Research Article

Hybrid Model Based on Genetic Algorithms and SVM Applied to Variable Selection within Fruit Juice Classification

C. Fernandez-Lozano,[1] **C. Canto,**[1] **M. Gestal,**[1] **J. M. Andrade-Garda,**[2]
J. R. Rabuñal,[1] **J. Dorado,**[1] **and A. Pazos**[1]

[1] *Information and Communications Technologies Department, Faculty of Computer Science, University of A Coruña,*
  Campus Elviña s/n, 15071, A Coruña, Spain
[2] *Analytical Chemistry Department, Faculty of Sciences, University of A Coruña, Campus da Zapateira s/n,*
  15008, A Coruña, Spain

Correspondence should be addressed to C. Fernandez-Lozano; carlos.fernandez@udc.es

Given the background of the use of Neural Networks in problems of apple juice classification, this paper aim at implementing a newly developed method in the field of machine learning: the Support Vector Machines (SVM). Therefore, a hybrid model that combines genetic algorithms and support vector machines is suggested in such a way that, when using SVM as a fitness function of the Genetic Algorithm (GA), the most representative variables for a specific classification problem can be selected.

1. Introduction

The evolution of technology has made accessible a wide range of information in almost any area of human activity. The increased connectivity and the development of Internet have allowed access to large volumes of data. Moreover, the increase of the processing capacity of computers, as well as the current low cost of computer storage, allows preserving up to the last generated byte.

However, even having the means necessary for the data preservation, it is common to find a large number of irrelevant information when solving a problem. Only a fraction of the data has different and significant profiles, whereas the rest is redundant information, unwanted noise, or worse, information captured or stored incorrectly. In recent years, various variable selection methods have been developed with the aim of mitigating these problems as far as possible [1].

The feature selection or extraction is first introduced in the early 1960s, becoming a research topic widely studied and used over the last four decades [1–3]. The need for these techniques arises, as noted, from the emergence of certain predictive variables that may become irrelevant or redundant.

A predictive variable is considered irrelevant if its content does not provide any information that could clear uncertainty about the original set. Similarly, it is redundant when its value can be determined from other predictive variables. The variable selection methods are aimed at detecting redundant and/or trivial variables for a given problem. The purpose of using these techniques is to identify an "optimal" subset, made up of a minimum number of variables necessary to find a valid solution to a problem.

As a result of applying these methods, some benefits are obtained, consolidating their position throughout time [4–6]. Among others, it is worth mentioning that the variable selection methods allow decreasing the number of samples required to obtain optimum results in virtually any classification or clustering problem. This leads, for example, to the creation of a faster prediction model when it comes to analyzing a lower number of variables. Similarly, cost reduction is obtained, taking into account a temporal perspective as well as the one regarding the system complexity for data acquisition, since we are dealing with a smaller amount of information. As a result, the complexity of the problem to be solved is reduced.

In this paper, we propose the use of a prediction hybrid system combining GA and SVM. First, we will briefly mention the most representative selection techniques in recent times. Then, we will describe the methods used to develop the proposed system. Next, we will explain concisely the developed model, as well as the results obtained with it. These results, related to the classification of juice samples according to their sugar concentration, will be compared to those obtained in previous studies from the use of ANN, allowing a comparative study of both methods.

2. State of the Art

In recent years, the methodology used by different organizations has been forced to evolve due to the fact that storage and processing capacities have substantially increased. This progress has led researchers to develop a large number of selection and learning techniques that allow improving their working methods.

The variable selection techniques have become a very useful tool. Among the existing techniques it is worth mentioning, as the most representative ones, Principal Component Analysis (PCA)—a mathematical method [7], and Partial Least Squares (PSA) [8], which behave similarly from the point of view of variable selection.

The main component analysis method is characterized as a statistical technique for the information synthesis or dimension reduction [7]. We are dealing with a linear algorithm whose operation is based on the correlation between variables. The aim of these methods is to minimize the number of variables as much as possible with the minimum loss of information. However, its use entails a series of drawbacks. The most important one comes from the set of variables obtained from their use, as these latter one do not belong to the initial set but are a linear combination of them.

On the other hand, PLS is a mathematical approach used to establish a model that relates the information from two different datasets. The idea is to look not only for the directions with a larger amount of information within the set of predictive variables but also to select those which have a stronger relationship with the variables to be predicted [8, 9]. This is why it is said that the PLS models are governed by a criterion of predictive ability rather than by the fitness of the model to the data.

The use of these or other mathematical methods involves a series of drawbacks that encourage the search for new techniques, aimed at reducing them. Among the common disadvantages of using these methods, it is worth mentioning two main drawbacks. On the one hand, it is necessary to have a broad knowledge of the field to be discussed, since, depending on certain features (linearity, interdependence, etc.) it will be possible to apply only some methods or others. On the other hand, such methods tend to have a single valid solution, not a set of solutions, so that they would not be efficient in problems in which there are several global solutions, or a global solution and local variables.

In several studies, evolutionary methods such as variable selection techniques were used [10, 11] due to the fact that they minimize the drawbacks of using the above-mentioned techniques. From the field of evolutionary computation and, more specifically, from the area of Gas, there has been a significant number of approaches towards variable selection methods.

That is why, in order to increase the reliability of the variable selection methods, we have opted for new strategies, among which we consider of great importance the hybrid methods that employ GAs and, on the one hand, prediction mechanisms, among which artificial neural networks (ANNs) are worth mentioning.

The ANNs are methods of nonparametric prediction-classification that allow obtaining a high degree of accuracy in many problems, especially those related to human knowledge [12–14]. They are based on a computational model established according to biological neural systems that try to simulate the operation of human brain neurons at small scale. The training of the weights of the ANN through traditional neuronal network learning methods has proven to be a very effective process. The creation of a hybrid model combining GAs and ANN allows the use of the former system as a guideline for improving the latter. As a result, GAs are employed to optimize the number of entries into the neural predictor classifier, thus decreasing the generalization error and consequently the size of the models.

There have been several studies that have opted for the combination of evolutionary methods with classification mechanisms [6, 13, 15–17]. However, even after having developed very efficient models, ANNs have the disadvantage of undergoing training processes on a strong stochastic basis, leading to nonrepeatability of the process. This is why many researchers prefer to choose new techniques based on robust statistical principles such as support vector machines (SVM) [18–23].

Developed by Vapnik [24, 25] and based on the statistical learning theory, the support vector machines are fast becoming one of the most used methods of prediction classification. Although their use is fairly recent, a considerable number of researchers have already reported states of the art of their performance in a variety of applications in pattern recognition, regression estimation, and prediction of time series. For example, the study carried out by Min et al. [26] combined the SVM and the GAs with the aim of predicting the business failure risk and avoiding bankruptcy, with a particular model being tested in predicting the crisis in Taiwan by Wu et al. [27]. Tan et al. [11], besides combining GAs and SVM, opted for the use of PLS for the identification of mitochondrial proteins. There are also some comparisons of SVM with regression methods [28]. Huerta et al. [29] combined a GA with SVM for gene selection and classification of microarray data. Venegas [30] used the SVM in the classification of academic texts according to their lexical-semantic content; in the study carried out by Donís et al. [31] the SVM were used to estimate the creep rupture stress of ferritic steel. Pérez et al. [32] applied the SVM to a typical classification problem of failure identification in distribution systems of electric power. In Fernandez-Lozano et al. [15, 16], a hybrid approach combines GAs and SVM for protein identification in two-dimensional gel electrophoresis images. Tong et al. [33] proposed a GA based ensemble SVM classifier built

on gene pairs. A GA-based for solving two dual quadratic programming problems with a twin parametric-margin SVM in the primal space is proposed by Wang et al. [34]. Won et al. [35] combined a novel GA-SVM to find features from biological sequences. A particle swarm optimization (PSO) [36] to optimize a GA-SVM method to predict single nucleotide polymorphisms (SNPs) and to select tag SNPs as pointed by Ilhan and Tezel [37] or to solve the heating system planning problem is presented [38]. Zhang et al. combined PSO with SVM for classifying magnetic resonance imaging (MRI), brain images [39]. Ocak [40] combined a GA with a SVM in a medical decision support system for the evaluation of fetal wellbeing. Uzer et al. [41] combined an artificial bee colony algorithm with SVM for classification. A hybrid approach with SVM and microarray data is presented by Li et al. [42]. A cross-study comparison of classification methods including ANN and SVM for predicting metastasis in breast cancer is presented by Burton et al. [43]. Other approaches using evolutionary computation techniques are presented in [44, 45].

3. Methods

3.1. Selection of Variables. Currently, there are several variable selection techniques. One of the guidelines used for their classification is given according to the approach used: the indirect or filter approach on the one hand [46] and the direct or wrapper approach [47] on the other hand. The filter techniques select subsets of variables in a preprocessing step regardless of the classification problem. On the contrary, the wrapper methods use machine learning to assign a rating to the subsets of samples according to their predictive ability. There is a third group of techniques, called embedded, which perform the selection of variables during the very learning process or classification of samples.

Once the type of variable selection technique to be used is defined, it is necessary to establish a mechanism which allows carrying out the search for significant variables. Ideally, the selection should be performed in terms of the entire subset of variables that can be formed, but however, this would involve analyzing numerous combinations, with their corresponding computation time loss. That is why we use search strategies that provide results as close as possible to the overall optimum value. As shown in Section 3, different methods have been developed in order to explore the set of variables, among which the best known is the one based on the use of GAs [48, 49].

Several studies have proposed the combination of GAs and selection and prediction methods such as selection techniques and data prediction, respectively. One of the selection-prediction methods most used in these hybrid models is the ANNs [50]. However, these models are not free of drawbacks. One of the criticisms associated with the use of the ANNs is due to their "black-box" characteristic, as it is difficult to understand their internal operation and the process which leads them to determine the appropriate solution when it comes to a set of patterns. Another drawback of their use as a selection or prediction method, and perhaps the most important one, is the one concerning the exact nonreproducibility of their training (or at least of their difficulty) as a result of the stochastic process by which the weights are initialized. GAs are search techniques inspired by Darwinian Evolution and developed by Holland in the 1970s [51]. GAs for feature selection were first proposed by Siedlecki and Sklansky [52]. Many studies have been done on GA for feature selection since then [6, 53], concluding that GA is suitable for finding optimal solutions to large problems with more than 40 features to select from. GA for feature selection could be used in combination with a classifier such SVM, optimizing it.

Today, there are other prediction-selection techniques that allow making up for the shortcomings caused by the use of ANNs. This paper is aimed at emphasizing a method recently developed within the field of machine learning, the SVM. This is a classification and regression algorithm family which currently shows comparable or better results than those obtained with ANNs or other statistical models in problems of pattern recognition, prediction, classification, or data mining. From their inception to the present day, the SVM have evolved in such a way that they have become a successful tool when dealing with highly dimensional data.

3.2. Genetic Algorithms. The evolutionary computation (EC) has reintroduced concepts of evolution and genetics to solve problems, mainly those related to optimization tasks [48, 54]. However, it is also worth mentioning the important influence of the studies carried out by Turing and Samuel in the 1950s, "Can machines think?" and "How can computers learn to solve problems without being explicitly programmed?" [55, 56].

Broadly speaking, the EC methods are search and optimization techniques consisting of the application of heuristic rules based on principles of natural evolution. In other words, these are algorithms that look for solutions according to properties of genetics and evolution. Among these properties, it is worth mentioning the survival of the fittest individuals (which implies that once the best solutions to a problem are reached, they will keep being this way) and heterogeneity (we mean basic heterogeneity, so that algorithms could have numerous types of information when creating solutions).

3.2.1. General Outline of Operation. The evolutionary algorithms base their operation on a relatively simple outline, as shown in Figure 1. This iterative outline will refine solutions which will gradually be closer and closer to obtaining an overall solution of the problem.

But prior to the implementation of the evolutionary process specified by the algorithm, it is necessary to undertake two issues, perhaps the most important ones in the whole process: determining how to represent the solutions (encoding) and specifying a method which would allow us to know how good a solution is (fitness function).

One of the most used branches in the EC is made up of genetic algorithms (GAs) [49, 57]. In this case, the encoding of the solutions is performed through value chains (the chains being of fixed or variable length, with the values of these being bits, whole and real numbers, etc.).

The critical step when setting up an evolutionary algorithm is the definition of the fitness function. This function
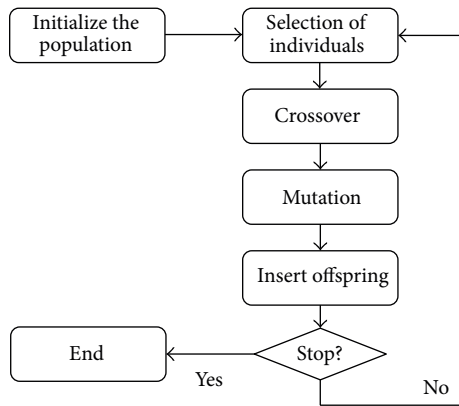
FIGURE 1: General outline of the operation of an evolutionary algorithm.

will have to evaluate every genetic individual, indicating a real value representing the goodness of the solution provided by the individual. This function will be responsible for guiding the search process in either direction. Precisely because we are dealing with a function responsible for verifying the goodness of each solution, this is an inherent aspect linked to the problem to be solved.

The above discussed evolution of solutions will happen due to crossover and mutation genetic operators which simulate processes of sexual and asexual reproduction that occur in a natural environment. Next, we will summarize each one of the remaining steps of which such algorithms consist.

Initialization. The power of evolutionary algorithms lies in the massively parallel exploration of the search space. This can occur due to the existence of numerous solutions, each exploring an area of the search space. The set of solutions, randomly initialized, is called genetic population.

Selection. The selection algorithms will be responsible for choosing the individuals which will and which will not have the opportunity to reproduce [58]. Since this is a simulation of what occurs in the natural environment, the fittest individuals have to be given more opportunities to reproduce. Therefore, the selection of an individual is related to its fitness value. However, the reproduction options of less fit individuals should not be completely eliminated, because in a few generations the population would become homogeneous in this way.

Crossover. Once the individuals are selected, they are recombined to produce offspring that will be inserted into the next generation. Their importance for the transition between generations is high because the usually employed crossover rates are around 90%.

The main idea of the crossover is based on that, if two individuals, properly adapted to their environment, are selected and the offspring obtained shares genetic information from both, there is a possibility that the inherited information is precisely the cause of their parents' goodness. By sharing the good features of two individuals, the offspring, or at least part of them, should have better characteristics than each parent separately.

Mutation. An individual's mutation causes that the value of one of its genes or nodes, usually only one of them varies randomly.

Although individuals can be selected directly from the current population and mutated before being introduced into the new population, the mutation is often used together with the crossover operator. Thus, the behavior that occurs in the natural environment is simulated, since when generating the offspring there is always some kind of error, usually with no consequence throughout the transmission of the genetic load from parents to offspring. The mutation causes sometimes a reduction in the individual's fitness value (which can be remedied in subsequent generations). However, the new information contributes directly to a significant increase of the goodness of the solutions or it can be a part of a better solution in future generations.

Replacement. The traditional operation of ANNs often includes the use of a temporary population. This latter is being filled by copying individuals and with the offspring generated due to crossover operations (and mutation, if that is the case). When this temporary population is complete—in this case it is said to have passed to a new generation—it becomes the population of the current generation, ruling out the previous one and repeating the process from a new empty temporary population. Such algorithms are usually called generational algorithms.

However, there is another approach, called steady-state algorithms. This option consists of working with a single population, which undergoes selections and insertions, ruling out the use of a temporary population. In this case, since the number of individuals in the population remains constant, it should be noted that a new individual cannot be added unless another is eliminated before that.

Stopping Criterion. As previously explained, the evolution process of solutions is essentially an iterative process. Therefore, it will be necessary to specify a criterion that allows establishing when the execution is completed. Once more, there are different options, but the most common ones are shown as follows.

(i) The fittest individuals in the population represent solutions good enough so that the problem could be solved.

(ii) The population has converged. A gene has converged when 95% of the population has the same value (or a very similar one) for that gene. Once all the genes reach convergence it is said that the population has converged. When this phenomenon happens, the average goodness of the population is close to the goodness of the fittest individual.

(iii) The difference of the best solutions found between different generations is reduced. This may indicate, at the very best, that the population has reached an overall solution or on the contrary that the population has come to a standstill at a local minimum value.

(iv) A predetermined maximum number of generations have been reached.

It may be worth mentioning that the advantage of such techniques is the simplicity of their implementation. No technical knowledge is required to solve the problem, only one way that allows evaluating a possible solution (in order to define the fitness function). Moreover, it should be also noteworthy the simplicity of the ideas taken from the natural environment, on which the evolution of solutions is based.

In addition, this type of techniques is easily adaptable to multimodal problems (those with multiple solutions) [59] or multiobjective problems (those in which different criteria are optimized simultaneously) [60].

When the computational cost is a criterion to be considered due to its inherently parallel operation, we are dealing with easily distributable techniques (or at least the evaluation of the solutions, which often becomes a hurdle) with a marked improvement in the response time arising from such distribution.

Finally, we should note that these techniques, unlike others, always provide a solution to the problem raised and, in addition, this solution will be improving as implementation is carried out over time.

3.3. Support Vector Machines. The support vector machines are general methods for solving problems of classification, regression, and estimation. They are learning systems based on the studies performed by Vapnik on the statistical learning theory [24, 25]. From their inception to the present day, they have become the subject of continuous research and application. The interest raised by this method has increased considerably, becoming a referent for the other disciplines of machine learning and data mining.

At first, the SVM were developed to solve problems of binary classification (two classes), but currently, and throughout their evolution, they have widened their field of action, dealing with any kind of problems. The SVM are aimed at finding a linear optimal hyperplane distributing the data into two or more classes, so that all those elements which belong to the same class are located on the same side. This is equivalent to solving a classical quadratic programming problem, which guarantees the existence of a single solution and a reasonable efficiency for real problems with thousands of examples and attributes.

Intuitively, it seems obvious to come to the conclusion that when solving a linear classification problem, there is a high probability of obtaining several solutions which could correctly classify the information, as shown in Figure 2.

Therefore, the question to be answered is which of the alternatives is the ideal one? In his studies, Vapnik answered this question by defining the concept of optimal hyperplane. "A hyperplane is said to be optimal if it maximizes the margin over all hyperplanes (see Figure 3) [61].
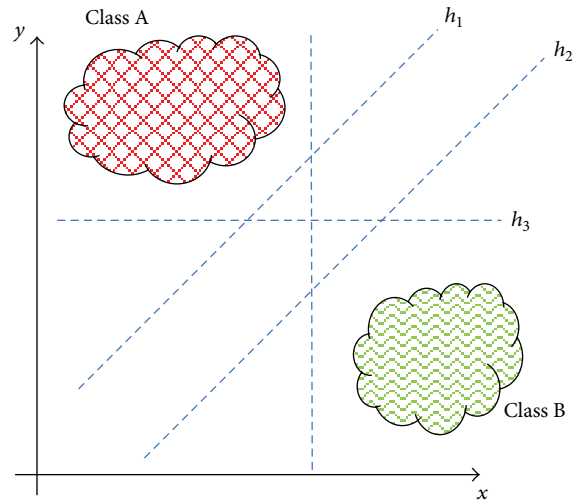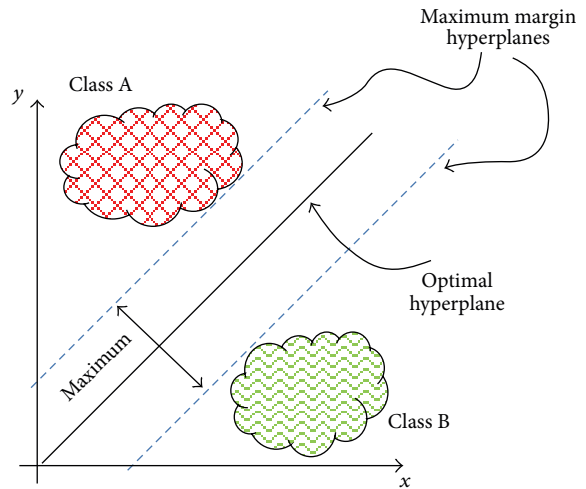

FIGURE 2: Linearly separable classification.

FIGURE 3: Linearly separable classification; hyperplane and separation margin.

Once defined the concept of optimal hyperplane, and after carrying out several studies, it was observed that the hyperplane could be defined only if considering certain data from the training set. These characteristic points are called "support vectors," and they are those instances of each class which are closest to the hyperplane with maximum margin.

However, in most of the existing problems, the data are not linearly separable, so that the implementation of the above-mentioned process does not achieve a good result. To solve this drawback, we should tackle these problems with different strategies, thus achieving a linear separation but in a different space. To this end, a transformation of input variables is performed in a dimensional space greater than the one to which they belong (the greater dimensional space being a Hilbert space):

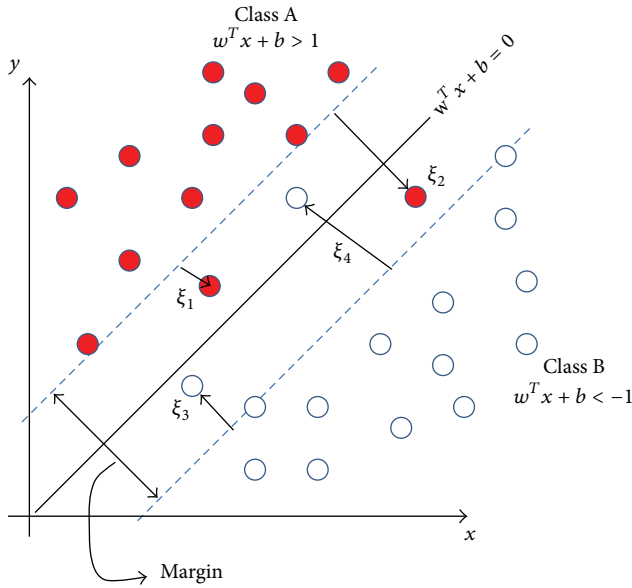$$x : \mathbb{R}^n \longmapsto \varphi(x) : \mathscr{H}. \tag{1}$$

FIGURE 4: Slack variables.

The next step is to find a hyperplane (actually a scalar product of vectors that can be expressed as a function of the input space x) in this new dimension that allows separating the data linearly. The result of this scalar product is called kernel, and the most common ones are as follows:

(i) linear kernel: $K(x_i, x_j) = x_i^T x_j$,

(ii) Gaussian kernel: $K(x_i, x_j) = \exp(\| x_i - x_j \|^2 / 2\sigma^2)$,

(iii) polynomial kernel (order n): $K(x_i, x_j) = (x_i^T x_j + 1)^n$.

In general, a kernel is any function $K(u, v)$ that verifies Mercer's theorem [62], that is, any function that verifies

$$\int_{u,v} K(u, v)\, g(u)\, g(v)\, du\, dv > 0 \tag{2}$$

for every $g(\cdot)$ function of integrable square.

Given the above, if the transformed function gives rise to a linearly separable space search, Vapnik and Chervonesky [63] showed that maximizing the separation margin between classes is equivalent to the minimization of the Euclidean norm of the weight vector. That is, considering the approximation of the set $\{x_i, y_i\} \text{con} x \in \mathbb{R}^n$, $y \in \mathbb{R}$ by the linear function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$,

$$f(x) = w^T x + b = \sum_{i=1}^{n} w_i x_i + b, \tag{3}$$

where $w \in \mathfrak{R}^n$ (weight vector) $yb \in \mathfrak{R}^n$ (vector of bias) are the parameters which define the hyperplane, as the lowest training and complexity error is obtained looking for the minimal $w \in \mathfrak{R}^n$.

But what happens when some datum is still not linearly separable in this new dimension? As shown in Figure 4, in this case the solution is to introduce a new set of slack

variables: ξ_i, $i = 1, \ldots, N$, representing an estimate of the error on the optimal hyperplane.

Therefore, the problem leads to the search of a classification function $f(x)$ that minimizes the sum of these losses reflected by the slack variables.

In this case, the function to be minimized would be as follows:

$$\Phi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i, \tag{4}$$

where C is a variable empirically specified by the user with the aim of controlling the tradeoff between the model complexity and the number of not separable data. More specifically, the greater is the value of this parameter, the higher is the assigned penalty to errors. Depending on the value of C, the margins of a boundary decision will vary their forms. As a result, we can conclude that the higher is the value, the narrower is the margin and the lower is the classification error in the training phase. On the contrary, the wider is the margin, the higher is the classification error in the training phase.

4. Proposed System

The method proposed in this work is based on creating a hybrid model that combines a GA and support vector machines, with the aim of classifying samples before selecting the minimum number of significant variables.

The GAlib library [64], developed by Matthew Wall in 1996 and last modified in 2007, was used to encode the genetic algorithms. Similarly, the WinSVM code was used to implement vector machines, a code developed by Sewell [65]. WinSVM provides as output a mean squared error (MSE) to measure the distance between the samples incorrectly classified on the optimal hyperplane. Thus, the obtained MSE value will be deterministic; hence, this will be one of the criteria which will be subsequently considered when comparing different executions.

According to those mentioned in the previous section, the SVM are extremely useful when trying to make dichotomous classification of data, that is, to distinguish between two classes. However, this idea can be generalized to identify, among a set of n categories, to which a certain datum belongs. In this work we have chosen to raise the following approach: considering that in the total data set n categories $(C_1, C_2, \ldots, C_n)$ can be defined for each possible C_i category existing in the input set, and an SVM is created. This latter will try to distinguish whether a datum belongs to the given C_i category or to the remaining set. Finally, to determine to which specific category each datum belongs, we have simply implemented all the defined SVM and we have selected that output which indicated a greater degree of belonging to a particular class (e.g., that in which the MSE value is lower).

With the aim of creating the above-mentioned hybrid model, we have modified the traditional operation of the GA in such a way that it could generate a population of individuals of varying length [66]. To this end, we have implemented an initialization function that will be responsible
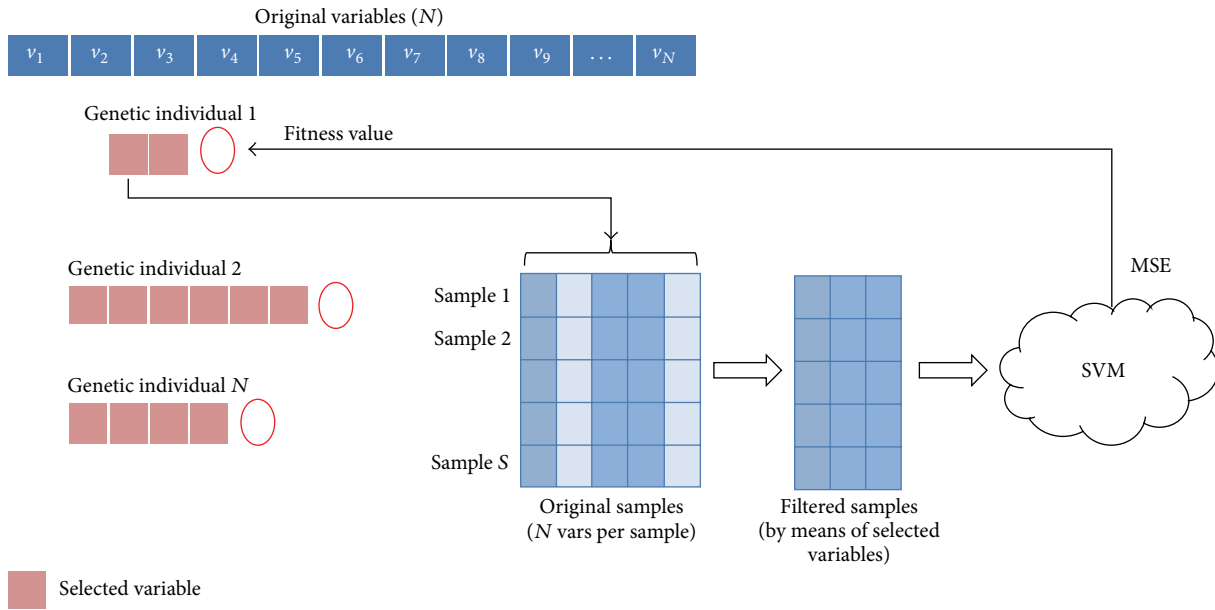
FIGURE 5: Evaluation of the genetic individuals.

```
M = data dimension
N = population size
for i = 1 to N
    perm = randomPermutation(M)
    for g = 1 tonumberOfGenes
        individual[i]·Gene[g] = perm[g]
    end for
end for
```

ALGORITHM 1

for, firstly, either performing a random generation or with a predetermined size, of the length of each individual making up the population of the genetic algorithm and, secondly, initializing the value of each gene. For this purpose, we have selected from the total set of variables a subset of random size so that, if the variables generated are different, the subset will be assigned to the individual. Otherwise, a new random combination will be generated until the condition stipulated is met. This procedure is repeated until all individuals in the GA population have an assigned subset. Therefore, the result is a set of individuals as shown in Algorithm 1.

Once the GA population is created, it will be assessed by a fitness function.

In this paper, we suggest the use of the SVM as a fitness function of the genetic algorithm. Thus, for each individual, and depending on the indicated positions, a training and validation set is created from the initial data. These sets will be applied to the SVM which, once the prediction is made, will yield a mean squared error (MSE) to be used as a scale to determine the fittest individual (see Figure 5).

Nevertheless, when the information refers to a nonbinary (or dichotomic) classification, it is necessary to modify a small aspect. It will be supposed that the problem has N classes $\{c_1, c_2, \ldots, c_n\}$. In this case an individual SVM is applied, using the variables specified by the genetic individual, to discriminate between each of the classes c_i and the rest a MSE being obtained. In this case of multiclass problems, the value of kindness of the individual will be the sum of the MSE obtained for every classification.

This choice of the fitness function allows, among other advantages, a repeatability of results, which hardly ever applies to other techniques such as artificial neural networks, used in similar problems tackled in previous studies [13, 67, 68].

Therefore, this paper is a step forward regarding previous studies on selection of variables in an experimental field on which numerous tests have been carried out. Still, as shown below, the obtained results significantly improved the previous ones.

5. Materials

5.1. Data Description. Nowadays the society awareness has evolved into the need for a more and more healthy diet in order to improve the quality of life. Undoubtedly, the juice manufacturing industry, influenced by these new circumstances, has enjoyed a boom in both production and sales. However, the increasing production of these industries leads to an increase regarding the level of adulteration of their products. Consequently, the search for new methods that allow identifying the exact amount of pure juice used to produce these products has become an issue of great importance in recent years [69].

In order to prevent and detect adulteration in food, this latter must be subjected to an increasingly strict series of quality control tests. This is due to the fact that the commonly used analysis techniques have become obsolete

TABLE 1: Samples with high concentration.

	Range: 20%–100%		
%	Training	Test	Commercial
20%	20	6	
25%	19	18	2
50%	16	13	
70%	14	1	
100%	17	6	19
Total	**86**	**44**	**21**

with their development and progress. Different techniques such as HPLC (*high performance liquid chromatography*) gas chromatography, or isotope methods are too slow and relatively expensive and, therefore, they are not suitable for carrying out routine analysis. On the other hand, IR (infrared) spectroscopy provides a quick and cheap alternative, which, besides these already mentioned characteristics, provides great information about the main components of the juice.

Therefore, in order to perform testing, we have used information that allows verifying the authenticity of the apple juice quality. Using various types of apples, such as Golden Delicious, Gloster, Granny, Smith, Reineta, Royal Gala, and Starking, their juice has been extracted for subsequent centrifuging, filtering and classification using Fourier transform mid-infrared attenuated total reflectance (FTMIR-ATR). As a result, we have obtained a series of samples of diluted pure juice, which will be used to obtain the different training and validation sets.

Two sets of samples will be considered: one for samples with high concentration of juice (see Table 1) and one for samples with low concentration of juice (see Table 2). Consequently, beverages with a low concentration of juice fall within what is called energy drinks (among which soft drinks are included), while beverages with a higher concentration receive the generic name of juices. The samples made up of 20% of diluted juice (the boundary between what is considered low and high concentrations) are found in both sets.

All samples were characterized by means of an infrared spectroscopy. As a result of this characterization, a spectrum (as shown in Figure 6) is obtained for each sample, which represents the amount of energy absorbed (or absorbance) for a total of 176 wavelengths or variables.

The objective is to determine the amount of juice in a sample using only the information provided by this spectrum. However, using all the information of the spectrum does not lead to fully satisfactory results, as shown below. That is why the need for applying a process of selection of variables is arisen. This would have two obvious advantages. On the one hand, it is time-saving (both when obtaining the spectrum and in the subsequent classification, since a smaller amount of information is involved). On the other hand, and perhaps most importantly, the expert is provided further information about which part of the spectrum, that is, what specific type
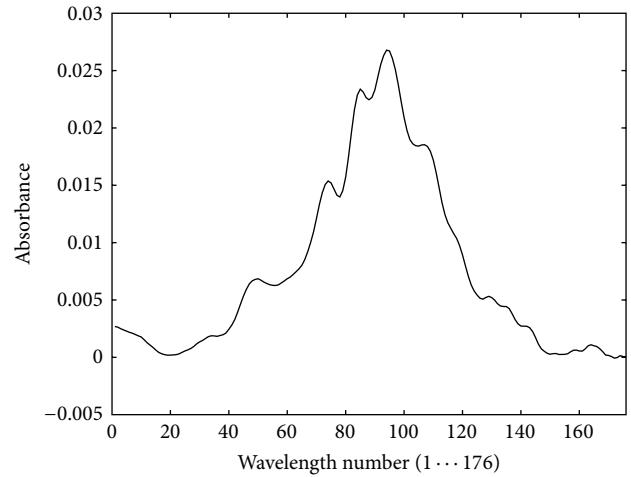

FIGURE 6: IR spectrum, specific to a sample.

TABLE 2: Samples with low concentration.

	Range: 2%–20%		
%	Training	Test	Commercial
2%	19	1	
4%	17	1	
6%	16	13	
8%	22	6	
10%	21	6	1
16%	20	6	1
20%	19	6	
Total	**134**	**39**	**2**

of sugar (fructose, sucrose, etc.), provides more information to carry out this classification.

5.2. Parameter Setting. The first experiment that should be performed before running the developed model is the choice of setting parameters for both vector machines and GA. To this end, we have applied the following procedure.

To determine the setting parameters of the SVM algorithm, the steps proposed by the author have been followed, using the "optimize" option provided by the algorithm itself. This function is based on a random generation of different combinations of the SVM parameters, which are then applied to the initial data obtaining a mean squared error (MSE) as a result of the application.

A total of 100 different combinations were generated for this purpose, and from among them, and taking as scale the training MSE, we have chosen those with the best results in the training cases. However, when performing this first test, there were no entirely satisfactory solutions (see Table 3). On the other hand, this fact is not really significant since the final aim in this phase is to determine the configuration parameters of the algorithm, not to carry out a real data classification.

Tables 3 and 4 show the three best combinations obtained from the 100 tested (for both high and low concentration),

Table 3: Setting of SVM parameters: samples with high concentration.

	SVM parameters			MSE train
	C	Epsilon	Radial	
1°	10000	0,000001	5	$3,83E-07$
2°	10000	0,0001	2	$4,49E-07$
3°	10000	0,0001	2	$4,49E-07$

Table 4: Setting of SVM parameters: samples with low concentration.

	SVM parameters			MSE train
	C	Epsilon	Radial	
1°	1000	0,000001	2	$5,26E-07$
2°	1000	0,000001	10	$5,95E-07$
3°	1000	0,0001	2	$5,95E-07$

selecting as optimal the one whose training mean squared error is lower. It is noted that the penalty variable C, in both cases, has a relatively high value ($C = 10,000$ for high concentration and $C = 1,000$ for low concentration), leading to, as mentioned above, a considerable reduction of the error training (3.83E-07 and 5.26E-07, resp.) finally obtained.

The configuration of the genetic algorithm parameters, carried out similarly as in the case of the SVM, should be selected after performing a series of tests by varying their value. From the range of tests, the configuration shown in Table 5 was taken as optimal.

6. Results and Discussion

First, to allow the comparison of results, a reference model should be established. In this case, we have chosen the original set of 176 variables provided by the IR spectrometer to build different classification models starting from the former. More specifically, we have chosen some of the most widely used models in the field of analytical chemistry with this type of data (partial least squares: PLS, SIMCA or potential functions), together with a model generated from the use of ANN and another one using SVM for the classification. Tables 6 and 7 show the results obtained with each of these models.

At first, given the great performance offered by the ANNs, we had chosen this technique to guide the search for the GA. However, as discussed below, they have the great disadvantage of the repeatability of results, which leads to the fact that the variables selected as the most relevant are different in each iteration.

On the other hand, the results obtained in this first test using SVM can seem discouraging. However, this can be explained according to the above classification criteria. Thus, for a given concentration, SVM are applied for each type of concentration (20%, 25%, . . .) obtaining a good or bad classification result regarding the individual classification. A good result in the final classification of the sample will be taken into account only if the datum is always assigned (i.e., for all the developed SVM) to the right category. For example,

Table 5: Genetic algorithm configuration.

(i) GA: simple	(i) Selection: wheel-roulette
(ii) Number of individuals: 100	(ii) Number of generations: 100
(iii) Crossover rate: 90%	(iii) Crossover: uniform
(iv) Mutation rate: 10%	(iv) Mutation: uniform

Table 6: High concentration: classification errors using 176 variables.

	Training (86 samples)	Validation (44 samples)	Commercial (21 samples)
PLS	11	5	1
SIMCA	15	12	1
Potential functions	4	6	0
ANN	0	1	0
SVM	0	39	21

Table 7: Low concentration: classification errors using 176 variables.

	Training (134 samples)	Validation (39 samples)	Commercial (2 samples)
PLS	29	11	0
SIMCA	19	14	0
Potential functions	4	9	0
ANN	0	4	0
SVM	0	33	1

a classification of a datum with 20% juice is considered correct only if the SVM that discriminate the category of 20% allocate it to that category, while the remaining defined SVM (each for the discrimination of one category) always assign the datum to the class of "other categories" (among which there is one of 20%). If, on the contrary, the sample is incorrectly characterized by a SVM, then the end result will be regarded as wrong.

As mentioned earlier, the results achieved in this test using the SVM are not satisfactory. Undoubtedly, any of the techniques employed for performing the classification (ANN, PLS, . . .) achieve better results than the SVM.

Previous studies, based on the use of a tool widely employed in the study of chemical data as Procrustes rotation [70], have allowed establishing that the information provided by two of the original variables would be enough to categorize the data. The existence of overinformation and the difficulty that this involves when generalizing may be some of the causes of the poor results provided by SVM. Thus, selecting the most important variables from their original set we may obtain similar or better results than using all the data. This is due to the fact that, in many cases, not all the original variables have the same influence, or they contain redundant information, when determining the actual content of juice in a sample.

Therefore, a second criterion of comparison or a reference model is established, in this case one consisting of the classifications made only from the two significant variables

TABLE 8: Classification errors using variables selected by Procrustes rotation. High concentration.

	Training (86 samples)	Validation (44 samples)	Commercial (21 samples)
PLS	6	6	1
SIMCA	44	27	8
Potential functions	5	6	0
ANN	6	8	0
SVM	5	3	3

TABLE 9: Classification errors using variables selected by Procrustes rotation. Low concentration.

	Training (134 samples)	Validation (39 samples)	Commercial (2 samples)
PLS	29	9	0
SIMCA	36	17	0
Potential functions	29	13	0
ANN	28	17	2
SVM	13	4	1

TABLE 10: Classification Errors using GA + ANN: high concentration.

High concentrations			
Selected variables	Train (86 samples)	Test (44 samples)	Commercial (21 samples)
[52, 141]	4	8	1
[102, 129]	10	14	2
[23, 67]	5	16	1
[18, 120]	3	12	2

TABLE 11: Classification errors using GA + ANN: low concentration.

Low concentrations			
Selected variables	Train (134 samples)	Test (39 samples)	Commercial (2 samples)
[52, 141]	19	17	2
[102, 129]	10	14	2
[23, 67]	5	16	1
[18, 120]	3	12	2

previously selected by Procrustes (specifically, the variable 94 and the variable 95).

The results for each of these models are shown in Tables 8 and 9.

In this case, the proper operation of SVM is shown with a smaller version of the data set provided by IR spectroscopy, reducing the number of errors obtained so far in other studies. Therefore, its use as a measure of goodness seems feasible when guiding the implementation process of variable selection by means of GAs.

As previously mentioned, the initial goal consists of determining by GAs a series of sets obtained from the original variables that allow the correct classification of a sample. Furthermore, by not providing a single solution—as does Procrustes Rotation—further information is provided to the expert about the importance of different regions of the spectrum. First, the GA is configured to generate individuals made up of only 2 genes, each of them representing each of the variables to be considered for performing the classification.

In previous studies [13, 17, 67, 71], due to the versatility of ANNs, this technique has been chosen as the evaluation function used by the GA. Thus, the GA selects variables considered as *significant*, which will be provided as inputs to an ANN that is responsible for classifying the samples according to their amount of juice. The MSE obtained in the training process of the ANN is used as fitness to determine the goodness of each individual gene.

Tables 10 and 11 show the best results obtained with this technique after various tests and configurations. We can observe that the results improve significantly compared to those obtained when classification is made according to the variables selected by Procrustes Rotation.

However, this approach raises a problem. Due to random initialization process of an ANN, the obtained results are

hardly repeatable. To solve this problem, this paper proposes the use of SVM as a fitness function employed to guide the search for the GA. On this basis, we propose two approaches for defining the fitness function. On the one hand, the training mean squared error (MSETrain) is used as a fitness function, while on the other hand, the sum between the training MSE and the validation MSE (MSETrain + MSETest) is used as a fitness function.

Depending on the selected option as a fitness function, we have obtained the results shown in Tables 12 and 13, which include the four best solutions reached during the execution of the hybrid system using the two possibilities raised here as a fitness function. They show the predictive variables, as well as the errors made during the training and validation phases. As it can be seen by analyzing the results, regardless of the fitness function chosen, the obtained results are not only improved when the classification is done starting from the variables selected by Procrustes, but they are comparable to those obtained from the initial data set (unlike other techniques). In addition, compared to the use of ANN, we can count on the advantage offered by the repeatability of results.

Comparing these results with those previously shown (Tables 10 and 11), we can observe that the results in the training phase are very similar, regardless of the use of ANN or SVM in the evaluation function. However, the results obtained by SVM on the validation and testing data significantly improve the ones obtained with ANN, coming to the conclusion that greater generalization ability is achieved.

On the other hand, according to the results presented in Tables 12 and 13, one can sense that the use of the sum of both MSE (MSETrain + MSETest) as a fitness function of the hybrid system is able to reduce the number of errors made during the classification. The graph shown in Figure 7 performs a comparison between the two implemented functions. The data displayed correspond to the average of errors made during the SVM classification of all the executions carried out

TABLE 12: Classification errors using GA + SVM: high concentration.

Selected variables	Train (86 samples)	Test (44 samples)	Commercial (21 samples)
	MSE train		
[4, 69]	5	5	2
[81, 84]	9	5	2
[68, 28]	4	6	2
[91, 101]	4	6	1
	MSE train + MSE test		
[9, 105]	5	3	3
[9, 113]	5	4	9
[34, 101]	4	5	11
[86, 104]	5	5	4

TABLE 13: Classification errors using GA + SVM: low concentration.

Selected variables	Train (134 samples)	Test (39 samples)	Commercial (2 samples)
	MSE train		
[1, 107]	15	5	1
[5, 91]	16	5	1
[2, 107]	17	5	1
[11, 97]	14	6	1
	MSE train + MSE test		
[5, 73]	13	4	1
[27, 90]	14	4	1
[2, 94]	15	4	1
[27, 88]	14	5	2

with the hybrid system. A slight superiority is observed when using the sum of both MSE as a fitness function, especially in the validation phase.

7. Conclusions

This paper has proposed the development of a hybrid model combining GAs and SVM. The aim pursued by using this approach is the identification, from the initial data set, of the smallest number of variables possible that allow determining optimally the amount of juice within a sample. To this end, the method proposed herein uses a GA that employs SVM as a fitness function. This would lead to the generation of a fast, effective model that is capable of reproducing the data obtained.

With the aim of demonstrating the effectiveness of the proposed system, a series of experiments have been raised, with their corresponding validation tests. Observing the values obtained, we can come to the conclusion that the system improves significantly the results obtained so far with other classification methods.

As noted above, SVM are useful and effective when performing prediction data with a margin of error, previously cited as ξ. This evaluation mechanism has the undeniable advantage of the repeatability of results. Thus, each set of vari-
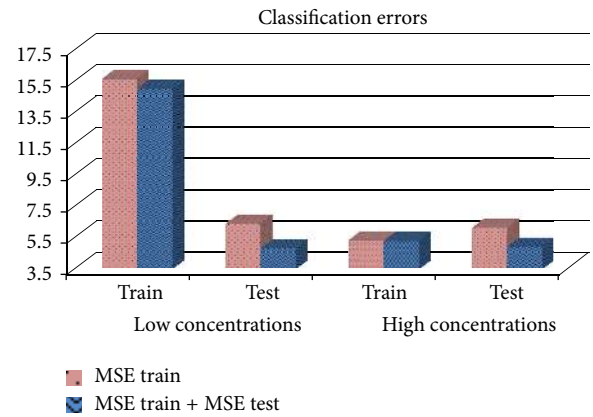

FIGURE 7: Classification errors according to the used fitness function.

ables will always have the same prediction system associated with the same fitness level, contrary to what would happen in the case of using ANN as a measure of goodness for genetic individuals.
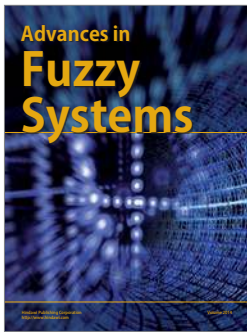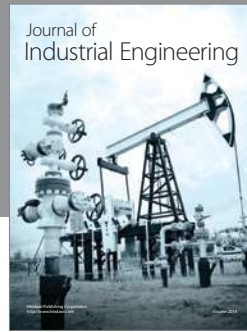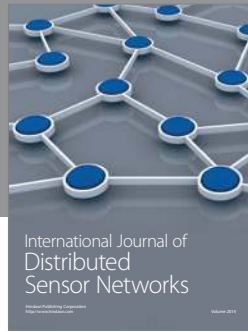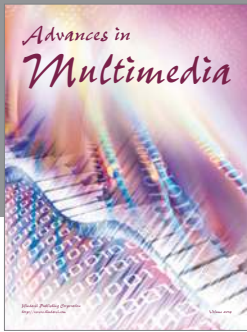
Acknowledgments

References

[1] I. Guyon, A. Elisseefi, and L. P. Kaelbling, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[2] P. Lewis, "The characteristic selection problem in recognition systems," *IRE Transactions on Information Theory*, vol. 8, pp. 171–178, 1962.

[3] G. S. Sebestyen, *Decision-Making Processes in Pattern Recognition*, ACM Monograph Series, 1962.

[4] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[5] F. Alonso-Atienza, J. L. Rojo-Álvarez, A. Rosado-Muñoz, J. J. Vinagre, A. García-Alberola, and G. Camps-Valls, "Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection," *Expert Systems with Applications*, vol. 39, no. 2, pp. 1956–1967, 2012.

[6] C. Fernandez-Lozano, J. A. Seoane, M. Gestal Pose, T. R. Gaunt, and C. Campbell, "Texture classification using kernel-based techniques," in *International Work Conference on Artificial Neural Network*, G. J. Ignacio Rojas and J. Cabestany, Eds., pp. 427–434, Springer, Puerto de la Cruz, Spain, 2013.

[7] I. T. Jolliffe, *Principal Component Analysis*, Springer, New York, NY, USA, 2 edition, 2002.

[8] M. Barker and W. Rayens, "Partial least squares for discrimination," *Journal of Chemometrics*, vol. 17, no. 3, pp. 166–173, 2003.

[9] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Proceedings of the International Conference on Subspace, Latent Structure and Feature Selection*, pp. 34–51, Springer, Bohinj, Slovenia, 2006.

[10] Z. Ramadan, D. Jacobs, M. Grigorov, and S. Kochhar, "Metabolic profiling using principal component analysis, discriminant partial least squares, and genetic algorithms," *Talanta*, vol. 68, no. 5, pp. 1683–1691, 2006.

[11] F. Tan, X. Feng, Z. Fang, M. Li, Y. Guo, and L. Jiang, "Prediction of mitochondrial proteins based on genetic algorithm—partial least squares and support vector machine," *Amino Acids*, vol. 33, no. 4, pp. 669–675, 2007.

[12] J. A. Freeman and D. M. Skapura, *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley, Reading, Mass, USA, 1991.

[13] M. Gestal, M. P. Gómez-Carracedo, J. M. Andrade et al., "Selection of variables by genetic algorithms to classify apple beverages by artificial neural networks," *Applied Artificial Intelligence*, vol. 19, no. 2, pp. 181–198, 2005.

[14] C. M. Bishop, *Neural Networks For Pattern Recognition*, Oxford university press, New York, NY, USA, 1995.

[15] C. Fernandez-Lozano, J. A. Seoane, P. Mesejo, Y. S. G. Nashed, S. Cagnoni, and J. Dorado, "2D-PAGE Texture classification using support vector machines and genetic algorithms," in *Proceedings of the 4th International Conference on Bioinformatics Models, Methods and Algorithms*, pp. 5–14, Scitepress, 2013.

[16] C. Fernandez-Lozano, J. A. Seoane, M. Gestal Pose, D. Rivero, J. Dorado, and A. Pazos, "A texture-based classification method for proteins in two-dimensional electrophoresis gel images," in *Proceedings of the International Conference on Computer Vision Theory and Applications VISSAP*, pp. 401–404, Scitepress, Barcelona, Spain, 2013.

[17] M. P. Gómez-Carracedo, M. Gestal, J. Dorado, and J. M. Andrade, "Chemically driven variable selection by focused multimodal genetic algorithms in mid-IR spectra," *Analytical and Bioanalytical Chemistry*, vol. 389, no. 7-8, pp. 2331–2342, 2007.

[18] K.-Y. Chen and C.-H. Wang, "Support vector regression with genetic algorithms in forecasting tourism demand," *Tourism Management*, vol. 28, no. 1, pp. 215–226, 2007.

[19] P.-F. Pai and W.-C. Hong, "Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms," *Electric Power Systems Research*, vol. 74, no. 3, pp. 417–425, 2005.

[20] K. K. Kandaswamy, G. Pugalenthi, S. Möller et al., "Prediction of apoptosis protein locations with genetic algorithms and support vector machines through a new mode of pseudo amino acid composition," *Protein and Peptide Letters*, vol. 17, no. 12, pp. 1473–1479, 2010.

[21] D.-K. Kang and M.-J. Kim, "Performance enhancement of SVM ensembles using genetic algorithms in bankruptcy prediction," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE '10)*, pp. V2154–V2158, August 2010.

[22] X. Chen and L. Wu, "Nonlinear demodulation and channel coding in EBPSK scheme," *The Scientific World Journal*, vol. 2012, Article ID 180469, 7 pages, 2012.

[23] Y. F. Huang and S. Y. Chen, "Extracting physicochemical features to predict protein secondary structure," *The Scientific World Journal*, vol. 2013, Article ID 347106, 8 pages, 2013.

[24] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[25] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, Chichester, UK, 1998.

[26] S.-H. Min, J. Lee, and I. Han, "Hybrid genetic algorithms and support vector machines for bankruptcy prediction," *Expert Systems with Applications*, vol. 31, no. 3, pp. 652–660, 2006.

[27] C.-H. Wu, G.-H. Tzeng, Y.-J. Goo, and W.-C. Fang, "A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy," *Expert Systems with Applications*, vol. 32, no. 2, pp. 397–408, 2007.

[28] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[29] E. Huerta, B. Duval, and J. K. Hao, "A hybrid GA/SVM approach for gene selection and classification of microarray data," in *Applications of Evolutionary Computing*, F. Rothlauf, J. Branke, S. Cagnoni et al., Eds., pp. 34–44, Springer, Berlin, Germany, 2006.

[30] R. Venegas, "Academic text classification based on lexical-semantic content," *Revista Signos*, vol. 40, no. 63, pp. 239–271, 2007.

[31] C. A. Donís, E. Valencia Morales, and C. Morell Pérez, "Support vector machine model for regression applied to the estimation of the creep ruptura stress in ferritic steels," *Revista Facultad de Ingenieri'a Universidad de Antioquia*, no. 47, pp. 53–58, 2009.

[32] L. Pérez, J. Mora, and J. Bedoya, "A linear approach to determining an SVM-based fault locator's optimal parameters," *Ingeniería e Investigación*, vol. 29, pp. 76–81, 2009.

[33] M. Tong, K. H. Liu, C. Xu, and W. Ju, "An ensemble of SVM classifiers based on gene pairs," *Computers in Biology and Medicine*, vol. 43, pp. 729–737, 2013.

[34] Z. Wang, Y. H. Shao, and T. R. Wu, "A GA-based model selection for smooth twin parametric-margin support vector machine," *Pattern Recognition*, vol. 46, pp. 2267–2277, 2013.

[35] K. J. Won, C. Saunders, and A. Prügel-Bennett, "Evolving fisher kernels for biological sequence classification," *Evolutionary Computation*, vol. 21, pp. 83–105, 2013.

[36] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[37] T. Ilhan and G. Tezel, "A genetic algorithm-support vector machine method with parameter optimization for selecting the tag SNPs," *Journal of Biomedical Informatics*, vol. 46, pp. 328–340, 2013.

[38] R. J. Ma, N. Y. Yu, and J. Y. Hu, "Application of particle swarm optimization algorithm in the heating system planning problem," *The Scientific World Journal*, vol. 2013, Article ID 718345, 11 pages, 2013.

[39] Y. Zhang, S. Wang, G. Ji, and Z. Dong, "An MR brain images classifier system via particle swarm optimization and kernel support vector machine," *The Scientific World Journal*, vol. 2013, Article ID 130134, 9 pages, 2013.

[40] H. Ocak, "A medical decision support system based on support vector machines and the genetic algorithm for the evaluation of fetal well-being," *Journal of Medical Systems*, vol. 37, aricle 9913, 2013.

[41] M. S. Uzer, N. Yilmaz, and O. Inan, "Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification," *The Scientific World Journal*, vol. 2013, Article ID 419187, 10 pages, 2013.

[42] L. Li, H. Chen, C. Liu et al., "A robust hybrid approach based on estimation of distribution algorithm and support vector machine for hunting candidate disease genes," *The Scientific World Journal*, vol. 2013, Article ID 393570, 7 pages, 2013.

[43] M. Burton, M. Thomassen, Q. Tan, and T. A. Kruse, "Gene expression profiles for predicting metastasis in breast cancer: a cross-study comparison of classification methods," *The Scientific World Journal*, vol. 2012, Article ID 380495, 11 pages, 2012.

[44] M. P. Poland, C. D. Nugent, H. Wang, and L. Chen, "Genetic algorithm and pure random search for exosensor distribution optimisation," *International Journal of Bio-Inspired Computation*, vol. 4, pp. 359–372, 2012.

[45] J. Muñuzuri, P. Cortés Achedad, M. Rodríguez, and R. Grosso, "Use of a genetic algorithm for building efficient choice designs," *International Journal of Bio-Inspired Computation*, vol. 4, pp. 27–32, 2012.

[46] D. Koller and M. Sahami, "Toward optimal feature selection," in *Citeseer*, pp. 284–292, 1996.

[47] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[48] J. H. Holland, *Adaptation in Natural and Artificial Systems*, 1975.

[49] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[50] S. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New York, NY, USA, 1999.

[51] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, 1975.

[52] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, no. 5, pp. 335–347, 1989.

[53] M. Kudo and J. Sklansky, "A comparative evaluation of medium- and large-scale feature selectors for pattern classifiers," *Kybernetika*, vol. 34, no. 4, pp. 429–434, 1998.

[54] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, "Genetic programming III: darwinian invention and problem solving [book review]," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 251–253, 1999.

[55] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433–460, 1950.

[56] A. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 11, pp. 601–617, 1967.

[57] M. Mitchell, *An Introduction to Genetic Algorithms*, The MIT press, 1998.

[58] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," *Urbana*, vol. 51, pp. 61801–62996, 1991.

[59] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *in Proceedings of the 2nd International Conference on Genetic Algorithms on Genetic Algorithms and their Application*, pp. 41–49, L. Erlbaum Associates, 1987.

[60] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems," *Evolutionary computation*, vol. 7, no. 3, pp. 205–230, 1999.

[61] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1–3, pp. 131–159, 2002.

[62] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, Cambridge, Mass, USA, 2002.

[63] V. Vapnik and A. Chervonenkis, "The necessary and sufficient conditions for consistency in the empirical risk minimization method," *Pattern Recognition and Image Analysis*, vol. 1, pp. 283–305, 1991.

[64] M. Wall, "GAlib: a C++ library of genetic algorithm components," Mechanical Engineering Department, Massachusetts Institute of Technology, 1996, http://lancet.mit.edu/ga/.

[65] M. Sewell, "WinSVM: a windows implementation of a support vector machine," 2005, http://winsvm.martinsewell.com/.

[66] U. Maulik and S. Bandyopadhyay, "Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 5, pp. 1075–1081, 2003.

[67] M. Gestal, A. Cancela, J. Andrade, and M. Gomez-Carracedo, "Several approaches to variable selection by means of genetic algorithms," in *Intelligent Information Technologies: Concepts, Methodologies, Tools and Applications*, V. Sugaraman, Ed., Information Science Reference, pp. 274–292, Hershey, New York, NY, USA, 2007.

[68] M. Gestal, M. P. Gómez-Carracedo, J. M. Andrade et al., "Classification of apple beverages using artificial neural networks with previous variable selection," *Analytica Chimica Acta*, vol. 524, no. 1-2, pp. 225–234, 2004.

[69] D. R. Heldman and D. B. Lund, *Handbook of Food Engineering*, CRC Press, New York, NY, USA, 2007.

[70] W. J. Krzanowski, *Principles of Multivariate Analysis: A User's Perspective*, Oxford University Press, New York, NY, USA, 2000.

[71] M. P. Gómez-Carracedo, M. Gestal, J. Dorado, and J. M. Andrade, "Linking chemical knowledge and genetic algorithms using two populations and focused multimodal search," *Chemometrics and Intelligent Laboratory Systems*, vol. 87, no. 2, pp. 173–184, 2007.