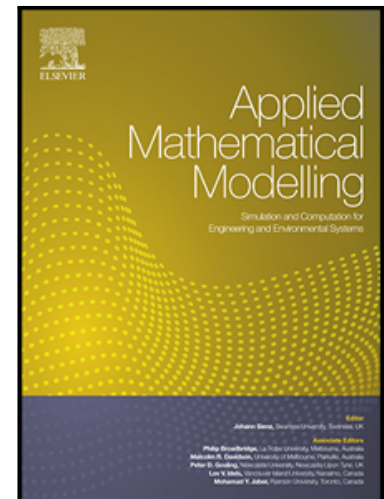
PII: S0307-904X(19)30118-0
DOI: <https://doi.org/10.1016/j.apm.2019.02.033>
Reference: APM 12686

To appear in: *Applied Mathematical Modelling*

Received date: 11 July 2018
Revised date: 9 February 2019
Accepted date: 25 February 2019

Please cite this article as: Carmen Del Vecchio, Gianfranco Fenu, Felice Andrea Pellegrino, Michele Di Foggia, Massimo Quatralè, Luca Benincasa, Stefania Iannuzzi, Alessandro Acernese, Pasquale Correrà, Luigi Glielmo, Support Vector Representation Machine for superalloy investment casting optimization, *Applied Mathematical Modelling* (2019), doi: <https://doi.org/10.1016/j.apm.2019.02.033>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



Highlights

- Original and innovative application of Machine Learning techniques to investment casting industry
- Key performance indicators to simultaneously increase quality and reduce defects in final products (i.e. turbine blades)
- Innovative solutions of Support Vector Representation Machine technique to define set-points of Key performance indicators

Support Vector Representation Machine for superalloy investment casting optimization[☆]

Carmen Del Vecchio^{a,*}, Gianfranco Fenu^b, Felice Andrea Pellegrino^b, Michele Di Foggia^c, Massimo Quatralè^c, Luca Benincasa^c, Stefania Iannuzzi^c, Alessandro Acernese^c, Pasquale Correrà^c, Luigi Glielmo^a

^aDepartment of Engineering, University of Sannio, Benevento, Italy

^bDepartment of Engineering and Architecture, University of Trieste, Italy

^cEuropea Microfusioni Aerospaziali S.p.A., Morra De Sanctis (AV), Italy

Abstract

Machine learning techniques have been widely applied to production processes with the aim of improving product quality, supporting decision-making, or implementing process diagnostics. These techniques proved particularly useful in the investment casting manufacturing industry, where huge variety of heterogeneous data, related to different production processes, can be gathered and recorded but where traditional models fail due to the complexity of the production process. In this study, we apply Support Vector Representation Machine to production data from a manufacturing plant producing turbine blades through investment casting. We obtain an instance ranking that may be used to infer proper values of process parameter set-points.

Keywords: Machine learning, investment casting, process optimization, key performance indicators

1. Introduction

Turbine blades are critical components in aeronautics and gas industry. Blades operate under extreme and complex environmental conditions; they are subjected to air dynamics and centrifugal forces causing tensile and bending stresses: for example, in

[☆]Partially supported by PON03PE-00111-01 "MATEMI" financed by the Italian Ministry for University and Research

*Corresponding author

Email address: c.delvecchio@unisannio.it (Carmen Del Vecchio)

aeroplane engines, turbines often reach 1050°C for several thousand hours [1, 2].

Turbine performance significantly depends on the shape and dimensions of blades as well as their material strength. To warrant resistance against intense mechanical loads, the manufacturing process must satisfy high quality requirements. Indeed, blades must be produced respecting stringent dimensional and geometrical tolerances as well as ensuring material integrity and mechanical properties (e.g. high temperature tensile strength, endurance strength and creep strength). To attain production specifications, blades are manufactured through the investment casting or “lost-wax” process often employed in the production of high quality and net-shaped complex parts [3, 4].

In the investment casting technique, a pattern of the desired blade shape, usually made of wax, is formed by injecting molten wax into a metallic die (see Figure 1). The wax mold is subsequently invested with ceramic or refractory slurry, which then solidifies to build a ceramic shell around the wax pattern. Wax is then removed from the shell by melting or combustion, leaving a hollow void within the shell, which exactly matches the blade shape. After the resulting refractory shell has been hardened by heating, the actual casting operation is performed filling the shell with molten alloy. After the molten metal solidifies, the shell is broken to obtain perfectly shaped components, that will finally be refined [5].

Investment casting is a high precision and very expensive production process. Major efforts have been made to reduce scraps in the shell building stage and enhance product quality (see for example [6]); less significant improvements have been achieved in casting operation, even though this production stage is characterized by a significant scrap rate—varying from 20% to 30% according to product characteristics—and determines most finite product mechanical characteristics [7, 8]. The lack of remarkable advancements aimed to enhance product quality by improved casting is due to the complex metallurgical phenomena arising in alloy injection and solidification, which have not been fully understood yet. Moreover, blade final characteristics depend on several different elements related to each production stage [9].

Scientific and industrial efforts to improve blade production have been mainly addressed to develop advanced casting simulation tools; these software reproduce molding stage and help to visualize several process variables, including flow of the molten metal

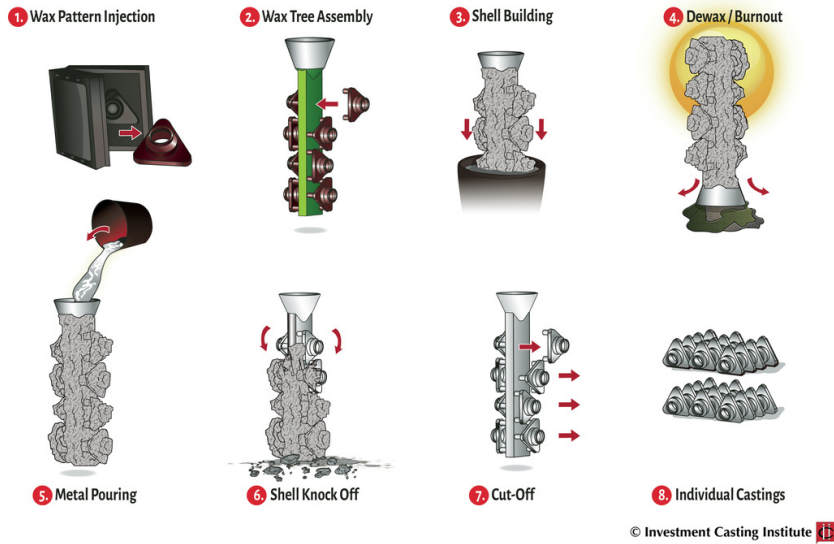


Figure 1: Main processes of investment casting

in the cavity, heat transfer, solidification, grain formation, shrinkage, and stress evolution [10]. However, simulation tools are unsuitable to correlate process variables (e.g. casting temperature at different heights in the furnace, casting pressure, alloy composition) to blade attributes—both defects and desirable qualities—hence they are not helpful in predicting final product quality from process parameter values. To the best of our knowledge, only few recent studies [11, 12] have tried to relate the performance of blade materials with parameter changes in the production process of superalloys and ceramic coatings. In [11] the author looks for possible correlations between product failures (typically blade rift) and possible causes such as structural defects of materials or mechanical fatigue. Our study instead aims at relating product characteristics (such as material structures obtained during alloy melting and solidification process) to production features. Reference [12] studies possible relations of a specific blade defect—the margin plate warpage deformation—to the solidification phase via numerical simulation analysis of deformation.

In an attempt to improve blade quality and reduce defects or scraps, we developed

a systematic strategy to determine the relationship between final product quality and casting process parameters. We studied and analyzed production data from Europea Microfusioni Aerospaziali (EMA), an investment casting foundry placed in Southern Italy producing turbine blades for jet-engines in civil and defence aerospace, marine and energy industry (see [13]). To achieve that goal, we relied on a machine learning technique, precisely the combination of a standard binary Support Vector Machine (SVM), and a one-class SVM called Support Vector Representation Machine (SVRM,[14]). Machine learning techniques have been widely applied to production processes with the aim of improving product quality, supporting decision-making, or implementing process diagnostics (see [15, 16] for an overview). SVM formulations (see for example [17] for the use of SVM in data mining applications) have often been successfully applied [18], mostly based on its scalability properties [19]. We chose SVMs because they are a theoretically sound tool, known to deal well with non linearly separable data while limiting overfitting [20]. Moreover, they are trainable in polynomial time by solving a convex quadratic programming problem. The choice of SVRM was motivated by the particular outcome of the SVM training, precisely the upper triangular form of the confusion matrix. Precisely, the outcome made clear that a subset of the data could be separated perfectly, hence a description of that subset could provide information about the meaningful features. The novelty of the paper is of methodological nature. Indeed, although the Support Vector Representation Machine is an existing tool, it is employed here in a novel way. In particular, rather than solving a classification/rejection problem [14], here the data are first classified using standard SVM and then a subset is ranked. The ranking criterion is the distance from an implicitly defined vector that acts as a kind of cluster center. The ranking is meaningful because the decision surface of the original classification problem leaves to one side points belonging to the same class, hence identifies a subset of the data that are “easy to classify”. Our machine learning approach proved particularly useful in the investment casting manufacturing industry, where huge variety of heterogeneous data, related to different production processes, can be gathered and recorded but where traditional models (i.e. continuous or discrete time models) fail due to the complexity of the production process. Moreover, we compared the results of SVRM with an alternative method—namely decision trees method—

aiming at inferring rules from data; the comparison of the two methodologies **show** the benefits deriving from the application of SVRM. Finally the developed methodology has been applied to EMA manufacturing data as part of a research project funded by Italian Ministry of University and Research.

The paper is organized as follows. In Section 2 we give a brief description of the overall data set characteristics and detail the analyzed EMA plant data; then, we describe the methodology adopted and customized to EMA production process. Results of data analysis are reported in Section 3. **In Section 4 we show how to apply to EMA data the decision trees method; we discuss the results obtained by applying this methodology and we compare them with the outcomes of SVRM.** Comments and concluding remarks can be found in Section 5.

2. Analysis of products: searching for relationships between process data and quality

Data to be measured in process plant analysis can be classified as follow:

- quality data, that is a set of data comprising measurements carried out during the conformity check, and related to attributes (quality or defect) of each monitored product. Therefore, for each statistical unit, the quality data will consist of information such as the compliance with dimensional tolerances, the presence or absence of defects (i.e. cracks, alloys grains undesirable structures), an assessment of the extent of non-compliance together with the maximum admissible threshold for the extent of the defect (if applicable).
- process data, that is all parameters detected during the production process. This set is composed of strictly technological variables, directly related to the materials used in the various stages (e.g. the chemical composition of the superalloy and the ceramic coating), or indirectly referring to the settings of the plants (for example the time trend of the temperature reference in the furnace);

These two sets of data have been the source from which we identified the relationship between process parameters and product attributes. In the analysis we performed for

EMA plant we focused on a specific product (referred to as *Part Number #A* (PN#A)) and collected production data over approximately 5 months, obtaining 1224 samples. A list of parameters related to the process has been provided by EMA (a pre-selection of the most significant has been made based on available process knowledge). These parameters, in the following referred to as “production features” or simply “features”, are shown in Table 1, along with a general brief description. The features of Table 1 are the sole features under analysis. Samples having complete data (i.e. each feature

Table 1: List of employed features.

symbol	feature description
T_{furn}	furnace temperature, acquired by means of a pyrometer
T_{x_1}	temperature value in the furnace, acquired using a probe at position P_{x_1}
Δ_{temp}	difference between T_{furn} and T_{x_1}
W_{load}	weight of the metal bar to be casted
SOAK	SOAK time interval (minutes)
N_{Probe}	casting activities, performed starting from the last furnace check-up
T_{x_2}	temperature value in the furnace, acquired using a probe at position P_{x_2}

has a value) are 1189. The analysis was carried out on these complete samples.

EMA’s objective is to relate some product attributes –here classified with codes A01, A02, A03, A04, A05, A06 – to production’s features, that is identify those production features that may be more likely responsible for product attributes. For the sake of simplicity from here on we will refer to the analyzed products attributes simply as “attributes”. Different attributes have different code. The attributes codes and their occurrence in the dataset are reported in Table 2. Some analysis are carried out per single attribute, some other by aggregating all the different attributes. In details, the occurrences of the A04-A06 attributes represent around 1 – 5 ‰ of the available

Table 2: Number of samples belonging to each attribute class; the percentages of each attribute class vs the whole amount respectively of labeled samples and vs the total number of samples showing attributes have been provided.

Label	n. of samples	% vs lab. samples	% vs overall samples
A01	40	27.027	3.364
A02	79	53.378	6.644
A03	20	13.514	1.682
A04	1	0.676	0.084
A05	2	1.351	0.168
A06	6	4.054	0.505
total	148		

samples, when each attribute is considered separately. Moreover, EMA process experts considered the first three attributes (namely A01, A02, and A03) more significant than the latter ones for the production process of the specific part number PN#A. Owing the EMA experts remark and the low occurrence rate of attributes A04-A06, it was decided to restrict the single attribute analysis to the three most frequent, namely A01, A02, A03. In the aggregate case, all attributes have been considered. Any description of product's attributes can't be given here due to nondisclosure agreement.

2.1. Methodology

The analysis is based on Statistical Learning tools [17, 20]. More specifically, it consists of two basic steps:

1. Designing and training of Support Vector Machine (SVM) classifiers;
2. instance ranking based on Support Vector Representation Machine.

The basic idea is that if a classifier can be trained to associate an attribute class to a feature vector with good performance, then:

- the features contain sufficient information for predicting the attributes;

- the structure of the classifier obtained can be exploited to characterize the feature vectors associated with the attribute.

In the following, we describe the steps 1 and 2 in more detail.

2.2. Designing and training of classifiers

By classifier we mean a function from the feature space to a finite set of classes. The function (called the decision function) maps a feature vector to an integer (defining the class):

$$f : \mathbb{R}^n \rightarrow \{0, 1, \dots\}$$

where n is the number of features. We only consider binary attributes, thus the classifiers take the form:

$$f : \mathbb{R}^n \rightarrow \{0, 1\}.$$

The classes 0 and 1 respectively represent the two possible values of a given attribute. In cases where the analysis is aggregated¹, we still use two classes: in such cases, the class 0 indicates that all the attributes take on zero value and class 1 that at least one attribute is non-zero. For simplicity, in the following we will say that “the attribute is present” rather than “the attribute takes on positive value”, and similarly we will say that “the attribute is absent” in place of “the attribute takes on zero value”.

Remark. *The proposed methodology may be applied to a set of binary attributes and results in a characterization of those instances of the training set that are likely not to possess any of the attributes. Thus, all the attributes are considered “non desirable” (for instance, they represent product defects). In the remainder of the paper we deal exactly with this case. Alternatively When some or all of the attributes are desirable, it is sufficient to properly re-define the binary attribute (if attribute A is desirable, define a new attribute $\tilde{A} \triangleq \text{not}(A)$ which is non desirable) and the same methodology still applies.*

¹namely, we consider more than one attribute

We consider a *supervised learning* setting, where a training set (TS) is assigned

$$\mathcal{T} = \{(x_i, y_i), i = \dots, N\}$$

namely a set of feature vectors x_i , each corresponding to a label y_i containing the attribute class. The *training* is the process by which, based on the training set, the decision function is chosen.

2.3. Type classifiers and their implementation

Among the possible classifiers it was decided to employ the Support Vector Machine (SVM) [21] due to the well-known capability of dealing with non linearly separable patterns while limiting the overfitting. Moreover, SVMs can be trained (in deterministic polynomial time) by solving a convex quadratic program. The SVM classifiers implement a decision function of the form

$$f(x) = \text{sign} \left(\sum y_i \alpha_i K(x_i, x) + b \right)$$

where the coefficients α_i and b are found during training and $K(x, y)$ is a kernel function acting as a generalized dot product. The kernel is a design parameters and its choice affects the shape of the resulting decision surface. In the case in which $K(x, y)$ is precisely the scalar product $\langle x, y \rangle$ between x and y , the classifier generates a linear decision surface (i.e. partitioning the feature space into two half-spaces delimited by a hyperplane). For general $K(x, y)$ the separation surfaces can "bend" and adapt to data that are not linearly separable. Preliminary experiments have been conducted with a linear SVM, getting poor performance in terms of accuracy (i.e. the ratio between instances correctly classified and the total evaluated instances). Thus, in the subsequent experiments a non-linear kernel has been used. Precisely, a Radial Basis Function kernel (RBF) has been used:

$$K(x, y) = \exp \left(-\gamma \|x - y\|^2 \right) \quad (1)$$

since it has been proven effective in a large number of applications [21, 22, 23]. As it is well-known, using a kernel amounts to implicitly define a map Φ between the feature space and a space (usually of larger size) in which a separating hyperplane is sought.

All trainings were performed in Matlab environment using the SVM implementation provided by the external tool LIBSVM [24], which is a free and well-established implementation. However, any SVM implementation providing binary classifiers with RBF kernels could be used. As a preprocessing, the features have been standardized, so that each feature exhibits zero mean and unit variance.

2.4. Selection of hyperparameters

The training of an SVM classifier consists in solving an optimization problem (precisely a convex quadratic programming problem). In the case of RBF kernel, such a problem, in addition to the training set, depends on two hyperparameters: the value γ appearing in (1), which defines the kernel, and a regularization parameter C , attributing a cost to instances of the training set that are incorrectly classified. In details, for each SVM classifier the choice of hyper-parameters was initially carried out by means of a 5-fold cross-validation [25, 26], by a grid search within a range of reasonable values. During this phase, the whole set of available data has been used, taking into account the data imbalance, as described in Sect. 2.5. The best hyper-parameters were those corresponding to the minimum value of the 5-fold cross-validation error. As a second step of training of the SVM classifiers, a further training of each SVM has been performed, selecting for every SVM machine the best hyper-parameters of the previous phase. Every SVM classifier has been trained using the whole set of data (see [26] Sect. 7.10 for a detailed analysis of such training approach).

2.5. Unbalanced training set

Whether the 3 attributes are considered individually or aggregated, the training set is *unbalanced*, i.e. it presents many more instances of a class (precisely, class 0) than the other. Among the various techniques available to cope with this problem (that may introduce a bias toward the most represented class), we adopted the SVM-WEIGHT method for its high effectiveness (see [27] for an overview of different approaches able to take into account the problem of class imbalance and for an analysis of SVM-WEIGHT performance). In [27] the authors suggest the SVM-WEIGHT method as the first choice especially when the cardinality of the available data-set is about few thousands

of samples, or less, for computational efficiency. The SVM-WEIGHT approach consists of a suitable modification of the cost function of the quadratic programming problem, by assigning different penalty values respectively to false negatives and to false positives. In particular, we decided to employ a higher penalty value in the case of false negatives than in the case of false positives. By using such a penalty values imbalance, we were looking for an SVM with the lowest number (possibly none) of false negatives.

2.6. Support Vector Representation Machine

Given a classifier with good performances in predicting the class of a product attribute, it may be desirable to analyze the classifier decision function to get some insight on how the feature values affect the prediction. Since the feature values represent production parameters and conditions, this information may be exploited to refine the process. In other words, the analysis of the decision function could help in answering the question “what combinations of feature values (i.e. production conditions) result, typically, in products with a given value of a given attribute?” However, the decision function may be rather complex (in the SVM case it is the weighted sum of many possibly non-linear functions) and going back from the prediction to the feature values may be difficult. Another possibility to gain insight useful to improve production process is to consider all those instances correctly classified and selecting a subset of these lying away from the decision boundary (in other words, instances that are easy to be classified correctly). These instances can then be analyzed by the production specialists. The method of Support Vector Representation Machine (SVRM) [21] allows to perform the mentioned selection. First we state two properties of the RBF kernel type:

- the vectors $\Phi(x)$ (where $\Phi(\cdot)$ is the feature map implicitly defined by the kernel) [21] have unit length

$$\langle \Phi(x), \Phi(x) \rangle = K(x, x) = e^0 = 1 \quad \forall x,$$

thus $\|\Phi(x)\|_2 = 1$ (recall that $K(x, y)$ is a dot product between $\Phi(x)$ and $\Phi(y)$)

- $\Phi(x)$ and $\Phi(y)$ lie in the same orthant $\forall x, y$, because

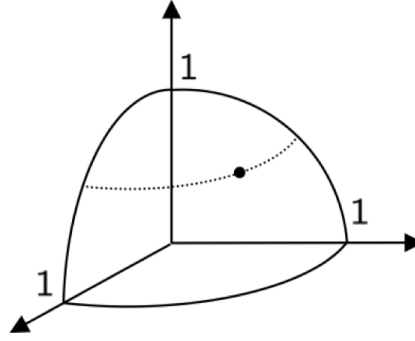


Figure 2: The map Φ maps the input vectors to the first orthant of the extended feature space

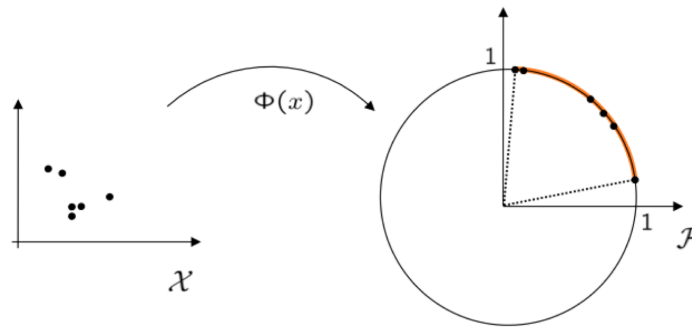


Figure 3: The arc of minimum length containing the images of all the training points.

$$\langle \Phi(x), \Phi(y) \rangle = K(x, y) > 0 \quad \forall x, y$$

The two properties guarantee that all the images of the input feature vectors according to the map Φ lie on the surface of the unit sphere of the extended feature space, and precisely in the first orthant, as shown in Fig. 2. The SVRM is a particular type of domain representation that uses the RBF kernel. The basic idea is to locate the smaller arc (or, more generally, the smaller solid angle) that contains all of the images of a given set \mathcal{A} of feature points (represented in orange in Fig. 3). This arc can be found as follows:

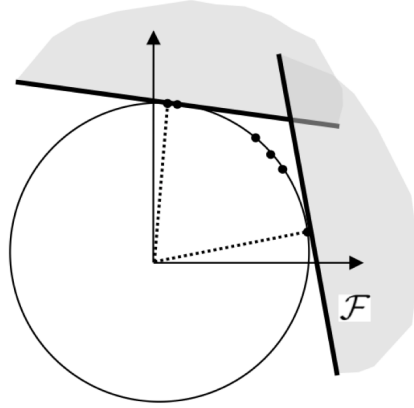


Figure 4: To each transformed input vector corresponds a tangent half-plane; their intersection is determined by the most external planes.

- find the vector of least Euclidean norm that satisfies the inequalities

$$\langle h, \Phi(x_i) \rangle \geq 1, \quad i = 1, \dots, N$$

- since images of points belonging to \mathcal{A} have unit norm:

$$\langle h, \Phi(x_i) \rangle = \|h\| \cos \angle(h, \Phi(x_i)), \quad i = 1, \dots, N$$

- as consequence we get constraints on the angles between the vector and each of the images:

$$-\pi/2 \leq \angle(h, \Phi(x_i)) \leq \pi/2, \quad i = 1, \dots, N$$

In other words we search for the vector of least Euclidean norm that lies in the intersection of all the half-planes (spaces) obtained as in Fig. 4, one for each point of \mathcal{A} . It is not difficult to guess that the solution h depends only on the most external image vectors (highlighted in orange in Fig. 5). It can be shown [21] that the vector sought may be written as

$$h = \sum_{i=1}^N \alpha_i \Phi(x_i)$$

where $\alpha = [\alpha_1, \alpha_2, \dots]$ is the solution of a quadratic programming problem, namely:

$$\max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)$$

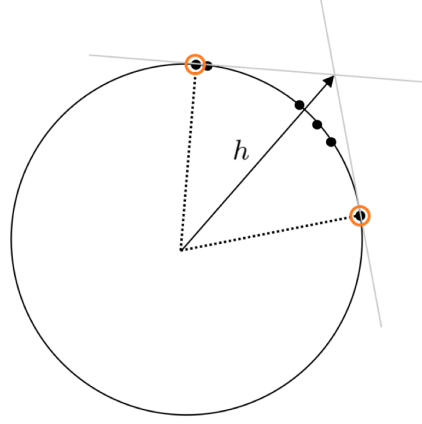


Figure 5: The vector h defines the arc of minimum length containing the images of all the input vectors, and depends on the most external images only.

subject to the constraints

$$0 \leq \alpha_i \leq C, \quad \forall i = 1, \dots, N$$

where C is a regularization parameter as in standard SVM. For obvious geometric reasons, a generic feature vector x belongs to the arc iff

$$\langle h, \Phi(x) \rangle \geq 1$$

which is equivalent to the following:

$$\sum \alpha_i K(x_i, x) \geq 1$$

The SVRM can be applied to the case at hand in the following manner:

1. Train an SVM to discriminate between the products with or without a given attribute on the basis of a certain TS
2. Take the subset \mathcal{A} of the TS containing the instances that are correctly classified as without attribute
3. Build SVRM using *the same kernel of SVM* and the subset \mathcal{A} , identifying in this way a vector h and a minimum arc length, as specified previously

4. Order the vectors of \mathcal{A} based on how close they are to the vector h . To this aim, the index

$$\sum \alpha_i K(x_i, x)$$

can be used, that precisely represents the scalar product in the feature space between the image of x and the vector h . This similarity criterion can be used because if the kernel is RBF, all images of any vector x have unit norm. So the index depends only on the angle between h and $\Phi(x)$, and increases with decreasing angle.

In Table 7 we report the first, tenth and fiftieth percentile of the instances ordered in the proposed way, using the “true negatives” obtained when discriminating between the presence or absence of any attribute (see the confusion matrix reported in Table 6). The feature values for these instances provide guidance for determining the set point of the various production parameters.

3. Results

3.1. Designing and training of SVM classifiers

As already mentioned, the attributes under analysis were A01, A02, A03. A total of 4 classifiers have been trained:

- one to discriminate between the presence or absence of A01;
- one to discriminate between the presence or absence of A02;
- one to discriminate between the presence or absence of A03;
- one to discriminate between the presence or absence of any attribute.

Each training was preceded by a selection of the γ and C hyper-parameters based on 5-fold cross validation. Tables 3-6 show the parameters and performances of the classifiers, evaluated on the training set (remember that the final aim of the work is to obtain a Support Vector Representation, that is a domain description and therefore it is not necessary to evaluate the performance on set different from the training set).

Table 3: Parameters and performance for the A01-dataset: data with label A01 belong to class P, whereas the other data belong to class N. Note the lack of misclassifications for data in the class P.

number of samples	1189		
class N	1149		
class P	40		
% P/(N+P)	3.3642		
γ	2.5000		
C	0.0410		
Accuracy	93.6922 % (1114/1189)		
Confusion matrix	predicted N	predicted P	
	actual N	1074	75
	actual P	0	40
precision	1.0000	0.3478	
recall	0.9347	1.0000	
F-measure	0.9663	0.5161	

3.2. Instance ranking via Support Vector Representation Machine

The confusion matrices reported in Tables 3-6 (where the rows mean actual values, and the columns are predicted values) show that there are no false negative. In other words, *there are no instances erroneously classified as without attribute, for any classifier*. The situation is shown in Fig. 6, in which the solid circles represent non-attribute instances and those empty instances with attribute, while the segment represents the separation surface. So it is reasonable to adopt the Support Vector Representation Machine to characterize the non-attribute instances that are correctly classified, that is to characterize the set highlighted in Fig. 7. In particular, the SVRM method was used to derive the first percentile of the most “central” instances of said set (in the sense already explained in Section 2.6). Precisely, for each instance without attribute and correctly classified as such, the following index was calculated

$$\sum \alpha_i K(x_i, x) \quad (2)$$

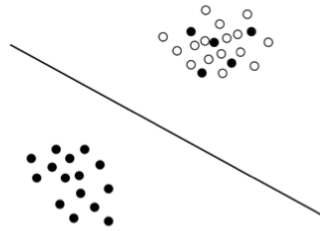


Figure 6: White and black dots represent attribute and non-attribute instances respectively. The decision function is such that all the instances with attribute are correctly classified.

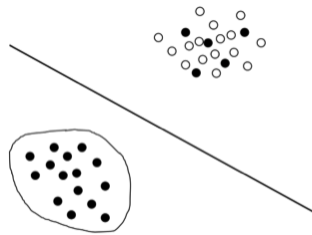


Figure 7: White and black dots represent attribute and non-attribute instances respectively. The selected set of non-attribute instances highlighted may be characterized by means of SVRM.

Table 4: Parameters and performance for the A02-dataset: data with label A02 belong to class P, whereas the other data belong to class N. Note the lack of misclassifications for data in the class P.

number of samples	1189	
class N	1110	
class P	79	
% P/(N+P)	6.6442	
γ	2.7000	
C	0.0810	
Accuracy	88.6459 % (1054/1189)	
Confusion matrix	predicted N	predicted P
	actual N	975 135
	actual P	0 79
precision	1.0000	0.3692
recall	0.8784	1.0000
F-measure	0.9353	0.5392

and the instances were sorted according to it, in descending order. In this way it was possible to obtain the first percentile (more precisely, the instances in the top 1%). By way of example, a comparison is made between the average values of the features for the first percentile (relative to the attributes A01, A02, A03) and the average values for the instances with attribute. The comparison indicates differences that do not seem negligible in some features (boldface in Tables 10-13). What is the most appropriate way to derive operational indications from these results (that are, basically, an instance ranking) is not fully clear, at the moment, because it is essential to have a thorough knowledge of the process. Reasonably, the production parameters should be set as close as possible to the values they take for the top ranked instances. Bearing in mind that the parameter setting for each instance should be considered as a whole, some insights for individual parameters can be drawn. For instance, Table 11 suggests that a low T_{x2} temperature is likely to reduce the occurrence of attribute A02, resulting

Table 5: Parameters and performance for the A03–dataset: data with label A03 belong to class P, whereas the other data belong to class N. Note the lack of misclassifications for data in the class P.

number of samples	1189	
class N	1169	
class P	20	
% P/(N+P)	1.6821	
γ	2.1000	
C	0.0810	
Accuracy	96.8881 % (1152/1189)	
Confusion matrix	predicted N	predicted P
	actual N	1132 37
	actual P	0 20
precision	1.0000	0.3509
recall	0.9683	1.0000
F-measure	0.9839	0.5195

in a “safety interval”. A “safety region” could be constructed when more than a single parameter is considered; this may be done, for instance, taking the convex hull of the parameters setting of the top ranked instances. However, further investigation is necessary, involving the production specialists, especially for dealing with possibly conflicting suggestions corresponding to different parameters/attributes.

3.3. Significance to EMA of SVRM results

We explicitly notice that EMA products’ attributes are undesirable characteristics (defects) thus the ultimate goal of our analysis was to suggest production parameters that may relate to them.

The discussion with EMA experts about SVRM results brought additional insights. From Table 2 it is clear that attributes A01 and A02 have a greater weight (i.e. they are more frequent) than others. Data in Tables 10 and 11 show that these attributes relate to the independent variables T_{x_2} and T_{furn} . Their measures refer to the process

Table 6: Parameters and performance of the classifier considering attributes altogether. Data with the presence of at least one attribute belong to class P, whereas the other data belong to class N. Note the lack of misclassification for data in class P.

number of samples	1189		
class N	1041		
class P	148		
% P/(N+P)	12.4474		
γ	2.0000		
C	0.2027		
Accuracy	79.7309 % (1041/1189)		
Confusion matrix	predicted N	predicted P	
	actual N	800	241
	actual P	0	148
precision	1.0000	0.3805	
recall	0.8691	1.0000	
F-measure	0.9839	0.5195	

where shells are filled with the molten alloy and in particular to the furnaces where the metal is poured. EMA experts responsible for the furnaces confirmed that according to their observation and practical experience both variables are relevant. In the past EMA personnel did perform some internal analysis aiming at relate attributes A01 and A02 to furnace parameters using standard multivariate statistical techniques (such as those reported in [28]); however they did not obtain any relevant correlation.

Moreover SVRM results lead to some suggestions that EMA staff could exploit for practical applications. According to Tables 10-11, the variable T_{x_2} is significant for the two attributes. Specifically, the best ranked negative instances have quite different mean temperature. However, Table 11 shows that non defective instances have a moderate value of T_{x_2} , suggesting that by controlling that temperature some benefits could be provided. EMA process experts consider such a control as feasible and implementable. Its potential benefits are remarkable: the decrease in the occurrences of A01 and A02

would lead to a reduction of scraps' cost. Referring scraps to the same production period when data have been analyzed, scraps costs related to non conforming products could be reduced up to 79% which translates in saving in production costs of 4% related to defect A01 and 7% for A02. EMA staff could either decide to make more specific analysis —as suggested in Section 3.2— or to start experiments on the production process to validate this study.

4. Comparison with alternative approaches

From a practical standpoint, the proposed methodology aims at inferring, from historical data, suitable values of production parameters in order to avoid some undesirable attributes. Thus, in principle, any technique aiming at inferring rules from data can be considered as an alternative. Decision trees are recognized as “simple” and “useful for interpretation” [17] thus seem an adequate approach for the problem at hand; in that case, indeed, interpreting the structure of the classifier is instrumental to getting the set point values of the production parameters. **As a first comparison, in Table 8 we report the performance, in terms of AUC score, of the SVM classifiers and those of decision tree classifiers. The latter have been trained to detect separately the presence or the absence of the attributes A01, A02, A03, based on the same features of the SVMs.** We have used the porting of the CART algorithm [29] into the Matlab framework (namely the *fitctree* function), adopting a 10-fold cross validation based optimization of the decision tree parameters. Moreover, with the aim of taking into account the class unbalancing, we set a proper misclassification cost matrix, for the decision tree to be trained, employing a weighting approach similar to that applied in the SVM cases (misclassification cost inversely proportional to the number of elements belonging to a particular class). Notice, by comparing AUC scores, that the performance of the decision trees are similar to those of the SVM, although slightly worst. We don't report the whole set of confusion matrices of the decision tree classifiers, but it's worth noting that, as for the SVM, the confusion matrices are upper triangular, confirming that a subset of the instances not carrying the considered attribute are separated from all the remaining instances by the decision surface.

To discuss the possibility of exploiting the decision tree for inferring parameter values, we take, as an example, the attribute A02. Using the training procedure described above, we obtained a decision tree whose confusion matrix is reported in Table 9. Notice, by comparing to the confusion matrix of Table 4, that the performance of the decision tree are close to those of the SVM; moreover, the confusion matrix is upper triangular, as in the SVM case. In other words, the situation is again the one represented in Fig. 6. **As for the interpretation of the obtained classifier, a graphical representation of the decision tree is provided in the supplementary material; the tree has 24 levels, 187 nodes and 94 leaves, thus suitable ranges of production parameters can't be inferred by simply inspecting the decision tree (or at least this would require a lot of time).** Moreover, at training step much more complex trees could be obtained. Thus, the decision tree alone cannot be considered as a systematic approach for this kind of problems, because there is no guarantee that the resulting tree will be human-readable. As a consequence, general methods based on decision trees must encompass some further step of analysis of the decision tree obtained in the first place. A first possibility is to follow [30] i.e. formulating an *inverse classification problem* where the features of an incomplete set are sought in such a way that will result in a desired classification outcome. In [30], the inverse classification problem is cast as the relaxation of a combinatorial problem and requires the discretization of the feature domain and the training of another decision tree. We do not go any further in this direction here but we stress that the approach requires a considerable implementation effort. Another possibility could be performing an instance ranking, to find the “most central” feature, i.e. the same approach we propose here by means of SVRM. In the case of decision trees, however, an appropriate ranking criterion is far from clear, since there is no notion of “distance from the decision surface”. Intuition could suggest, as a ranking criterion, the number of decision nodes to be evaluated for classifying a given instance: the more the nodes, the more difficult is the instance to be classified and therefore it is closer to the decision boundary. However, this heuristic seems not solid, since the number of nodes can vary, for a given instance, depending on some arbitrary and equivalent choices in the tree construction (different trees may lead to the same classifier). On the contrary, using the approach proposed in this paper, it is only necessary to compute, for each instance, the value (2) which is an

elementary computation, based on the kernel and the support vectors obtained as the output of the training phase.

5. Conclusion

We have presented an analysis, based on the Support Vector Representation Machine approach, which aims to characterize a set of instances belonging to a class of interest. That characterization is expressed in terms of an inequality that the instances must satisfy. The instances are ordered based on their distance from the failure to satisfy the inequality, and subsequently some statistics are computed on the percentiles thus obtained (the first percentile being the one farthest from the boundaries of the considered set). The proposed methodology can be applied when a classifier exhibiting zero false negative can be trained with the given data, meaning that a subset of instances that do not possess a given attribute can be discriminated with no error. The underlying idea is that those instances that are easily classified as not possessing an attribute, may suggest a production parameter setting likely not to produce that attribute. The study reported is intended as preliminary with respect to an analysis involving experts in the production process to derive operational indications (in terms of set-points of certain process parameters) to increase the presence of the attribute in case it is desirable or to decrease it in the opposite case. Such a further analysis is matter of future work.

Acknowledgment

The authors are indebted to Francesco Fabiano for his precious help during the data acquisition.

The research whose results have been reported in this manuscript was partially supported by PON03PE-00111-01 "MATEMI" financed by the Italian Ministry for University and Research

References

- [1] A. Krimpenis, P. Benardos, G.-C. Vosniakos, A. Koukouvitaki, Simulation-based selection of optimum pressure die-casting process parameters using neural nets

- and genetic algorithms, *The International Journal of Advanced Manufacturing Technology* 27 (5-6) (2006) 509–517.
- [2] A. S. Sabau, Alloy shrinkage factors for the investment casting process, *Metallurgical and Materials Transactions B* 37 (1) (2006) 131–140.
- [3] M. P. Boyce, *Gas turbine engineering handbook*, Elsevier, 2011.
- [4] N. Kumar, C. Besant, M. Ristic, Process variables determining the dimensional and metallurgical properties of directionally solidified, cored turbine blades produced by investment casting of super-alloys, *The International Journal of Advanced Manufacturing Technology* 3 (2) (1988) 3–19.
- [5] S. Kalpakjian, S. Schmid, *Manufacturing processes for engineering materials–5th edition*, agenda 12 (2014) 1.
- [6] A. S. Sabau, Alloy shrinkage factors for the investment casting process, *Metallurgical and Materials Transactions B* 37 (1) (2006) 131–140.
- [7] J. Aguilar, A. Schievenbusch, O. Kätzlitz, Investment casting technology for production of TiAl low pressure turbine blades–process engineering and parameter analysis, *Intermetallics* 19 (6) (2011) 757–761.
- [8] S. Pattnaik, D. B. Karunakar, P. Jha, Developments in investment casting process – A review, *Journal of Materials Processing Technology* 212 (11) (2012) 2332 – 2348. doi:<https://doi.org/10.1016/j.jmatprotec.2012.06.003>.
- [9] M. James, Design, manufacture and materials; their interaction and role in engineering failures, *Engineering Failure Analysis* 12 (5) (2005) 662–678.
- [10] M. Khan, A. Sheikh, A comparative study of simulation software for modelling metal casting processes., *International Journal of Simulation Modelling (IJSIMM)* 17 (2).
- [11] M. Durand-Charre, *The microstructure of superalloys*, Routledge, 2017.

- [12] F. Qiu, K. Bu, J.-H. Song, G.-L. Tian, X.-D. Zhang, Dimensional control of nickel-based single crystal turbine blade investment casting by process control optimization, *International Journal of Metalcasting* (2017) 1–11.
- [13] Europea Microfusioni Aerospaziali, Europea Microfusioni Aerospaziali, <https://emaht.com/>, accessed: 2018-07-06.
- [14] C. Yuan, D. Casasent, Support vector machines for class representation and discrimination, in: *Proceedings of the International Joint Conference on Neural Networks*, 2003., Vol. 2, 2003, pp. 1611–1616 vol.2. doi:10.1109/IJCNN.2003.1223940.
- [15] B. Esmailian, S. Behdad, B. Wang, The evolution and future of manufacturing: A review, *Journal of Manufacturing Systems* 39 (2016) 79–100.
- [16] G. Köksal, İ. Batmaz, M. C. Testik, A review of data mining applications for quality improvement in manufacturing industry, *Expert systems with Applications* 38 (10) (2011) 13448–13467.
- [17] G. James, D. Witten, T. Hastie, R. Tibshirani, *An introduction to statistical learning*, Vol. 112, Springer, 2013.
- [18] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al., Top 10 algorithms in data mining, *Knowledge and information systems* 14 (1) (2008) 1–37.
- [19] I. W. Tsang, J. T. Kwok, P.-M. Cheung, Core vector machines: Fast svm training on very large data sets, *Journal of Machine Learning Research* 6 (Apr) (2005) 363–392.
- [20] V. Vapnik, *The nature of statistical learning theory*, Springer science & business media, 2013.
- [21] S. Abe, *Support vector machines for pattern classification*, Vol. 2, Springer, 2005.
- [22] P. Culverhouse, R. Williams, B. Simpson, C. Gallienne, B. Reguera, M. Cabrini, S. Fonda-Umani, T. Parisini, F. Pellegrino, Y. Pazos, et al., Hab buoy: a new

instrument for in situ monitoring and early warning of harmful algal bloom events, *African Journal of Marine Science* 28 (2) (2006) 245–250.

- [23] G. Pin, M. Filippo, F. A. Pellegrino, G. Fenu, T. Parisini, Approximate model predictive control laws for constrained nonlinear discrete-time systems: analysis and offline design, *International Journal of Control* 86 (5) (2013) 804–820.
- [24] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, *ACM transactions on intelligent systems and technology (TIST)* 2 (3) (2011) 27.
- [25] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern classification*, John Wiley & Sons, 2012.
- [26] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, 2nd Edition, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2009, corrected 12th printing, 2017.
- [27] Y. Tang, Y.-Q. Zhang, N. V. Chawla, S. Krasser, Svms modeling for highly imbalanced classification, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39 (1) (2009) 281–288.
- [28] D. C. Montgomery, *Introduction to statistical quality control*, John Wiley & Sons (New York), 2009.
- [29] L. Breiman, J. Friedman, R. A. Olshen, C. J. Stone, *Classification and regression trees*, Chapman and Hall/CRC, 1984.
- [30] C. C. Aggarwal, C. Chen, J. Han, On the inverse classification problem and its applications, in: *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, IEEE, 2006, pp. 111–111.

Table 7: Complete set of attributes: min., max. mean and median value of each feature, for data in the first, tenth, fiftieth percentile when considering the whole subset of data correctly classified as “without any attribute”, i.e. the true negatives in Table 6.

minimum value							
percentile	T_{x_1}	T_{x_2}	T_{furn}	Δ_T	W_{load}	$SOAK$	N_{Probe}
1	1455	1304	1415	40	2.43	21	6
10	1454	1283	1323	131	2.35	21	4
50	1454	1283	1313	141	2.34	21	1
100	1454	1246	1310	144	2.25	18	1
maximum value							
percentile	T_{x_1}	T_{x_2}	T_{furn}	Δ_T	W_{load}	$SOAK$	N_{Probe}
1	1455	1313	1415	40	2.44	21	35
10	1455	1354	1439	16	2.49	22	38
50	1455	1354	1441	14	2.49	22	39
100	1455	1354	1441	14	2.58	31	40
mean value							
percentile	T_{x_1}	T_{x_2}	T_{furn}	Δ_T	W_{load}	$SOAK$	N_{Probe}
1	1455.000	1309.625	1415.000	40.000	2.436	21.000	17.125
10	1454.888	1312.112	1405.875	49.013	2.433	21.075	17.875
50	1454.932	1305.763	1406.135	48.797	2.425	21.050	17.015
100	1454.849	1305.311	1394.156	60.693	2.424	21.406	16.282
median value							
percentile	T_{x_1}	T_{x_2}	T_{furn}	Δ_T	W_{load}	$SOAK$	N_{Probe}
1	1455	1313	1415	40	2.44	21	7
10	1455	1304	1415	40	2.44	21	14
50	1455	1304	1415	40	2.43	21	16
100	1455	1298	1413	42	2.43	21	15

Table 8: AUC scores of SVM and decision tree classifiers on the A01, A02 and A03 datasets.

	A01 dataset	A02 dataset	A03 dataset
SVM classifier	0.967363	0.939189	0.984175
Decision tree	0.945170	0.920270	0.970488

Table 9: Confusion matrix of the decision tree classifier on the A02 dataset.

Confusion matrix	predicted N	predicted P
actual N	933	177
actual P	0	79

Table 10: Mean value of each feature when only the top 1% of “no attribute A01” data has been considered versus mean value of features when considering the whole set of no-A01 labelled data (i.e. the true negatives in Table 3).

mean value (no-A01)							
	T_{x_1}	T_{x_2}	T_{furn}	Δ_T	W_{load}	$SOAK$	N_{Probe}
top 1 %	1455.000	1315.455	1414.091	40.909	2.429	21.000	25.364
whole set	1454.804	1313.628	1391.203	63.601	2.425	21.196	17.493

Table 11: Mean value of each feature when only the top 1% of “no attribute A02” data has been considered versus mean value of features when considering the whole set of no-A02 labelled data (i.e. the true negatives in Table 4).

mean value (no-A02)							
	T_{x_1}	T_{x_2}	T_{furn}	Δ_T	W_{load}	$SOAK$	N_{Probe}
top 1 %	1454.700	1298.400	1389.200	65.500	2.420	21.300	33.000
whole set	1454.804	1313.628	1391.203	63.601	2.425	21.196	17.493

Table 12: Mean value of each feature when only the top 1% of “no attribute A03” data has been considered versus mean value of features when considering the whole set of no-A03 labelled data (i.e. the true negatives in Table 5).

mean value (no-A03)							
	T_{x_1}	T_{x_2}	T_{furn}	Δ_T	W_{load}	$SOAK$	N_{Probe}
top 1 %	1455.000	1335.750	1411.500	43.500	2.435	21.000	16.750
whole set	1454.804	1313.628	1391.203	63.601	2.425	21.196	17.493

Table 13: Mean value of each feature when only the top 1% of all data corresponding to “no any attribute” has been considered versus mean value of features when considering the whole set of production data correctly classified as “without any attribute” (i.e. the true negatives in Table 6).

mean value (no attributes)							
	T_{x_1}	T_{x_2}	T_{furn}	Δ_T	W_{load}	$SOAK$	N_{Probe}
top 1 %	1455.000	1309.625	1415.000	40.000	2.436	21.000	17.125
whole set	1454.804	1313.628	1391.203	63.601	2.425	21.196	17.493