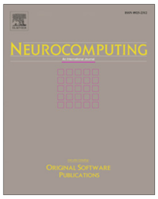
Neurocomputing 408 (2020) 189–215

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



A comprehensive survey on support vector machine classification: Applications, challenges and trends

Jair Cervantes [a,*], Farid Garcia-Lamont [a], Lisbeth Rodríguez-Mazahua [b], Asdrubal Lopez [c]

[a] UAEMEX (Autonomous University of Mexico State), Texcoco, 56259, Mexico
[b] Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, Veracruz 94320, Mexico
[c] UAEMEX (Autonomous University of Mexico State), Zumpango 55600, Mexico

ARTICLE INFO

Article history:
Received 2 June 2019
Revised 24 August 2019
Accepted 1 October 2019
Available online 08 May 2020

Keywords:
SVM
Classification
Machine learning

ABSTRACT

In recent years, an enormous amount of research has been carried out on support vector machines (SVMs) and their application in several fields of science. SVMs are one of the most powerful and robust classification and regression algorithms in multiple fields of application. The SVM has been playing a significant role in pattern recognition which is an extensively popular and active research area among the researchers. Research in some fields where SVMs do not perform well has spurred development of other applications such as SVM for large data sets, SVM for multi classification and SVM for unbalanced data sets. Further, SVM has been integrated with other advanced methods such as evolve algorithms, to enhance the ability of classification and optimize parameters. SVM algorithms have gained recognition in research and applications in several scientific and engineering areas. This paper provides a brief introduction of SVMs, describes many applications and summarizes challenges and trends. Furthermore, limitations of SVMs will be identified. The future of SVMs will be discussed in conjunction with further applications. The applications of SVMs will be reviewed as well, especially in the some fields.

1. Introduction

Machine Learning is a highly interdisciplinary field which builds upon ideas from cognitive science, computer science, statistics, optimization among many other disciplines of science and mathematics. In machine learning, classification is a supervised learning approach used to analyze a given data set and to build a model that separates data into a desired and distinct number of classes [1].

There are many good classification techniques in the literature including k-nearest-neighbor classifier [2,3], Bayesian networks [4,5], artificial neural networks [6–10], decision trees [11,12] and SVM [13–15]. K-nearest-neighbor methods have the advantage that they are easy to implement, however, they are usually quite slow if the input data set is very large. On the other hand, these are very sensitive to the presence of irrelevant parameters [2,3].

Decision trees have also been widely used in classification problems. These are usually faster than neural networks in the training phase, however, they do not have flexibility to modeling the parameters [11,12]. Neuronal networks are one of the most used techniques [16–20]. Neural networks have been widely used in a large number of applications as a universal approach. However, many factors must be taken into account to building a neural network to solve a given problem: the learning algorithm, the architecture, the number of neurons per layer, the number of layers, the representation of the data and much more. In addition, these are very sensitive to the presence of noise in the training data [21,22].

From these techniques, SVM is one of the best known techniques to optimize the expected solution [13,15]. SVM was introduced by Vapnik as a kernel based machine learning model for classification and regression task. The extraordinary generalization capability of SVM, along with its optimal solution and its discriminative power, has attracted the attention of data mining, pattern recognition and machine learning communities in the last years. SVM has been used as a powerful tool for solving practical binary classification problems. It has been shown that SVMs are superior to other supervised learning methods [23–29]. Due to its good theoretical foundations and good generalization capacity, in recent years, SVMs have become one of the most used classification methods.

Decision functions are determined directly from the training data by using SVM in such a way that the existing separation (margin) between the decision borders is maximized in a highly

[*] Corresponding author.
E-mail addresses: jcervantesc@uaemex.mx (J. Cervantes), fgarcial@uaemex.mx (F. Garcia-Lamont).

https://doi.org/10.1016/j.neucom.2019.10.118
0925-2312/© 2020 Elsevier B.V. All rights reserved.

dimensional space called the feature space. This classification strategy minimizes the classification errors of the training data and obtains a better generalization ability, i.e., classification skills of SVMs and other techniques differ significantly, especially when the number of input data is small. SVMs are a powerful technique used in data classification and regression analysis. A notable advantage of SVMs lies in the fact that they obtain a subset of support vectors during the learning phase, which is often only a small part of the original data set. This set of support vectors represents a given classification task and is formed by a small data set.

The rest of this paper is divided as follows: in Section 2 the theoretical basis of SVM are presented; in addition, their characteristics, advantages and disadvantages are described. In Section 3 weaknesses of SVM are introduced and reviewed. In Section 4 a set of SVM implementations are presented. Section 5 shows some applications of SVM in real world problems. Finally, Section 6 closes the paper with trends and challenges.

2. Theoretical basis of SVMs

The principal objective in pattern classification is to get a model which maximizes the performance for the training data. Conventional training methods determine the models in such a way that each input–output pair is correctly classified within the class to which it belongs. However, if the classifier is too fit for the training data, the model begins to memorize training data rather than learning to generalize, degrading the generalization ability of the classifier.

The main motivation of SVM is to separate several classes in the training set with a surface that maximizes the margin between them. In other words, SVM allows to maximizing the generalization ability of a model. This is the objective of the Structural Risk Minimization principle (SRM) that allows the minimization of a bound on the generalization error of a model, instead of minimizing the mean squared error on the set of training data, which is the philosophy often used by the methods of empirical risk minimization.

In this Section, we discuss Support Vector Machines, in which training data are linearly separable in the input space and the case where training data are not linearly separable.

2.1. Linearly separable case

Training a SVM requires a set of n examples. Each example consists of a pair, an input vector x_i and the associated label y_i. Assume that a training set X is given as:

$$(\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2),\ldots,(\mathbf{x}_n,\mathbf{y}_n) \tag{1}$$

i.e., $X = \{x_i, y_i\}_{i=1}^{n}$ where $x_i \in R^d$ and $y_i \in (+1,-1)$. For reasons of visualization, we will consider the case of a two-dimensional input, i.e., $x \in \mathbb{R}^2$. The data are linearly separable and there are many hyperplanes that can perform the separation. Fig. 1 shows several decision hyperplanes that perfectly separate the input data set. It is clear that there are infinite hyperplanes that could perform this work. However, the generalization ability depends on the location of the separation hyperplane and the hyperplane with maximum margin. This hyperplane is called *optimal separation hyperplane [14]*. The decision level, i.e., the hyperplane that separates the input space is defined by the equation $w^T x_i + b = 0$.

The simplest case of SVM is the linearly separable case in the feature space. We optimize the geometric margin by setting the functional margin $kappai = 1$(also called *Canonical Hyperplane* [30]), therefore, the linear classifier $y_i = 1$,

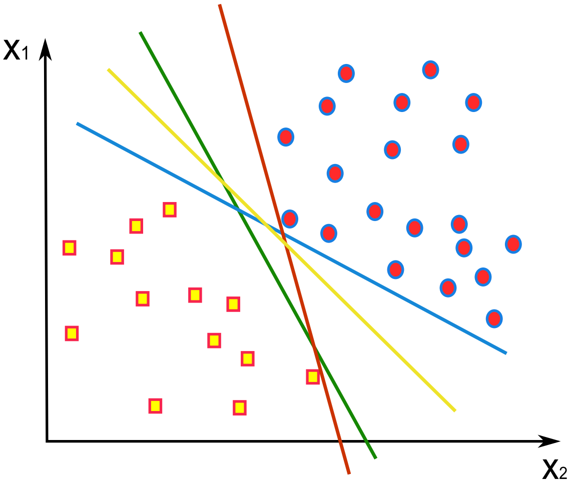


Fig. 1. Separation hyperplanes.

$$\begin{aligned} \langle w \cdot x^{+} \rangle + b &= 1 \\ \langle w \cdot x^{-} \rangle + b &= -1 \end{aligned} \tag{2}$$

These can be combined into a set of inequalities:

$$y_i(\langle w \cdot x_i \rangle + b) \geqslant 1 \forall i \tag{3}$$

The geometric margin of x^{+} y x^{-} is

$$\begin{aligned} \gamma_i &= \tfrac{1}{2}\left(\left\langle \tfrac{w}{\|w\|} \cdot x^{+} \right\rangle - \left\langle \tfrac{w}{\|w\|} \cdot x^{-} \right\rangle\right) \\ &= \tfrac{1}{2\|w\|}\left[\langle w \cdot x^{+} \rangle - \langle w \cdot x^{-} \rangle\right] \\ &= \tfrac{1}{\|w\|} \end{aligned} \tag{4}$$

where w defines the optimal separation hyperplane and b is the bias. The distance between the hyperplane and the training data closest to the hyperplane is called margin. The generalization ability is maximized if the optimal separation hyperplane is selected as the separation hyperplane. Optimizing the geometric margin means minimizing the norm of the vector of weights. When solving the problem of quadratic programming we try to find the optimal hyperplane and two parallel hyperplanes (H_1 and H_2).

The distance between H_1 and H_2 is maximized and there is no data between the two hyperplanes. When the distance between $H_1$and H_2 is maximized, some data points can be over $H_1$and some data points can be over H_2.These points of data are called *support vectors [14,30]*, since they participate directly in defining the separation hyperplane, the other points can be removed or changed without crossing the planes H_1 and H_2 and will not modify in any way the generalization skill of the classifier, i.e., the solution of an SVM is given only by this small set of support vectors. Any hyperplane can be represented by w, x and b, where w is a vector perpendicular to the hyperplane. Fig. 2 shows the geometric representation of the quadratic programming problem showing H (optimal separator) and hyperplanes H_1 and H_2. In this way, the original problem of optimization is as follows.

Proposition 1. For the linearly separable case $S = [(x_1, y_1) \cdots (x_l, y_l)]$, if (w, b) is the solution

$$\begin{aligned} &\min_{w,b} \langle w \cdot w \rangle = \|w\|^2 \\ &\text{subject to}: y_i(\langle w \cdot x_i \rangle + b) \geqslant 1 \end{aligned} \tag{5}$$

then the maximal margin is given by $\gamma = \frac{1}{\|w\|}$.

We change this to the dual problem using the Lagrange formulation. There are two reasons to do this. The first lies in the fact that the conditions given in the Eq. (5) will be replaced by Lagrange

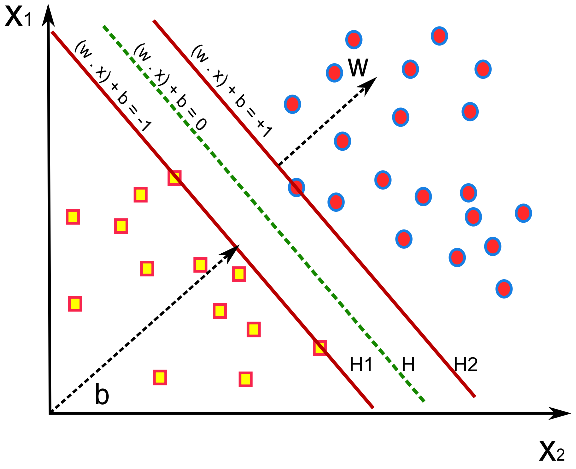


Fig. 2. Optimal classifier.

multipliers, which are much easier to handle. The second is the reformulation of the problem, the training data will only appear in the form of dot product between vectors. This is a fundamental property that will allow to generalize the procedure in the non-linear case. In this way, the Lagrangian is given by:

$$L(w,b,\alpha) \equiv \frac{1}{2}\langle w\cdot w\rangle - \sum_{i=1}^{l}\alpha_i[y_i(\langle w\cdot x_i\rangle + b - 1] \tag{6}$$

where α_i are the Lagrange's multipliers.

The dual is found in two steps: first, taking the derivative with respect to w and b

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{l}\alpha_i y_i x_i = 0 \rightarrow w = \sum_{i=1}^{l}\alpha_i y_i x_i \tag{7}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = -\sum_{i=1}^{l}\alpha_i y_i = 0 \rightarrow \sum_{i=1}^{l}\alpha_i y_i = 0 \tag{8}$$

and second, substituting Eqs. (7) and (8) in the original Lagrangian (6)

$$\begin{aligned} L(w,b,\alpha) &= \tfrac{1}{2}\langle w\cdot w\rangle - \sum_{i=1}^{l}\alpha_i[y_i(\langle w\cdot x_i\rangle + b) - 1] \\ &= \tfrac{1}{2}\left\langle \sum_{i=1}^{l}\alpha_i y_i x_i \cdot \sum_{i=1}^{l}\alpha_i y_i x_i \right\rangle - \sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j(\langle x_j\cdot x_i\rangle + b) - \sum_{i=1}^{l}\alpha_i \\ &= \tfrac{1}{2}\sum_{i,j=1}^{l}\alpha_i y_i\alpha_j y_j\langle x_i\cdot x_j\rangle - \sum_{i,j=1}^{l}\alpha_i y_i\alpha_j y_j\langle x_j\cdot x_i\rangle - \sum_{i=1}^{l}\alpha_i y_i b + \sum_{i=1}^{l}\alpha_i \\ &= -\tfrac{1}{2}\sum_{i,j=1}^{l}\alpha_i y_i\alpha_j y_j\langle x_i\cdot x_j\rangle + \sum_{i=1}^{l}\alpha_i \end{aligned} \tag{9}$$

The data points with $\alpha_i > 0$ are called "support vectors" and these vectors define the hyperplanes H_1, H_2. At all other training data $\alpha_i = 0$. Support vectors are the critical elements of the training data and these are the closest to the decision hyperplane.

Remark 1. SVMs obtain a hyperplane by training the initial data set, which perfectly separates these data and is defined by a small set of support vectors. If all other points (non-support vectors) were eliminated (or moved around without crossing H_1or H_2) and the training was repeated, the same hyperplane defined by the same set of support vectors would be found. Therefore, the original problem of optimization is as follows.

Proposition 2. To the linearly separable case $S = [(x_1, y_1)\cdots(x_l, y_l)]$, if α_i^* is the solution to the quadratic optimization problem

$$\begin{aligned} &\max_{\alpha_i} -\tfrac{1}{2}\sum_{i,j=1}^{l}\alpha_i y_i\alpha_j y_j\langle x_i\cdot x_j\rangle + \sum_{i=1}^{l}\alpha_i \\ &\text{s.t. : } \sum_{i=1}^{l}\alpha_i y_i = 0 \end{aligned} \tag{10}$$

Then $\|w\|^2$ defines the minimum $w^* = \sum_{i=1}^{l}\alpha_i^* y_i x_i$ and the geometric margin $\gamma^* = \frac{1}{\|w^*\|}$ is maximized.

2.2. Karush-Kuhn-Tucker conditions

Karush-Kuhn-Tucker conditions (KKT) [31,32] play a very important role in the theory of optimization, because they give the conditions to obtain an optimal solution to a general optimization problem.

Theorem 1. *Given an optimization problem with convex domain* $\Omega \subseteq \mathbb{R}^n$,

$$\begin{aligned} &\text{minimize} \quad f(w), \quad w\in\Omega \\ &\text{s.t.} \qquad\quad g_i(w) \leqslant 0, i = 1,\ldots,k, \\ &\qquad\qquad\quad h_i(w) = 0, i = 1,\ldots,m, \end{aligned} \tag{11}$$

with $f \in C^1$ convex, the necessary and sufficient conditions for a normal point w^* to be optimal are the existence of α^*, β^* such that

$$\begin{aligned} &\tfrac{\partial L(w^*,\alpha^*,\beta^*)}{\partial w} = 0 \\ &\tfrac{\partial L(w^*,\alpha^*,\beta^*)}{\partial \beta} = 0 \\ &\alpha_i^* g_i(w^*) = 0, i = 1,\ldots,k, \\ &g_i(w^*) \leqslant 0, i = 1,\ldots,k, \\ &\alpha_i^* \geqslant 0, i = 1,\ldots,k. \end{aligned} \tag{12}$$

From KKT conditions if the training set is linearly separable, it is verified that

$$\|w^*\|^2 = \langle w^*\cdot w^*\rangle = (\sum_{i\in sv}\alpha_i^*)^{-\frac{1}{2}} \tag{13}$$

Therefore, the maximum distance of a hyperplane is:

$$\frac{1}{\|w^*\|} = \left(\sum_{i\in sv}\alpha_i^*\right)^{-\frac{1}{2}} \tag{14}$$

2.3. Soft margin hyperplanes

The learning problem presented before is valid for the case where the data is linearly separable, which means that the training data set has no intersections. However, these problems are rare in the real life. At the same time, there are problems in which the linear separation hyperplane can give good results even when the data set has intersections. However, quadratic programming solutions as given above can not be used in the case of intersection because the condition $y_i(\langle w\cdot x_i\rangle + b) \geqslant 1, \forall i$ can not be satisfied in the case of intersection (see Fig. 3). The points that are in the intersection can not be correctly classified and for any misclassified data x_i, its corresponding α_i will tend to infinite.

To find a classifier with maximum margin, the algorithm presented above should be changed allowing a soft margin (Fig. 4), therefore, it is necessary to introduce non-negative *slack* variables $\xi_i(\geqslant 0)$ in the Eq. (3)

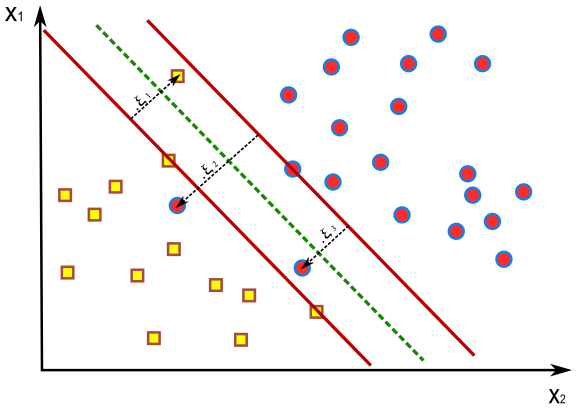


Fig. 3. Soft margin hyperplanes.

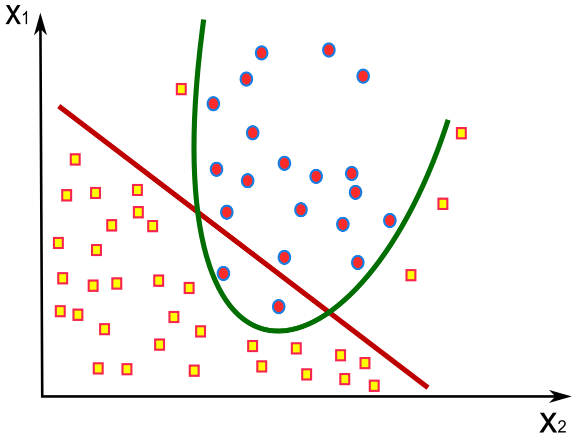


Fig. 4. Non linearly classifier.

$$y_i(\langle w^T \cdot x_i\rangle + b) \geqslant 1 - \xi_i \quad \forall i \tag{15}$$

Using the slack variables ξ_i, the feasible solution always exists.

For the training data x_i, if $0 < \xi_i < 1$, the data do not have the maximum margin, but can be correctly classified. On the other hand, the width of this soft margin can be controlled by the penalty parameter C, which determines the relationship between the training error and the Vapnik–Chervonenkis dimension.

Definition 1. (**Dimension Vapnik–Chervonenkis -VC-**) The VC dimension describes the capacity of a set of functions implemented in a learning machine. For binary classification, h is the maximum number of points in which two classes can be separated in all the 2^h possible ways using the functions of the learning machine.

Choosing a large Cprovides a small number of classification errors and a large w^Tw. It is clear that taking $C = \infty$ requires that the number of misclassified data be zero. However, in this case it is not possible, since the problem may be feasible only for some value $C < \infty$. Introducing non-negative “soft variables ” $\xi_i (i = 1, l)$ to the optimization problem, now instead of the conditions of the Eq. (5), the separation hyperplane should satisfy

$$\begin{aligned} &\min_{w,b,\xi_i} \langle w \cdot w\rangle + C\sum_{i=1}^{l}\xi_i^2 \\ &\text{s.t.} : y_i(\langle w \cdot x_i\rangle + b) \geqslant 1 - \xi_i \\ &\qquad \xi_i \geqslant 0 \end{aligned} \tag{16}$$

i.e., subject to

$$\langle w \cdot x_i\rangle + b \geqslant +1 - \xi_i, y_i = +1, \xi_i \geqslant 0 \tag{17}$$

$$\langle w \cdot x_i\rangle + b \leqslant -1 + \xi_i, y_i = -1, \xi_i \geqslant 0 \tag{18}$$

If $\xi_i < 0, y_i(\langle w \cdot x_i\rangle + b) \geqslant 1 - \xi_i \geqslant 1$, then, we do not consider the condition $\xi_i < 0$.

For the maximum soft margin with Norm-2 (with the diagonal $\frac{1}{C}\delta_{ij}$) the original Lagrangian is given by:

$$L(w, b, \xi_i, \alpha) = \frac{1}{2}\langle w \cdot w\rangle - \sum_{i=1}^{l}\alpha_i[y_i(\langle w \cdot x_i\rangle + b) - 1 + \xi_i] + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \tag{19}$$

The dual is found in two steps: in the same way as in the linearly separable case, first differentiating with respect to w and b, and then replacing it in the original Lagrangian

$$\begin{aligned} &\max_{\alpha_i} -\tfrac{1}{2}\sum_{i,j=1}^{l}\alpha_i y_i \alpha_j y_j[\langle x_i \cdot x_j\rangle + \tfrac{1}{C}\delta_{ij}] + \sum_{i=1}^{l}\alpha_i \\ &\text{s.t.} : \sum_{i=1}^{l}\alpha_i y_i = 0 \end{aligned} \tag{20}$$

The Kuhn-Tucker condition is

$$\alpha_i^*[y_i(\langle w^* \cdot x_i\rangle + b^*) - 1 + \xi_i] = 0 \tag{21}$$

That is, the quadratic optimization problem is practically the same as in the separable case with the only difference of the modified heights of the Lagrange multipliers α_i. The parameter C is determined by the user. The selection of an appropriate C is done experimentally using some cross-validation technique [30,14,33].

2.4. Kernels

In an SVM, the optimal hyperplane is determined to maximize the generalization ability of the model. But, if the training data are not linearly separable, the classifier obtained may not have a high generalization ability, even if the hyperplanes are optimally determined i.e., to maximize the space between classes, the original input space is transformed into a highly dimensional space called “feature space”.

The basic idea in designing non-linear SVMs is to transform the input vectors $x \in \mathbb{R}^n$ into vectors $\Phi(x)$ of a highly dimensional feature space [30] F (where Φ represents the mapping: $\mathbb{R}^n \to \mathbb{R}^f$) and solve the problem of linear classification in this feature space

$$x \in \mathbb{R}^n \to \Phi(x) = [\phi_1(x), \phi_2(x), \ldots, \phi_n(x)]^T \in \mathbb{R}^f \tag{22}$$

The set of hypotheses considered will be

$$f(x) = \sum_{i=1}^{l} w_i \phi_i(x) + b \tag{23}$$

where $\phi : X \to F$ is a non-linear mapping from an input space to a *feature space*, i.e., the learning procedure consists of two steps: first, a non-linear mapping transforms the data within a feature space F and later, a SVM is used to classify the data in a feature space.

A property of linear learning machines is that they can be expressed in a dual representation, this means that the Eq. (23) can be expressed as a linear combination of the training data points. Therefore, the decision rule can be evaluated using dot products

$$f(x) = \sum_{i=1}^{l}\alpha_i y_i \langle \phi(x_i) \cdot \phi(x)\rangle + b \tag{24}$$

If there is a way to capture the product $\langle \phi(x_i) \cdot \phi(x)\rangle$in the feature space directly as a function of the original input data, this makes it possible to join the two necessary steps to build a non-linear learning machine. This method of direct computation is called kernel function [13].

Definition 2. A kernel is a function K, such that for each $x, z \in X$

$$K(x,z) = \langle \phi(x) \cdot \phi(z) \rangle \tag{25}$$

where ϕ is a mapping of $X$to a feature space F.

The key to the approach is to find a kernel function that can be evaluated efficiently. Once we have such a decision function, the rule can be evaluated

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i K \langle x_i \cdot x_j \rangle + b \tag{26}$$

A kernel function must respect the following properties: for any $x, y, z \in X$ and $\alpha \in R$

1. $x \cdot x = 0$ only if $x = 0$
2. $x \cdot x > 0$ otherwise
3. $x \cdot y = y \cdot x$
4. $(\alpha x \cdot y)$ = $\alpha(x \cdot y)$
5. $(z + x) \cdot y$ = $(z \cdot y) + (x \cdot y)$

Moreover, kernel functions must fulfill the condition given by Mercer theorem. Some of the most used kernel functions are:

1. **Linear kernel:** $K(x_i, x_j) = (x_i \cdot x_j)$;
2. **Polynomial kernel:** $K(x_i, x_j) = (x_i \cdot x_j + 1)^p$;
3. **Gaussian kernel:** $K(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}$;
4. **RBF kernel:** $K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2}$;
5. **Sigmoid kernel:** $K(x_i, x_j) = tanh(\eta x_i \cdot x_j + v)$;

We checked the type of kernel used in some real-world applications. Table 1 shows a summary of the four main kernels found. In some papers, more than one kernel was applied, in these cases we consider the kernel that produces the better results. Without a doubt, Gaussian RBF function is the most commonly used in many different type of applications.

There is no unanimous conclusion about which kernel is better or worse for specific applications, some authors such as [34] have performed tests to identify the performance of SVM with different kernels, reaching the general conclusion that the polynomial and the Gaussian RBF function are the best option for acoustic signals. On the other hand, Kasnavi et al. [35] found that Gaussian RBF and hyperbolic tangent are the best for genome wide prediction. For other applications, less common kernels produce better results than the most popular, for example, in [36], Hasan concludes that Laplace kernel is the ideal one for intrusion detection. Furthermore, personalized kernels can be designed for certain applications, such is the case of the introduced in [37,38].

The observation in which all the authors agree, is that the selection of the kernel should be based on the characteristics of data, and that to obtain good results it is necessary to determine the optimal parameters of the kernel used.

2.5. Mercer's condition

Mercer's theorem [60,13] determines the conditions of functions to be kernels. Given a finite input space $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and assuming that $K(\mathbf{x}, \mathbf{z})$ is a symmetric function of X then

$$\mathbf{K} = (K(\mathbf{x}_i \cdot \mathbf{x}_j))_{i,j=1}^{n} \tag{27}$$

Since $\mathbf{K}$ is symmetric there exists an orthogonal matrix $\mathbf{V}$ such that $\mathbf{K} = \mathbf{V}\Lambda\mathbf{V}'$, where Λ is the diagonal matrix that contains the eigenvalues λ_t of $\mathbf{K}$, with its corresponding eigenvectors $\mathbf{v}_t = (v_{ti})_{i=1}$. Assuming that all eigenvalues are non-negative and considering the mapping

$$\phi : \mathbf{x}_i \mapsto (\sqrt{\lambda_t} v_{ti})_{t=1}^{n} \in \mathbb{R}^n, i = 1, \ldots, n. \tag{28}$$

$$\langle \phi(x_i) \cdot \phi(x_j) \rangle = \sum_{t=1}^{n} \lambda_t v_{ti} v_{tj} = (\mathbf{V}\Lambda\mathbf{V}')_{ij} = \mathbf{K}_{ij} = \mathbf{K}(\mathbf{x_i} \cdot \mathbf{x_j}) \tag{29}$$

this implies that $K(\mathbf{x}, \mathbf{z})$ is a kernel function corresponding to the ϕ mapping. The requirement that the eigenvalues of $\mathbf{K}$ be non-negative is necessary, since if we have a negative eigenvalue λ_s in the eigenvector $\mathbf{v}_s$, the point

$$\mathbf{z} = \sum_{i=1}^{n} v_{si} \phi(\mathbf{x}_i) = \sqrt{\Lambda}\mathbf{V}'\mathbf{v}_s \tag{30}$$

in the feature space could have square norm

$$\|\mathbf{z}\|^2 = \langle \mathbf{z} \cdot \mathbf{z} \rangle = \mathbf{v}_s'\mathbf{V}\sqrt{\Lambda}\sqrt{\Lambda}\mathbf{V}'\mathbf{v}_s = \mathbf{v}_s'\mathbf{V}\Lambda\mathbf{V}'\mathbf{v}_s = \mathbf{v}_s'\mathbf{K}\mathbf{v}_s = \lambda_s$$
$$< 0, \tag{31}$$

contradicting the geometry of this space. This brings us to the next proposition.

Proposition 3. Let X be a finite input space with a symmetric function on X $K(\mathbf{x}, \mathbf{z})$.We say that $K(\mathbf{x}, \mathbf{z})$ is a kernel function if and only if the matrix

$$\mathbf{K} = (K(\mathbf{x_i} \cdot \mathbf{x}_j))_{i,j=1}^{n} \tag{32}$$

is positive semi-definite (has non-negative eigenvalues).

Allowing a slight generalization of a dot product in a Hilbert space [14] by entering a weight $\lambda_i$for each dimension

$$\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \cdot \phi_i(\mathbf{z}) = K(\mathbf{x}, \mathbf{z}), \tag{33}$$

therefore, the feature vector would be

$$\phi(\mathbf{x}) = (\phi_i(\mathbf{x}), \phi_i(\mathbf{x}), \ldots, \phi_i(\mathbf{x}), \ldots). \tag{34}$$

Mercer's theorem gives the necessary and sufficient conditions so that a symmetric continuous function $K(\mathbf{x}, \mathbf{z})$ is represented

$$K(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \cdot \phi_i(\mathbf{z}) \tag{35}$$

with non-negative λ_i, which is equivalent to $K(\mathbf{x}, \mathbf{z})$ being a dot product in the feature space $F \supseteq \phi(\mathbf{X})$, where F is the space l_2 of all the sequences

$$\psi = (\psi_1, \psi_2, \ldots \psi_i, \ldots). \tag{36}$$

for which

$$\sum_{i=1}^{\infty} \lambda_i \psi_i^2 < \infty. \tag{37}$$

This implicitly induces a space defined by the feature vector and as a consequence a linear function in F can be represented by

$$f(x) = \sum_{i=1}^{\infty} \lambda_i \psi_i \phi_i(\mathbf{x}) + b = \sum_{j=1}^{l} \alpha_j y_j K(\mathbf{x}, \mathbf{x}_j) + b \tag{38}$$

where the first expression is the original representation and the second is the dual [30]. The relationship is given by

$$\psi = \sum_{j=1}^{l} \alpha_j y_j \phi(\mathbf{x}_j). \tag{39}$$

Table 1
Popular kernels for SVM.

Kernel name	Expression	Parameters	Characteristics	Some applications
Polynomial	$K(x_i, x_j) = (< x_i, x_j > +1)^r$	$r \in \mathbb{Z}^+$	This kernel allows to map the input space into a higher dimensional space that is a combination of products of polynomials. Despite its high computational load, this kernel is frequently applied to data that has been normalized (norm L_2) [39].	Fault diagnosis of centrifugal pumps [40], natural language processing [41,42
Gaussian radial basis function (RBF)	$K(x_i, x_j) = \exp^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$	σ	This kernel is one of the most widely used in applications, it can be considered a general-purpose translation-invariant kernel. Other related functions are the exponential kernel and Laplacian kernel. Parameter σ must be carefully chosen.	Electric power load forecasting [43], hyperspectral/image classification [44][45][46][47][48], clustering (One-class SVM) [49], bankruptcy prediction [50], classification of electroencephalography signals [51], bioemetric identification [52], health applications [53], intrusion detection [54], stream flow predictions [55]
Linear	$K(x_i, x_j) =< x_i, x_j > +1 = x_i^T x_j + 1$	None	This is the simplest kernel function. It represents the non-kernelized version of SVM. Datasets with many features usually become linearly separable problems. Therefore, the choice of this kernel can be a good option in these cases.	Stock prediction [56], malware detection [57]
Hiperbolic tangent	$K(x_i, x_j) = tanh(< x_i, x_j > \beta + b)$	β, b	This kernel is also known as sigmoid kernel, it is also used as activation function in neural networks. β can be seen as a scaling parameter of the product $x_i^T x_j$, and b a shift. These parameters affect considerably the performance of SVM. [58] showed that using $\beta > 0$ and $b < 0$ guarantees this kernel be conditional positive definite.	Audio classification [59]

In the original representation, the number of terms in the sum is equal to the dimensionality in the feature space, while in the dual there are l terms. The analogy with the finite case is similar. The contribution from the functional analysis leads us to the problem for integral equations of the form

$$\int_X K(\mathbf{x}, \mathbf{z})\phi(\mathbf{z})d\mathbf{z} = \lambda\phi(\mathbf{x}) \tag{40}$$

where $K(\mathbf{x}, \mathbf{z})$ is the symmetric and positive kernel function, and X is an compact space.

Theorem 2. (Mercer) Let X be a compact subset of $\mathbb{R}^n$. Assuming that $\mathbf{K}$ is a symmetric continuous function such that the integral operator $T_K : L_2(X) \to L_2(x)$, [14]

$$(T_K f)(\cdot) = \int_X K(\cdot, \mathbf{x})f(\mathbf{x})d\mathbf{x}, \tag{41}$$

is positive, that is

$$\int_{X \times X} K(\mathbf{x}, \mathbf{z})f(\mathbf{x})f(\mathbf{z})d\mathbf{x}d\mathbf{z} \geqslant \mathbf{0}, \tag{42}$$

for all $f \in L_2(X)$. then $K(\mathbf{x}, \mathbf{z})$ can be expanded in a uniformly convergent series (on $X \times X$) in terms of functions $\phi_j \in L_2(X)$, normalized in such a way that $\|\phi_j\|_{L_2} = 1$ and positive eigenvalues associated $\lambda_j \geqslant 0$,

$$K(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{\infty} \lambda_j \phi_j(\mathbf{x})\phi_j(\mathbf{z}). \tag{43}$$

2.6. Non-linearly separable case

The linear classifiers presented in the two previous sections are very limited. In most real life data sets, the data points not only overlap or intersect when generating a separation hyperplane, but the genuine separation of these data is given by non-linear hyper-surfaces.

The approach presented above can be easily extended to create non-linear decision functions. The reason for this extension is that an SVM can create a non-linear hyper surface of decision, capable of classifying non-linearly separable data. Generally, for n - dimensional input patterns, instead of a non-linear curve, an SVM will create a non-linear separation hyper-surface.

The problem of optimization using kernels is as follows [30,13]:

Proposition 4. Given a data set $S = [(x_1, y_1) \cdots (x_l, y_l)]$, a feature space $\phi(x)$defined by the kernel $K(x, z) = \langle\phi(x) \cdot \phi(z)\rangle$, the solution of

$$\begin{aligned} &\max_{\alpha_i} -\tfrac{1}{2}\sum_{i,j=1}^{l} \alpha_i y_i \alpha_j y_j \left[K(x_i, x_j) + \tfrac{1}{C}\delta_{ij}\right] + \sum_{i=1}^{l} \alpha_i \\ &\qquad \text{subject to a} : \sum_{i=1}^{l} \alpha_i y_i = 0 \end{aligned} \tag{44}$$

is $\alpha_i^*, f(x) = \sum_{i=1}^{l} \alpha_i^* y_i K(x_i, x) + b^*$,where b^* is choosing such that

$$\begin{aligned} &y_i f(x_i = 1 - \xi^* = 1 - \tfrac{\alpha^*}{C} \\ &w^* = \sum_{i=1}^{l} \alpha_i^* y_i K(x, x), \end{aligned} \tag{45}$$

The decision function $sgn[f(x)]$ is equivalent to the hyperplane in the feature space defined by the kernel $K(x,z)$ which solves the optimization problem. Then, the geometric margin is given by

$$\gamma^* = \left(\sum_{i \in sv} \alpha_i^* - \frac{1}{C}\langle \alpha^* \cdot \alpha^* \rangle\right)^{-\frac{1}{2}} \tag{46}$$

Using the kernel

$$K'(x,z) = K(x,z) + \frac{1}{C}\delta_x(z) \tag{47}$$

The soft margin in L1

$$\begin{aligned} &\min_{w,b,\xi_i} \langle w \cdot w \rangle + C\sum_{i=1}^{l}\xi_i \\ &\text{s.t.}: y_i(\langle w \cdot x_i \rangle + b) \geqslant 1 - \xi_i \\ &\qquad \xi_i \geqslant 0 \end{aligned} \tag{48}$$

where the original Lagrangian is

$$L(w,b,\xi_i,\alpha) = \frac{1}{2}\langle w \cdot w \rangle - \sum_{i=1}^{l}\alpha_i[y_i(\langle w \cdot x_i \rangle + b) - 1 + \xi_i] + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\gamma_i\xi_i \tag{49}$$

the dual is given by

$$w(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i y_i \alpha_j y_j \langle x_i \cdot x_j \rangle + um_{i=1}^{l}\alpha_i \tag{50}$$

This is the same maximum margin, but

$$C - \alpha_i - \gamma_i = 0, \gamma_i \geqslant 0 \Rightarrow C \geqslant \alpha_i \tag{51}$$

with Kuhn-Tucker conditions

$$\begin{aligned} &\gamma_i\xi_i = 0 \ \text{o}(\alpha_i - C)\xi_i = 0 \\ &\alpha_i[y_i(\langle w \cdot x_i \rangle + b) - 1 + \xi_i] = 0 \end{aligned} \tag{52}$$

where $\xi_i \neq 0, \gamma_i = 0, \Rightarrow C = \alpha_i$, with $\xi_i = 0$ the maximum margin, α_i is positive and can increased to C, therefore $C \geqslant \alpha_i \geqslant 0$.

Proposition 5. Given a data set $S = [(x_1, y_1) \ldots (x_l, y_l)]$, a feature space $\phi(x)$defined by the kernel $K(x,z) = \langle \phi(x) \cdot \phi(z) \rangle$, the solution

$$\begin{aligned} &\max_{\alpha_i} -\tfrac{1}{2}\sum_{i,j=1}^{l}\alpha_i y_i \alpha_j y_j K(x_i,x_j) + \sum_{i=1}^{l}\alpha_i \\ &\quad \text{s.t.}: \sum_{i=1}^{l}\alpha_i y_i = 0, C \geqslant \alpha_i \geqslant 0 \end{aligned} \tag{53}$$

is given by $\alpha_i^*, f(x) = \sum_{i=1}^{l}\alpha_i^* y_i K(x_i, x) + b^*$, the decision function $sgn[f(x)]$ is equivalent to the hyperplane in the feature space defined by the Kernel $K(x,z)$, which solves the optimization problem. The geometric margin is given by

$$\gamma^* = \left(\sum_{i \in sv}\alpha_i^*\right)^{-\frac{1}{2}} \tag{54}$$

When the bound of α_i is C, the problem of maximum margin arises.

Choosing C is the same as getting v in

$$\begin{aligned} &\max_{\alpha_i} -\tfrac{1}{2}\sum_{i,j=1}^{l}\alpha_i y_i \alpha_j y_j K(x_i,x_j) \\ &\quad \text{s.t.}: \sum_{i=1}^{l}\alpha_i y_i = 0, \\ &\qquad \sum_{i=1}^{l}\alpha_i \geqslant v, \\ &\qquad \tfrac{1}{l} \geqslant \alpha_i \geqslant 0 \end{aligned} \tag{55}$$

For a non-optimal solution, $\hat{\alpha}$ is the current value of the dual variables. The vector of weights is calculated by $\frac{\partial L}{\partial w} = 0$. The solution $\hat{w} \cdot \hat{w}$ satisfies the original conditions with $\frac{1}{2}\|\hat{w}\|^2 + c\sum_{i=1}^{l}\xi_i, \inf_{w,b}L(w,b,\hat{\alpha})$as a feasible dual solution

$$L = \frac{1}{2}\langle w \cdot w \rangle - \overset{l}{\underset{i=1}{m}} \alpha_i[y_i(\langle w \cdot x_i \rangle + b) - 1], \tag{56}$$

$$w = \sum_{i=1}^{l}\hat{\alpha}_i y_i x_i, \tag{57}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{l}\alpha_i y_i = 0 \tag{58}$$

Computing the difference between the original and dual feasible solutions $C - \alpha_i = \gamma_i$

$$\begin{aligned} &\tfrac{1}{2}\|w\|^2 + c\sum_{i=1}^{l}\xi_i - \inf_{w,b}L(w,b,\hat{\alpha}) \\ &= \sum_{i=1}^{l}\hat{\alpha}_i[y_i(\sum_{j=1}^{l}y_j\alpha_j K\langle x_j \cdot x_i \rangle + b - 1 + \xi_i] + \sum_{i=1}^{l}\gamma_i\xi_i \\ &= C\sum_{i=1}^{l}\xi_i + \sum_{i=1}^{l}\hat{\alpha}_i[y_i(\sum_{j=1}^{l}y_j\alpha_j K\langle x_j \cdot x_i \rangle + b) - 1] \\ &= \sum_{i,j=1}^{l}\hat{\alpha}_i y_i y_j \alpha_j K\langle x_j \cdot x_i \rangle - \sum_{i=1}^{l}\hat{\alpha}_i + C\sum_{i=1}^{l}\xi_i \\ &= \sum_{i=1}^{l}\hat{\alpha}_i - 2w(\alpha) + C\sum_{i=1}^{l}\xi_i \end{aligned} \tag{59}$$

3. Weaknesses of SVM

Despite the generalization capacity and many advantages of the SVM, they have some very marked weaknesses, among which are: the selection of parameters, algorithmic complexity that affects the training time of the classifier in large data sets, development of optimal classifiers for multi-class problems and the performance of SVMs in unbalanced data sets.

3.1. Algorithmic complexity

Maybe the principal disadvantage of SVM is due to its excessive computational cost in large data sets, because the training kernel matrix grows in quadratic form with the size of the data set, which provokes that training of SVM on large data sets is a very slow process. Support Vector Machines (SVM) have demonstrated highly competitive performance in many real-world applications. However, despite its good theoretical foundations and generalization performance, SVM is not suitable for large data set classification. Training an SVM is usually posed as a quadratic programming (QP) problem to find a separation hyperplane which implicates a matrix of density $n \times n$, where the n is the number of points in the data set. This needs huge quantities of computational time and memory for large data sets, so the training complexity of SVM is highly dependent on the size of a data set [61,62].

According to the strategy used, the training methods for SVM can be categorized into data selection, decomposition, geometric, parallel implementations and heuristics. Their core ideas and the most representative algorithms are presented in this section.

Data selection methods for SVM intent to decrease the size of data sets by removing the instances that do not contribute to the definition of the optimal separating hyperplane. The latter depends completely on instances which are located closest to the separation boundary [63], and correspond to those whose Lagrange multipliers are greater than zero in the Karush–Kuhn–Tucker conditions

(60). These instances are called support vectors (SVs). Generally, the number of SVs is a small portion compared with the size of training sets.

$$\begin{aligned} &\alpha_i = 0 \Rightarrow y_i(\langle \omega, x_i \rangle + b) \geqslant 1, \xi_i = 0 \\ &0 < \alpha_i < C \Rightarrow y_i(\langle \omega, x_i \rangle + b) = 1, \xi_i = 0 \\ &alpha_i = C \Rightarrow y_i(\langle \omega, x_i \rangle + b) \leqslant 1, \xi_i \geqslant 0 \end{aligned} \tag{60}$$

Simple random sampling (SRS) is probably the most basic strategy to reduce the size of training sets. It consists in choosing a number of instances and then training a SVM with them. The works presented in [64–66] show that uniform random sampling is the optimal robust selection scheme in terms of several statistical criteria. However, although SRS is computationally cheap, the standard deviation of classification accuracy is large in most cases [66].

A more sophisticated form of this type of sampling consists in assigning to each instance a probability to be chosen. Once a number of instances is randomly selected, a SVM is trained with them. After this, the probabilities are updated, increasing those whose instances have been miss-classified [67–69]. This process is repeated several times.

Some data selection methods have been developed by computing the distance between the instances and the optimal hyperplane. Several metrics for measuring distance have been used in previous works: These measures include the Euclidean [70], Mahalanobis [71] and Hausdorff [72,73] distances. Most of the current distance-based methods are inspired on two observations: (1) the instances closest to those ones with opposite label have high chances to be SVs [72] and (2) instances far from hyperplane do not contribute to the definition of the decision boundary [74]. A problem with naive implementations that require to compute all distances between objects is that this task has a temporal and a spatial complexity of $O(n^2)$.

The Condensed Nearest Neighbor (CNN) [75] chooses instances near to class frontiers, reducing the size of training sets. However, CNN is not noise tolerant. Reduced Nearest Neighbor (RNN) [76], Selective Nearest Neighbor (SNN) [77] and Minimal Consistent Set (MCS) are methods based on CNN, and therefore, they have also problems with noisy data sets. RNN, SNN and MCS are more costly than CNN.

Neighborhood properties of SVs have also been exploited to reduce size of training sets. Wang and Kwong [78] used neighborhood entropy, while in [79] only the patterns in the overlap region around the decision boundary are selected. The method presented in [80] follows this trend but use fuzzy C-mean clustering to select samples on the boundaries of class distribution, whereas [72] uses hyper spheres. Clustering has been proved to be an effective method to collaborate with SVM on classifying large data sets. For example, hierarchical clustering [81,82], k-means [83] and parallel clustering [84]. Clustering-based methods can reduce the computations burden of SVM, however, the clustering algorithms themselves are still complicated for large data set. Rocchio bundling is a statistics-based data reduction method [85]. The Bayesian committee machine is also reported to be used to train SVM on large data sets, where the large data set is divided into m subsets of the same size, and m models are derived from the individual sets [86]. But, it has higher error rate than normal SVM and the sparse property does not hold.

The basis for decomposition methods lies in the fact that the training time can be reduced if only the active constraints of the QP problem are taken into account [87]. A similar idea to active set methods for optimization is applied in decomposition methods. In the active set approach, two sets are used: the working set and the set of fixed variables. The optimization is made only on the working set. For the case of SVM, the working set is usually composed of instances that violate the Karush–Kuhn–Tucker conditions. Apart of the proved convergence [88], a clear advantage of decomposition is that memory requirement is linear in the number of training examples; but on the other hand, because only a fraction of variables is being considered in each iteration, it is time consuming [89,90] if elements in the active set are not carefully selected. One of the first decomposition methods was Chunking [74]. It consists in repetitively obtaining the maximum margin hyperplane from an amount of instances (called the chunk) and then forming a new chunk with the SVs from the previous solution and some new instances. Probably the most famous decomposing algorithm is the SMO [15]. Sequential minimal optimization (SMO) is a fast method to train SVM [91,84]. Training SVM requires the solution of the QP optimization problem. SMO breaks this large QP problem into a series of smallest possible QP problems. It considers the smallest size working set: only two training samples, and it is faster than the projected conjugate gradient (PCG) chunking algorithm. Dong et al. [61] introduced a parallel optimization step where block diagonal matrices are used to approximate the original kernel matrix so that SVM classification can be split into hundreds of sub-problems. A recursive and computational superior mechanism referred as adaptive recursive partitioning was proposed in [92], where the data are recursively subdivided into smaller subsets. Genetic programming is able to deal with large data sets that do not fit in main memory [93]. Neural networks technique can also be applied for SVM to simplify the training process [94]. LibSVM [95] is an algorithm based on SMO with the improvement of a more advanced mechanism of selection of the working set by using the second order information method previously shown in [96]. The SVM^{light} [97] is another important state-of-the-art decomposition method.

Variants of SVM speed up the training time of SVM at expense of loosing accuracy [89]. These methods work by changing the original QP problem formulation. Most of the variants methods conclude with a system of linear equations solved efficiently if the number of features is moderate, i.e., around 100. A representative method in this category is the least square SVM (LS-SVM) [98] which changes the original QP problem by using a linear system of equations that can be solved explicitly or by using a conjugate gradient method. Other important methods are the PSVM (Proximal SVM) [99] and reduced SVM (RSVM) [100].

Parallel implementation of the QP problem is difficult because there is a strong dependence between data [101]. Most parallel methods for training SVM divide the training set into independent subsets to train SVM in different processors, as in [101–103]. In [61], the kernel matrix of SVM is approximated by block diagonal matrices so that the original optimization problem can be decomposed into hundreds of sub problems, which are easy to solve in a parallel fashion. Other parallel implementations can be found in [104–108].

Geometric methods for SVM are based on that computing the optimal separating hyperplane is equivalent to find the closest pair of points belonging to convex hulls [63,109,110]. Recent advances on geometric methods can be found in [111–115].

Among all heuristic methods, the alpha seeding [116] consists of providing initial estimates of the α_i values for the starting of the QP problem. Alpha seeding seems to be a practical method to improve training time of SVM. Recently, an improvement of this method has been proposed in [117].

According to the reviewed literature, there are currently just few methods that combine decision tree (DT) for instance selection in a similar way to the presented in this research. In [118], the patterns by ordered projections (POP) algorithm was presented. It uses projections of instances on the axis of attributes to find the minimal number of elements to represent hyper-rectangles which contain instances of the same class (entropy zero). A disadvantage of POP is that the reduction of the size of data sets is very low [119].

In [120], a method that approximates the decision boundary of SVM using a DT to speed up SVM in its testing phase was proposed. In [120], an SVM is used in some leaves of a DT. The idea is to reduce the number of test data points that require SVM's decision.

Recently, in [121,26], the combination of a DT and SVM was proposed. The underlying idea is to train an SVM first, and then use the predictions of the model obtained to modify the class of examples in the training set. A DT is afterward trained using the modified set. The SVM is used as a pre-processor for improving the performance of DT, when dealing with the problem of imbalance.

3.2. Development of optimal classifiers for multi-class problems

Support Vector Machines were originally designed to solve binary classification problems [122]. The problem of multi-classification for SVM, does not present an easy solution. In order to apply SVM to multi-classification problems, it is necessary to change the problem to multiple binary classification problem. There are two basic types of algorithms to solve the multiclass classification based in SVM.

1. "one versus one" method (OVO). This method constructs $\frac{k(k-1)}{2}$ hyperplanes for k-class problem, where each one is trained with just two-class data sets. Thus given n training data $(x_1, y_1), \dots, (x_n, y_n)$, where $x_i \in R^d, i = 1, \dots, n$ and $y_i \in \{1, \dots, k\}$is the class of x_i, the i^{th} SVM solves the problem:

$$\begin{aligned} &\min_{w^{ij}, b^{ij}, \xi^{ij}} \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_{j=1}^{l} \xi_t^{ij} \\ &(w^{ij})^T \phi(x_t) + b^{ij} \geqslant 1 - \xi_t^{ij}, \quad \text{if}\}y_t = i \\ &(w^{i})^T \phi(x_j) + b^{i} \leqslant -1 + \xi_t^{ij}, \quad \text{if}\}y_t = j \\ &\xi_t^{ij} \geqslant 0, j = 1, \dots, n \end{aligned} \tag{61}$$

At the prediction phase, a voting scheme is applied to get the class of x. The $K(K1)/2$ classifiers are applied to an unseen sample x. If the k^{th} classifier says that x belongs to the i class, then one vote for the i class is added by one and the label of x is predicted in the class that got the highest number of votes. The principal disadvantage of OVO is that some times the ambiguities in some regions of its input space can provoke that two regions receive the same number of votes.

2. "one versus all" method (OVA). This algorithm constructs k hyperplanes for k-class problem. The i-th hyperplane is trained with the samples between the i-th class and the rest data. Thus given n training data $(x_1, y_1), \dots, (x_n, y_n)$, where $x_i \in R^d, i = 1, \dots, n$ and $y_i \in \{1, \dots, k\}$is the class of x_i, the i^{th} SVM solves the problem:

$$\begin{aligned} &\min_{w^{p}, b^{p}, \xi^{p}} \frac{1}{2}(w^{p})^T w^{p} + C\sum_{j=1}^{l} \xi_i^{p} \\ &(w^{p})^T \phi(x_i) + b^{p} \geqslant 1 - \xi_i^{p}, \quad \text{if}\}y_i = p \\ &(w^{p})^T \phi(x_i) + b^{p} \leqslant -1 + \xi_i^{p}, \quad \text{if}\}y_i \neq p \\ &\xi_i^{p} \geqslant 0, i = 1, \dots, n \end{aligned} \tag{62}$$

where x_i are mapped to high dimensional space by means of ϕ, where $\phi$is a *Kernel* that satisfies the Mercer's condition [14], and C is a penalty parameter. The k decision functions are given by

$$\begin{aligned} &(w^1)^T \phi(x) + b^1 \\ &(w^2)^T \phi(x) + b^2 \\ &\vdots \\ &(w^k)^T \phi(x) + b^k \end{aligned} \tag{63}$$

Given a sample x to classify, the label of the class that has the largest value of the decision function is chosen as:

$$\textit{class of } x = \arg \max_{i=1,2,\dots,k} ((w^p)^T \phi(x) + b^p) \tag{64}$$

where w_i and b_i depict the hyperplane of the ith SVM.

Currently, most of methods use one-against-all or one-against-rest approaches to facing multi-class problems with SVM. However, there are numerous researchers who have faced the problem and have developed algorithms to try to solve the problem [123–131].

3.3. Performance of SVMs in imbalanced datasets

In imbalanced data sets, the correct classification of minority class objects is a challenging problem. Normal classification methods, such as support vector machines, do not work well for these skewed data sets because is difficult to get the optimal separation hyperplane for an SVM trained with imbalanced data.

The imbalance in data sets affects considerably the performance of most classifiers. In general, the model extracted from this type of data sets is biased towards the minority class. As a result, the accuracy on the minority classes is hampered. The imbalance in data sets is a recurrent problem in many domains, some examples are: fraud detection problems [132], classification of protein sequences [133–135], medical diagnosis of rare and dangerous diseases [136], intrusion detection and text classification [137,138], discrimination between earthquakes and nuclear explosions [139]. Support Vector Machines were introduced by Vapnik [13] as a kernel based machine learning model for classification and regression tasks. The generalization capabilities and discriminative power of SVM have attracted the attention of practitioners and theorists in last years. SVM has strong theoretical foundations, and, in general, it presents high classification accuracy in real-world applications. However, recent experiments [140–142] show that the performance of SVM is severely affected when it is applied on imbalanced data sets. This is more evident when the ratio between the majority and the minority class is large. The first disadvantage of SVM on imbalanced data sets is due to the margin obtained is biased towards the minority class.

There are several solutions of SVM classification for imbalanced data [143]. The techniques used to minimize the negative effect of imbalanced data sets on classifiers can be categorized as external and internal. The first techniques balance the data sets before training a classifier [144,145,142]. The second techniques modify the model or architecture of classification methods [146–148]. Principal external techniques are under sampling and over sampling. In general, under sampling consists in selecting, randomly, a small number of objects from majority class [146]. Over sampling techniques generate artificial examples of the minority class. Other methods use evolutionary algorithms to balance the data sets [143,149,150]. However, to add artificial data points to the minority class is a promising technique to tackle the problem of imbalance. Chawla et al. [140] proposed Synthetic Minority Over sampling Technique (SMOTE), which generates artificial objects to be included as members of the minority class. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any or all of the k minority class nearest neighbors. It does not cause any information loss and could potentially find hidden minority regions. The disadvantage of this method is that it creates noise for the classifiers which could result in a loss of performance because SMOTE makes the assumption that the instance between a positive class instance and its nearest neighbors is also positive [151].

Classification for imbalanced data sets has been studied by machine learning community since the last decade [142]. There are many methods that are applied to imbalanced data sets in order to improve the performance of classifiers [135]. Generally, these methods are divided into two categories: external methods and internal methods. External methods involve a pre processing of training data sets in order to make them balanced. Internal methods deal with modifications of the learning algorithms in order to reduce their sensitiveness to class imbalance. In other words, external methods attempt to balance the data sets by considering the number of examples for each class, whereas internal methods consider the costs associated with misclassification and include these costs in the model. The most popular external methods are under-sampling, over-sampling and SMOTE. The under sampling and over sampling method [146] balances the data sets by randomly selecting small number of objects from majority class, and doubling the objects in the minority class. The main drawback is that some important points, such as support vectors, may be neglected by the random algorithm. [141] pointed out that the under sampling strategy is not a good choice for SVM, and that the over sampling cannot improve the final accuracy. The synthetic minority over sampling technique [140] generates artificial data in the minority class by multiplying a random number in each original object. [152] showed that SMOTE is better than under sampling and over sampling. Several proposals inspired in SMOTE have been proposed, see, for example, [153–155]. In [146] an SVM with different costs and SMOTE is used. That method introduces a scheme to penalize classification errors. The majority class is assigned a high cost and the minority class is assigned a low cost. This combination makes denser the distribution of the minority class, and pushes the separating hyperplane to the minority class. In [135], a penalization criterion is used to produce a similar effect on the separating hyper plane. In [142], a kernelized version of SMOTE is proposed. Other kernel-modification methods have been proposed in [142,156,157]. Other proposals can be found in [158,159].

Although sampling methods and cost-sensitive learning methods seem to dominate the current research efforts in imbalanced learning, Genetic Algorithm (GA)-based approaches have also been pursued by the community. These algorithms use GAs in order to balanced data sets. [149] used a GA for under-sampling the majority class, the algorithm tackles the difficulties of SVM learning on large data sets, because the method significantly reduce the size of the training set without loss of performance. Batuwita and Palade [160] applied Fuzzy SVM (FSVM) to improve the performance on imbalanced data sets to handle the problem of outliers and noise by assigning different fuzzy-membership values based on their importance. In [161], the authors proposed a efficient resampling method selecting the most informative data examples located closer to the class boundary region by using the separating hyperplane found by training an SVM model on the original imbalanced data set, and then use only those examples in resampling. An excellent survey about classification on imbalanced data sets can be found in [162]. In [143], a GA is used to balance skewed data sets. That method produces better results than simple random sampling. Bazi and Melgani [163] used PSO algorithm in order to improve the performance of SVM. That method was applied for the classification of electrocardiogram signals and parameters estimation. In [159], the authors propose a classification system in order to detect the most important rules, and the rules which perturb the performance of classifier. That system uses hierarchical fuzzy rules and a GA. Garcia et al. [150] implemented an algorithm which performs an optimized selection of examples from data sets. The learning algorithm is based on the nested generalized exemplar method and GA to generate and select the best suitable data to enhance the classification performance over imbalanced domains. GA is used to guide the search process. Although generating new instances in the minority class can improve the performance in SVM classification [144], this process could introduce noise to the data set, and to select randomly instances that helps to improve the performance in this area is almost impossible without using a genetic algorithm because the search space is huge.

4. SVM implementations

Currently there are several implementations of SVM in the literature. Table 2 shows a list of the most used SVM implementations. SVMs must solve a quadratic programming problem to find a hyperplane that separates the classes. The main reason for multiple implementations is because computational time depends mainly on the heuristics used to divide the problem into small fragments. In small data sets, the computational time of the SVMs is not important, however the computational complexity of the SVMs is almost cubic, so that in large data sets the training time is enormous and it is very important to use some algorithm that face this challenge. This section briefly shows some approaches used to improve the training time of SVM.

Data reduction: In most cases the SVM solution is given by a small subset of data called support vectors and not by the entire data set. The basic idea is to eliminate data less likely to be support vectors and preserve the data more likely to be support vectors and train an SVM with them.

Chunking: It is based on the sparsity of the SVM. In most cases the solution of the SVM is given by a small subset of data and not by the entire data set [164–167]. Moreover, an α_i point can only be optimal if it fully satisfies the conditions of KKT. The algorithm starts selecting an arbitrary subset of the data called chunk. The quadratic optimization problem is solved on this small "chunk" and the next chunk is obtained with the resulting support vectors and the points violating the KKT conditions. The process is stopped until all the training data are considered and the chunk get all the SV. This algorithm reduces the complexity of SVM by reducing the large problem to a sequence of smaller optimization problems, iteratively determining the support vectors.

Decomposition: These methods are similar to chunking methods. However, in decomposition methods the size of the sub problems is fixed. Decomposition methods were designed to reduce the complexity to computing the full kernel matrix by solving a sequence of smaller quadratic programming sub problems. Decomposition methods tackle the problem of training an SVM by optimizing iteratively only on the variables belonging to a subset of tractable size. This is the so-called working or active set. The variables that do not belong to the working set are fixed and form the so-called fixed set. Decomposition methods can be classified into primal and dual methods. They aim for dual(primal) feasibility, while maintaining primal (dual) feasibility and complementary slackness.

A clear advantage in this scheme, in addition to its proved convergence [168,169], is that its memory requirements grow linearly with the number of training examples. On the other hand, because only a fraction of the variables is being considered in each iteration, it is time consuming [89] if elements in the working set are not carefully selected. It has been observed that the active set method can oscillate nearby the solution [170].

The most important element in decomposition methods for them to converge quickly is the selection of the subset of variables

Table 2
SVM Implementations

Implementation	Developer	Source code	University	Web page
SVMTorch	Ronan Collobert and Samy Bengio	C++	Universite de Montreal	[176]
Pegasos	Shai Shalev-Shwartz	C++	The Hebrew University of Jerusalem	[177]
LibSVM	Chih-Chung Chang and Chih-Jen Lin	C and Java	National Taiwan University	[178]
SVMLight	Thorsten Joachims	C	Cornell University	[179]
Incremental SVM	Chris Diehl	M	Carnegie Mellon	[180]

in the working set [171]. One method, commonly used,consists in selecting those samples that violated the most KTT conditions [172–174].

Sequential Minimal Optimization: The Sequential Minimal Optimization algorithm (SMO) [15] is obtained from the idea of the decomposition method to the extreme, by optimizing a minimum subset of only two points in each iteration. The power of this technique lies in the fact that the two-point optimization problem admits an analytical solution, eliminating the need to use an iterative quadratic programming optimizer as part of the algorithm [97].

The condition $\sum_{i=1}^{l}\alpha_i y_i = 0$ always requires that the number of multipliers which can be optimized at each step is 2. Each time that a multiplier is updated, at least one other multiplier needs to be adjusted in order to maintaining the condition true. In each step, SMO chooses two elements α_i and α_j to optimize them, finds the optimal value of those two parameters, and updates the vector α. The choice of the two points is determined by a heuristic, while the Optimization of the two multipliers is performed analytically.

Experimentally, SMO performance is very well. This is because the time of kernel computing can be reduced, which directly improves its performance. Although it needs more iterations to converge, each iteration uses only a few operations, so it converges very fast. In addition to the convergence time, another feature of the algorithm is that it does not need to store the kernel matrix in memory since matrix operations are not involved.

The SMO algorithm performs well for large data sets because it scales well with the size of the training set. The authors argue that SMO is a strong candidate to become the standard algorithm of SVM training [15].

Shrinking: The shrinking heuristics are designed to speed up the optimization reducing the number of kernel values needed to update the gradient vector. The algorithm is based on the fact that An optimal solution α of the SVM dual problem may contain some bounded elements(i.e.,$\alpha_i = 0$ or C). The principal propose of shrinking technique is to reduce the size of the problem by temporarily eliminating the bounded elements α_i (Joachims, 1999).

Working selection: The selection of an initial set of variables as a working set is important when optimizing SMO so that the current iteration moves towards the minimum. There are many ways to select the pair of indices (i, j) representing the working set for each iteration of the SMO algorithm. Although maximal gain working set selection may reduce the number of iterations, it makes each iteration very slow. Practical working set selection schemes can to achieve a good compromise between the number of iterations and the speed of each iteration [175,165–167].

Most of the implementations are designed to solve classification and regression problems. In addition, the implementations have many benefits, among which we could highlight some as:

1. These can work on large data sets, the implementations can handle several hundred-thousands of training examples and many thousands of support vectors.
2. They can supports standard kernel functions and in some cases it is possible to define your own kernel function.
3. They can perform efficient multi-class classification.
4. They efficiently compute cross validation for model selection.
5. In some cases they can use weighted SVM for unbalanced data.
6. They provide probability estimates.

In Table 2 is shown some of the most popular SVM implementations to large data sets. All current implementations use one or more of the approaches described above to reduce SVM training time.

SVMLigth uses Working selection and Shrinking techniques. SVMLigth is one of the most popular SVM implementations. The algorithm is very fast and has been applied to solve classification and regression problems in large data sets.

SVMTorch uses Working selection and Shrinking to improve the training time of SVM the authors argue that the implementation can efficiently solve large scale regression problems.

Pegasos use decomposition methods to reduce the training time of SVM. Pegasos is essentially an Stochastic Subgradient Descent optimization algorithm that solves the primal formulation, which means it needs the actual feature vectors. Pegasos can be used to train a non-linear SVM only if can be represent the kernel as a dot product of finite-dimensional feature vectors.

LIBSVM algorithm is based on the SMO algorithm, however, LIBSVM has a more advanced work set selection algorithm. Most decomposition methods obtain an initial data set from the entire data set, this data set is optimized in each iteration, then the value of the objective function is improved in each iteration. The iterative process ends when a stop criterion derived from Karush–Kuhn–T ucker conditions is satisfied or a required accuracy is reached. LIBSVM uses a search direction algorithm which maximizes the increase in the objective function in each iteration. The algorithm starts by making a first approximation of the objective function by obtaining a vector α. From this first approximation compute $\alpha' = \alpha + \lambda u$, where the direction u has only two non zero coefficients. The algorithm uses two search directions, a search direction u^{ij} for positive λ and a search direction $-u^{ij} = u^{ji}$ for negative λ. The most effective search direction for each iteration will be the direction that maximizes the increase in the objective function.

Incremental SVM: is a framework for incremental learning and adaptation of support vector machine classifiers that aims to simplify the model selection task by perturbing the SVM solution as the regularization and kernel parameters are adjusted.

5. Applications in real-world problems

SVM applications have been used to solve many real-world problems, in this section we describe some of them.

5.1. Text (and hypertext) categorization

The authors in [181] presented the implementation of a text document classification framework that uses the SVM approach in the training phase and the Euclidean distance in the classification phase. In the proposed approach, the support vectors for each category are identified from the training data points during training phase using SVM. During classification, when a new data point is mapped into the original vector space, the average distances between the new data point and the support vectors from different categories are measured using the Euclidean distance. The classification decision is made based on the category of support vectors which has the lowest average distance with the new data point, making the classification decision irrespective of the efficacy of hyper-plane formed by applying the particular kernel function and soft margin parameter.

In [182], is evaluated three machines learning methods, k-nearest neighbor, SVM and adaptive resonance associative map are evaluated for Chinese document categorization. Based on two Chinese corpora, a series of controlled experiments evaluated their learning capabilities and efficiency in mining text classification knowledge. SVM is highly efficient in learning from well-organized samples of moderate size, although on relatively large and noisy data the efficiency of SVM and adaptive resonance associative map are comparable.

In reference [183], the authors showed that in the case of text classification, term-frequency transformations have a larger impact on the performance of SVM than a kernel itself. It is discussed the role of importance-weights, which is not totally understood given the model complexity and calculation cost. It is also shown that the time consuming lemmatization or stemming can be avoided even when classifying highly inflection language.

SVM is one of the techniques used in active learning to reduce data labeling effort in different fields of pattern recognition. In [184], it was presented a batch mode active learning using SVM for text classification, since most of the related works applying active learning methods to automatic text classification are focused on requesting the label of an unlabeled document in each iteration.

The authors in [185] bring forward linear SVM together with distributional clustering of words to realize its potential in text categorization realm. Distributional clustering has been presented as an efficient alternative to the feature selection conventionally used in text categorization. Distributed clustering together with linear SVM brings down the dimensionality of text documents without any compromise in classification performance. In this study, linear SVM and its extension fuzzy SVM were employed together with distributed clustering for text categorization.

The authors in [186] proposed an approach for term weighting in very short documents that is used with an SVM classifier. The paper focuses on market research and social media documents. In both data sources, the average length of a document is below twenty words. As the documents are short, each word occurs usually only once within a document. Thus, it was proposed an approach for term weighting that does not use term frequency within a document but substitutes it with other word statistics.

In high dimensions and large-scale multi-class textual data, it is common to ignore the semantic between words with the traditional feature selection method. The authors in reference [187] introduced the categories information into the existing LDA (Latent Dirichlet Allocation) model feature selection algorithm and construct SVM multi-class classifier on the implicit topic-text matrix.

In [188], it was presented a text classifier using positive and unlabeled examples. The challenge of this problem as compared with the classical text classification problem is that no labeled negative documents are available in the training example set. Many more reliable negative documents are identifier by an improved 1-DNF algorithm. Then, a set of classifiers are built by iteratively applying the SVM algorithm on a training data set, which is augmented during iteration. Later, different from previous PU-oriented text classification works, the weighted vote of all classifiers generated in the iteration steps is adopted to construct the final classifier instead of choosing one of the classifiers as the final classifier. The authors propose an approach to evaluate the weighted vote for all classifiers generated in the iteration steps to construct the final classifier based on particle swarm optimization.

In reference [189], the authors proposed the combined dichotomy transformations, a text categorization system that combines binary classifiers that are trained with different dichotomy sets using dichotomy transformation, where the number of training examples increases exponentially when they are compared with the original set. This property is desirable because each classifier can be trained with different data without reducing the number of examples or features. Thus, it is possible to compose an ensemble with diverse and strong classifiers. Experiments are preformed using SVM, random subspace, boostexter and random forest.

The authors in [190] promoted a new benchmark called RTAnews, which is a data set of multi-label Arabic news articles for text categorization. They conducted an extensive comparison of most of the well-known multi-label learning algorithms for Arabic text categorization in order to have baseline results and show the effectiveness of these algorithms for Arabic text categorization on RTAnew. The evaluation involves several algorithms, such as binary relevance, classifier chains, calibrated ranking, SVM, k-nearest neighbors (KNN), random forest and four adaptation-based algorithms. The results demonstrate that adaptation-based algorithms are faster than transformation-based algorithms.

5.2. Image classification

In [191], the image is enhanced with the help of median filter, Gaussian filter and un-sharp masking. After that, morphological operations and the entropy based segmentation are used to find the region of interest and finally the KNN and SVM classification techniques are employed for the analysis of kidney stone images. Kidney stone detection is a sensitive topic. There are various problems associated with this topic like low resolution images, similarity of kidney stone and prediction of stone in the new image kidney. Ultrasound images have low contrast and are difficult to detect and extract the region of interest. Therefore, the image has to go through the preprocessing which normally contains image enhacement.

Qiao et al. [192] presented a method which combines low and high frequency Curvelet coefficients with feature vectors based on the traditional features to make up for contour and texture feature in details. Commonly used parameters optimization algorithms in SVM are cross validation grid search, genetic algorithm and PSO algorithm. In reference [192], the smart algorithm is used for parameter optimization, making it simple and rapid.

The authors in [193] developed a handle device featured with low cost and high performance to enhance early detection of melanoma at the primary healthcare. SVM is a common classifier that shows high accuracy for classifying melanoma within the diagnosis system and is considered as the most compute-intensive task in the system. Therefore, the authors propose a dynamic hardware system for implementing a cascade SVM classifier on FPGA for early melanoma detection. A multi-core architecture is proposed to implement a two-stage cascade classifier using two classifiers obtaining high accuracies.

Orthogonal moments are the projections of image function onto particular kernel function. They play vital role in digital image feature extraction being rotation, scaling, translation invariant, robust to image noise and contain minimal information redundancy. Fractional-order moments are superclass of integer order and more efficient underrated. Kaur et al. [194] proposed fractional-order Zernike moments along with SVM to recognize grape leaf diseases.

Comparative analysis with integer-order Zernike moments along with other feature selection methods has been explored.

In [195], the authors presented a framework for person-independent expression recognition by combining multiple types of facial features via multiple kernel learning in multiclass SVM. Approaches based on multiple kernel learning jointly learn the same kernel weights with l1-norm constraint for all binary classifiers, whereas the proposed framework learns one kernel weight vector per binary classifier in the multiclass-SVM with lp-norm constraints, which considers both sparse and non-sparse kernel combinations within multiple kernel learning. The authors studied the effect of lp-norm of multiple kernel learning in multiclass algorithm for learning the kernel weights and evaluated the recognition results.

In reference [28], it was addressed the recognition of Indian signs based on dynamic hand gesture recognition techniques in real-time scenario. The captured video is converted to HSV color space for pre-processing and then the skin pixels are segmented. Hu moments and motion trajectory are extracted from the image frames and the classification of gestures is performed by using SVM.

The authors of reference [196] proposed a system consisting of three modules: digital zoom, adaptive skin detection and hand gesture recognition. The last module recognizes both static and dynamic hand gesture. The region of interest next to the detected user face is for fist/waving hand gesture recognition. An efficient algorithm using SVM is developed to classify the dynamic hand gestures under complex background, motion history image and four groups of novel Haar-like features are investigated to classify the dynamic and right hand gestures.

In reference [197], three robust approaches for feature extraction for gender classification were presented. The first approach is based on using discrete cosine transform and consists of two different methods for calculating features values. The second approach is based on the extraction of texture features using the gray-level co-occurrence matrix. The third approach is based on 2D-wavelet transform. The extracted features vectors are classified using SVM. K-fold cross validation is used in training the SVM.

The authors in [198] introduced a method for spectral-spatial classification of hyperspectral images. The proposed technique consists of two steps: 1) a probabilistic SVM pixelwise classification of the hyperspectral image is applied; 2) spatial contextual information is used for refining the classification results obtained in the first step, by means of a Markov random field regularization.

In reference [199], the feature selection methods for mass classification of mammograms are addressed. A procedure based on SVM recursive feature elimination is integrated with a normalized mutual information feature selection to avoid their disadvantages. Different initialization methods are investigated with spatial constraints as the initialization step. Different feature selection methods with a minimum redundancy-maximum relevance filter are used to select features and to compare mass classification results using the selected features.

The contribution of reference [200] concerns histogram intersection kernel SVM for image classification. The intersection coordinate descent and a deterministic and scalable histogram intersection kernel solver are proposed. The intersection coordinate descent is faster than general purpose SVM solvers and other fast histogram intersection kernel SVM training methods.

The use of color in QR codes brings extra data capacity, but also inflicts tremendous challenges on the decoding process due to chromatic distortion-cross-channel color interference and illumination variation. The authors of reference [201] proposed two approaches to solve these problems: LSVM-CMI and QDA-CMI which jointly model these different types of chromatic distortion. Extended from SVM and QDA (Quadratic Discriminant Analysis), respectively, LSVM-CMI and QDA-CMI optimize over a particular objective function and learn a color classifier.

5.3. Bioinformatics (protein classification and cancer classification

The authors in reference [202] proposed a semi-supervised SVM-based feature selection, which simultaneously exploits the knowledge from unlabeled and labeled data. Experimental results on the gene expression data of lung cancer show that semi-supervised SVM-based feature selection achieves the higher accuracy and requires shorter processing time compared with the well-known supervised method.

In [203], it was presented a clinical decision support system aimed to save lives, time and resources in the early diagnostic process. Segmentation, feature extraction and lesion classification are the important steps in the proposed system. The system analyzes the images to extract the affected area using a segmentation method. The underlying features which indicate the difference between melanoma and benign images are obtained through specialized texture analysis methods. Self-SVM is employed for classification which shows improved classification rate.

Zhang et al. [204] applied an 1-norm SVM with the squared loss to implement fast gene selection for cancer classification. The 1-norm SVM square loss performs gene selection and classification at the same. The approach is used as a gene selector and adopts a subsequent classifier to classify the selected genes.

The authors of reference [205] proposed to identify significant attributes in a well-established prostate cancer gene expression data set. Different statistical and artificial intelligence-based feature selection methods are paired with neural networks, Naive Bayes, AdaBoost and J48. Naive Bayes and AdaBoost achieve the best accuracy with SVM attribute selection. By investigating National Center Biotechnology Information database, 21 out of 24 attributes that belong to SVM attribute selection have a reference to cancer/tumor, establishing a link between feature selection and biological plausibility.

In reference [206], the performance of SVM classification to stratify the Gleason score of prostate cancer in the central gland was assessed, based on image features across multi parametric magnetic resonance imaging. Fifty-five variables are computed in the SVM classification. The classification model is developed with 10-fold cross-validation and is further validated mutually across two separated data sets.

The authors of reference [207] proposed an algorithm based on deep neural network and emotional learning process. Firstly, principal component analysis is applied for feature reduction; then, the features are extracted using a deep neural; different classifiers are implemented: multi-layer perceptron, SVM, decision tree and Gaussian mixture model. Experimental results show that, generally, using emotional learning increased the accuracy, where the highest accuracy is obtained using SVM.

The use of convolutional neural networks in the classification and diagnosis of medical image problems is becoming common and popular. Nevertheless, the training of convolutional neural networks requires a large data set of images. In [208], it was proposed to overcome this problem by using transfer learning to extract images features for further classification. Three architectures of convolutional neural networks are tested where the features are selected according to their gain ratios and used as input to the SVM classifier.

Mazo, Alegre and Trujillo [209] classified automatically cardiovascular tissues using texture information and SVM. Also, several cardiovascular organs are recognized following the same process. The texture of histological images is described using local binary patterns, local binary patterns invariant to rotation and Haralick features and different concatenations between them. SVM is selected as the classifier with a higher area under the curve that represents both higher recall and precision. A linear SVM allows the separation of four classes of tissue: cardiac muscle of the heart, smooth muscle of the muscular artery, loose connective tissue and smooth muscle of the large vein and the elastic artery.

In reference [210], the authors presented a method based on ultrasound RF time series analysis and an extended version of SVM classification for generating probabilistic cancer maps that can augment ultrasound images of prostate and enhance the biopsy process. The RF time tries are formed by recording sequential ultrasound RF echoes backscattered from tissue while the imaging probe and the tissue are stationary in position.

5.4. Hand-written character recognition

Bhowmik et al. [29] proposed recognition of hand-written Bangla characters based on SVM hierarchical classification schemes. It is observed that there are groups of characters having similar shapes. These groups are determined in two different ways on the basis of confusion matrix obtained from SVM classifier. Three different two-stage hierarchical learning architectures are proposed using the grouping schemes.

In reference [211], binary area matrix calculation was presented. The performance of the binary zone area matrix is measured individually and with combinations of other existing features. The proposed method for character recognition is applied to the eighteen different scripts authorized by the Government of India Sridhar. The recognition of the characters is performed using SVM classifiers.

Jebri et al. [212] proposed an optical character recognition system for Arabic characters. In the first phase the characters are extracted, in the second phase histograms of oriented gradient are used for feature extraction. The final phase employs SVM for character classification.

In [213], an approach for handwritten digit recognition is developed that uses a small number of patterns for training phase. The recognition performance is improved by using the bag of visual words technique to construct images feature vectors. Each visual word is described by scale invariant feature transform method. For learning feature vectors, SVM classifiers are employed.

The authors of reference [27] presented a system to recognize handwritten character for the Gujarati language. SVM with linear, polynomial & RBF kernel, k-NN with different values of k and multi-layer perceptron are employed to classify strokes using hybrid feature set.

In reference [214], it was proposed a framework of providing handwritten character recognition as a service via internet, based on cloud computing technology. SVM classifier is used, along with other classifiers, for large scale character recognition, writing adaptation technology and handwriting Chinese word/text recognition.

Bertolini et al. [215] investigated the efficacy of a writer-independent classifier based on dissimilarity for multi-script writer identification. Multi-script writer identification consists in identifying a person of a given text written in one script from the samples of the same person written in another script. The authors performed experiments on Arabic and English samples, features are extracted using the texture descriptors local binary patterns and local phase quantization; with these features SMVs with Gaussian kernel are used as classifiers. The free parameters of the system for the SVM are chosen using 5-fold cross validation; the parameters are determined through a grid search.

The authors of reference [216] performed experiments for handwritten character recognition using and comparing the performance of multi-layer feed forward back propagation neural network and SVM classifier. The neural network is trained with the pixels of character images resized into 7050 pixels, which is directly subjected to training. In other words, each resized image has 3500 pixels and these pixels are fed for the neural network training. For the SVM, 25 features are extracted from each character and these features are employed to train the SVM, where the polynomial kernel is used.

In reference [217], it was described a method for offline writer identification, using RootSIFT descriptors computed densely at the script contours. GMM (Gaussian Mixture Model) supervectors are used to describe the characteristic handwriting of an individual scribe. Exemplar-SVMs are proposed to train a document-specific similarity measure.

In [218], the authors proposed the block wise local binary count as descriptor for offline text independent writer identification of handwritten documents. The proposed operator characterizes the writing style of each writer by a set of histograms calculated from all the connected components in the writing. Each histogram is constructed by calculating the occurrence distribution of pixels corresponding to the writing within small blocks in each connected component extracted and cropped from the input handwriting sample. The samples are classified according to their normalized histogram feature vectors through the nearest-neighbor rule using the Hamming distance and SVM.

5.5. Face detection

Je, Kim and Yang Bang [219] proposed the automatic detection of human face in digital video using an SVM ensemble to improve the detection performance. The SVM ensemble consists of several SVMs trained using training samples via a bootstrap technique. They are aggregated in order to make a collective decision via a majority voting scheme.

One of the problems of face detection is the large variations because of some factors, like viewpoint, extreme illuminations and expression changes, leading to large intra-class variations and making the detection algorithms not robust enough. The authors of reference [220] proposed a locality-sensitive SVM using kernel combination algorithm to solve the problems mentioned before. The locality-sensitive SVM is employed to construct a local model on each local region, which handles the classification task. Then, multiple local convolutional neural networks are employed to jointly learn local facial features because of the strength of convolutional neural networks learning characteristic.

Face recognition plays an important role in video surveillance; these systems are exposed to challenging operational environments. The appearance of faces changes when captured under unconstrained conditions due to variations in pose, scale, illumination, occlusion, blur, etc. In [221], the authors developed a multi-classifier system based on multiple face representation and domain adaptation. An individual-specific ensemble of exemplar-SVM classifiers is thereby designed to improve robustness to intra-class variations. During enrollment of a target, an ensemble is used to model the single reference, where multiple face descriptors and random feature subspaces allow generating a diverse pool of patch-wise classifiers. These ensembles are adapted to the operational domains; the exemplar-SVMs are trained using labeled face patches extracted from the reference still versus patches extracted from cohort and other non-target stills mixed with unlabeled patches extracted from the corresponding face trajectories captured with surveillance cameras.

In reference [222], the authors proposed a methodology to solve the problem of full illumination variation by the combination of histogram equalization and Gaussian low-pass filter. So as to process illumination normalization, feature extraction is applied with consideration of both Gabor wavelet and principal component analysis. An SVM classifier is used for face classification.

The authors of reference [223] presented an automatic gender recognition algorithm based on machine learning methods. It consists of two stages: adaptive feature extraction and SVM classification. The algorithm consists of the following steps: color space transform, image scaling, adaptive feature set calculation and SVM classification with preliminary kernel transformation.

In [224], the authors proposed a method for face detection based on principal component analysis and SVM. Firstly, the potential face area of the image is filtered using statistical feature, which is generated by analyzing local histogram distribution; and then, SVM classifier is used to detect face feature in the test image. PCA is employed to reduce dimension of sample data; after PCA transform, the feature vectors, which are used for training SVM classifier, are generated.

Kumar, Kar and Chandra [225] employed mean and median filters which are normally used to reduce noise present in an image and for preserving useful detail in the image. Adaptive filtering is more selective which helps for preserving edges and other high frequency parts of an image. Once the noise from an image has been removed the image is sent to the SVM trained for identification of faces.

While a number of face spoof detection techniques have been proposed, their generalization ability has not been adequately addressed. In reference [226], it was proposed a robust face spoof detection algorithm based on image distortion analysis. Four different features: spectacular reelection, blurriness, chromatic moment and color diversity, are extracted to form the feature vector. An ensemble classifier, consisting of multiple SVM classifiers trained for different face spoof attacks is used to distinguish between genuine and spoof faces.

Waring and Liu [227] presented a face detection method using spectral histograms and SMV. Each image window is represented by its spectral histogram, which is a feature vector consisting of histograms of filtered images. Using statistical sampling, the authors showed systematically the representation groups face images together; in comparison, commonly used representations often do not exhibit this necessary and desirable property. The high performance of the approach is attributed to the desirable properties of the spectral histogram representation and good generalization of the SVMs.

Face direction detection plays an important role in human–computer interaction and has a wide application. Current detection methods are mainly focused on extracting specific patterns from user's optical images, which raises concerns on privacy invasion and these detection techniques do not usually work in dark environments. In reference [228], the authors developed an activity recognition system guided by an unobtrusive sensor. By using a low pixel infrared thermopile array sensor, the proposed system is capable of identifying five facing directions through the SVM classifier.

5.6. Protein fold and remote homology detection

Rahman et al. [229] presented a computational model that introduces ways to extract features from protein sequences, but also optimizes classification of trans-Golgi and cis-Golgi proteins. After feature extraction, random forest model is employed to rank the features based on the importance score obtained from it. After selection of the top ranked features, the Golgi proteins are classified using SVM.

In [230], a computational approach was tried to find the evolutionarily related fold of the receptor-associated proteins. Through the structural and sequence-based analysis, various protein folds were found that are very close to the receptor-associated protein folds. Remote homolog data sets were used potentially to develop different SVM methods to recognize the homologous receptor-associated protein fold.

Mei [231] presented SVM ensemble based transfer learning model for membrane proteins discrimination, to reduce the data constraints on computational modeling. This method investigates the effectiveness of transferring the homolog knowledge to the target membrane proteins under the framework of probability weights ensemble learning. As compared to multiple kernel learning based transfer learning model, the method takes the advantages of sparseness based SVM optimization on large data; hence, is more computationally efficient for large protein data analysis.

In reference [232] it was developed an intelligent prediction system for protein sub cellular localization using fluorescence microscopy images. The proposed prediction system uses a feature extraction strategy and ensemble classifications. The feature extraction mechanism exploits statistical and text based image descriptors, whereas ensemble classification is performed using the majority voting based ensemble of SVMs. The contribution of this work lies in the individual exploitation of the feature spaces generated from both individual gray level co-occurrence matrices and sexton images as well as in the manner the extracted features are exploited using the learning capabilities of SVM, which is utilized as base classifier in majority voting based ensemble.

In [233], the performances of SVM and neural networks for lipid binding proteins identifications were compared. Fivefold cross-validation and independent evaluation tests are used to assess the validity of the two methods. The results indicated that SVM outperforms neural network.

Remote homology detection at amino acid level is a complex problem in bio-informatics. Customary detection methods may be replaced by SVM based approaches where a sequence is represented by significant feature vectors. Two approaches are presented in [234]: 1) 2-mers are generated from individual amino acid for protein sequences and various physicochemical parameters are used to generate the feature vector; 2) the properties of amino acid are used to create the feature vectors using 3-mers in the similar manner. Principal component analysis is employed for dimensionality reduction and SVM is applied for classification.

The authors in reference [235] proposed a profile-based representation for sequences called Ngram. This representation extends the traditional Ngram scheme and permits considering all of the evolutionary information in the profile. Ngrams are extracted from the whole profile, equipping them with a weight directly computed from the corresponding evolutionary frequencies. Two different approaches are proposed to model the representation and to derive a feature vector which can be effectively used for classification using an SVM.

Bedoya and Tischer [236] presented a method for remote protein homology detection. Usually, the discriminative methods concatenate the values extracted from physicochemical properties to build a model that separates homolog and non–homolog examples. Each discriminative method uses a specific strategy to represent the information extracted from the protein sequence and a different number of indices. After the vector representation is obtained, SVMs are often employed. The contribution of this work lies on reducing the high dimensionality of the feature vector using models that are defined at the 3D level.

Homology-based methods have been developed to detect protein structural classes from protein primary sequence information, these methods are divided into three types: 1) discriminative classifiers, generative models for protein families and pairwise sequence comparisons. SVMs have shown being fast speed during training, more accurate and efficient compared to neural networks. In reference [237] it was presented a comprehensive method based on two-layer classifiers. The first layer detects up to superfamily and family in SCOP (Structural Classification of Proteins) hierarchy using optimized binary SVM classification rules. It employs the Bio-kernel, which incorporates the biological information in the classification process. The second layer uses discriminative SVM algorithm with string kernel that detects up to protein fold level in SCOP hierarchy.

5.7. Generalized predictive control

Wang and Kwong [78] presented a surge control strategy for centrifugal compressor using nonlinear model predictive control based on lease-squared SVM in order to increase efficiency of centrifugal compressor. The nonlinear predictive models of compressor's discharge pressure and mass flow are developed by the lease-square SVM.

In [238], an SVM-based multi-model predictive control is proposed, in which SVM classification combines well with SVM regression. Each working environment is modeled by SVM regression and the SVM network-based model predictive control algorithm corresponding to each environment is developed, and then a multi-class SVM model is established to recognize multiple operating conditions. For control, the current environment is identified by the multi-class SVM model and then the corresponding controller is activated at each sampling instant.

The lease-squared SVM has been successfully used to predictive control on small samples, nonlinear data. It was presented the predictive model methodology in reference [239]. The control results are influenced by the parameters on the lease-squared SVM. They are optimized by genetic algorithm, according to root mean square relative error. To successfully control the depth and attitude of an autonomous underwater vehicle, it is important to study the algorithm of the control. The autonomous underwater vehicle control on lease-square SVM is including the direction, depth, trims and roll movement.

In reference [240], the authors proposed a systematic data-driven method for the design of quantized explicit model predictive control for time-varying output tracking in nonlinear systems. The design involves: sampling the admissible state space; at each sampled point, solving for optimal quantized model predictive control actions and determining feasibility of the intrinsic mixed-integer nonlinear programming problem, and constructing the quantized explicit model predictive control surface using multi-class SVMs.

Chu et al. [241] developed a rapid modeling method for centrifugal compressor based on model migration and SVM. The base model of an existing old compressor is revised to fit for the new compressor by SVM. This method is evaluated by a simulation case and the results show that, compared with the pure SVM, the migrated model can fit the new compressor faster with better accuracy.

The authors of reference [242] analyzed whether trust can be used as a predictor of cross-functional team performance by proposing a prediction model. The inputs of the model are both team structural and contextual factors and project process factors, which are two major sources that form team trust. The output of the model is different of team performance, which consists of internal performance and external performance. The SVM techniques are used to establish the model. The authors give reference for managers to dynamically control and predict team performance during project period.

5.8. Complex classifications problems

SVM have been successfully used in multiple fields of application. However, there are very complex applications where further research is still needed to obtain satisfactory results. This section lists some of those applications:

5.8.1. Plant species classification

The classification of plant species from digital images has been a subject studied in recent years and today, very important results have been obtained in data sets of images with fully controlled environments and with very specific characteristics. However, important results are still not obtained in images of plants in uncontrolled and partially controlled environments or in data sets with very few images [243–245].

Moreover, current studies are carried out on very small data sets. However, the vast majority of the species in the world is around ($369K$ species) and the performance of state-of-the-art machine learning algorithms on these data sets is unknown and presumably much lower [246].

5.8.2. Classification of credit card fraud

Currently, the credit-card fraud is turning into a substantial challenge for financial institutions and service providers. Reported studies on the use of modern data-driven and learning-based methods to detect credit-card fraud are relatively few [247–250]. To solve these kind of problems is not common due to the imbalance in the data set (many examples of one class and very few of another class). The ratio of non-fraudulent transactions to fraudulent transactions is around 99.83% to 0.17% respectively.

The principal challenge is to design new algorithms to cope with the disadvantage of working with imbalanced data sets because performance of machine learning algorithms changes a lot when they are trained with unbalanced data sets. These algorithms tend to show a bias for the majority class, treating the minority class as a noise in the data set.

5.8.3. Classification and staging of melanoma

Melanoma is the most aggressive type of skin cancer, its diagnosis is unstable in 25%. Therefore, research has been done for the analysis of melanoma through computer vision images and great progress has been observed. Support vector machines is a robust machine learning model that shows high accuracy with different classification problems, although there is a disadvantage of the SVM classifier in many cases as integrated detection systems and some image processing since the SVM model is computationally expensive and time consuming. Although the implementation of SVM in software produces high accuracy rates and with real-time limitations, the accuracy in the detection of melanoma and early diagnosis can help to reduce mortality rates and treatment costs. Dermatoscopic images acquired are used in computational analysis for the detection of skin cancer, however there are limitations of image quality such as noise, shadows, artifacts that compromise the robustness of skin image analysis [251–253].

6. Trends and challenges

Large amounts of data are generated and collected at each moment. The supervised and unsupervised learning methods of machine learning are the responsible for transforming these data into useful information. SVMs have proven to be one of the best supervised learning methods in various applications; however, since SVM development several challenging problems have been identified to be able to use this classifier with very large data sets, also in dynamic environments such as data streams with concept drift, in multi-class problems, in data sets with few tagged data, and the selection of the right kernel and adjusting its parameters efficiently. These challenges are more difficult to solve when two or more of them are presented simultaneously. In the following, we explain in brief some of the challenge problems of SVM classifier.

- **Multi-class SVM**. The mathematical formulation of the SVM classifier is designed for two classes, these are the positive and the negative. Such a formulation restricts the direct application of SVM to binary problems. However, many real-world data sets have multi-class output.

  For more than two decades several methods have been proposed to extend the capabilities of SVM to face multi-class problems. The earliest and commonest methods use one-against-all or one-against-one approach.

  The basic idea of one-against-all or one-against-rest approach is to train an amount of SVMs equal to the number of classes. Each binary classifier is constructed considering the samples of one class as the positive, and the rest of instances as belonging to the negative class. The prediction is assigned to the model that produces the largest value after evaluating its decision function. One-against-one approach [124,125] builds binary SVMs using only the samples that belong to two different classes, therefore, the number of models obtained is $\frac{K(K-1)}{2}$, with K equals to the number of classes. To decide the class of a previously unseen instance, a voting scheme between all models is used. Voting strategy to achieve high classification accuracy is a key component for facing multi-class problems with SVM. To compensate the errors in the predictions of individual SVM binary classifiers, differential evolution was applied in [126].

  Other methods are similar to the work presented by Bredensteiner in [127]. The proposal of this approach is to construct a piecewise-nonlinear classification function. Each piece of this classifier can be a polynomial, a radial basis function, or a classifier such as a neural network. Recently, Tang [128] proposed a method that maps the K classes to K vertices of a $(K-1)$-dimensional regular simplex so that the K-class classification becomes a $(K-1)$-output learning task.

  Shao et al. [129] proposed a method that use a binary tree classifier which is build with a method that maximizes the distance between the classes in each partition, for this purpose twin support vector machines are trained. Although the reported results show the effectiveness of this method for facing multi-class problems, a drawback is that it needs to train $2^{K-1}-1$ SVM at each stage. In [130], a decision tree like algorithm capable of tackling multi-class data sets was presented. Similar to other approaches, a binary SVM is used to split the data at each level of the decision tree. Different from other proposals, a kernelized clustering algorithm is used to create the sets of positive and negative samples.

  Santosa [131] proposed a method based on the one-against-rest and one-against-one approaches for multi-class problems with SVM. To train the SVMs the Cross entropy method is utilized. Cross entropy is a stochastic optimization method that consists in improving solutions iteratively by means of an specific random mechanism. Usually, initial solutions are generated randomly following a normal distribution with mean an standard deviation established arbitrarily. A subset of the best solutions is used to update the mean and standard deviation, these new parameters are used to generate the next set of solutions. This process is repeated until a stopping criteria is satisfied. Four experiments were done in [131]. The results of experiments showed that Cross entropy has less computational complexity than the standard quadratic programming SVM, besides, it produces comparable results in terms of generalization error.

  Most of methods use one-against-all or one-against-rest approaches for facing multi-class problems with SVM. For a large number of classes, new heuristic, stochastic or hybrid methods need to be designed to improve classification accuracy.

- **Multi-task SVM**

  Multi-task learning (MTL) is a recently area of machine learning based on the assumption that if different tasks are related among them, then jointly learning these multiple tasks can lead to better performance than learning them independently, i.e., the idea is to leveraging useful information among the tasks. The determination of the relatedness between tasks is usually the key to the formulation of MTL [254].

  For supervised MTL with SVM, most of current approaches use multiple multitask binary problems, this can be seen as the opposite to the philosophy of MTL because the relationships between classes are ignored. Ji and Sun [255] proposed to cast multitask multiclass problems into a constrained optimization problem with a quadratic objective function, this approach produces accurate prediction, as shown in the results of experiments.

  Deeper studies on supervised MTL with SVM to determine its usefulness in specific real-world applications is a current trend.

- **Large-scale problems**. The training of an SVM basically consists in solving a QP problem, this task is a high computational burden when the number of instances is large. In the last past few years scholars have proposed some methods to enable SVM on large data sets. Most common approaches are based on the following strategies:

  - *Under sampling.*

    The underlying idea with under sampling is to select a small yet significant number of samples from a large data set to train SVM. In order for the generated model to perform well in classifying new data, it is necessary to design an adequate data selection strategy. Among these strategies random sampling and support vectors candidate selection are the most studied, these consists in the identification of data that have a high probability of being support vectors.

    In [256–258] the SVs candidates set is formed by picking the points from all CHs. This type of strategy is based on the observation that SVs are the closest points and they correspond to convex hull [113]. Other approaches [259,260] use the fact that SVs are usually the closest to opposite class or far from the class center. On the other hand, some authors propose to train a classifier different from SVM [26,244,261,262] or a linear SVM [263,264] to select SV candidates, and then train with them. Clustering methods [265–267,73] or heuristic methods [263,268,269[also have been studied.

  - *Alternative optimization methods.*

    The QP problem associated to SVM has one global optimal solution. In order to search this solution, interior point methods, gradients methods, decomposition methods, are used in many works. These methods can converge slow in many cases. To improve performance the stochastic gradient descent method called PEGASOS was proposed in [270], the run-time of PEGASOS does not depend directly on the size of the training set. Recently, Wang et al. [271] have adapted PEGASOS to twin support vector machines. Experiments showed that the method can handle large scale problems easily.

  - *Transform QP problem into a simpler problem*

    Instead of solving a QPP that is costly in terms of memory and computing time, an approach is to transform such problem into a one easier to solve. The Sequential Minimal Optimization (SMO) algorithm proposed by Platt [15] decomposes the QP problem into a series of smallest possible sub-problems that are solved analytically. This is one of the most successfully methods implemented in some libraries. The Cholesky decomposition to solve SVM iteratively was proposed in [272]. De Lima et al. [273] proposed a Sherman–Morrison–Woodbury formulation which is employed to reduce the complexity of nonlinear Improvements on least squares twin multi-class support vector machine. The solution requires solving two systems of linear equations, instead of solving two QPPs.

New methods to put the QPP of SVM in a simpler form, and then solve it in more efficient ways have not been discovered yet. .

- *Geometric approach*. The convex-hull (CH) of a set of points X is the smallest convex set that contains all the elements in X. Usually, CH is very small compared with the data set X. Mavrofarakis in [274] took advantage of the geometry of SVM to propose a framework for training SVM using projections of points that belong to CH. For the linearly inseparable case, Mavrofarakis proposed to use reduced or shrinked CHs, transforming the problem into a linearly separable one. Different from this approach, Liu et al. [275] use measures in the projected high dimensional feature space for SVM since this is where the separating hyperplanes are determined to select a subsets of instances.

- *Parallel algorithms*. With the availability of tools to perform parallel computing using low-cost computer clusters, progress has been made in parallelizing algorithms to train SVM. In [276], the MapReduce framework is employed to train SVM in a distributed way to predict protein–protein interactions. Efficient implementations of SVM in dedicated hardware architectures, distributed systems or GPUs can help to apply SVM classifiers on large-scale problems or in dynamic environments.

- **On-line SVM**

  Data streams such as sensor networks, financial markets, social networks, and healthcare monitoring systems are very common in these days [277]. In these environments, the distribution of classes is changing dynamically (this phenomena is known as concept drift), therefore, the predictor must be updated according to the changes. A classifier needs to be trained very fast and easily upgradeable, these two requirements make difficult to apply SVM classifier on data streams.

  Zheng et al. [278] proposed to use a type of sampling from a data stream, the method learns the prototypes and continuously adjusts prototypes to the data concept, an SVM is then trained with these elements. Recently, Wang and Xing [279] noticed that training SVM with old and new prototypes can have a bad influence on its performance due to losing much information about the data. They proposed to use a Representative Prototype Area (RPA), that retains the representative data of all historical data. In the RPA, each class is maintained by an Online Incremental Feature Map (OIM) which learns a suitable representative set from the stream data automatically. Experiments showed that the algorithms can deal with data sets with millions of samples.

  Liu et al. [280] pointed that On-line learning is more difficult on data over distributed environments, also when data privacy is required.

- **Kernel choice and parameter optimization**

  In order to apply SVM successfully on a data set, it is necessary to select appropriate kernel and also tune its parameters. The basic approach to tune the parameters of kernels is the so-called grid search algorithm. It is a process that searches exhaustively through a manually specified subset of the hyper-parameter space of the targeted algorithm [281]. Other approach is based on analyzing the class separability [282]. Candelieri et al. [283] proposed a parallel global optimization model to optimize the hyper-parameters of Support Vector Machine. Wang [284] created a method for parameter selection of SVM with Gaussian kernel (one of the most commonly used). Other approaches use evolutionary algorithms [285].

  Automatic kernel choice and calibration of this parameters in a low computational cost fashion is a problem that has not been completely solved. For some applications, such as speech recognition [286], recognition based on image set [287], the design of new kernel functions can improve the performance of SVM. This is another important issue.

- **Semi-supervised and transductive SVM**

  Supervised learning methods, such as SVM and others, need the data be labeled. Collecting unlabeled data is usually easy, cheap, and it can be done automatically; however, the manual labeling of data is a slow and error-prone process, or even unfeasible in some circumstances, such as on-line applications. New strategies to enable SVM on partially labeled data sets are necessary. Recently, extensions to apply SVM on partially labeled data have been attracted the attention of researchers and practitioners. One of the most representative algorithms is Transductive SVM (TSVM), the idea is to find an hyperplane that separates the labeled samples with a large margin, but at the same time that ensures that the unlabeled instances will be as far as possible from the margin. In some tasks, semi-supervised SVM can be applied successfully by exploiting the information contained in data [288]. Davy et al. [289] used a sequential optimization algorithm to detect abnormal events. However, not in all cases is feasible to take advantage of the characteristics of the problem.

  Chevikalp and Franc [290] replaced the Hinge loss that is used for labeled data with a Ramp loss (a loss function is a measure of the distance between predictions and the real class, Hinge and Ramp are examples of loss functions). Also, they solved the optimization problem in the primal space (most of approaches do this on dual space) by using a stochastic gradient algorithm. To give an idea of the speed of this state-of-the-art method, it takes 38 s to train a TSVM with 2,000 labeled samples and 8,000 unlabeled ones. Each instance with 100 attributes. It is reported that this algorithm is able to train a TSVM with about 400,000 instances each one with 2,000 features, however, the training time is not reported.

  Li et al. [291] pointed that in TSVM the unlabeled examples harm the performance of the classifier. To address this problem, they proposed two algorithms to optimize the margin distribution of TSVM via maximizing the margin mean and minimizing the margin variance simultaneously. Results showed that this solution is robust to the outliers and noise.

Among the plethora of classification methods, SVM has been one of the most popular methods in a wide variety of applications, this due to its good classification accuracy. In spite of this outstanding performance, some important problems of this classifier have been identified since it was published. Although most of these problems have been studied for more than ten years, and there are many proposals to solve them, every time it becomes more necessary to develop better solutions.

Deep learning Vs SVM

Deep learning is a set of machine learning algorithms that attempts to model high-level abstractions in data using computational architectures that support multiple non-linear and iterative transformations of data expressed in matrix or tensorial form [292]. In recent years the popularity and expansion of Deep Learning has grown due to its potential utility in different types of applications in the “real world”, mainly because it obtains high success rates with “unsupervised” training. Many studies have been conducted with deep learning especially in healthcare, finance, speech

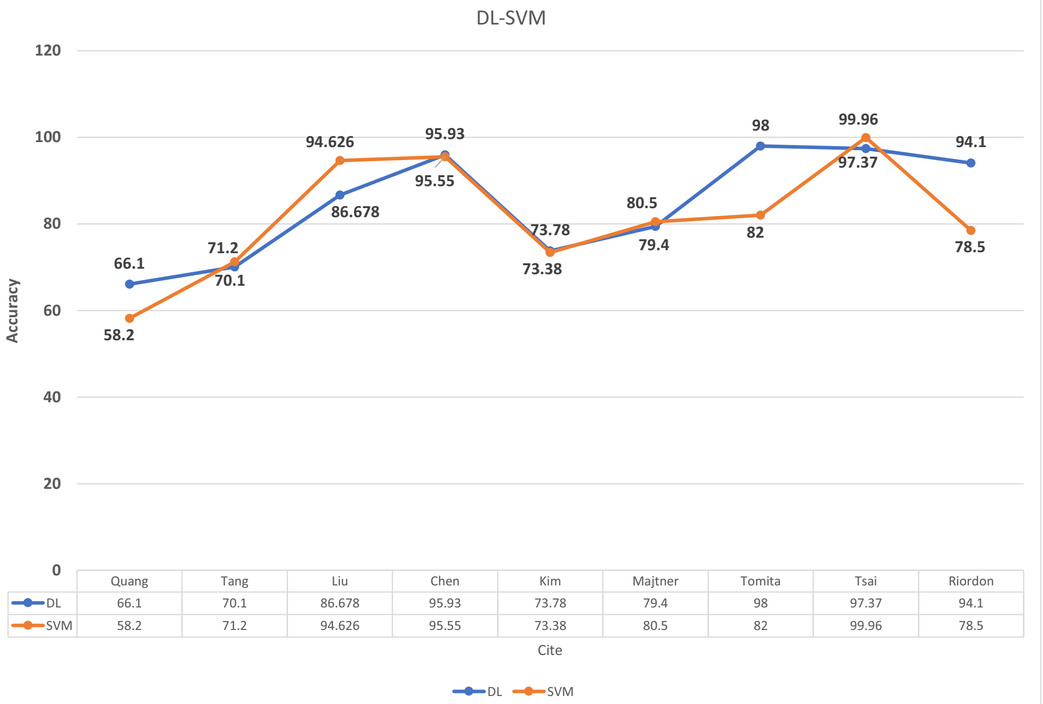


Fig. 5. Performance of SVM vs Deep Learning.

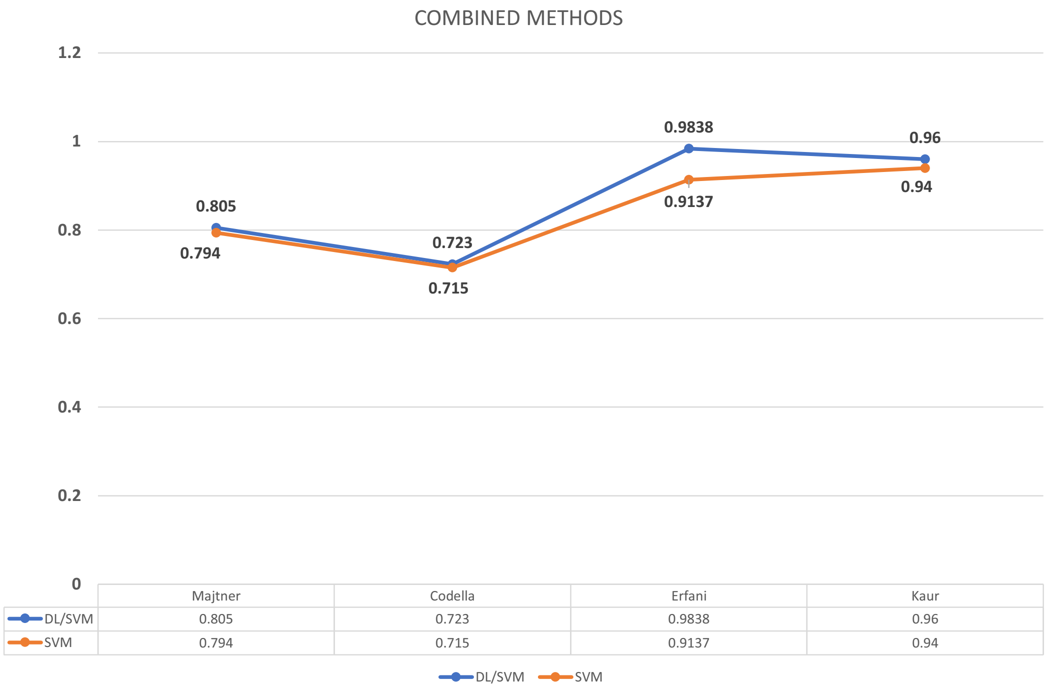


Fig. 6. Performance of combined SVM and Deep learning.

recognition, augmented reality, digital image processing and more complex 3D and video applications.

In this Section is shown a comparative of performance between deep learning and SVM from several authors. The Figs. 5 and 6 show the results obtained by different authors in different applications [293–301]. The Fig. 5 shows the comparison of the performance obtained using some SVM algorithm and deep learning. In the Figure.

Some authors propose the use of deep learning combined with SVM. The reported results improve the results in most cases. In some cases, the combination is done by training one of the SVM classes from the characteristics learned by the convolutional neural network, a linear kernel can be replaced by non-linear ones without losing precision. The Fig. 5 shows the results obtained with combinations of SVM and Deep learning [298,302,303,194].

It is important to note that SVM and deep learning have similar performances on average. Together they can work in synergy improving performance in different applications.

6.1. Impact of SVM

We analyzed the impact of SVM in the literature. Fig. 7 shows the distribution of research papers of SVM by year.

Most of the SVM papers have been written mainly for solving problems in normal-sized data sets and balanced data sets, where the SVM does not have problems. We used Science direct and IEEE Xplore search engines to retrieve publications, published only in journals, containing the term SVM, the search produced more than 13,000 results. Fig. 7 shows the number of publications in book chapters and journals per year, from 1998 to 2018.

For the purpose of identifying two applications of SVM (Large data sets and imbalanced data sets), we searched publications related to applications of SVM using the search engine IEEE Xplore and Science direct. We selected the publications that satisfy the following criteria: • The papers analyzed are the published in journals and book chapters. • Papers from 1998 up to 2018 were considered in the study. • Publications were selected that contain at least one of the search terms in the title, abstract and/or list of key- words.

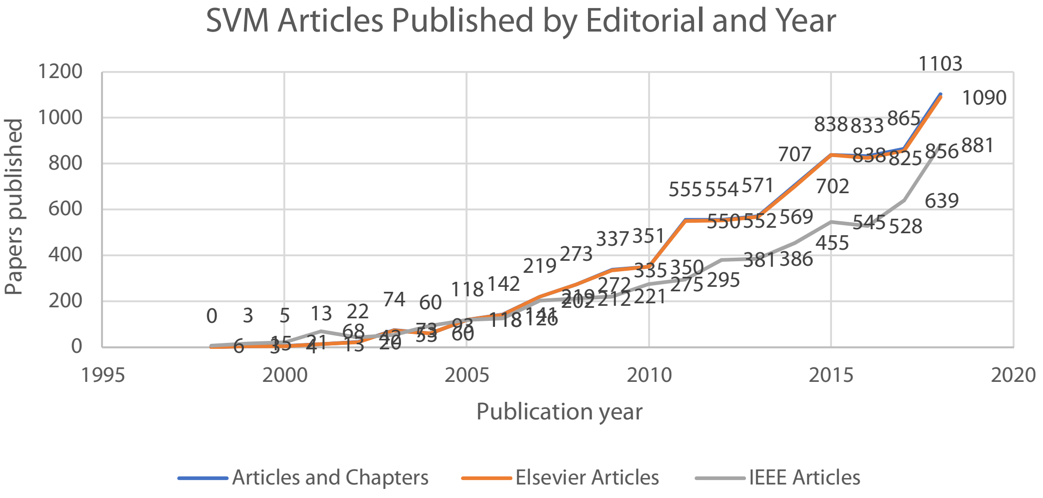


Fig. 7. Number of publications, in book chapters and journals, per year which contain the search term SVM.

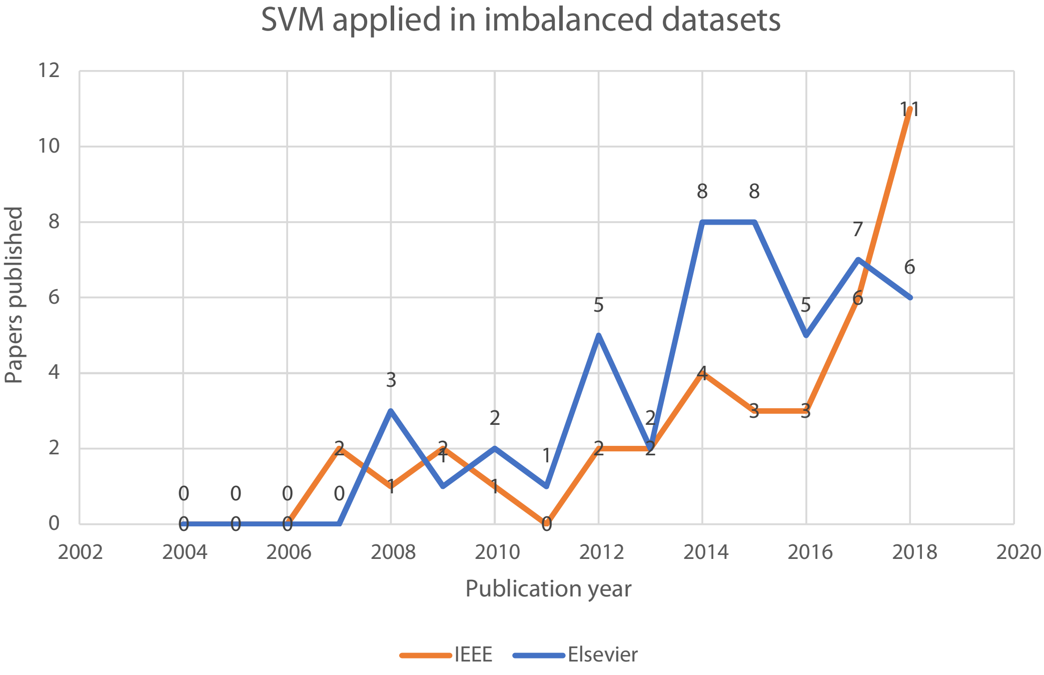


Fig. 8. Number of publications, in book chapters and journals, per year which contain the search terms SVM and Imbalanced data sets.

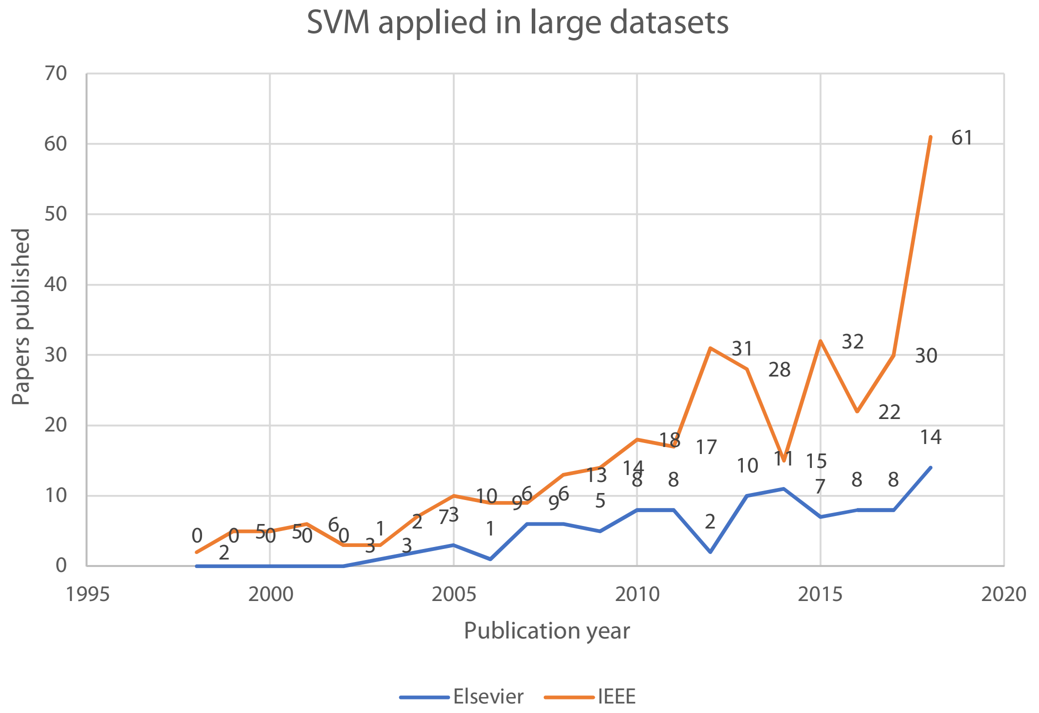


Fig. 9. Number of publications, in book chapters and journals, per year which contain the search terms SVM and Large data sets.

We found 440 publications of SVM on Large data sets and 85 publications of SVM on imbalanced data sets. The Figs. 8 and 9 show the number of publications, in book chapters and journals.

7. Conclusions

Due to its good theoretical foundations and generalization capacity among other advantages, the SVMs have been implemented in many real-world applications. SVM algorithms have been implemented in many research fields like: Text (and hypertext) categorization, Protein fold and remote homology detection, Image classification, Bioinformatics (protein classification and cancer classification), Hand-written character recognition, Face detection, Generalized predictive control and many more. Many researchers have shown that SVMs are better than other current classification techniques. However, despite SVM has some limitations related to: parameter selection, algorithmic complexity, multiclass data sets and imbalanced data sets, SVM has been implemented in many real life classification problems due to its good theoretical foundations and generalization performance.

It is important to mention that SVM is not so popular when the data sets are very large because some SVM implementations demand huge training time or in other cases when the data sets are imbalanced, the accuracy of SVM is poor, we have presented some techniques when the data sets are imbalanced. This paper describes in detail the principal disadvantages of SVM and many algorithms implemented to face these disadvantages and cites the works of researchers who have faced these disadvantages.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson, 2005.
[2] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Statist. 46 (Aug 1992) 175–185.
[3] Y. Zhang, G. Cao, B. Wang, X. Li, A novel ensemble method for k-nearest neighbor, Pattern Recogn. 85 (2019) 13–25.
[4] B.G. Marcot, T.D. Penman, Advances in bayesian network modelling: integration of modelling technologies, Environ. Modell. Software 111 (Jan 2019) 386–393.
[5] B. Drury, J. Valverde-Rebaza, M.-F. Moura, A. de Andrade Lopes, A survey of the applications of bayesian networks in agriculture, Engineering Applications of Artificial Intelligence, 2017, vol. 65, pp. 29–42. .
[6] D. Huang, Systematic theory of neural networks for pattern recognition, Publishing House of Electronic Industry of China (1996).
[7] D.-S. Huang, Radial basis probabilistic neural networks: model and application, Int. J. Pattern Recognit Artif Intell. 13 (07) (1999) 1083–1101.
[8] D.-S. Huang, A constructive approach for finding arbitrary roots of polynomials by neural networks, IEEE Trans. Neural Networks 15 (March 2004) 477–491.
[9] D.-S. Huang, H.H.S. Ip, K.C.K. Law, Z. Chi, Zeroing polynomials using modified constrained neural network approach, IEEE Trans. Neural Networks 16 (May 2005) 721–732.
[10] D.-S. Huang, Radial basis probabilistic neural networks: model and application, Int. J. Pattern Recognit Artif Intell. 13 (07) (1999) 1083–1101.
[11] A. Trabelsi, Z. Elouedi, E. Lefevre, Decision tree classifiers for evidential attribute values and class labels, Fuzzy Sets Syst., 2018. .
[12] M. Fratello, R. Tagliaferri, Decision trees and random forests, in: Encyclopedia of Bioinformatics and Computational Biology, Elsevier, 2019, pp. 374–383.
[13] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, 1998.
[14] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.
[15] J. Platt, Sequential minimal optimization: a fast algorithm for training support vector machines, tech. rep., 1998. .
[16] D.S. Huang, J.X. Du, A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks, IEEE Trans. Neural Networks 19 (Dec 2008) 2099–2115.
[17] X.-F. Wang, D.-S. Huang, A novel multi-layer level set method for image segmentation, J. Univers. Comput. Sci 14 (14) (2008) 2428–2452.
[18] Z.-Q. Zhao, D.-S. Huang, A mended hybrid learning algorithm for radial basis function neural networks to improve generalization capability, Appl. Math. Model. 31 (7) (2007) 1271–1281.
[19] J.-X. Du, D.-S. Huang, X.-F. Wang, X. Gu, Shape recognition based on neural networks trained by differential evolution algorithm, Neurocomputing 70 (4) (2007) 896–903, Advanced Neurocomputing Theory and Methodology.

[20] J. Du, D. Huang, G. Zhang, Z. Wang, A novel full structure optimization algorithm for radial basis probabilistic neural networks, Neurocomputing 70 (1–3) (2006) 592–596.

[21] M. Zhang, H. Qu, X. Xie, J. Kurths, Supervised learning in spiking neural networks with noise-threshold, Neurocomputing 219 (Jan 2017) 333–349.

[22] L. Huang, Y. Cui, D. Zhang, S. Wu, Impact of noise structure and network topology on tracking speed of neural networks, Neural Networks 24 (Dec 2011) 1110–1119.

[23] B.-Y. Sun, D.-S. Huang, H.-T. Fang, Lidar signal denoising using least-squares support vector machine, IEEE Signal Process. Lett. 12 (Feb 2005) 101–104.

[24] P. Chen, B. Wang, H.-S. Wong, D.-S. Huang, Prediction of protein b-factors using multi-class bounded SVM, Protein Peptide Lett. 14 (Feb 2007) 185–190.

[25] X. Liang, L. Zhu, D.-S. Huang, Multi-task ranking SVM for image cosegmentation, Neurocomputing 247 (Jul 2017) 126–136.

[26] J. Cervantes, F. García Lamont, A. López-Chau, L. Rodríguez Mazahua, J. Sergio Ruíz, Data selection based on decision tree for SVM classification on large data sets, Appl. Soft Comput. J. (2015).

[27] V.A. Naik, A.A. Desai, Online handwritten gujarati character recognition using svm, mlp, and k-nn, in: 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, pp. 1–6. .

[28] J.L. Raheja, A. Mishra, A. Chaudhary, Indian sign language recognition using svm, Pattern Recogn. Image Anal. 26 (Apr 2016) 434–441.

[29] T.K. Bhowmik, P. Ghanty, A. Roy, S.K. Parui, Svm-based hierarchical architectures for handwritten bangla character recognition, Int. J. Document Anal. Recogn. (IJDAR) 12 (2009) 97–108.

[30] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Disc. 2 (2) (1998) 121–167.

[31] W. Karush, Minima of functions of several variables with inequalities as side conditions (Master's thesis), Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939. .

[32] H.W. Kuhn, A.W. Tucker, Nonlinear programming, in: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, (Berkeley, Calif.), University of California Press, 1951, pp. 481–492..

[33] S. Haykin, Neural Networks: A Comprehensive Foundation (second ed.), Prentice Hall, 1998.

[34] M. Achirul Nanda, K. Boro Seminar, D. Nandika, A. Maddu, A comparison study of kernel functions in the support vector machine and its application for termite detection, Information 9 (2018) 5.

[35] S. Kasnavi, M. Aminafshar, M. Shariati, N. Emam Jomeh Kashan, M. Honarvar, The effect of kernel selection on genome wide prediction of discrete traits by support vector machine, Gene Reports 11 (2018) 279–282.

[36] M. Hasan, S. Xu, M. Kabir, S. Ahmad, Performance evaluation of different kernels for support vector machine used in intrusion detection system, Int. J. Comput. Networks Commun. 8 (6) (2016) 39–53.

[37] K. Chui, M. Lytras, A novel moga-svm multinomial classification for organ inflammation detection, Appl. Sci. (Switzerland) 9 (11) (2019).

[38] S. Saeed, H. Ong, Performance of svm with multiple kernel learning for classification tasks of imbalanced datasets, Pertanika J. Sci. Technol. 27 (1) (2019) 527–545.

[39] J. Pennington, F.X. Yu, S. Kumar, "Spherical random features for polynomial kernels, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS'15, (Cambridge, MA, USA), MIT Press, 2015, pp. 1846–1854. .

[40] M.A.S. ALTobi, G. Bevan, P. Wallace, D. Harrison, K. Ramachandran, Fault diagnosis of a centrifugal pump using mlp-gabp and svm with cwt, Eng. Sci. Technol. Int. J. 22 (3) (2019) 854–861.

[41] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, C.-J. Lin, Training and testing low-degree polynomial data mappings via linear svm, J. Mach. Learn. Res. 11 (Aug. 2010) 1471–1490.

[42] Y. Goldberg, M. Elhadad, splitsvm: Fast, space-efficient, non-heuristic, polynomial kernel computation for nlp applications, 2008, pp. 237–240. .

[43] C. Xia, M. Zhang, J. Cao, A hybrid application of soft computing methods with wavelet svm and neural network to electric power load forecasting, J. Electr. Syst. Inf. Technol. 5 (3) (2018) 681–696.

[44] K. Chen, C. Li, B. Kuo, M. Wang, Applying automatic kernel parameter selection method to the full bandwidth rbf kernel function for hyperspectral image classification, in: 2014 IEEE Geoscience and Remote Sensing Symposium, 2014, pp. 3442–3445. .

[45] B. Kuo, H. Ho, C. Li, C. Hung, J. Taur, A kernel-based feature selection method for svm with rbf kernel for hyperspectral image classification, IEEE J. Selected Top Appl. Earth Observ. Remote Sensing 7 (Jan 2014) 317–326.

[46] O. Chapelle, P. Haffner, V. Vapnik, Support vector machines for histogram-based image classification, IEEE Trans. Neural Networks 10 (5) (1999) 1055–1064, cited By 967.

[47] B. Schölkopf, K.-K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, Comparing support vector machines with gaussian kernels to radial basis function classifiers, IEEE Trans. Signal Process. 45 (11) (1997) 2758–2765, cited By 796.

[48] J. Cheng, T. Liu, Y. Zhou, Y. Xiong, Road junction identification in high resolution urban sar images based on svm, Adv. Intell. Syst. Comput. 994 (2020) 597–606.

[49] Y. Xiao, H. Wang, W. Xu, Parameter selection of gaussian kernel for one-class svm, IEEE Trans. Cybern. 45 (May 2015) 941–953.

[50] A. Chaudhuri, K. De, Fuzzy support vector machine for bankruptcy prediction, Appl. Soft Comput. J. 11 (2) (2011) 2472–2486, cited By 85.

[51] Q. Xu, H. Zhou, Y. Wang, J. Huang, Fuzzy support vector machine for classification of eeg signals using wavelet-based features, Med. Eng. Phys. 31 (7) (2009) 858–865.

[52] N. Tran, D. Tran, S. Liu, L. Trinh, and T. Pham, Improving svm classification on imbalanced datasets for eeg-based person authentication, in: F. Martínez Álvarez, A. Troncoso Lora, J.A. Sáez Muñoz, H. Quintián, E. Corchado (Eds.), International Joint Conference: 12th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2019) and 10th International Conference on EUropean Transnational Education (ICEUTE 2019), (Cham), Springer International Publishing, 2020, pp. 57–66..

[53] V.J. Kadam, S.S. Yadav, S.M. Jadhav, Soft-margin svm incorporating feature selection using improved elitist ga for arrhythmia classification, in: A. Abraham, A.K. Cherukuri, P. Melin, and N. Gandhi, (Eds.), Intelligent Systems Design and Applications, (Cham), Springer International Publishing, 2020, pp. 965–976. .

[54] X. Dai, N. Wang, W. Wang, Application of machine learning in bgp anomaly detection, vol. 1176, 2019. .

[55] A. ur Rauf, A. Ghumman, S. Ahmad, H. Hashmi, Performance assessment of artificial neural networks and support vector regression models for stream flow predictions, Environ. Monitor. Assessment 190(12) (2018). .

[56] B.M. Henrique, V.A. Sobreiro, H. Kimura, Stock price prediction using support vector regression on daily and up to the minute prices, J. Finance Data Sci. 4 (3) (2018) 183–201.

[57] B. Sanjaa and E. Chuluun, Malware detection using linear svm, in: Ifost, vol. 2, 2013, pp. 136–138. .

[58] H.-T. Lin, C.-J. Lin, A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods, Neural Comput. 06 (2003).

[59] L. Grama, L. Tuns, C. Rusu, On the optimization of svm kernel parameters for improving audio classification accuracy, in: 2017 14th International Conference on Engineering of Modern Electric Systems (EMES), 2017, pp. 224–227.

[60] R. Courant, D. Hilbert, Methods of Mathematical Physics, vol. 1, Wiley-VCH, 1989.

[61] J. Xiong Dong, A. Krzyzak, C. Suen, Fast svm training algorithm with decomposition on very large data sets, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 27, 2005, pp. 603–618. .

[62] X. Li, J. Cervantes, W. Yu, A novel svm classification method for large data sets, in: 2010 IEEE International Conference on Granular Computing, 2010, pp. 297–302.

[63] K.P. Bennett, E.J. Bredensteiner, Geometry in learning, in: In Geometry at Work, 1997.

[64] Y.-J. Lee, S.-Y. Huang, Reduced support vector machines: a statistical theory, IEEE Trans. Neural Networks 18 (1) (2007) 1–13.

[65] F. Zhu, J. Yang, N. Ye, C. Gao, G. Li, T. Yin, Neighbors distribution property and sample reduction for support vector machines, Appl. Soft Comput. 16 (2014) 201–209.

[66] X. Li, J. Cervantes, W. Yu, Fast classification for large data sets via random selection clustering and support vector machines, Intelligent Data Anal. 16 (6) (2012) 897–914.

[67] B. Gartner, E. Welzl, A simple sampling lemma: analysis and applications in geometric optimization, Discrete Computat. Geometry 25 (4) (2001) 569–590.

[68] S. Canu, L. Bottou, S. Canu, Training invariant support vector machines using selective sampling, MIT Press, 2007.

[69] J.L. Balcázar, Y. Dai, J. Tanaka, O. Watanabe, Provably fast training algorithms for support vector machines, Theory Comput. Syst. 42 (4) (2008) 568–595.

[70] Y.-G. Liu, Q. Chen, R.-Z. Yu, Extract candidates of support vector from training set, in: Machine Learning and Cybernetics, 2003 International Conference on, vol. 5, nov. 2003, pp. 3199–3202. .

[71] A. Shigeo, I. Takuya, Fast training of support vector machines by extracting boundary data, in: ICANN '01: Proceedings of the International Conference on Artificial Neural Networks, (London, UK), Springer-Verlag, 2001, pp. 308–313..

[72] D. Wang, D. Yeung, E. Tsang, Weighted mahalanobis distance kernels for support vector machines, IEEE Trans. Neural Networks 18 (sept. 2007.) 1453–1462.

[73] D. Wang, L. Shi, Selecting valuable training samples for SVMs via data structure analysis, Neurocomputing 71 (13) (2008) 2772–2781.

[74] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, COLT '92, (New York, NY, USA), pp. 144–152, 1992, ACM..

[75] P.E. Hart, The condensed nearest neighbor rule, IEEE Trans. Inf. Theory 14 (1968) 515–516.

[76] G. Gates, The Reduced Nearest Neighbor Rule, IEEE Trans. Inf. Theory, vol. IT-18(3), 1972, pp. 431–433. .

[77] G.L. Ritter, H.B. Woodruff, S.R. Lowry, T.L. Isenhour, An algorithm for a selective nearest neighbor decision rule (corresp.), IEEE Trans. Inf. Theory 21 (6) (1975) 665–669.

[78] R. Wang, S. Kwong, Sample selection based on maximum entropy for support vector machines, in: Machine Learning and Cybernetics (ICMLC), 2010 International Conference on, vol. 3, 2010, pp. 1390–1395. .

[79] H. Shin, S. Cho, Neighborhood property-based pattern selection for support vector machines, Neural Comput. 19 (March 2007) 816–855.

[80] X. Jiantao, H. Mingyi, W. Yuying, F. Yan, A fast training algorithm for support vector machine via boundary sample selection, in: International Conference

on Neural Networks and Signal Processing, 2003. Proceedings of the 2003 IEEE, 2003.
[81] H. Yu, J. Yang, J. Han, Classifying large data sets using svms with hierarchical clusters, KDD (2003).
[82] M.A. Awad, L. Khan, F.B. Bastani, I.-L. Yen, An effective support vector machines (svms) performance using hierarchical clustering, in: 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, pp. 663–667. .
[83] J. Cervantes, X. Li, W. Yu, Support vector machine classification based on fuzzy clustering for large data sets, in: Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2006, pp. 572–582.
[84] R. Collobert, S. Bengio, Svmtorch: support vector machines for large-scale regression problems, J. Mach. Learn. Res. 1 (2001). .
[85] L. Shih, J.D.M. Rennie, Y.-H. Chang, D.R. Karger, Text bundling: Statistics based data-reduction, ICML (2003).
[86] V. Tresp, A bayesian committee machine, Neural Comput. 12 (Nov 2000) 2719–2741.
[87] M. Doumpos, An experimental comparison of some efficient approaches for training support vector machines, Oper. Res. Int. Journal 4 (2004) 45–56, https://doi.org/10.1007/BF02941095.
[88] N. List, S. Hans-Ulrich, A general convergence theorem for the decomposition method, in: COLT, 2004, pp. 363–377. .
[89] G. Wang, A survey on training algorithms for support vector machine classifiers, in Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management – Volume 01, vol. 1 of NCM '08, (Washington, DC, USA), pp. 123–128, IEEE Computer Society, sept. 2008. .
[90] C.W. Hsu, C.J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Networks (2002).
[91] J. Platt, Fast training of support vector machines using sequential minimal optimization, Adv. Kernel Methods: Support Vector Mach., 1998, pp. 185–208. .
[92] S.-W. Kim, B. Oommen, Enhancing prototype reduction schemes with recursion: a method applicable for "large data sets, IEEE Trans. Syst., Man Cybern., Part B (Cybernetics) 34 (jun 2004.) 1384–1397.
[93] G. Folino, C. Pizzuti, G. Spezzano, GP ensembles for large-scale data classification, IEEE Trans. Evol. Comput. 10 (Oct 2006) 604–616.
[94] G.-B. Huang, K. Mao, C.-K. Siew, D.-S. Huang, Fast modular network implementation for support vector machines, IEEE Trans. Neural Networks 16 (Nov 2005) 1651–1663.
[95] C. Chih-Chung, L. Chih-Jen, Libsvm: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–27.
[96] R.-E. Fan, P.-H. Chen, C.-J. Lin, Working set selection using second order information for training support vector machines, J. Mach. Learn. Res. 6 (December 2005) 1889–1918.
[97] T. Joachims, Making large-scale svm learning practical, Adv. Kernel Methods-Support Vector Learn. (1998) 169–184. .
[98] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (June 1999) 293–300.
[99] G. Fung, O.L. Mangasarian, Incremental support vector machine classification, in: 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 77–86. .
[100] Y. Jye Lee, O.L. Mangasarian, Rsvm: Reduced support vector machines, in: Data Mining Institute, Computer Sciences Department, University of Wisconsin, 2001, pp. 00–07. .
[101] H.P. Graf, E. Cosatto, L. Bottou, I. Durdanovic, V. Vapnik, Parallel support vector machines: the cascade svm, in: Advances in Neural Information Processing Systems, MIT Press, 2005, pp. 521–528.
[102] L. Bao-Liang, W. Kai-An, W. Yi-Min, Comparison of parallel and cascade methods for training support vector machines on large-scale problems, in: Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, vol. 5, 2004, pp. 3056–3061. .
[103] R. Collobert, Y. Bengio, S. Bengio, Scaling large learning problems with hard parallel mixtures, Int. J. Pattern Recogn. Artif. Intell.(IJPRAI) 17 (3) (2003) 349–365.
[104] S. Qiu, T. Lane, Parallel computation of rbf kernels for support vector classifiers, in: Proc. 5th SIAM International Conference on Data Mining (SDM05), 2005, pp. 334–345. .
[105] T. Serafini, L. Zanni, G. Zanghirati, Some improvements to a parallel decomposition technique for training support vector machines, in: B. Di Martino, D. Kranzlmller, J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface, vol. 3666 of Lecture Notes in Computer Science, 2005, pp. 9–17. .
[106] T. Eitrich, B. Lang, On the optimal working set size in serial and parallel support vector machine learning with the decomposition algorithm," in: AusDM '06: Proceedings of the fifth Australasian conference on Data mining and analystics, (Darlinghurst, Australia, Australia), Australian Computer Society, Inc., 2006, pp. 121–128..
[107] F. Poulet, Multi-way distributed svm algorithms, in: Parallel and Distributed computing for Machine Learning. In: conjunction with the 14th European Conference on Machine Learning (ECML'03) and 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03), (Cavtat-Dubrovnik, Croatia), 2003. .
[108] G. Zanghirati, L. Zanni, A parallel solver for large quadratic programs in training support vector machines, Parallel Comput. 29 (4) (2003) 535–551.
[109] Y. Ming-Hsuan, A. Narendra, A geometric approach to train support vector machines, IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1 (2000) 1430.
[110] Crisp, Burges, A geometric interpretation of υ-svm classifiers, NIPS 12 (2000) 244–250.
[111] M.E. Mavroforakis, M. Sdralis, S. Theodoridis, A geometric nearest point algorithm for the efficient solution of the svm classification task, IEEE Trans. Neural Networks 18 (sept. 2007.) 1545–1549.
[112] Z. Liu, J.G. Liu, C. Pan, G. Wang, A novel geometric approach to binary classification based on scaled convex hulls, IEEE Trans. Neural Networks 20 (July 2009) 1215–1220.
[113] E. Osuna, O.D. Castro, Convex Hull in Feature Space for Support Vector Machines, in: Proceedings of the 8th Ibero-American Conference on AI: Advances in Artificial Intelligence, IBERAMIA 2002, (London, UK, UK), Springer-Verlag, 2002, pp. 411–419. .
[114] X. Peng, Efficient geometric algorithms for support vector machine classifier, in Natural Computation (ICNC), 2010 Sixth International Conference on, vol. 2, 2010, pp. 875–879. .
[115] Z.-Q. Zeng, H.-R. Xu, Y.-Q. Xie, and J. Gao, A geometric approach to train svm on very large data sets, in: Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on, vol. 1, 2008, pp. 991–996. .
[116] D. DeCoste, K. Wagstaff, Alpha seeding for support vector machines, in: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '00, ACM, New York, NY, USA, 2000, pp. 345–349.
[117] D. Feng, W. Shi, H. Guo, L. Chen, A new alpha seeding method for support vector machine training, in: L. Wang, K. Chen, Y. Ong (Eds.), Advances in Natural Computation, vol. 3610 of Lecture Notes in Computer Science, Springer, Berlin/, Heidelberg, 2005, p. 418.
[118] J.C.R. Santos, J.S. Aguilar-Ruiz, M. Toro, Finding representative patterns with ordered projections, Pattern Recogn. 36 (4) (2003) 1009–1018.
[119] Y. Caises, A. González, E. Leyva, R. Pérez, Combining instance selection methods based on data characterization: an approach to increase their effectiveness, Inf. Sci. 181 (20) (2011) 4780–4798.
[120] M. Arun Kumar, M. Gopal, A hybrid svm based decision tree, Pattern Recognition, Dec. 43 (2010) 3977–3987.
[121] A. Almas, M. Farquad, N. Avala, J. Sultana, Enhancing the performance of decision tree: A research study of dealing with unbalanced data, in: Digital Information Management (ICDIM), 2012 Seventh International Conference on, 2012, pp. 7–10. .
[122] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Networks 13 (March 2002) 415–425.
[123] J.C. Platt, N. Cristianini, J. Shawe-Taylor, Large margin dags for multiclass classification, NIPS (1999).
[124] S. Knerr, L. Personnaz, G. Dreyfus, Single-layer learning revisited: a stepwise procedure for building and training a neural network, in: Neurocomputing, 1990. .
[125] H.G.K. Ulrich, Pairwise classification and support vector machines, Adv. Kernel Methods (1999).
[126] A.A. Aburomman, M.B. Ibne Reaz, A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems, Inf. Sci. (2017).
[127] E.J. Bredensteiner, K.P. Bennett, Multicategory classification by support vector machines, Computat. Optim., 1999. .
[128] L. Tang, Y. Tian, P.M. Pardalos, A novel perspective on multiclass classification: regular simplex support vector machine, Inf. Sci. 480 (2019) 324–338.
[129] Y.-H. Shao, W.-J. Chen, W.-B. Huang, Z.-M. Yang, N.-Y. Deng, The best separating decision tree twin support vector machine for multi-class classification, Proc. Comput. Sci. 17 (2013) 1032–1038.
[130] L. Zhang, W.-D. Zhou, T.-T. Su, L.-C. Jiao, Decision tree support vector machine, Int. J. Artif. Intell. Tools (2007).
[131] B. Santosa, Multiclass classification with cross entropy-support vector machines, Proc. Comput. Sci. (2015).
[132] F. Provost, T. Fawcett, Mach. Learn. 42 (3) (2001) 203–231.
[133] J. Cervantes, X. Li, W. Yu, "Splice site detection in DNA sequences using a fast classification algorithm," in 2009 IEEE International Conference on Systems, in: 2009 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2009.
[134] G. Dror, R. Sorek, R. Shamir, Accurate identification of alternatively spliced exons using support vector machine, Bioinformatics 21 (Nov 2004) 897–901.
[135] K. Veropoulos, C. Campbell, N. Cristianini, Controlling the sensitivity of support vector machines, IJCAI 1999 (1999).
[136] I. Kononenko, Machine learning for medical diagnosis: history, state of the art and perspective, Artif. Intell. Med. 23 (Aug 2001) 89–109.
[137] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surveys 34 (Mar 2002) 1–47.
[138] S. Tan, Neighbor-weighted k-nearest neighbor for unbalanced text corpus, Expert Syst. Appl. 28 (May 2005) 667–671.
[139] R.M. Kebeasy, A.I. Hussein, S.A. Dahy, Discrimination between natural earthquakes and nuclear explosions using the aswan seismic network, Ann. Geophys. 41(1998). .
[140] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[141] S. Köknar-Tezel, L.J. Latecki, Improving SVM classification on imbalanced data sets in distance spaces, in: 2009 Ninth IEEE International Conference on Data Mining, IEEE, Dec 2009. .

[142] Z.-Q. Zeng, J. Gao, Improving SVM classification with imbalance data set, in: Neural Information Processing, Springer, Berlin, Heidelberg, 2009, pp. 389–398.

[143] S. Zou, Y. Huang, Y. Wang, J. Wang, C. Zhou, SVM learning from imbalanced data by GA sampling for protein domain prediction, in: 2008 The 9th International Conference for Young Computer Scientists, IEEE, 2008. .

[144] M. Farquad, I. Bose, Preprocessing unbalanced data using support vector machine, Decis. Support Syst. 53 (Apr 2012) 226–233.

[145] Y. Liu, X. Yu, J.X. Huang, A. An, Combining integrated sampling with SVM ensembles for learning from imbalanced datasets, Inf. Process. Manage. 47 (Jul 2011) 617–631.

[146] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, in: Machine Learning: ECML 2004, Springer, Berlin Heidelberg, 2004, pp. 39–50. .

[147] X. Yang, Q. Song, A. Cao, Weighted support vector machine for data classification, in: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, IEEE, 2005.

[148] Y.-M. Huang, S.-X. Du, Weighted support vector machine for classification with uneven training class sizes, in: 2005 International Conference on Machine Learning and Cybernetics, IEEE, 2005. .

[149] J.M. Choi, A Selective Sampling Method for Imbalanced Data Learning on Support Vector Machines (Ph.D. thesis), Ames, IA, USA, 2010. AAI3413682..

[150] S. Garcıa, J. Derrac, I. Triguero, C.J. Carmona, F. Herrera, Evolutionary-based selection of generalized instances for imbalanced classification, Knowl.-Based Syst. 25 (Feb 2012) 3–12.

[151] B.X. Wang, N. Japkowicz, Boosting support vector machines for imbalanced data sets, Knowl. Inf. Syst. 25 (Mar 2009) 1–20.

[152] N.V. Chawla, A. Lazarevic, L.O. Hall, K.W. Bowyer, SMOTEBoost: Improving prediction of the minority class in boosting, in: Knowledge Discovery in Databases: PKDD 2003, Springer, Berlin Heidelberg, 2003, pp. 107–119. .

[153] H.M. Nguyen, E.W. Cooper, K. Kamei, Borderline over-sampling for imbalanced data classification, Int. J. Knowl. Eng. Soft Data Paradigms 3 (1) (2011) 4.

[154] S. Hu, Y. Liang, L. Ma, Y. He, MSMOTE: Improving classification performance when training data is imbalanced, in: 2009 Second International Workshop on Computer Science and Engineering, IEEE, 2009. .

[155] H. Guo, H.L. Viktor, Learning from imbalanced data sets with boosting and data generation, ACM SIGKDD Explorations Newsletter 6 (jun 2004.) 30.

[156] G. Wu, E.Y. Chang, Class-boundary alignment for imbalanced dataset learning, 2003. .

[157] G. Wu, E. Chang, KBA: kernel boundary alignment considering imbalanced data distribution, IEEE Trans. Knowl. Data Eng. 17 (2005) 786–795.

[158] D.A. Cieslak, N.V. Chawla, Learning decision trees for unbalanced data, in: Machine Learning and Knowledge Discovery in Databases, Springer, Berlin Heidelberg, pp. 241–256. .

[159] A. Fernández, M.J. del Jesus, F. Herrera, Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets, Int. J. Approximate Reasoning 50 (Mar 2009) 561–577.

[160] R. Batuwita, V. Palade, FSVM-CIL: fuzzy support vector machines for class imbalance learning, IEEE Trans. Fuzzy Syst. 18 (2010) 558–571.

[161] R. Batuwita, V. Palade, Efficient resampling methods for training support vector machines with imbalanced datasets, in: The 2010 International Joint Conference on Neural Networks (IJCNN), IEEE, 2010. .

[162] N. Rout, D. Mishra, M.K. Mallick, Handling imbalanced data: a survey, in: Advances in Intelligent Systems and Computing, Springer Singapore, 2017, pp. 431–443. .

[163] F. Melgani, Y. Bazi, Classification of electrocardiogram signals with support vector machines and particle swarm optimization, IEEE Trans. Inf Technol. Biomed. 12 (2008) 667–677.

[164] R. En Fan, P. Hsuen Chen, T. Joachims, Working set selection using second order information for training svm, J. Mach. Learn. Res., p. 2005. .

[165] C.-C. Chang, C.-J. Lin, LIBSVM, ACM Trans. Intell. Syst. Technol. 2 (Apr 2011) 1–27.

[166] C.-C. Chang, C.-J. Lin, Training v-support vector classifiers: theory and algorithms, Neural Comput. 13 (sep 2001.) 2119–2147.

[167] P.-H. Chen, R.-E. Fan, C.-J. Lin, A study on SMO-type decomposition methods for support vector machines, IEEE Trans. Neural Networks 17 (2006) 893–908.

[168] N. List, H.U. Simon, A general convergence theorem for the decomposition method, in: Learning Theory, Springer, Berlin, Heidelberg, 2004, pp. 363–377.

[169] D.G. Luenberger, Y. Ye, Linear and Nonlinear Programming (International Series in Operations Research & Management Science Book 116), Springer, 2008.

[170] R. Fletcher, Practical Methods of Optimization, John Wiley & Sons Ltd, 2000.

[171] C.-W. Hsu, C.-J. Lin, Mach. Learn. 46 (1/3) (2002) 291–314.

[172] E. Osuna, R. Freund, F. Girosi, An improved training algorithm for support vector machines, in: Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop, IEEE. .

[173] R. Topor, K. Salem, A. Gupta, K. Goda, J. Gehrke, N. Palmer, M. Sharaf, A. Labrinidis, J.F. Roddick, A. Fuxman, R.J. Miller, W.-C. Tan, A. Kementsietsidis, P. Bonnet, D. Shasha, J.F. Roddick, A. Gupta, R. Peikert, B. Ludäscher, S. Bowers, T. McPhillips, H. Naumann, K. Voruganti, J. Domingo-Ferrer, B. Carterette, P.G. Ipeirotis, M. Arenas, Y. Manolopoulos, Y. Theodoridis, V.J. Tsotras, B. Carminati, J. Jurjens, E.B. Fernandez, M. Kantarcıoglu, J. Vaidya, I. Ray, A. Vakali, C. Sirangelo, E. Pitoura, H. Gupta, S. Chaudhuri, G. Weikum, U. Leser, D. W. Embley, F. Giunchiglia, P. Shvaiko, M. Yatskevich, E.Y. Chang, C. Parent, S. Spaccapietra, E. Zimányi, G. Anadiotis, S. Kotoulas, R. Siebes, G. Antoniou, D. Plexousakis, J. Bailey, F. Bry, T. Furche, S. Schaffert, D. Martin, G. Speegle, K. Ramamritham, P.K. Chrysanthis, K.-U. Sattler, S. Bressan, S. Abiteboul, D. Suciu, G. Dobbie, T.W. Ling, D. Suciu, S. Basu, R. Govindan, M.H. Böhlen, C.S. Jensen, J. Wang, K. Vidyasankar, A. Chan, S. Mankovski, S. Elnikety, P. Valduriez, P. Valduriez, P. Valduriez, Y. Velegrakis, M.A. Nascimento, M. Huggett, A.U. Frank, Y. Zhang, G. Xu, C.S. Jensen, R.T. Snodgrass, A. Fekete, M. Herzog, K. Morfonios, Y. Ioannidis, E. Wohlstadter, M. Matera, F. Schwagereit, S. Staab, K. Fraser, J. Zhou, M.F. Mokbel, W.G. Aref, M.F. Mokbel, W.G. Aref, M. M. Moro, V.J. Tsotras, M. Schneider, P. Kalnis, G. Ghinita, M.F. Goodchild, S. Shekhar, J. Kang, V. Gandhi, M. Schneider, Y. Manolopoulos, Y. Theodoridis, V. J. Tsotras, N. Mamoulis, B. George, S. Shekhar, M. Scholl, A. Voisard, N. Mamoulis, R.H. Güting, Y. Tao, D. Papadias, P. Revesz, G. Kollios, E. Frentzos, Y. Theodoridis, A.N. Papadopoulos, B. Thalheim, J. Pehcevski, B. Piwowarski, S. Theodoridis, K. Koutroumbas, N. Palmer, G. Karabatis, D. Chamberlin, P.A. Bernstein, M. Böhlen, J. Gamper, C.S. Jensen, R.T. Snodgrass, P. Li, K. Subieta, S. Harizopoulos, E. Zhang, Y. Zhang, T. Johnson, K. Morfonios, Y. Ioannidis, H.-A. Jacobsen, A. Gupta, S.E. Fienberg, J. Jin, R. Sion, C.D. Paice, N. Hardavellas, I. Pandis, E. Rasmussen, K. Voruganti, K. Goda, H. Yoshida, K. Voruganti, K. Voruganti, H. Yoshida, H. Yoshida, G. Graefe, K. Goda, H. Yoshida, B. Reiner, K. Hahn, K. Goda, K. Wada, K. Voruganti, H. Yoshida, K. Voruganti, H. Yoshida, T. Risch, J. Han, B. Ding, L. Golab, M. Stonebraker, B. Lahiri, S. Tirthapura, E. Vee, Y. Ahmad, U. Çetintemel, M. Cherniack, S. Zdonik, A. Fekete, M.P. Consens, M. Lalmas, K.-U. Sattler, M. Lalmas, R. Baeza-Yates, D. Hiemstra, R. Baeza-Yates, H.-A. Jacobsen, P. Krögerand, A. Zimek, N. Craswell, C.K.-S. Leung, M. Crochemore, T. Lecroq, A. Shoshani, J. Lin, H. Yu, D. Lomet, H. Hinterberger, N. Li, P.B. Gibbons, J. Domingo-Ferrer, M. Kacimi, T. Neumann, Support vector machine, in Encyclopedia of Database Systems, Springer, US, 2009, pp. 2890–2892.

[174] A. Shilton, M. Palaniswami, D. Ralph, A. Tsoi, Incremental training of support vector machines, IEEE Trans. Neural Networks 16 (Jan 2005) 114–131.

[175] S. Venkateshan, A. Patel, K. Varghese, Hybrid working set algorithm for SVM learning with a kernel coprocessor on FPGA, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, 2015, pp. 2221–2232. .

[176] R. Collobert, S. Bengio, Svmtorch, 2001,http://bengio.abracadoudou.com/SVMTorch.html. .

[177] S. Shalev-Shwartz, Pegasos, 2007,https://www.cs.huji.ac.il/shais/code/index.html. .

[178] C.-C. Chang, C.-J. Lin, Libsvm, 2018,https://www.csie.ntu.edu.tw/cjlin/libsvm/. .

[179] T. Joachims, Svmlight, 2008,http://svmlight.joachims.org/. .

[180] C. Diehl, Isvm, 2006.https://github.com/diehl/Incremental-SVM-Learning-in-MATLAB. .

[181] L.H. Lee, C.H. Wan, R. Rajkumar, D. Isa, An enhanced support vector machine classification framework by using euclidean distance function for text document categorization, Appl. Intell. 37 (Aug 2011) 80–99.

[182] J. He, Appl. Intell. 18 (3) (2003) 311–322.

[183] E. Leopold, J. Kindermann, Mach. Learn. 46 (1/3) (2002) 423–444.

[184] S. Hoi, R. Jin, M. Lyu, Batch mode active learning with applications to text categorization and image retrieval, IEEE Trans. Knowl. Data Eng. 21 (sep 2009.) 1233–1248.

[185] M.A. Kumar, M. Gopal, Text categorization using fuzzy proximal SVM and distributional clustering of words, in Advances in Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg, 2009, pp. 52–61.

[186] G. Dias, T. Honkela, Term weighting in short documents for document categorization, keyword extraction and query expansion, 2012. .

[187] K. Li, J. Xie, X. Sun, Y. Ma, H. Bai, Multi-class text categorization based on LDA and SVM, Proc. Eng. 15 (2011) 1963–1967.

[188] T. Peng, W. Zuo, F. He, SVM based adaptive learning method for text classification from positive and unlabeled documents, Knowl. Inf. Syst. 16 (2007) 281–301.

[189] R.H. Pinheiro, G.D. Cavalcanti, I.R. Tsang, Combining binary classifiers in different dichotomy spaces for text categorization, Appl. Soft Comput. 76 (Mar 2019) 564–574.

[190] B. Al-Salemi, M. Ayob, G. Kendall, S.A.M. Noah, Multi-label arabic text categorization: a benchmark and baseline comparison of multi-label learning algorithms, Inf. Processing Manage. 56 (Jan 2019) 212–227.

[191] J. Verma, M. Nath, P. Tripathi, K.K. Saini, Analysis and identification of kidney stone using kth nearest neighbour (knn) and support vector machine (svm) classification techniques, Pattern Recogn. Image Anal. 27 (Jul 2017) 574–580.

[192] Z. Qiao, X. Kewen, W. Panpan, H. Wang, Lung nodule classification using curvelet transform, lda algorithm and bat-svm algorithm, Pattern Recogn. Image Anal. 27 (Oct 2017) 855–862.

[193] S. Afifi, H. GholamHosseini, R. Sinha, Dynamic hardware system for cascade svm classification of melanoma, Neural Comput. Appl., 2018. .

[194] P. Kaur, H.S. Pannu, A.K. Malhi, Plant disease recognition using fractional-order zernike moments and svm classifier, Neural Comput. Appl., 2019. .

[195] X. Zhang, M.H. Mahoor, S.M. Mavadati, Facial expression recognition using $l_p$lp-norm mkl multiclass-svm, Mach. Vis. Appl. 26 (May 2015) 467–483.

[196] C.-C. Hsieh, D.-H. Liou, Novel haar features for real-time hand gesture recognition using svm, J. Real-Time Image Proc. 10 (Jun 2015) 357–370.

[197] M.A. Berbar, Three robust features extraction approaches for facial gender classification, Visual Comput. 30 (Jan 2014) 19–31.

[198] Y. Tarabalka, M. Fauvel, J. Chanussot, J.A. Benediktsson, Svm- and mrf-based method for accurate classification of hyperspectral images, IEEE Geosci. Remote Sens. Lett. 7 (2010) 736–740. .

[199] X. Liu, J. Tang, Mass classification in mammograms using selected geometry and texture features, and a new svm-based feature selection method, IEEE Syst. J. 8 (2014) 910–920.

[200] J. Wu, Efficient hik svm learning for image classification, IEEE Trans. Image Process. 21 (Oct 2012) 4442–4453.

[201] Z. Yang, H. Xu, J. Deng, C.C. Loy, W.C. Lau, Robust and fast decoding of high-capacity color qr codes for mobile applications, IEEE Trans. Image Process. 27 (Dec 2018) 6093–6108.

[202] J.C. Ang, H. Haron, H.N.A. Hamed, Semi-supervised svm-based feature selection for cancer classification using microarray gene expression data, in: M. Ali, Y.S. Kwon, C.-H. Lee, J. Kim, and Y. Kim, (Eds.), Current Approaches in Applied Artificial Intelligence, (Cham), Springer International Publishing, 2015, pp. 468–477..

[203] A. Masood, A. Al-Jumaily, K. Anam, Texture analysis based automated decision support system for classification of skin cancer using sa-svm," in: C.K. Loo, K.S. Yap, K.W. Wong, A. Teoh, and K. Huang (Eds.) Neural Information Processing, (Cham), Springer International Publishing, 2014, pp. 101–109..

[204] L. Zhang, W. Zhou, B. Wang, Z. Zhang, F. Li, Applying 1-norm svm with squared loss to gene selection for cancer classification, Appl. Intell. 48 (Jul 2018) 1878–1890.

[205] S.S. Tirumala, A. Narayanan, Classification and diagnostic prediction of prostate cancer using gene expression and artificial neural networks, Neural Comput. Appl., 2018. .

[206] J. Li, Z. Weng, H. Xu, Z. Zhang, H. Miao, W. Chen, Z. Liu, X. Zhang, M. Wang, X. Xu, Q. Ye, Support vector machines (svm) classification of prostate cancer gleason score in central gland using multiparametric magnetic resonance images: a cross-validated study, Eur. J. Radiol. 98 (2018) 61–67.

[207] N. Jafarpisheh, M. Teshnehlab, Cancers classification based on deep neural networks and emotional learning approach, IET Syst. Biol., 12 (2018) 258–263(5). .

[208] L.H. Vogado, R.M. Veras, F.H. Araujo, R.R. Silva, K.R. Aires, Leukemia diagnosis in blood slides using transfer learning in cnns and svm for classification, Eng. Appl. Artif. Intell. 72 (2018) 415–422.

[209] C. Mazo, E. Alegre, M. Trujillo, Classification of cardiovascular tissues using lbp based descriptors and a cascade svm, Comput. Methods Programs Biomed. 147 (2017) 1–10.

[210] M. Moradi, P. Abolmaesumi*, D.R. Siemens, E.E. Sauerbrei, A.H. Boag, P. Mousavi, Augmenting detection of prostate cancer in transrectal ultrasound images using svm and rf time series, IEEE Trans. Biomed. Eng., 56 (2009) 2214–2224..

[211] P.M. Dinesh, R.S. Sabenian, Comparative analysis of zoning approaches for recognition of indo aryan language using svm classifier, Cluster Computing, 2017. .

[212] N.A. Jebril, H.R. Al-Zoubi, Q. Abu Al-Haija, Recognition of handwritten arabic characters using histograms of oriented gradient (hog), Pattern Recogn. Image Anal. 28 (Apr 2018) 321–345.

[213] G.A. Montazer, M.A. Soltanshahi, D. Giveki, Farsi/arabic handwritten digit recognition using quantum neural networks and bag of visual words method, Opt. Memory Neural Networks 26 (Apr 2017) 117–128.

[214] Y. Gao, L. Jin, C. He, G. Zhou, Handwriting character recognition as a service: a new handwriting recognition system based on cloud computing, in: 2011 International Conference on Document Analysis and Recognition, 2011, pp. 885–889.

[215] D. Bertolini, L.S. Oliveira, R. Sabourin, Multi-script writer identification using dissimilarity, in: in 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 3025–3030.

[216] P. Kumar, N. Sharma, A. Rana, Article: Handwritten character recognition using different kernel based svm classifier and mlp neural network (a comparison), Int. J. Comput. Appl. 53 (2012) 25–31. Full text available.

[217] V. Christlein, D. Bernecker, F. Honig, A. Maier, E. Angelopoulou, Writer identification using gmm supervectors and exemplar-svms, Pattern Recogn. 63 (2017) 258–267.

[218] A. Chahi, I.E. Khadiri, Y.E. Merabet, Y. Ruichek, R. Touahni, Block wise local binary count for off-line text-independent writer identification, Expert Systems with Applications, vol. 93, 2018, pp. 1–14.

[219] H.-M. Je, D. Kim, S. Yang Bang, Human face detection in digital video using svmensemble, Neural Process. Lett. 17 (2003) 239–252. .

[220] Q.-Q. Tao, S. Zhan, X.-H. Li, T. Kurihara, Robust face detection using local cnn and svm based on kernel combination, Neurocomputing 211 (2016) 98–105. SI: Recent Advances in SVM. .

[221] S. Bashbaghi, E. Granger, R. Sabourin, G.-A. Bilodeau, Dynamic ensembles of exemplar-svms for still-to-video face recognition, Pattern Recogn. 69 (2017) 61–81.

[222] M. Li, X. Yu, K.H. Ryu, S. Lee, and N. Theera-Umpon, Face recognition technology development with gabor, pca and svm methodology under illumination normalization condition, Cluster Comput., 2017. .

[223] L. Shmaglit, V. Khryashchev, Gender classification of human face images based on adaptive features and support vector machines, Optical Memory Neural Networks 22 (Oct 2013) 228–235.

[224] J. Zhang, X. Zhang, S. Ha, in: 2008 Fourth International Conference on Natural Computation, 2008, pp. 29–33.

[225] S. Kumar, A. Kar, M. Chandra, Svm based adaptive median filter design for face detection in noisy images, in: 2014 International Conference on Signal Processing and Integrated Networks (SPIN), 2014, pp. 695–698.

[226] D. Wen, H. Han, A.K. Jain, Face spoof detection with image distortion analysis, IEEE Trans. Inf. Forensics Secur. 10 (April 2015) 746–761.

[227] C.A. Waring, X. Liu, Face detection using spectral histograms and svms, IEEE Trans. Syst., Man, Cybern. Part B (Cybernetics) 35 (June 2005) 467–476.

[228] Z. Chen, Y. Wang, H. Liu, Unobtrusive sensor-based occupancy facing direction detection and tracking using advanced machine learning algorithms, IEEE Sens. J. 18 (Aug 2018) 6360–6368.

[229] M.S. Rahman, M.K. Rahman, M. Kaykobad, M.S. Rahman, isGPT: an optimized model to identify sub-golgi protein types using SVM and random forest based feature selection, Artif. Intell. Med. 84 (Jan 2018) 90–100.

[230] S.M. Krishnan, Using chou's general PseAAC to analyze the evolutionary relationship of receptor associated proteins (RAP) with various folding patterns of protein domains, J. Theor. Biol. 445 (May 2018) 62–74.

[231] S. Mei, SVM ensemble based transfer learning for large-scale membrane proteins discrimination, J. Theor. Biol. 340 (Jan 2014) 105–110.

[232] M. Tahir, A. Khan, Protein subcellular localization of fluorescence microscopy images: employing new statistical and texton based image features and SVM based ensemble classification, Inf. Sci. 345 (2016) 65–80.

[233] M.R. Bakhtiarizadeh, M. Moradi-Shahrbabak, M. Ebrahimi, E. Ebrahimie, Neural network and SVM classifiers accurately predict lipid binding proteins, irrespective of sequence homology, J. Theor. Biol. 356 (2014) 213–222.

[234] M. Routray, S. Vipsita, Remote homology detection using physicochemical parameters and physicochemical properties, in: 2017 International Conference on Information Technology (ICIT), IEEE, 2017. .

[235] P. Lovato, M. Cristani, M. Bicego, Soft ngram representation and modeling for protein remote homology detection, IEEE/ACM Trans. Comput. Biol. Bioinf. 14 (Nov 2017) 1482–1488.

[236] O. Bedoya, I. Tischer, Reducing dimensionality in remote homology detection using predicted contact maps, Comput. Biol. Med. 59 (Apr 2015) 64–72.

[237] H.M. Muda, P. Saad, R.M. Othman, Remote protein homology detection and fold recognition using two-layer support vector machine classifiers, Comput. Biol. Med. 41 (2011) 687–699. .

[238] Z. Bao, Y. Sun, Support vector machine-based multi-model predictive control, J. Control Theory Appl. 6 (Aug 2008) 305–310.

[239] X. Song, K. Cao, S. Gao, C. Chen, J. Huang, The research of the AUV navigation control system based on the LS-SVM, in: 2017 IEEE International Conference on Unmanned Systems (ICUS), IEEE, 2017. .

[240] A. Chakrabarty, G.T. Buzzard, S.H. Zak, Output-tracking quantized explicit nonlinear model predictive control using multiclass support vector machines, IEEE Trans. Industr. Electron. 64 (May 2017) 4130–4138.

[241] F. Chu, B. Dai, W. Dai, R. Jia, X. Ma, F. Wang, Rapid modeling method for performance prediction of centrifugal compressor based on model migration and SVM, IEEE Access 5 (2017) 21488–21496.

[242] L. Zhang, X. Zhang, SVM-based techniques for predicting cross-functional team performance: using team trust as a predictor, IEEE Trans. Eng. Manage. 62 (Feb 2015) 114–121.

[243] L.D. Jalili, A. Morales, J. Cervantes, J.S. Ruiz-Castilla, "Improving the performance of leaves identification by features selection with genetic algorithms, in: Communications in Computer and Information Science, Springer International Publishing, 2016, pp. 103–114.

[244] J. Cervantes, F. García-Lamont, L. Rodríguez-Mazahua, A.L. Chau, J.S.R. Castilla, A.A. Trueba, F. Garcia-Lamont, L. Rodriguez, A. López, J.S.R. Castilla, A.A. Trueba, PSO-based method for SVM classification on skewed data sets, Neurocomputing 228 (2017) 187–197.

[245] J. Cervantes, F.G. Lamont, L.R. Mazahua, A.Z. Hidalgo, J.S.R. Castilla, Complex identification of plants from leaves, in Intelligent Computing Methodologies, Springer International Publishing, 2018, pp. 376–387.

[246] M. Sanderson, P. Clough, "Plantclef."https://www.imageclef.org/PlantCLEF2019, 2004. .

[247] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, O. Caelen, Sequence classification for credit-card fraud detection, Expert Syst. Appl. 100 (2018) 234–245.

[248] N. Carneiro, G. Figueira, M. Costa, A data mining based system for credit-card fraud detection in e-tail, Decis. Support Syst. 95 (Mar 2017) 91–101.

[249] M. Zareapoor, P. Shamsolmoali, Application of credit card fraud detection: Based on bagging ensemble classifier, Procedia Comput. Sci. 48 (2015) 679–685.

[250] S. Bhattacharyya, S. Jha, K. Tharakunnel, J.C. Westland, Data mining for credit card fraud: a comparative study, Decis. Support Syst. 50 (Feb 2011) 602–613.

[251] A. Hekler, J.S. Utikal, A.H. Enk, C. Berking, J. Klode, D. Schadendorf, P. Jansen, C. Franklin, T. Holland-Letz, D. Krahl, C. von Kalle, S. Fröhling, T.J. Brinker, Pathologist-level classification of histopathological melanoma images with deep neural networks, Eur. J. Cancer 115 (Jul 2019) 79–83.

[252] S.H. Kassani, P.H. Kassani, A comparative study of deep learning architectures on melanoma detection, Tissue Cell 58 (2019) 76–83.

[253] S. Afifi, H. GholamHosseini, R. Sinha, A system on chip for melanoma detection using FPGA-based SVM classifier, Microprocess. Microsyst. 65 (Mar 2019) 57–68.

[254] X. He, G. Mourot, D. Maquin, J. Ragot, P. Beauseroy, A. Smolarz, E. Grall-Maës, Multi-task learning with one-class SVM, Neurocomputing 133 (2014) 416–426. .

[255] Y. Ji, S. Sun, Multitask multiclass support vector machines: Model and experiments, Pattern Recogn. 46 (3) (2013) 914–924.

[256] A. López Chau, X. Li, W. Yu, Convex and concave hulls for classification with support vector machine, Neurocomputing 122 (2013) 198–209. .

[257] A.L. Chau, X. Li, W. Yu, J. Cervantes, Support vector candidates pre selection strategy based on non convex hulls, in: Program and Abstract Book - 2010 7th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE 2010, 2010, pp. 345–350. .

[258] A. Reeberg de Mello, M.R. Stemmer, F.G. Oliveira Barbosa, Support vector candidates selection via Delaunay graph and convex-hull for large and high-dimensional datasets, Pattern Recogn. Lett. (2018).

[259] L. Guo, S. Boukir, Fast data selection for SVM training using ensemble margin, Pattern Recogn. Lett. 51 (2015) 112–119.

[260] C. Liu, W. Wang, M. Wang, F. Lv, M. Konan, An efficient instance selection algorithm to reconstruct training set for support vector machine, Knowl.-Based Syst. 116 (2017) 58–73.

[261] A. López Chau, L. López-Garcia, J. Cervantes, X. Li, W. Yu, Data Selection Using Decision Tree for SVM Classification, in: IEEE 24th International Conference on Tools with Artificial Intelligence, ICTAI 2012, Athens, Greece, November 7–9, 2012, 2012, pp. 742–749. .

[262] P. Arumugam, P. Jose, Efficient Decision Tree Based Data Selection and Support Vector Machine Classification, Materials Today: Proceedings 5(1), Part 1 (2018) 1679–1685. .

[263] G. Taskin Kaya, O.K. Ersoy, M.E. Kamasak, Support vector selection and adaptation for remote sensing classification, IEEE Trans. Geosci. Remote Sens. (2011).

[264] G.T. Kaya, O.K. Ersoy, M.E. Kamasak, Support vector selection and adaptation for classification of earthquake images, in: 2009 IEEE International Geoscience and Remote Sensing Symposium, vol. 2, Jul 2009, pp. II–851–II–854. .

[265] X. Li, J. Cervantes, W. Yu, Fast classification for large data sets via random selection clustering and Support Vector Machines, Intell. Data Anal. (2012).

[266] S. Ougiaroglou, K.I. Diamantaras, G. Evangelidis, Exploring the effect of data reduction on Neural Network and Support Vector Machine classification, Neurocomputing 280 (2018) 101–110.

[267] X.-J. Shen, L. Mu, Z. Li, H.-X. Wu, J.-P. Gou, X. Chen, Large-scale support vector machine classification with redundant data reduction, Neurocomputing 172 (2016) 189–197. .

[268] J.P. Yeh, C.M. Chiang, Reducing the Solution of Support Vector Machines Using Simulated Annealing Algorithm, in: 2017 International Conference on Control, Artificial Intelligence, Robotics Optimization (ICCAIRO), 2017, pp. 105–108.

[269] J. Nalepa, M. Kawulok, Adaptive memetic algorithm enhanced with data geometry analysis to select training data for SVMs, Neurocomputing 185 (2016) 113–132.

[270] S. Shalev-Shwartz, Y. Singer, N. Srebro, A. Cotter, Pegasos: primal estimated sub-gradient solver for SVM, Math. Program. (2011).

[271] Z. Wang, Y.-H. Shao, L. Bai, C.-N. Li, L.-M. Liu, N.-Y. Deng, Insensitive stochastic gradient twin support vector machines for large scale problems, Inf. Sci. 462 (2018) 114–131.

[272] R.D. Morales, Á.N. Vázquez, Improving the efficiency of IRWLS SVMs using parallel Cholesky factorization, Pattern Recogn. Lett. 84 (2016) 91–98.

[273] M.D. de Lima, N.L. Costa, R. Barbosa, Improvements on least squares twin multi-class classification support vector machine Neurocomputing (2018). .

[274] M.E. Mavroforakis, S. Theodoridis, A geometric approach to support vector machine (SVM) classification, IEEE Trans. Neural Networks 17 (3) (2006) 671–682.

[275] J. Liu, R. Danait, Instance Selection in the Projected High Dimensional Feature Space for SVM, in: 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), 2018, pp. 575–579.

[276] Z.-H. You, J.-Z. Yu, L. Zhu, S. Li, Z.-K. Wen, A MapReduce based parallel SVM for large-scale predicting protein–protein interactions, Neurocomputing 145 (2014) 37–43.

[277] A. Bifet, R. Gavaldà, G. Holmes, B. Pfahringer, Machine Learning for Data Streams with Practical Examples in MOA, MIT Press, Cambridge, MA, 2018.

[278] J. Zheng, F. Shen, H. Fan, J. Zhao, An online incremental learning support vector machine for large-scale data, Neural Comput. Appl. (2013).

[279] X. Wang, Y. Xing, An online support vector machine for the open-ended environment, Expert Syst. Appl. (2019).

[280] Y. Liu, Z. Xu, C. Li, Distributed online semi-supervised support vector machine, Inf. Sci. (2018).

[281] S. Chan, P. Treleaven, Chapter 5 – Continuous Model Selection for Large-Scale Recommender Systems, in: V. Govindaraju, V.V. Raghavan, and C.R. Rao, (Eds.), Big Data Analytics, vol. 33 of Handbook of Statistics, Elsevier, 2015, pp. 107–124. .

[282] S. Yin, J. Yin, Tuning kernel parameters for SVM based on expected square distance ratio, Inf. Sci. (2016).

[283] A. Candelieri, I. Giordani, F. Archetti, K. Barkalov, I. Meyerov, A. Polovinkin, A. Sysoyev, N. Zolotykh, Tuning hyperparameters of a SVM-based water demand forecasting system through parallel global optimization, Comput. Operat. Res. (2018).

[284] X. Wang, F. Huang, Y. Cheng, Super-parameter selection for Gaussian-Kernel SVM based on outlier-resisting, Measur.: J. Int. Meas. Confederation, 2014. .

[285] F. Friedrichs, C. Igel, Evolutionary tuning of multiple SVM parameters, Neurocomputing (2005).

[286] X. Zhang, X. Liu, Z.J. Wang, Evaluation of a set of new ORF kernel functions of SVM for speech recognition, Eng. Appl. Artif. Intell. (2013).

[287] G. Du, S. Tian, Y. Qiu, C. Xu, Effective and efficient Grassfinch kernel for SVM classification and its application to recognition based on image set, Chaos Solitons Fractals (2015).

[288] M. Dalponte, L.T. Ene, M. Marconcini, T. Gobakken, E. Næsset, Semi-supervised SVM for individual tree crown species classification, ISPRS J. Photogrammetry Remote Sens. 110 (2015) 77–87.

[289] M. Davy, F. Desobry, A. Gretton, C. Doncarli, An online support vector machine for abnormal events detection, Signal Processing (2006).

[290] H. Cevikalp, V. Franc, Large-scale robust transductive support vector machines, Neurocomputing (2017).

[291] Y. Li, Y. Wang, C. Bi, X. Jiang, Revisiting transductive support vector machines with margin distribution embedding, Knowl.-Based Syst. (2018).

[292] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (Aug 2013) 1798–1828.

[293] D. Quang, Y. Chen, X. Xie, DANN: a deep learning approach for annotating the pathogenicity of genetic variants, Bioinformatics 31 (Oct 2014) 761–763.

[294] Y. Tang, "Deep learning using support vector machines," ArXiv, vol. abs/1306.0239, 2013. .

[295] P. Liu, K.-K.R. Choo, L. Wang, F. Huang, SVM or deep learning? a comparative study on remote sensing image classification, Soft. Comput. 21 (Jul 2016) 7053–7065.

[296] Y. Chen, Z. Lin, X. Zhao, G. Wang, Y. Gu, Deep learning-based classification of hyperspectral data, IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. 7 (2014) 2094–2107. .

[297] Y. Kim, H. Lee, E.M. Provost, Deep learning for robust feature generation in audiovisual emotion recognition, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013..

[298] T. Majtner, S. Yildirim-Yayilgan, J.Y. Hardeberg, Combining deep learning and hand-crafted features for skin lesion classification, in: 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA), IEEE, Dec 2016. .

[299] K. Tomita, R. Nagao, H. Touge, T. Ikeuchi, H. Sano, A. Yamasaki, Y. Tohda, Deep learning facilitates the diagnosis of adult asthma, Allergol. Int., 2019. .

[300] M.-J. Tsai, Y.-H. Tao, I. Yuadi, Deep learning for printed document source identification, Signal Processing: Image Commun. 70 (Feb 2019) 184–198.

[301] J. Riordon, C. McCallum, D. Sinton, Deep learning for the classification of human sperm, Comput. Biol. Med. 111 (Aug 2019) 103342.

[302] N. Codella, J. Cai, M. Abedini, R. Garnavi, A. Halpern, J.R. Smith, Deep learning, sparse coding, and SVM for melanoma recognition in dermoscopy images, in Machine Learning in Medical Imaging, Springer International Publishing, 2015, pp. 118–126.

[303] S.M. Erfani, S. Rajasegarar, S. Karunasekera, C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, Pattern Recogn. 58 (Oct 2016) 121–134.

Jair Cervantes received the B.S. degree in Mechanical Engineering from Orizaba Technologic Institute, Veracruz, Mexico, in 2001 and the M.S degree in Automatic Control from CINVESTAV-IPN, México, in 2005. In 2009 he got a Ph.D. in Computer Science at Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN). His research interests include support vector machines, pattern classification, neural networks, fuzzy logic, clustering and genetic algorithms.

Farid García-Lamont received a B.Sc. degree in Robotics from ESIME-IPN, México in 2000; in 2004 obtained a M. Sc. in Automatic Control from the Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV-IPN), México. In 2010 received a PhD degree in Computer Science from CINVESTAV-IPN. His research interests are pattern recognition, applications of artificial intelligence and robotics.

Lisbeth Rodríguez-Mazahua received the B.S. degree in informatic and the M.S. degree in computer science from the Instituto Tecnológico de Orizaba, Veracruz, Mexico, in 2004 and 2007, respectively. In 2012 she got a Ph.D. in Computer Science at Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN). From 2012 to 2014, she was a Professor of computer science at Universidad Autónoma del Estado de México, Centro Universitario UAEM Texcoco. Since February 2014 she is doing a postdoctoral research at Instituto Tecnológico de Orizaba. Her current research interests include distribution design of databases, database theory, autonomic database systems, multimedia databases, data mining and Big Data.

Asdrúbal López-Chau, received his B.S. degree in Communications and electronics from ESIME-IPN, Mexico, in 1997; in 2000, received his M.S. the degree in Computer Engineering from CIC-IPN, Mexico; in 2013, his Ph.D. degree in Computer Science from CINVESTAV-IPN, Mexico. Since 2009, he is a computer science researcher at Universidad Autonoma del Estado de Mexico, CU UAEM Zumpango, Mexico.