

# Support Vector Machines for Quality Monitoring in a Plastic Injection Molding Process

Bernardete Ribeiro, *Member, IEEE*

**Abstract**—Support vector machines (SVMs) are receiving increased attention in different application domains for which neural networks (NNs) have had a prominent role. However, in quality monitoring little attention has been given to this more recent development encompassing a technique with foundations in statistic learning theory. In this paper, we compare  $C$ -SVM and  $\nu$ -SVM classifiers with radial basis function (RBF) NNs in data sets corresponding to product faults in an industrial environment concerning a plastics injection molding machine. The goal is to monitor in-process data as a means of indicating product quality and to be able to respond quickly to unexpected process disturbances. Our approach based on SVMs exploits the first part of this goal. Model selection which amounts to search in hyperparameter space is performed for study of suitable condition monitoring. In the multiclass problem formulation presented, classification accuracy is reported for both strategies. Experimental results obtained thus far indicate improved generalization with the large margin classifier as well as better performance enhancing the strength and efficacy of the chosen model for the practical case study.

**Index Terms**—Fault detection and diagnosis, kernel learning methods, model selection, radial basis function (RBF) neural networks (NNS), support vector machines (SVMs).

## I. INTRODUCTION

SUPPORT vector machines (SVMs) are a new learning-by-example paradigm spanning a broad range of classification, regression, and density estimation problems. This systematic approach motivated by statistical learning theory led to a class of algorithms characterized by the use of kernels, the absence of local minima, the sparseness of the solution and the capacity control obtained by acting on the margin. They combine ideas from various scientific branches such as mathematical programming, exploiting the quadratic programming for convex optimization, functional analysis, indicating adequate methods for kernel representations, and machine learning theory, exploring the large maximum classifiers concept [1]. They were first introduced by Vapnik and co-workers and are described in more detail in [2] and [3]. The roots of this approach, the so-called support vector (SV) methods of constructing the optimal separating hyperplane for pattern recognition, were already presented and had been used in machine learning in [4]. The SV technique was generalized for nonlinear separating surfaces in [5], and it was further extended for constructing decision rules in the non separable case [6]. The training task involves optimization of

a convex cost function conveying to a technique without local minima. Other benefits include the construction of a learning model which has an explicit dependence of the most informative patterns in the data (the SV). Advances in kernel methods [7] and learning theory [8] allowed generalizations of the technique. Extensions of the SVM's concept have been made by several authors as, for example, in [9]. Recently, a number of practical applications of SVMs have been reported [10]–[13].

In this paper, SVMs are applied within the framework of an industrial problem for fault detection and diagnosis in an injection molding process (IMP). Inherent complexities and nonlinearity of the process make mathematical modeling difficult, hindering the use of conventional methods for process modeling and condition monitoring. Neural networks (NNs) have successfully been applied in industry. Concerning injection molding, previous work has been done using feedforward networks as monitoring tools for improving product quality [14], [15]. The article [16] investigates the integration of analytical process knowledge and NNs as a solution for quality prediction of molded parts. With the aid of NNs the relationship between the process characteristics and product quality is modeled in [17]. In [18], prediction of mechanical product properties such as tensile modulus in injection molded plastics using NNs has been performed. In [19], a backpropagation network is trained to associate part quality with the corresponding data pattern produced during injection. In [20], a radial basis function (RBF) NN has been used to identify faults and to classify part defects following a series of runs in the molding machine. The approach in [21] uses NNs to learn from sample data and induces the values for change of the operating molding conditions, leading to the dimensional improvement of the injection molding parts.

Although, NNs are valuable machine learning modeling tools, provided enough rich data is available for modeling the process nonlinear relationships, they rely on empirical risk minimization. The theoretical understanding of models which are based on minimization of the generalization error increase the degree of confidence of their use, particularly in noisy and dynamic environments such as those found in industry. This and other problem design techniques lead to seek innovative ways within the field of machine learning.

We present a successful example of generating accurate models for multiple quality characteristics in injection molding. These models map process measurements into product quality. The increasing quality demands from the customers make clear the need to explore novel ways of quality control. The goal is to monitor in-process data as a means of indicating product

Manuscript received June 10, 2003; revised June 14, 2004. This paper was recommended by Associate Editor J. Wang.

The author is with the Center of Informatics and Systems, Department of Informatics Engineering, University of Coimbra, P-3030-290 Coimbra, Portugal (e-mail: bribeiro@dei.uc.pt).

Digital Object Identifier 10.1109/TSMCC.2004.843228

quality and to be able to respond quickly to unexpected process disturbances. Our approach, based on SVMs, exploits the first part of this goal. The approach is further compared with RBF NNs under the same classification problem in the given industrial data sets.

The rest of the paper is organized as follows. Section II introduces SVMs for classification and gives details on both the  $C$ -SVM and  $\nu$ -SVM model classifiers. Section III points out the main issues involved in the SVMs training and gives a summary of model selection techniques as well as the formulation of the multiclass handled problem in injection molding. Section IV is devoted to present experimental results. First, describes the industrial process set up and examines the process data sets and, second, reports results concerning both the SVM's and the RBF approaches. Finally, Section V presents the conclusions.

## II. SUPPORT VECTOR CLASSIFICATION

The problem of classification consists of estimating a function  $f: \mathbb{R}^N \rightarrow \{\pm 1\}$  using  $l$  i.i.d. input-output training data  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l) \in \mathbb{R}^N \times \{\pm 1\}$  from a data set  $\mathcal{D}$  such that  $f$  classifies correctly unobserved data  $(\mathbf{x}, y)$  (i.e.,  $f(\mathbf{x}) = y$  for examples  $(\mathbf{x}, y)$  generated from the some underlying probability distribution  $P(\mathbf{x}, y)$ ). In other words, the loss function  $L$  can be defined by (1)

$$L(y_i, f(\mathbf{x}_i)) = |1 - y_i f(\mathbf{x}_i)|_+ \quad (1)$$

where  $r_+ = \max\{0, r\}$   $r \in \mathbb{R}$ .

We give a very brief review of support vector classification (SVC); the reader is referred, for more details, to textbooks or survey papers in [22] and [23]. When the data is linearly separable there exists a vector  $\mathbf{w} \in \mathbb{R}^N$  and a scalar  $b \in \mathbb{R}$  such that  $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$  for all patterns in the training set ( $i = 1, \dots, l$ ). The optimal hyperplane separates points lying on opposites classes yielding to the maximum margin separation. A separating hyperplane which generalizes well can be found by solving the following quadratic programming (QP) problem (for  $i = 1, \dots, l$ ):

$$\begin{aligned} & \text{Minimize} && \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{Subject to} && y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i. \end{aligned} \quad (2)$$

This constrained optimization problem is solved by constructing a Lagrangian

$$\mathcal{L}_P(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1). \quad (3)$$

The Lagrangian has to be minimized with respect to the primal variables  $\mathbf{w}$  and  $b$  and maximized with respect to the dual variables  $\alpha_i$ . The Karush–Kuhn–Tucker (KKT) conditions lead to find the solution vector in terms of the training patterns,  $\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i$  for some  $\alpha_i \geq 0$ . Notice that  $\alpha_i \neq 0$  only for a subset of the training patterns, precisely those few vectors that lie on the margin, called the support vectors (SVs). Under certain conditions, a kernel function  $K(\cdot, \cdot)$  can be found such that  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ . An SVM uses then the convolution of the

scalar product to build, in input space, the nonlinear decision function

$$f(\mathbf{x}) = \text{sgn} \left( \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \right) \quad (4)$$

where  $b$  is found from the primal constraints and is computed by  $\alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1) = 0, i = 1, \dots, l$ , such that  $\alpha_i$  is not zero and  $\text{sgn}$  is the signal function.

### A. $C$ -SVM Classifier

When the training data is not linearly separable, a separating hyperplane does not exist. Besides, when real data sets are used, SVMs can fit noise and outliers leading to poor generalization. Thus, a hard margin classifier is no longer adequate. Introducing a soft margin, the learning task is essentially the same as indicated in (2) except for the introduction of the penalty term  $C$  and the slack variable  $\xi$ . The classifier tries then to separate the data by minimizing the objective function

$$\begin{aligned} & \text{Minimize} && \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{l} \sum_{i=1}^l \xi_i \\ & \text{Subject to} && y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \\ & && 0 \leq \alpha_i \leq C/l, \quad \xi_i \geq 0 \end{aligned} \quad (5)$$

for  $i = 1, \dots, l$ . In this sense, it acts by controlling the classifier capacity and the number of training errors. In other words, the task is now to minimize the sum of errors  $\sum_{i=1}^l \xi_i$  in addition to  $\|\mathbf{w}\|^2$ . Again this optimization problem can be transformed into an QP problem. The value of  $C$  can be found by experimentation in a validation set and cannot be determined from either the model or the data set.

### B. $\nu$ -SVM Classifier

The setting of error penalty in the original SVM's formulation is essentially based on trial-and-error which requires additional time consuming training. This shortcoming is partially overcome with the  $\nu$ -SVM by [24]. The  $\nu$ -SVC algorithm is a new class of SVMs where the minimization problem is

$$\begin{aligned} & \text{Minimize} && \frac{1}{2} \|\mathbf{w}\|^2 - \nu \rho + \frac{1}{l} \sum_i \xi_i \\ & \text{Subject to} && y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq \rho - \xi_i \\ & && \xi_i \geq 0, \quad i = 1, \dots, l, \quad \rho \geq 0 \end{aligned} \quad (6)$$

where  $\rho$  is the position of the margin and  $0 \leq \nu \leq 1$ . The new parameter  $\nu$  has been introduced to control the number of SVs and training errors.

## III. SVM TRAINING

This section reviews some key concepts that we will need for the the SVM's approach in the plastics injection molding machine problem.

### A. Problem Oriented Kernel Choice

The use of kernel methods [25] provide a powerful way of obtaining nonlinear algorithms capable of handling nonseparable

data sets in the original input space. The basic idea is to construct a mapping  $\phi$  into a higher dimensional feature space by the use of reproducing kernels. The kernel function is a positive definite function  $\mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$  that defines an embedding of input patterns into feature vectors. An inner product in feature space has an equivalent kernel in input space  $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$  provided  $K$  is a symmetric kernel function satisfying Mercer's condition [5]. This allows classification to be carried out in the feature space without knowing the explicit form of the mapping  $\phi$ . While Mercer's theorem clearly establishes the characterization of the kernel, its choice remains an open issue and in most cases is problem-oriented. In fact, its importance is two-fold: it determines both the effectiveness of data representation for a particular problem and the class of functions chosen by the SVMs for solving the problem. Using different kernels correspond to different functions to be chosen. Some examples include Gaussian RBF ( $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\|\mathbf{x} - \mathbf{y}\|^2)$  where  $\gamma > 0$  is related with the kernel width, polynomials of degree  $d$  ( $K(\mathbf{x}, \mathbf{y}) = (\gamma(\mathbf{x} \cdot \mathbf{y}) + r)^d$ ) with  $\gamma > 0$  and sigmoid ( $K(\mathbf{x}, \mathbf{y}) = \tanh(\gamma(\mathbf{x} \cdot \mathbf{y}) + r)$ ) where  $\gamma, r$ , and  $d$  are kernel parameters.

### B. Model Selection

In any predictive learning task, such as classification, an appropriate representation of examples as well as the model and parameter estimation method should be selected to obtain a high level of performance of the learning machine. Traditional statistical approach to estimating models from data is based on parametric estimation. The basic fact that an assumption of an underlying dependency with a simple known parametric form is an ensuing need, limits its applicability in practice. Recent approaches allow a wide class of models of varying complexity to be chosen. Then the task of learning amounts to selecting the model of optimal complexity and estimating parameters from training data [26]. Under the SVM's approach, the usually parameters to be chosen are the following:

- the penalty term  $C$  which determines the tradeoff between the complexity of the decision function and the number of training examples misclassified;
- the mapping function  $\phi$ ; and
- the kernel function such that  $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ .

In the case an RBF kernel is chosen, the other indeterminate is the kernel width which implicitly defines the high dimensional feature space. Theoretical model selection strategies give a reasonable estimate for the kernel parameter utilizing an upper bound on the generalization error predicted by the VC-theory [27]. In this sense, it can be used a theorem [2] stating that the generalization error bound ( $E$ ) is reduced as the margin  $\rho$  is increased. This upper bound is  $E = R^2/l\rho^2$ , where  $R$  is the radius of the smallest ball containing the  $l$  training data points in the nonlinear feature space. At the optimum (2) it is possible to show that  $\sum_i \alpha_i^0 = (1/\rho^2)$  where  $\alpha_i^0$  are the values of  $\alpha_i$  at the optimum. For an RBF kernel, the data lies in an hypersphere ( $R \simeq 1$ ) and the bound above can be written as  $\sum_{i=1}^l \alpha_i^0/l$ . Through sequential training of the SVMs, an estimate can be found by choosing the proper value of the kernel width for which

$E$  is minimized. This estimate can be poor in distributions where data lies in a flat ellipsoid since the radius  $R$  would be influenced by the largest deviations [25]. Experiments in [2] and [23] indicate that the bounds are very loose, though the minimum of the bound seems to approximately coincide with the minimum of the generalization error [28]. Taking into account unbalanced data distributions, the approach in [29] is based on rescaling data in the kernel feature space using the eigenvalues and eigenvectors of the covariance matrix  $K(\mathbf{x}_i, \mathbf{x}_j)$ . Another strategy is the leave-one-out cross validation approach. In this method of resampling, single elements are removed from the training set, while the support vector machine is trained on the remained ones and then tested on the removed datapoint. The prediction risk is estimated via cross validation and the model providing lowest estimated risk is chosen. It can be shown than for large data sets, cross validation is asymptotically equivalent to analytical model selection [30]. In this case, the computational cost of cross validation in terms of computational time and training time is high. This problem also occurs with NNs when using weight decay since both the regularization parameter and the number of hidden neurons have to be determined.

### C. Multiclass Problem Formulation

Although SVMs were originally designed for binary classification, there are several methods for the extension from the binary two-class problem to  $n$  classes while ongoing research is still being done. Basically, the two types of approaches usually followed are: 1) to modify the design of the SVMs in order to incorporate multiclass learning in the quadratic solving algorithm [31] and 2) to combine several binary classifiers [32], [33]. In the first case, the proposed methods treat all classes at once considering only one optimization problem, however the computational cost is high particularly if a large-scale real-life problem is considered. In the second case, several methods have been proposed where typically a multiclass classifier is constructed by combining binary classifiers. From this standpoint of view, the "one-against-all" is the standard approach to the  $n$ -class problem. It constructs  $n$  models of SVMs, where the training is as follows: the  $k$ th model is trained with all patterns in class  $k$  with positive labels (e.g., +1) and all the other patterns with negative labels (e.g., -1). After solving  $n$  decisions functions a given pattern is in the class with largest value of the decision function, i.e., it uses a winner takes-all like voting scheme. Another method is the "one-against-one" which consists on a pairwise classification. The basic idea is to use  $n * (n - 1)/2$  classifiers covering all pairs of classes instead of using only  $n$  classifiers as in the "one-against-all." In fact, in this algorithm,  $n$  hyperplanes are constructed, each hyperplane separating one class from another. In this sense, there are  $n$  decision functions  $f_k (1 \leq k \leq n)$  of the form (4). Thus, the  $n$ -class classification problem can be viewed as a discriminant vector  $\in \mathbb{R}^n$  where the index of the largest component is chosen as the elected class decision. The approach used in this paper uses the last but one formulation being  $n = 6$ , since six faults are identified for molded parts produced by the injection machine, as it will be further detailed.

#### D. Implementation

The SV optimization problem can be solved analytically when the size of training data is very small. A general package that solves linearly constrained convex programs can then be used. For larger problems, alternative techniques have to be chosen. Traditional optimization algorithms such as Newton, quasi-Newton, conjugate gradient, etc., cannot be directly applied. One common technique is to use a decomposition algorithm [34] so that only portions of the training data need to be handled at a time, i.e., the QP problem to be solved is decomposed into a series of smaller QP problems. As discussed in [35], from an optimization point of view, the decomposition method has the disadvantage of slow convergence. Therefore, for difficult models, more optimization knowledge and flexibility should be added as implemented in the SV package LIBSVM [36]. In this package, other formulations are also included such as  $\nu$ -SVM. Unlike the standard C-SVM technique which penalizes the total misclassification error, the  $\nu$ -SVM controls the number of SVs and training errors. The  $\nu$  parameter is more intuitive to tune than  $C$  which is difficult to select and can easily lead to model overfitting as shown henceforth in the paper.

### IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

We describe the computational experiments and discuss the results from applying SV algorithms,  $C$ -SVM and  $\nu$ -SVM, and RBF NNs, in a real application concerning plastics injection molding technology for automotive industry. One primary goal is to ensure a superior level of quality control of the molded parts via automation in order to respond to the high demanding from industry customers. To achieve this goal, a model for defects detection of molded parts is designed. The model consists on training a SVM in combination with multiple-input–multiple-output strategy to learn relationships between discrepancies of process parameters and parts quality properties. The model contains the essential components for quality control leading to the ultimate goal of the molding machine parameters setup.

A comparison of performance between RBF NNs and SVMs for handling the model is performed. All computations were done on a Pentium 3 Mobile 1.0-GHz machine running on Windows.

#### A. Industrial Problem Setup

IMP is one of the most important plastic manufacturing techniques and widely used extrusion processes [37]. In the IMP, a thermoplastic, in the form of granules, passes from a feed hopper into a barrel where it is heated and melted. It is then injected into a cold mold that is clamped tightly closed. When enough time has elapsed, the plastic returns to a solid state, the mold opens and the produced part is extracted. All the mold details are, by this way, reproduced in the finished part. Fig. 1 illustrates the molding machine, a sketch of the molding functioning process and two of the observed parts defects (edges and unfilled) for the automotive industry. While this industrial process seems quite simple from a theoretical point of view, a close look at its dynamics reveals, however, a high degree of complexity. In fact, the strong nonlinear relationships between

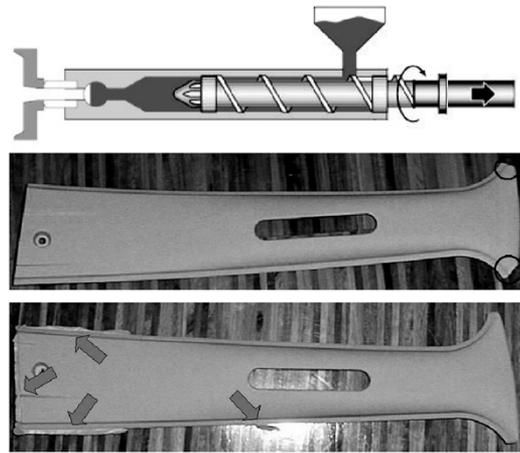


Fig. 1. IMP and automobile plastic molded parts defects.

TABLE I  
PLASTIC PARTS IDENTIFICATION FAULTS

Fault 1	Streaks
Fault 2	Stains
Fault 3	Burn marks
Fault 4	Edges
Fault 5	Unfilled parts
Fault 6	Warped parts

the process parameters and their reciprocal effects as well as the unpredictable behavior of the material under temperature–pressure fluctuations, makes not only the mathematical analysis of its behavior, but also the quality control of the molded parts difficult. Additionally, the lack of consistent relationships between the process parameters, the material properties and the molded parts quality, as well as the environmental fluctuations, introduce several sources of variation into the system, which prevent it from attaining an ideal state of constant quality.

The wide range of parameters to be manipulated during real time, polymers variety and production techniques make, no doubt, the process very difficult to maintain under control. Each variable (temperature, flow rate, pressure, ...) has a great impact in the final product quality and in the production rate. The choice of the control strategy, particularly the development of a new regulation tool to suggest process parameter values automatically, is essential to solve quality defects in injection molding ensuring energy savings and process safety.

#### B. Process Data Analysis

The industrial data we deal with concern plastic parts for the automotive industry which have been molded in Demag injection molding machine with Hostacom DM2 T06 polymer and with mold DN502. Currently, parts quality are monitored in the plant using statistical analysis of discrete data. Control of injection molding is currently mostly done by operator intuition in a trial-and-error basis mode. The operator controls the set points of the machine based upon his understanding of the effects of each of the controls on the quality of the parts. This situation leads to significant difficulties and variation in the quality of the parts and reliability of the process. Although there

TABLE II  
DISPERSION AND CENTRAL TENDENCY MEASURES FOR PROCESS DATA

Process Variables	Dispersion Measures			Central Tendency	
	Range	Std. Dev	Variance	Mean	Median
Cycle time	692.10	73.43	5391.92	43.77	31.65
Metering time	5.91	0.99	0.99	10.42	10.50
Injection time	3.05	0.47	0.22	2.40	2.10
Barrel temp.	17.00	3.52	12.41	228.16	227.00
Cushion	18.00	3.01	9.07	13.13	13.10
Injection Velocity	67.00	6.20	38.40	204.80	205.00

is ongoing research in the molding machine parameter setting, most published approaches include: mathematical models, numerical simulation, process windows, design of experiment (DOE), expert systems, NNs, case based reasoning (CBR), genetic algorithms (GAs) and evolutionary strategies. For an excellent review of the state-of-art on the topic, see [38]. Here, we present a new approach for quality monitoring of online molded parts by analyzing complete data patterns based on SVMs. Before going further into the proposed solution, some observations about the process data characterization are examined first. The data patterns consists of a series of discrete values. The process responses include sensor readings forming a time series of temperatures, pressures, etc. Twenty-six setup parameters have been registered and stored in each run, from barrel temperatures up to injection pressures. Since the total number of inputs is large, selection of features from these profiles were extracted for modeling purposes as well as the cycle length for the cyclical variables. Therefore, six process variables, namely, cycle time, metering time, injection time, barrel temperature before nozzle, cushion, and injection velocity have been identified. Regarding quality characteristics they are classified as mechanical properties, dimensions, or measurable characteristics, and attributes. In general, some of the main causes of quality problems are material related defects, i.e., black specks and splay, process related problems such as filling related defects, i.e., flash and short shots, packing and cooling related defects, i.e., sink marks and voids, and post mold related defects, i.e., warpage, dimensional changes, and weight. The parts quality measure was chosen to be based on six quality variables, namely, streaks, stains, burn marks, edges, unfilled parts and warpage which have been found to be representative of the range of defects that can occur so far in the plant. These data have been obtained using different process conditions across the runs when an entire sensor profile is analyzed. Table I establishes the corresponding fault identification assignement. Table II illustrates the tendency central measures and the dispersion measures for the data showing heterogeneity.

### C. Injection Molding Fault Models Design

1) *RBF NNs*: The standard RBF NN with a single output neuron realizes a mapping function,  $f : \mathbf{x} \rightarrow f(\mathbf{x})$ , where the  $N$ -dimensional input vector is submitted to the network and

the scalar output,  $f(\mathbf{x})$ , obtained to generate the classification decision. The RBF network output is then given by (7)

$$f(\mathbf{x}) = \sum_{i=1}^m w_i \phi_i(\mathbf{x}, \mathbf{c}_i) + b \quad (7)$$

where  $m$  is the number of hidden layer neurons,  $w_i$  and  $b$  are the weights. Each hidden neuron  $i$  has an associated kernel function  $\phi_i = \exp(-(\|\mathbf{x} - \mathbf{c}_i\|^2)/(2\sigma^2))$ ,  $\phi : \mathbb{R}^N \rightarrow \mathbb{R}$ , where  $\mathbf{c}_i \in \mathbb{R}^N$  and  $\sigma$  are the RBF centers and width, respectively. For a two-class classification problem, the classifier determines the class label of the input vector  $\mathbf{x}$

$$\hat{C}(\mathbf{x}) = \text{sgn}(f(\mathbf{x})). \quad (8)$$

Training of RBF NNs involves selecting the centers  $\mathbf{c}_i$ , estimating the weights  $w_i$ , and bias  $b$  that connect the hidden and the output layers. Although most straightforward approaches for training RBF networks are nonlinear optimization algorithms, most nonlinear algorithms suffer from long training times and local minima. Therefore, a hybrid learning algorithm is often used. It allows, in a first stage, to determine the centers and the kernel widths, by a self-organizing bottom-up process that places the centers only in those regions of the input space where data is present. In a second stage, the weights are then evaluated by the supervised LMS algorithm. Standard k-means clustering algorithm [39] allows to find the centers by determining the local minimum of the squared Euclidean distance,  $E$ , between  $l$  training vectors  $\mathbf{x}_j$  and the nearest of  $m$  centers  $\mathbf{c}_i$  given by

$$E = \sum_i^m \sum_j^l \mathbf{k}_{ij} (\mathbf{c}_i - \mathbf{x}_j)^T (\mathbf{c}_i - \mathbf{x}_j) \quad (9)$$

where  $\mathbf{k}_{ij}$  is an  $(m \times l)\{0, 1\}$  matrix with exactly one “1” per column which identifies the processing node to which the training vector belongs. A RBF NN with six inputs (cycle time, dosage time, injection time, cushion, peak melt temperature, ram velocity), six outputs (streaks, stains, burn marks, edges, unfilled parts, warped parts) and Gaussian hidden nodes was used. It was trained with the previously described algorithm on a training data set consisting of 120 input–output pairs. The testing data set was constituted by 80 input–output patterns. The outputs are the observed defects in molded parts obtained onsite

TABLE III  
RBF-NN: TRAINING ACCURACY WITH VARYING NUMBER OF GAUSSIAN HIDDEN UNITS

%Training Accuracy	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6
RBF Kernels	streaks	stains	burn marks	edges	unfilled parts	warped parts
60	65.10	76.56	92.20	88.29	99.89	97.35
80	73.55	77.49	93.95	93.11	99.89	82.98
100	77.27	84.35	93.61	93.16	99.58	85.98

TABLE IV  
RBF-NN: TESTING ACCURACY WITH VARYING NUMBER OF GAUSSIAN HIDDEN UNITS

%Testing Accuracy	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6
RBF Kernels	streaks	stains	burn marks	edges	unfilled parts	warped parts
60	53.09	70.61	85.55	72.90	99.86	99.95
80	67.60	68.18	87.33	77.92	88.68	99.04
100	64.03	81.26	90.26	84.65	99.46	99.60

TABLE V  
C-SVM WITH RBF KERNEL:  $C = 1$  AND  $C = 1000$

	C-SVM (C=1)			C-SVM (C=1000)			
	#Iter	SVs	Accuracy(%)	#Iter	SVs	Accuracy(%)	
Fault1	1831	93	75.00	Fault1	43588	68	86.67
Fault2	350	87	79.17	Fault2	18693	60	90.83
Fault3	233	51	86.67	Fault3	4387	27	97.50
Fault4	456	54	90.00	Fault4	31300	46	96.67
Fault5	102	22	98.33	Fault5	61	15	100.00
Fault6	37	17	99.17	Fault6	77	16	100.00

during running of the industrial molding machine upon convenient change of the respective process setups. The classification results are shown in Tables III and IV regarding the six faults tested. With 60 hidden Gaussian units, the root mean squared error (RMSE) obtained on the data testing set was 0.2465, after 183745 training iterations, whereas with 100 Gaussian units the RMSE = 0.1863, achieved on the testing data set after 127713 training iterations. An intermediate value of RMSE = 0.2165 and a number of training iterations of 121724 was obtained for the network with 80 hidden units. The configuration with 60 nodes almost fails to identify Fault 1. Besides, if we think about the sizes of the training and test data set, we easily realize that the configuration design of 100 hidden units leads to overfitting, since we have almost one unit per data point. The results regarding Fault 6 are due to an unbalanced number of this kind of fault in both data sets. In summary, the RBF classifier with 80 hidden units corresponds to the best configuration obtained so far and will be considered for further comparison in next section.

2) *C-SVM Classifier*: *C-SVM* classifiers are designed for binary classification. When dealing with several classes, as in the case of fault detection in the injection molding machine, an appropriate multiclass method is required. As mentioned in the Subsection III-C, one common approach is to formulate the multiclassification problem as a series of binary classification problems. The standard approach “one against all” has been used and six classifiers are then constructed. For a given test pattern, the classifier with the highest output is selected as the winner and the corresponding class label assigned.

Training SVMs requires the selection of parameters which influence the ensuing model performance. Therefore, to achieve a good model those parameters have to be chosen correctly. Examples, as stated earlier, are 1) cost function  $C$  and 2) the map-

ping function  $\phi$ . In our experiments, we have considered RBFs as the kernel function. The RBF kernel is very advantageous in complex nonseparable classification problems due to its ability of nonlinear input mapping. As aforementioned, it has the property that  $\phi(\mathbf{x}) \cdot \phi(\mathbf{y}) = \exp(-\gamma\|\mathbf{x} - \mathbf{y}\|^2)$ , and subsequently  $\gamma$  (defined as  $1/2\sigma^2$ ,  $\sigma$  being the kernel width) is an important parameter to be chosen.

a) “*Trial-and-Error*” *Parameters Selection*: In a first series of experiments we run the classifier with several values of  $C$  and  $\gamma$  somehow trying to guess which combination of parameters might be the best for a “good” model. That is, the one that could better express the causal relation among variables which govern the quality within the mold/machine platform. This is accessed through the evaluation of performance accuracy. One possible way is to divide the original data into a data training set and into a validation data set for model evaluation. We compare, in Table V, the number of iterations, the number of SVs and the testing accuracy of the learning machine with  $C$  set to 1 and 1000. The stopping tolerance for solving the optimization problem is set to  $10^{-3}$ . As verified in the latter case, the accuracy is higher for all the displayed faults. In case of fault 5 (Unfilled parts) and 6 (Warped parts), no misclassification errors are found. In Fig. 2, experimental results show for Fault1 a) the variation of number of SVs and b) testing accuracy as a function of kernel  $\gamma$  both parametrized with  $C$ . Larger  $C$  correspond to less number of SVs as well as higher testing accuracy although overfitting cannot thus be avoided. Further explanation is required for these results taking into account both  $C$  and  $\gamma$  parameters. In fact, minimizing  $1/2\|\mathbf{w}\|^2$  in (5) corresponds to maximize the margin  $2/\|\mathbf{w}\|^2$  between two classes of data. For nonseparable data, the penalty term  $C\sum_{i=1}^l \xi_i$  is able to reduce the training errors in the working data set. Therefore,

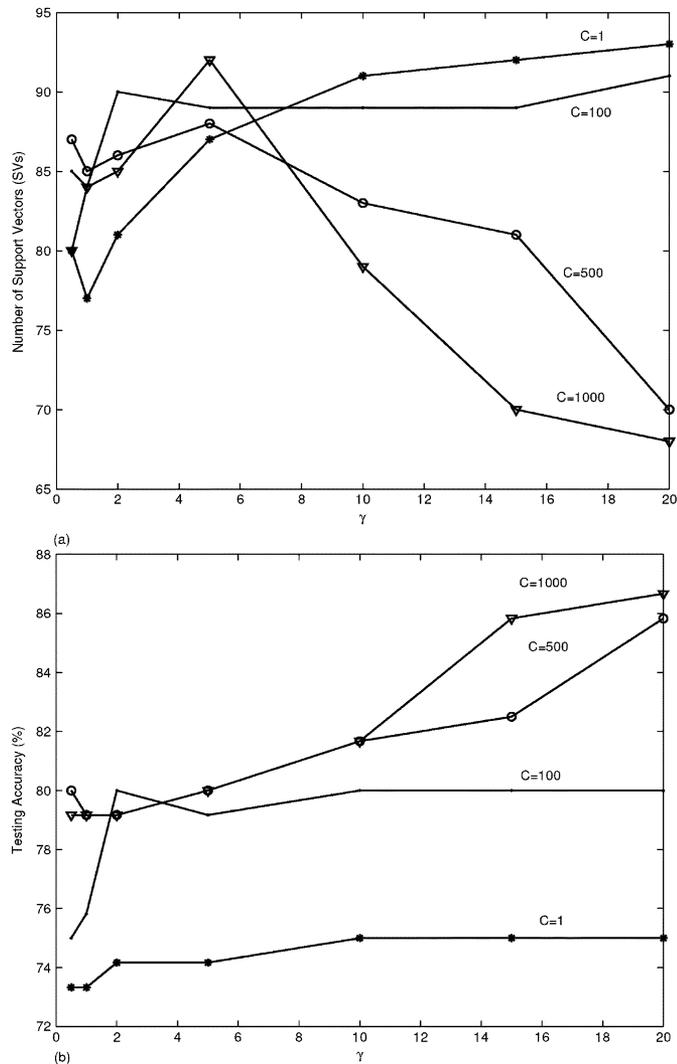


Fig. 2. Fault 1 model parametrized with  $C$ . (a) Number of SVs versus  $\gamma$ . (b) Accuracy versus  $\gamma$ .

the margin is an indicator of the generalization accuracy. In the absence of a method to compute the best tradeoff between the regularization term and the training errors, the balance sought by the SVM's technique is hard to find. Thus, a larger  $C$  corresponds to assign a higher penalty of training errors and clearly overfitting occurs. On the other hand, when the kernel parameter  $\gamma$  becomes higher, the greater the variety of the decision boundaries that can be formed originating a more complex model. The added flexibility decreases initially the generalization error as the model can better fit the data. However, there is the danger, as can be seen in the Fault 1 model, that this can lead to overfitting as well.

*b) Interactive Grid Search Model Selection:* Choosing the best parameters, especially if a systematic approach is not used and/or the problem knowledge do not aid for proper selection, can be timing consuming since we have to rely upon guessing and trial and error techniques. Therefore, an interactive grid search model selection has been accomplished for each one of the displayed faults and the generalized accuracy evaluated. Fig. 3 depicts the generalization graphic contours for the selected faults after a five-cross validation, thus, reducing

the search space as it might be noted by simply analyzing the graphs in the following. The efficient heuristic way of searching points in that space with small generalization errors will lead to a good understanding of the hyperparameter space [40]. We can then do a refined search of the  $(C, \gamma)$  pairs for proper model selection. Table VI illustrates the results obtained choosing the pair of parameters which satisfy clearly the restrictions to the handled problem. Said in another way, it is demonstrated that the pair  $(C, \gamma)$  in each fault case conveys to the learning model with the smallest capacity and, thus, the highest generalization. If we compare with RBF NNs we may conclude that SVMs are more accurate and allow better generalization than the former. Besides, in the SVM's technique there is no heuristic choice for model design as it is required in the RBF networks. Also, another issue concerns the different solution method comparison in both techniques; the QP problem in SVM's formulation is simpler to solve, particularly in cases of not too large data sets, conveying to a technique without local minima.

*3)  $\nu$ -SVM Classifier:* To support our experiments on the SVM's model building, we have used a  $\nu$ -SVM classifier, since, partially, it removes the difficulty in selecting the most effective error penalty. Besides, it has been shown that when  $C$  is large, there may have more numerical difficulties on using the decomposition method [41]. In the  $\nu$ -SVM method, there is no  $C$ , so intuitively we might think that this difficulty no longer exists. We compare  $C$ -SVM and  $\nu$ -SVM classifiers which basically perform in the same way but with different parameters. The range of  $C$  is from zero to infinity but  $\nu$  is always between  $[0, 1]$ . A nice property of  $\nu$  is that it is related to the ratio of SVs and the ratio of the training error. We can see that if  $\nu$  becomes smaller, the total number of SV decreases where the increase in  $C$  decreases the number of SVs. Table VII shows the results with  $\nu$ -SVM classifier obtained for the six faults tested. However, the use of uneven training data class sizes, which occurs in our run experiments particularly with Faults 5 and 6, leads to learning machines biased toward the classes with the largest training sizes (Faults 2, 3, and 4). To circumvent these effects as well as to adjust the error penalties of each class separately, a different  $\nu$  for each class was used. Fig. 4 illustrates an experimental study which shows the relation between  $C$  and  $\nu$  for various values of the kernel parameter  $\gamma$ .

## V. CONCLUSION

In this paper, we have discussed the application of the multiclass SVM's classifiers and compared with RBF NNs to predict faults in automotive parts from molding injection in plastics industry. The basic idea of the SVMs is to determine the structure of the classifier by minimizing the bounds of the training error and generalization error. The SVs close to the boundary decision surface determine the efficacy of the classifier, whereas in RBF NN classifier, the RBF centers are selected by k-means clustering techniques. The hidden layer of the RBF NN classifier can be viewed as a function that maps the input patterns from a nonlinear separable space. In the new space, the responses of the hidden layer neurons form new features vectors for pattern representation. Then the discriminative power is determined by RBF centers. Experiments

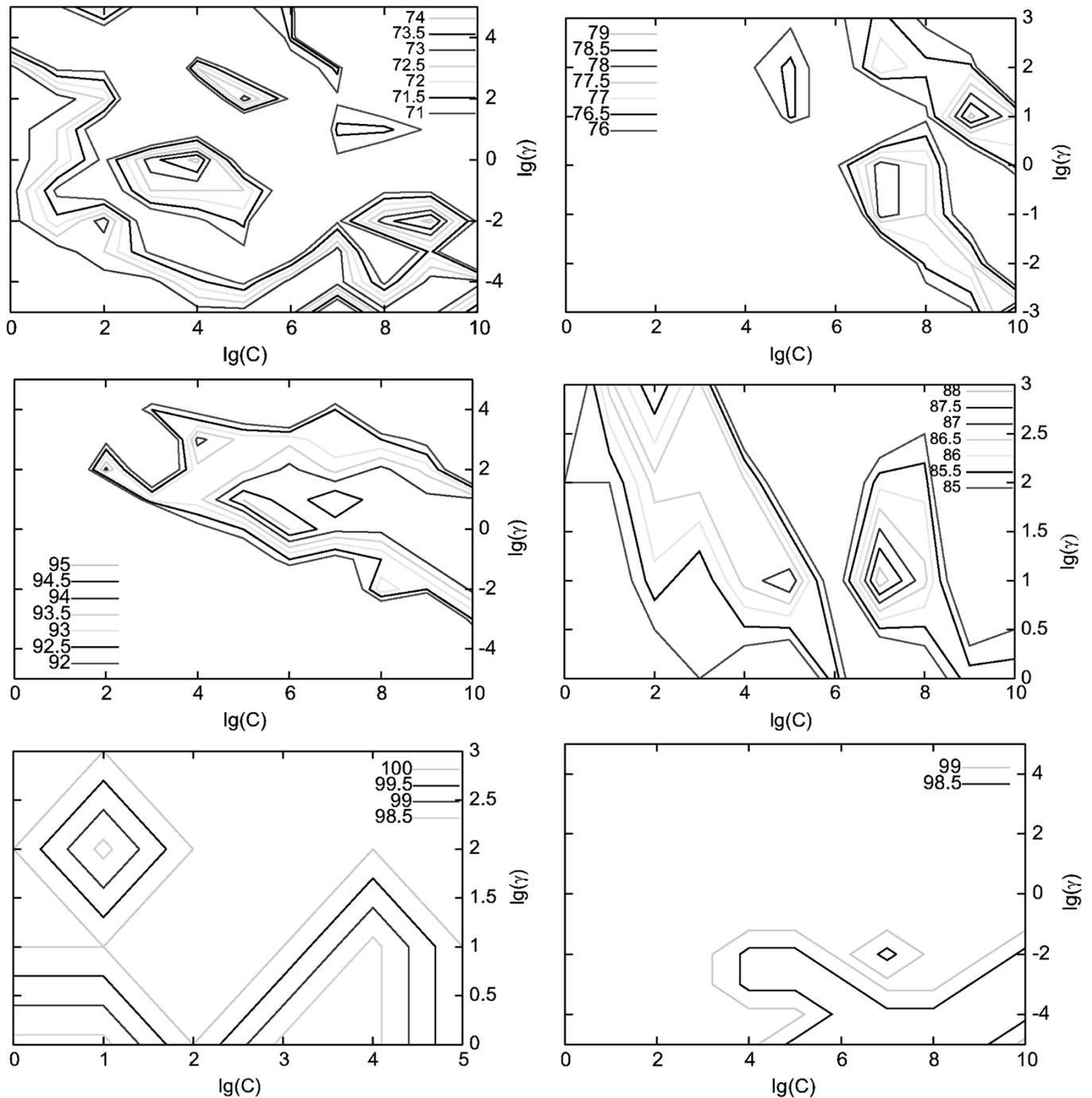


Fig. 3. Model Selection for Faults  $F_i, i = 1, 6$ . In case of  $F_1$ , the grid search is on  $C = 2^0, 2^2, \dots, 2^{10}$  and  $\gamma = 2^{-5}, 2^{-4}, \dots, 2^5$ .

TABLE VI  
GENERALIZED ACCURACY FOR  $C$  AND  $\gamma$  HYPERPARAMETERS OBTAINED  
BY MODEL SELECTION FOR FAULTS  $F_i, i = 1, 6$

Faults	C	$\gamma$	#Iter	SVs	Accuracy (%)	Accuracy (%) (cv)
Fault 1	16	1	3724	85	75.00	70.83
Fault 2	512	2	14221	65	85.83	76.67
Fault 3	32	1	1246	37	96.67	93.33
Fault 4	128	2	2856	42	92.5	87.50
Fault 5	4	2	115	20	100	98.33
Fault 6	16	0.25	8	4	100	99.16

indicate that in general the test accuracy of RBF NNs is a little worse than that of the SVMs in both cases of either the

$C$ -SVM or the  $\nu$ -SVM. Though RBF NNs trained with an hybrid learning algorithm keeps similarities to the SVMs due to the Gaussian kernel, determining ad hoc the number of RBF centers still downgrades the classifier performance. Within the chosen SVM's framework, the hyperparameter tuning, input selection and computation of posterior class probability for risk minimization decision making can be conducted in a unified way. We show that overfitting can be avoided by a criterious grid selection search on parameter's space. Regarding the implementation issues, SVMs are considerable faster than the training of RBF NNs, even for large-size problems, requiring less heuristics and, thus, being preferable. In addition, SVMs

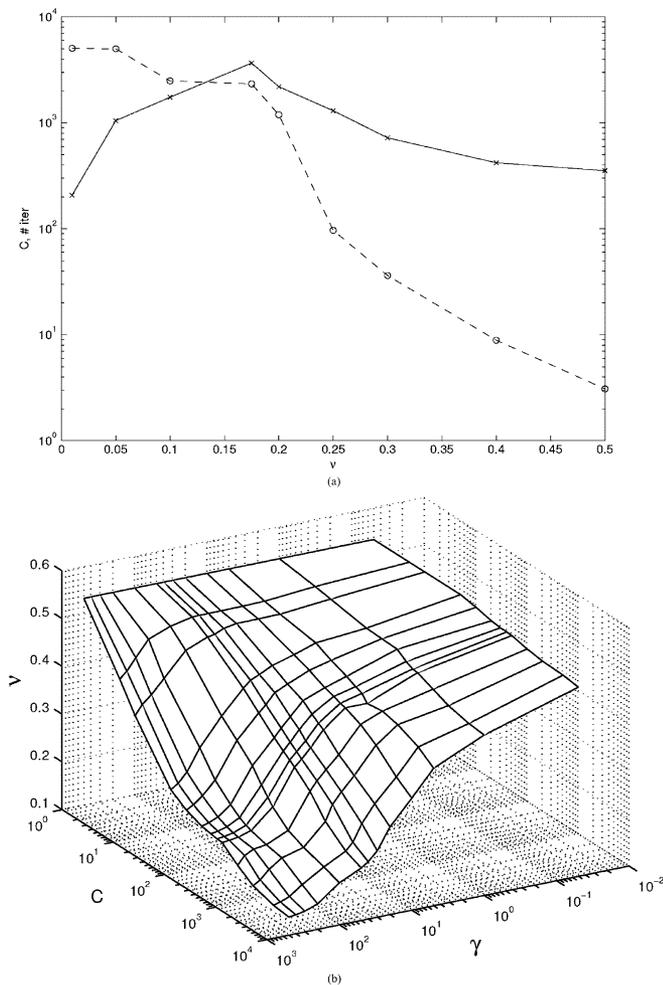


Fig. 4. (a)  $C$  and number of iterations versus  $\nu$ . (b) Relation between  $C$  and  $\nu$ . ( $\gamma$ -RBF kernel).

TABLE VII  
 $\nu$ -SVM CLASSIFIER: RBF KERNEL

	$\nu$ -SVM				
	SVs	Accuracy (%)	#Iter	$\nu$	$C$
Fault1	58	93.33	2190	0.200	1194.5
Fault2	76	92.50	2150	0.175	330.84
Fault3	59	98.33	1356	0.060	26.79
Fault4	66	95.83	718	0.150	8.25
Fault5	15	100.00	86	0.015	10.26
Fault6	15	100.00	96	0.095	9.93

attain sparse solutions in the sense that most of the coefficients  $\alpha_i$  are set to zero as a result of the optimization problem. This property is also computationally attractive. Besides the expected ratio of the number of SVs and the number of training points bounds the generalization error. Our results demonstrate that the SVMs have the potential to obtain a reliable distinction among production part defects, and to assist operators for making a correct diagnosis. An interesting aspect to be investigated is to use SVMs for feature extraction from large industrial data profiles for modeling purposes. The extension of the implementation to data sets in a larger scale, different

methods of problem multiclass formulation and possibly further subclassify the part defects are the focus of our current research.

#### ACKNOWLEDGMENT

The author would like to thank the Center of Informatics and Systems, University of Coimbra (CISUC).

#### REFERENCES

- [1] R. Herbrich, *Learning Kernel Classifiers: Theory and Algorithms*. Cambridge, MA: MIT Press, 2002.
- [2] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [3] B. Schölkopf, C. Burges, and A. Smola, *Advances in Kernel Methods—Support Vector Learning*. Cambridge, MA: MIT Press, 1999.
- [4] V. Vapnik and A. Chervonenkis, "A note on one class of perceptrons," *Autom. Remote Control*, vol. 25, 1964.
- [5] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. ACM Workshop on Computational Learning Theory*, 1992, pp. 144–152.
- [6] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, 1995.
- [7] K. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.
- [8] V. Cherkassky and F. Mulier, *Learning from Data—Concepts, Theory, and Methods*. New York: Wiley, 1998.
- [9] O. L. Mangasarian and D. R. Musicant, "Successive overrelaxation for support vector machines," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1032–1037, Sep. 1999.
- [10] S. Hua and Z. Sun, "A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach," *J. Molecular Biol.*, vol. 308, pp. 397–407, Apr. 2001.
- [11] K. Jonsson, J. Kittler, Y. P. Li, and J. Matas, "Support vector machines for face authentication," *Image Vis. Comput.*, vol. 20, pp. 369–375, Apr. 2002.
- [12] L. J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1506–1518, Nov. 2003.
- [13] L. H. Chiang, M. E. Kotanchek, and A. K. Kordon, "Fault diagnosis based on fisher discriminant analysis and support vector machines," *Comput. Chem. Eng.*, vol. 28, pp. 1389–1401, Jul. 2004.
- [14] O. G. Ali and Y.-T. Chen, "Design quality and robustness with neural networks," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1518–1527, Nov. 1999.
- [15] N. Costa and B. Ribeiro, "Artificial neural networks for data modeling of a plastic injection molding process," in *Proc. IEEE 6th Int. Conf. Neural Information Processing (ICONIP'99)*, Perth, Australia, pp. 1081–1087.
- [16] T. Petrova and D. Kazmer, "Incorporation of phenomenological models in a hybrid neural network for quality control of injection molding," *Polym. Plastics Technol. Eng.*, vol. 38, no. 1, pp. 1–18, 1999.
- [17] E. Schmidberger, O. Gutbrod, F. Heinzelmann, and D. Ohngemach, "Neural networks in injection molding," *Kunststoffe-Plast Europe*, vol. 85, no. 5, p. 620, 1995.
- [18] S. Kenig, A. Ben-David, M. Omer, and A. Sadeh, "Control of properties in injection molding by neural networks," *Eng. Appl. Artif. Intell.*, vol. 16, no. 6, pp. 819–823, 2001.
- [19] S. L. B. Woll, D. J. Cooper, and B. V. Souder, "Online pattern-based part quality monitoring of the injection molding process," *Polym. Eng. Sci.*, vol. 236, no. 11, pp. 1477–1488, 1996.
- [20] B. Ribeiro, "Fault detection in a thermoplastic molding injection process using neural networks," in *Proc. Int. Joint Conf. Neural Networks (IJCNN'99)*, vol. 5, Washington, DC, 1999, pp. 3352–3355.
- [21] H. Lau, A. Ning, K. Pun, and K. Chin, "Neural networks for the dimensional control of molded parts based on a reverse process model," *J. Mater. Process. Technol.*, vol. 117, no. 1–2, pp. 89–96, 2001.
- [22] N. Cristianini and J. Shawe-Taylor, *Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [23] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

- [24] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. Bartlett, "New support vector algorithms," *Neural Computat.*, vol. 12, pp. 1207–1245, 2000.
- [25] C. Campbell, "Kernel methods: A survey of current techniques," *Neuro-comput.*, vol. 48, pp. 63–84, 2002.
- [26] V. Cherkassy, "Model complexity control and statistical learning theory," *Natural Comput.*, vol. 1, no. 1, pp. 109–133, 2002.
- [27] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
- [28] J. T.-Y. Kwok, "The evidence framework applied to support vector machines," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1162–1173, Sep. 2000.
- [29] O. Chapelle, P. Haffner, and V. Vapnik, "SVM's for histogram-based classification," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1055–1065, Sep. 1999.
- [30] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge, U.K.: Cambridge Univ. Press, 1996.
- [31] J. Weston and C. Watkins, "Multiclass support vector machines," Univ. London, Egham, U.K., Tech. Rep. CSD-TR-98-04, 1998.
- [32] U. H.-G. Krebel, "Pairwise classification and support vector machines," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 255–268.
- [33] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems*, 12th ed. Cambridge, MA: MIT Press, 2000.
- [34] J. Platt, "Using analytic QP and sparseness to speed training of support vector machines," *Neural Inf. Process. Syst.*, vol. 11, 1999.
- [35] C.-J. Lin, "On the convergence of the decomposition method for support vector machines," Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2000.
- [36] C.-C. Chang and C.-J. Lin. (2001) LIBSVM: A Library for Support Vector Machines. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [37] *Injection Molding Handbook: The Complete Molding Operation Technology, Performance, Economics*, 2nd ed. London, U.K.: Chapman & Hall, 1995.
- [38] S. Mok, C. Kwong, and W. Lau, "Review of research in the determination of process parameters for plastic injection molding," *Adv. Polym. Technol.*, vol. 18, no. 3, pp. 225–236, 1999.
- [39] J. Moody and C. Darken, "Fast learning in networks locally-tuned processing units," *Neural Computat.*, vol. 1, no. 2, pp. 281–294, 1989.
- [40] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with gaussian kernel," *Neural Computat.*, vol. 7, no. 15, pp. 1667–1689, 2003.
- [41] C.-C. Chang and C.-J. Lin, "Training nu-support vector classifiers: Theory and algorithms," *Neural Computat.*, vol. 13, no. 9, pp. 2119–2147, 2001.



**Bernardete Ribeiro** (M'01) received the degree in chemical engineering, the M.Sc. degree in computer science, and the Ph.D. degree in electrical engineering from the University of Coimbra, Coimbra, Portugal, in 1975, 1987, and 1995, respectively.

She is an Assistant Professor in the Department of Informatics Engineering, Faculty of Science and Technology, University of Coimbra. Her current interests include neural networks and their applications to engineering systems, intelligent control, computational learning, and support vector machines. She has

more than 100 publications in these areas.