Pattern Recognition 110 (2021) 107649

Contents lists available at ScienceDirect

Pattern Recognition

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

Multi-objective adaptive differential evolution for SVM/SVR hyperparameters selection

Carlos Eduardo da Silva Santos[a,b], Renato Coral Sampaio[c,d], Leandro dos Santos Coelho[e], Guillermo Alvarez Bestard[c,d], Carlos Humberto Llanos[c,f,*]

[a] Federal Institute of Science, Education and Technology of Tocantins - IFTO, Brazil
[b] AE 310 Sul, Avenida LO 05, s/n, Plano Diretor Sul, TO
[c] University of Brasilia - UnB, Brazil
[d] UnB - Faculdade Gama, Área Especial de Indústria Projeção A, Gama, DF
[e] Industrial & Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR), and Department of Electrical Engineering, Federal University of Parana (UFPR), Curitiba, PR, Brazil
[f] UnB - Faculdade de Tecnologia, Campus Universitário Darcy Ribeiro, Brasília, DF, Brazil

ARTICLE INFO

Article history:
Received 28 October 2019
Revised 31 August 2020
Accepted 7 September 2020
Available online 11 September 2020

Keywords:
Support vector machines
Parameters selection problem
Multi-objective optimization
Differential evolution
Adaptive parameters strategy

ABSTRACT

Parameters Selection Problem (PSP) is a relevant and complex optimization issue in Support Vector Machine (SVM) and Support Vector Regression (SVR), looking for obtaining an optimal set of hyperparameters. In our case, the optimization problem is addressed to obtain models that minimize the number of support vectors and maximize generalization capacity. However, to obtain accurate and low complexity solutions, defining an adequate kernel function and the SVM/SVR's hyperparameters are necessary, which currently represents a relevant research topic. To tackle this problem, this work proposes a multi-objective metaheuristic named Adaptive Parameter control with Mutant Tournament Multi-Objective Differential Evolution (APMT-MODE). Its performance is first tested in a series of benchmarks for classification and regression problems using simple kernels such as Gaussian and polynomial kernels. In both cases, the APMT-MODE is able to yield more precise and more straightforward solutions using simple kernels. Then, the approach is used on a real case study to create a welding bead depth and width SVR models for a Gas Metal Arc Welding (GMAW) process. Additionally, a study on kernel functions was developed in terms of computational effort, aiming to assess its performance for embedded systems applications.

1. Introduction

Support Vector Machine (SVM) and Support Vectors Regression (SVR) are Machine Learning (ML) techniques that have been proven to have high generalization capacity and robustness in both data classification and regression. Before the SVM/SVR training process, it is necessary to set the hyperparameters that are composed of the regularization parameter (C) and the kernel parameters. These hyperparameters control the balance between the SVM/SVR's generalization capability and its complexity. The process of choosing them is called the *Parameters Selection Problem* (PSP), which is a complex optimization problem with several local minima. Thus, gradient directed algorithms may become trapped in a local optimum. Additionally, SVM/SVR have high sensibility over small changes of these parameters, making it an arduous training process.

Several works approach the PSP as a Single-Objective Problem (SOP), using metaheuristics based approaches. However, minimizing the SVM/SVR's error and complexity simultaneously, over the training set, is a problem with conflicting criteria. Therefore, our work tackles it as a Multi-Objective Optimization Problem (MOOP), where the complexity of SVRs/SVMs is measured by the Support Vector cardinality ($\#SV$) and the generalization capability is evaluated by the accuracy for SVMs, and the Mean Squared Error (MSE) for SVRs.

In our work, a novel metaheuristics method that combines an *Adaptive Parameter Control* (APC) with Mutant Tournament in a Multi-objective Differential Evolution (APMT-MODE) algorithm is

* Corresponding author.
E-mail addresses: carlosedu@ifto.edu.br (C.E.d.S. Santos), renatocoral@unb.br (R.C. Sampaio), leandro.coelho@pucpr.br (L.d.S. Coelho), guillermo@unb.br (G.A. Bestard), llanos@unb.br (C.H. Llanos).

https://doi.org/10.1016/j.patcog.2020.107649
0031-3203/

used to solve this problem introducing strategies to tune its steps along with some aspects proposed in [1]. In our case, the MOOP is addressed based on the Pareto Front, which merges multiple metrics in a vector of objective functions that may conflict with each other, characterizing a non-linear problem in the objectives space. Here, an important concept to measure the quality of solutions is the notion of *dominance*, as defined in Coello et al. [2], in which the optimization process yields a set of solutions where none of them dominate each other, known as the *non-dominated set*.

Usually, in multi-objective approaches, strategies such as crowding distance, hypervolume, ranking, or weighting of the objective functions are used to choose the best solutions to guide the metaheuristics over the search space [2]. However, our work uses the *Criticism of Lexicographic Ordering* (CLO) [2] to adapt the Differential Evolution (DE) [1] with a multi-objective approach and to properly tune the strategy to obtain solutions able to yield a better tradeoff between high precision and a low #SVs.

Most previous works that deal with multi-objective strategies and, therefore, somehow consider the SVM/SVR complexity do not pay attention to the kernel's computational complexity. This fact becomes essential in embedded systems with performance (timing) and power consumption restrictions. Some mathematical kernel operations such as exponential, sine, and cosine use algorithms such as the CORDIC [3] whose numeric values are approximated by series of sums and multiplications and have a higher computational cost than a polynomial kernel with simpler operations. Therefore, our multi-objective strategy seeks solutions that minimize the amount of #SVs and maximize the accuracy of the obtained models, which is essential in embedded system design.

To carry out our SVM complexity study, including kernel computational effort for embedded systems, the SVMs resulting from our proposed metaheuristic (APMT-MODE) were implemented in C language and executed in an ARM processor. Additionally, the APMT-MODE was also applied to a real case study for estimating weld bead width and depth for a Gas Metal Arc Welding (GMAW) process presented in [4]. As shown in the literature review of [5], there are no applications of SVMs/SVRs in this context. Since the welding process is highly nonlinear and sensitive to changes, a mathematical model with minimal errors and fast response is essential for real-time weld bead geometry control.

We emphasize that our technique is based on APC so that one can find good quality solutions for any application/kernel function combination. This strategy is relevant due to the fact that PSP has a complex search space problem with distinct properties for different combinations of applications and/or kernels. So, given an application and a kernel, the *mutation* and *crossover* parameters (F, C_r) of the DE can provide a suitable solution. However, these same parameters may not be ideal other applications/kernel. In this sense, we have performed statistical tests to compare the results obtained by the APMT-MODE with the results using the APMODE (the latter uses mutation strategies defined in [1]) which proves the advantages of the adaptive strategies proposed in this work.

Among the contributions of this paper, we highlight: (a) a novel Adaptive Parameter with Mutant Tournament Multi-objective Differential Evolution algorithm (the APMT-MODE) able to find good solutions for the SVM/SVR parameters selection problem using simple kernels; (b) the development of a strategy using CLO to deal with both objective functions in an efficient way that was properly tuned (using experimental information) to obtain good results as shown in Section 5; (c) a computational complexity study of the kernels for embedded systems applications; and (d) an SVR-based model addressed to estimate welding bead width and depth that is more accurate and faster than the original work presented in [4]. This study was made possible using our tool (APMT-MODE), which allows for collecting a large amount of information from SVM solutions with different kernels, with their parameters tuned using our MOOP strategies.

2. Related works

2.1. Single-objective approaches

In Ping Jiang et al. [6], the parameters of $\nu - SVM$ are tuned by the metaheuristic Single-objective Cuckoo Search (CS), which has the Mean Absolute Percentage Error - MAPE as its criterion. The $\nu - SVM$ regressor models were developed with a Gaussian kernel to predict wind speed for wind power generation. Alaa Tharwat et al. [7] propose mathematical models based on SVM to predict drug toxicity in the early stages of drug development, where the Whale Optimization Algorithm (WOA) is used to adjust the hyperparameters of the SVMs with a cost function that uses the square root of the product between the sensibility and specificity, employing the Gaussian kernel to handle the nonlinearity of the data.

Fei Ye *et al.* employs the Fly Optimization algorithm (FOA) with a mutation strategy that simultaneously configures the parameters of the SVM (with the Gaussian kernel) and the characteristics of the input vector. The authors used a linear combination of multiple objectives, namely: sensitivity, specificity, number of features, and the #SVs [8]. Therefore, although there are multiple objectives, the problem is modeled as an SOP. On the other hand, to detect Electroencephalograms (EEG) abnormalities, Abdulhamit Subasi et al. [9] propose the use of SVMs that are sensitive to the configuration of hyperparameters, where the hybrid GA-PSO algorithm was developed to solve the PSP of the SVMs. The kernel used was the ANOVA, and the optimization criterion is based on the weighted average of the sensitivity and specificity.

Hossam Faris et al. [10] applied the Multi-Verse Optimizer (MVO) metaheuristic to select the features and to define the hyperparameters of the SVMs. The problems are modeled as an SOP in which the cost function is the weighted sensibility mean for the specificity of the classifier model.

In the context of Internet of Things (IoT) applications, Shorman A. et al. [11] propose to detect botnet attacks using the Gray Wolf Optimization (GWO) algorithm to optimize the hyperparameters of One-Class SVMs (OCSVM) while defining the features that best describe the problem. Although the GWO is single-objective, its objective function tries to minimize the rate of false positives and maximize the rate of true positives. Otherwise, Kalita D. and Singh S. [12]. consider the PSP as a dynamic problem, where the SVM's quality degrades when new data joins the dataset. Therefore, the hyperparameters, previously optimal, must be updated to meet the demand of the application. To resolve the PSP dynamism, the Quantized Multiple PSO (Q-DMPSO) was developed employing multiple swarms by dividing the original flock into sub-swarms that communicate, based on the idea of communication-exclusion and anti-convergence.

Kouziokas G. et al. [13] define a hybrid structure SVR for predicting the growth of Gross Domestic Product (GDP). The SVR is developed in three stages: (a) training an Artificial Neural Network (ANN) by a PSO, (b) training an SVM with the Gaussian kernel tuned by a Bayesian Optimized (BO) algorithm, and (c) building the W-SVM kernel. The results show that the W-SVM's hybrid model is better than the classic ANN and SVM models and the BO-SVM and PSO-ANN models.

In R. Laref et al. [14], SVRs are widely used for the quantification of gasses through sensors called electronic-noses. The authors propose the Generalized Pattern Search (GPS) algorithm as a fast alternative to find the hyperparameters. Additionally, a study is carried on the relationship between the regularization parameter (C) and the ϵ, as well as on the relationship between the C and the γ (of the Gaussian kernel) parameters. The results con-

clude that this is a problem with several local minima and plateau regions.

These works exemplify much interest in SVMs/SVRs and their broad applicability. However, none of them approach the SVR's PSP as a MOOP. Also, there is no absolute consensus about which criterion (objective function) to be used in the optimization model of hyperparameters. Additionally, the Gaussian kernel is also observed to be used in most works.

2.2. Multi-objective approaches

A self-adaptive evolutionary algorithm is proposed by Igel [15], which updates individuals based on a standard distribution $N(0, 1)$ that does not change during the evolutionary process. The algorithm is the first Evolution Strategy (ES) with self-adaptation that uses non-dominant ordering called NSES (Self-adaptive Evolution Strategy Non-dominate Sorting). The NSES was developed to use the Gaussian kernel for classifiers and highlights the study of the possible objective functions of the MOOP. In [16], Igel et al. approach the PSP with three objective functions modeled by using the same algorithm over benchmarks, as well as the same objective functions, proposed on the NSGA-II (Non-sorting Genetic Algorithm, version II) to classify the presence of pedestrians in autonomous car images.

In Chatelain et al. [17], the NSGA-II is used to solve the PSP of SVMs where the objective functions considered are the False-Positive (FP) and False-Negative (FN), that is, it does not evaluate the complexity of the SVM. The decision variables are the γ of the Gaussian kernel and two regularization parameters because, in this case, each class is treated independently.

The Multi-objective Uniform Design - MOUD algorithm, proposed by W. Li et al. [18]. employs the *Uniform Design* technique, which is similar to the Symmetric Latin Hypercube Design (SLHD) [19]. The MOUD does not consider the complexity of the SVMs, but the sensitivity and the specificity as multi-objective criteria to obtain the optimal hyperparameters. The Gaussian and the polynomial kernels are used. Additionally, Miranda et al. [20] has developed a hybrid algorithm in which a *Meta-Learning* (ML) technique based on the characteristics of the problems generates a swarm of good quality solutions (as initial swarm for MOPSO). It employs the Gaussian kernel to generate classifiers, and the MOOP criteria are *#SV* and precision. One notable limitation is the fact that the search space is discretized by limiting it to 399 different configurations of C and γ (the hyperparameters adopted), which compromises the quality of the models.

The Multi-Objective Whale Optimization Algorithm (MOWOA), proposed by Xiong et al. [21], searches for defining the Gaussian kernel parameter and the number of inputs required to forecast the carbon price in three subsequent time units. The problem was modeled as a MOOP having the Root Mean of Squared Error (RMSE) of the three outputs of the model as objective functions. It is worth mentioning that the objective functions are not conflicting in the MOWOA defined by Xiong et al.

Therefore, It can be observed that all multi-objective algorithms previously described use the NSGA-II, and a Gaussian kernel to develop classifiers, i.e., a subcategory of the problem. It should be noted that for different kernels, one always has PSPs with different characteristics. Also, studies that model the PSP as a MOOP for SVRs have not been found in the literature, which makes evident the need for more careful research on the subject, using different kernels and algorithms that solve the PSP as a MOOP.

3. Support vector machines

The training process of an SVM is a convex quadratic optimization problem, as shown by Eq. (1), which is a necessary and sufficient condition to guarantee a global optimal solution.

$$\begin{aligned} \max \quad & L_c(\alpha_i) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0, \quad \xi_i \geq 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N, \end{aligned} \tag{1}$$

where C is a regularization parameter, N is the training set cardinality, α's are the Lagrange multipliers, ξ's are the slack variables and $K(\cdot)$ is the symmetric Positive Semidefinite (PSD) matrix. This matrix must satisfy Merce's conditions and is called *kernel matrix* or *kernel function* [22].

The training set is defined as $X = \{(\mathbf{x}_i, y_i)\}$, $i = 1, 2, \ldots, N$ such that $\mathbf{x}_i \in \mathbb{R}^n$ e $y_i \in \{-1, 1\}$ for classifiers and $y_i \in \mathbb{R}$ for the approximation function, where $\mathbf{x}_i$ represents the features vector, y_i is the label and N is the cardinality of the training set.

Originally SVMs were designed for classification tasks, but Vapnik et al. [23] extended the technique to treat regression problems by penalizing points out of a tube around the curve. This technique is named Support Vector Regression (SVR), and its training problem is expressed by Eq. (2),

$$\begin{aligned} \max \quad & L_r(\alpha_i) = \frac{1}{2} \sum_{i,j=1}^{n} K(\mathbf{x}_i, \mathbf{x}_j) \overline{\alpha}_i \overline{\alpha}_j + \epsilon \sum_{i=1}^{N} \overline{\alpha}_i - \sum_{i=1}^{N} y_i \overline{\alpha}_i \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{n} \overline{\alpha}_i = 0, \quad 0 \leq \alpha_i, \alpha_i^{\star} \leq C, \end{aligned} \tag{2}$$

where C is a regularization parameter that penalizes data points containing errors greater than ϵ by a positive constant. The ϵ is the ϵ-insensitive parameter that defines a tube around the curve to be approximated and $\overline{\alpha}_i$ are the decision variables, i.e. $\overline{\alpha}_i = \alpha_i - \alpha_i^{\star}$ where α_i and $\alpha_i^{\star}$ are the Lagrange multipliers.

The mathematical models for the SVM and SVR are given by Eqs. (3) and (4), respectively.

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{\#SV} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b \right), \tag{3}$$

$$f(\mathbf{x}) = \sum_{i=1}^{\#SV} \overline{\alpha}_i^* K(\mathbf{x}_i, \mathbf{x}) + b, \tag{4}$$

where *#SV* is the cardinality of the SV set, α_i^* and $\overline{\alpha}_i^*$ are the optimal values that satisfy Eqs. (1) and (2), respectively, and b is the bias that can be evaluated from optimal alphas.

3.1. Kernel issues

Radial Basis Functions (RBFs, in our case, Gaussian), polynomial, and sigmoid functions are generally used as kernels [22]. However, each one is best suited to specific applications where the choice criteria are: (a) low complexity, (b) high generalization capability, and (c) few parameters to be tuned. Previous works demonstrate that different kernels satisfy these requirements [24].

Moghaddam and Hamidzadeh [24] propose the Hermite orthogonal polynomial kernel, presenting good generalization capability with a low number of SVs and a lower computational effort when compared to the Gaussian, Wavelet and Chebyshev kernels for a list of benchmarks presented. Furthermore, combinations of the Hermitian kernel, such as the Hermite-Chebyshev (H-C) kernel, present a lower number of SVs. At the same time, the Hermite-Gaussian (H-G) has a lower error rate on all benchmarks. However, both combinations preset a higher computational effort. The proposed kernels defined in [24] along with the polynomial and Gaussian kernels are presented in Table 1.

Table 1
Kernel functions [24].

Kernel	Function $K(x, y)$	Parameters
Polynomial	$K(\mathbf{x},\mathbf{y}) = (\beta\mathbf{x}\cdot\mathbf{y}+\kappa)^d$	β, κ, d
Gaussian	$K(\mathbf{x},\mathbf{y}) = exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\gamma}\right)$	γ
Hermite	$K(\mathbf{x},\mathbf{y}) = \prod_{j=1}^{f}\sum_{i=0}^{n} He_i(x_j)He_i(y_j)$	n
H-G	$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\gamma}\right) + \prod_{j=1}^{f}\sum_{i=0}^{n} He_i(x_j)He_i(y_j)$	n, γ
H-C	$K(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=0}^{n} T_i(\mathbf{x})T_i^t(\mathbf{y})}{\sqrt{a-\langle\mathbf{x},\mathbf{y}\rangle}} + \prod_{j=1}^{f}\sum_{i=0}^{n} He_i(x_j)He_i(y_j)$	n, a

4. Multi-Objective optimization problem (MOOP)

Minimizing both the errors over the training set and the approximation function's complexity is crucial in SVMs/SVRs. However, these are conflicting criteria, and thus our work models this dilemma as a MOOP defined by Eq. (5),

$$\min_{\omega\in\Omega} \quad \mathbf{F}(\omega) = (E(\omega), C_{SV}(\omega)), \tag{5}$$

where **F** is the objective vector. For SVRs, $E(\omega)$ is the model MSE on the validation set (see Eq. (6)), C_{SV} is the support vector set cardinality (#SV), ω is the vector of hyperparameters, and Ω is the set of feasible ones.

$$E(\omega) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2. \tag{6}$$

For SVMs, $E(\omega)$ represents a metric of the error rate evaluation over validation set and ω is the hyperparameters vector. For APMT-MODE, the criteria in the definition of the classifier hyperparameters are both accuracy and #*SV*; while for regressors are the MSE and the #*SV*.

The representation adopted for both the regularization parameter and the Gaussian kernel parameter are $C = e^{C^*}$ and $\gamma = e^{\gamma^*}$, respectively, where C^* and γ^* are the decision variables for APMT-MODE. This fact allows the evaluation of more extensive search ranges with a reduced number of iterations [25]. All SVM/SVRs related work researched adopt the cardinality of the set of support vectors for hardware implementation as a metric of complexity. Still, severe divergences between models with the same amount of support vectors and different kernels can be achieved. As an example, one SVM/SVRs with the polynomial kernel has much lower computational effort than another with the same #SVs but with the Gaussian kernel.

4.1. Adaptive parameter with mutant tournament multi-objective differential evolution (APMT-MODE)

The DE algorithm, proposed by Storn and Price [26], simulates the evolutionary process to solve optimization problems with continuous variables. DE, like other evolutionary algorithms, also applies mutation, crossover, and selection operators over its population.

In our work, the APMT-MODE parameters F and C_r together with mutation and crossover operators are adjusted in runtime by a normal distribution $N(\mu, \sigma)$ whose mean is calculated concerning the quality of the individuals of the current generation, and the standard deviation σ is controlled by a decreasing function.

Seminal proposals to adapt DE to multi-objective are described in Del Ser-et al. [27]. In our case, the PSP of SVMs has two conflicting objective functions, namely: (a) the number of SVs (#SV) that measures the complexity, and (b) the error on the validation set (empirical error).

Given the multimodal characteristics of PSP, both *Cr* and *F* that control mutation and crossover strategies are controlled by an APC-based approach inspired by Fan and Zhang work [1].

Table 2
The Hyperparameters description.

Parameters/Kernel	Parameter description	Parameter Range
Gaussian	$\gamma = \exp(\gamma^*)$, $\gamma^* \in \mathcal{R}$	[−10, 10]
Hermitian	$n \in \mathbb{Z}_+$	[1, 10]
Polynomial	$d \in \mathbb{Z}_+$	[1, 10]
SVR e-insensitive	$\epsilon \in \mathcal{R}_+$	(0, 10]
Regularization	$C = \exp(C^*)$, $C^* \in \mathcal{R}$	[−10, 10]

4.2. Description of the proposed APMT-MODE metaheuristic

APMT-MODE allows the user to work with kernels shown in Table 2, and introduces three modifications over the DE proposed in Fan and Zhang work [1]: (a) the equation to control the standard deviation of parameter adjustment, (b) replacement of one of the mutation operators *DE/rand-to-best/1* (described in [1]) by the tournament mutation, and (c) adaptation of the original algorithm to treat the PSP as a MOOP, using the CLO method.

To compare the impact of our mutation operators, we have introduced a previous version keeping the classic DE mutation operator *DE/rand-to-best/1*. In this case, we have named this algorithm AP-MODE, which differs from APMT-MODE only in item (b).

In our proposed strategy, parameters *F* and *Cr*, and the individual's update are performed by four mutation and two crossover strategies that act on different subgroups of individuals that are updated every five iterations, eliminating the need of tuning for different data sets or kernel types [27].

The APMT-MODE algorithm consists of the following steps: *Initialize, Adaptive Parameters, Mutation, Crossover, Evaluation*, and *Selection*. Each of these steps is described in Fig. 1.

1. *Initialize*: Several DE implementations use the uniform distribution to initialize the population, which does not guarantee reasonable spread over the search space. Therefore, to generate the initial population, the Symmetric Latin Hypercube Design (SLHD) was applied [19].
2. *Adaptive Parameters Control*: Parameters *F* and *Cr* are updated by Eqs. (7) and (8), respectively, for each individual. Additionally, the cardinality of the subgroups of individuals that will be modified by crossover and mutation strategies is also updated. These updates occur every five iterations.

$$F_i^{G+1} = N\left(F_w^{G+1}, \sigma\right) \tag{7}$$

$$Cr_i^{G+1} = N\left(Cr_w^{G+1}, \sigma\right) \tag{8}$$

where N is a Normal distribution with standard deviation σ and mean of F_w^{G+1} and Cr_w^{G+1} given by Eqs. (9) and (10), respectively. In our approach the standard deviation is defined as $\sigma = 0.1 + 0.4 \times (1 - (G/G_{\max})^2)$, where, G is the current generation, and $G_{\max}$ is the maximum number of generations.

$$F_w^{G+1} = \sum_{i=1}^{NP} w_i^G \times F_i^G \tag{9}$$

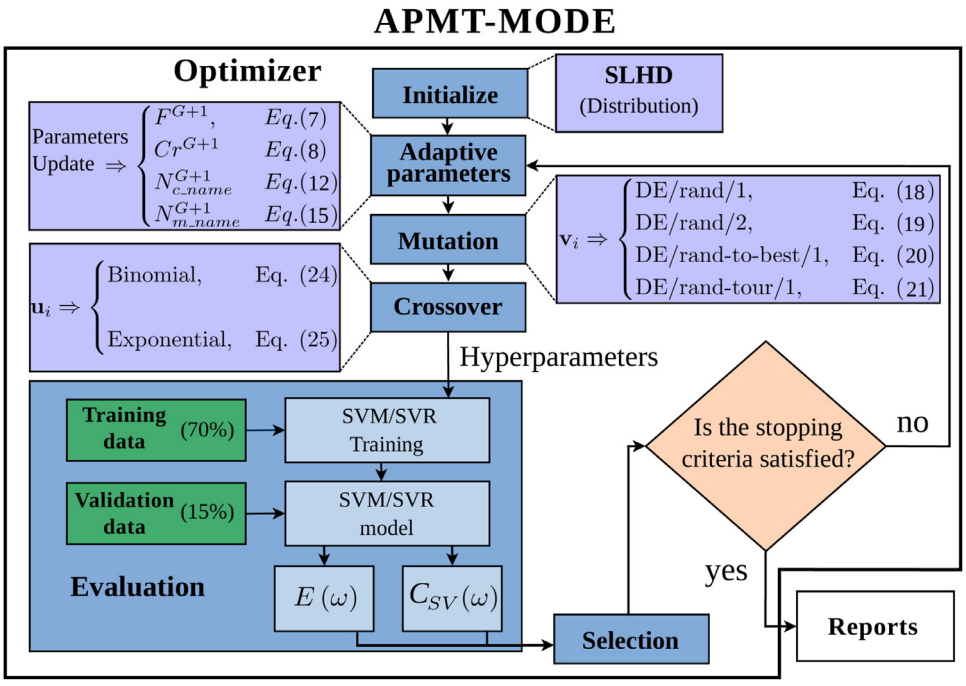


Fig. 1. The data flow diagram of APMT-MODE.

$$Cr_w^{G+1} = \sum_{i=1}^{NP} w_i^G \times Cr_i^G \tag{10}$$

where NP is the number of population individuals and the next w_i^G is computed by Eq. (11),

$$w_i^{G+1} = \frac{\mid f_j(\mathbf{x}_i^G) - \max\left(f_j(\mathbf{x}^G)\right) \mid}{\sum_{k=1}^{NP} \mid f_j(\mathbf{x}_k^G) - \max\left(f_j(\mathbf{x}^G)\right) \mid} \tag{11}$$

where f_j is the objective function chosen by CLO strategy, $\mathbf{x}_i$ is the i-th individual from the population, and G is the current generation. The APC also defines the cardinality of the subgroups of individuals that will be modified by the exponential and binary crossover strategies in the next generation ($N^{G+1}_{c\_name}$), calculated by Eq. (12).

$$N^{G+1}_{c\_name} = \begin{cases} N^G_{c\_name} + 1, & \text{if } En_{c\_name} > N^G_{c\_name} \\ N^G_{c\_name} - 1, & \text{if } En_{c\_name} < N^G_{c\_name} \\ N^G_{c\_name}, & \text{otherwise.} \end{cases} \tag{12}$$

The adaptive strategy benefits the crossover making the most significant difference between the worst individual and individuals in its set, evaluating the number of individuals for each crossover strategy by Eq. (13),

$$En_{c\_name} = round\left(NP \times \frac{S^{G+1}_{c\_name}}{S^{G+1}}\right) \tag{13}$$

where, En_{c_name} is the evaluated number of individuals to c_name crossover methodology, $round$ is a rounding function, $S^{G+1} = S^{G+1}_{bin\_cross} + S^{G+1}_{exp\_cross}$, $c\_name \in \{bin\_cross, exp\_cross\}$ (as defined in Eqs. 22 and 23), and $S^{G+1}_{c\_name}$ is computed by Eq. (14)

$$S^{G+1}_{c\_name} = \sum_{k=1}^{N^G_{c\_name}} \mid f_j(\mathbf{x}^G_{c\_name,k}) - \max\left(f_j(\mathbf{x}^G)\right) \mid . \tag{14}$$

Finally, the best number of individuals for each mutation strategy over the iteration process is defined by Eq. (15),

$$N^{G+1}_{m\_name} = \begin{cases} N^G_{m\_name} + 1, & \text{if } En_{m\_name} > N^G_{m\_name} \\ N^G_{m\_name} - 1, & \text{if } En_{m\_name} < N^G_{m\_name} \\ N^G_{m\_name}, & \text{otherwise,} \end{cases} \tag{15}$$

where m_name are indexes representing the mutation strategies (see Eqs. 18-21) and En_{m_name} is defined by Eq. (16)

$$En_{m\_name} = round\left(NP \times \frac{S^{G+1}_{m\_name}}{S^{G+1}}\right), \tag{16}$$

where $S^{G+1} = \sum_{m\_name=1}^{K} S^{G+1}_{m\_name}$, K is the amount of mutation strategies, and $S^{G+1}_{m\_name}$ is evaluated by Eq. (17),

$$S^{G+1}_{m\_name} = \sum_{k=1}^{N^G_{m\_name}} \mid f_j(\mathbf{x}^G_{m\_name,k}) - \max\left(f_j(\mathbf{x}^G)\right) \mid, \tag{17}$$

where f_j is the objective function employed in crossover strategies and m_name is same as Eq. (15).

3. *Mutation*: Produces NP mutants distributed among mutation strategies as evaluated by Eq. (15). The Eqs. 18–(21) show proposed mutation strategies,

$$\mathbf{v}_{i,DE/rand/1} = \mathbf{x}_{r_1} + F_i(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{18}$$

$$\mathbf{v}_{i,DE/rand/2} = \mathbf{x}_{r_1} + F_i(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) + F_i(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}) \tag{19}$$

$$\mathbf{v}_{i,DE/current-to-best/1} = \mathbf{x}_{r_1} + F_i(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) + F_i(\mathbf{x}_{best} - \mathbf{x}_i) \tag{20}$$

$$\mathbf{v}_{i,DE/rand-tour/1} = \mathbf{x}_{r_1} + F_i(\mathbf{x}_{t\_best} - \mathbf{x}_i) \tag{21}$$

where r_1, r_2, r_3, r_4, and r_5 are individuals randomly picked from the population and mutually different, i is the i-th individual, $\mathbf{v}$ represents the mutant vector, $\mathbf{x}_{best}$ is the best individual of the current population and $\mathbf{x}_{t\_best}$ is the individual with the best evaluation objective function of a subset randomly picked from the population. Note that the F_i parameter is adapted by step 2 of the algorithm.

Equation (21) defines a new strategy for MOOP in which the mutation searches the balance between exploration and exploitation. To that end, a subset of the population with 25% of

randomly picked individuals is created and the one with the best fitness is chosen to be a $\mathbf{x}_{t\_best}$. This tournament mutation ensures a chance for an individual with not so good fitness to guide the movement of the subset population, avoiding the local minima.

It can be observed that equations 18 - (21) are applied over disjunctive subsets randomly chosen from the population with *NP* individuals. Initially, the subsets have the same cardinality. Afterward, at every five iterations, the cardinality of the subsets is updated depending on Eq. (15).

4. *Crossover*: The binary crossover (*bin_cross*) verifies each position of the mutant vector and changes it, as shown in Eq. (22).

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & \text{if} \quad rand_{i,j} \leq Cr \text{ or } j = j_{rand} \\ x_{ij}^{G}, & \text{otherwise,} \end{cases} \tag{22}$$

where $\mathbf{x}$ are the parents, $\mathbf{u}$ are the offspring vectors, j is the j-th position of i-th individual, $rand_{i,j}$ is a uniform random number in the interval [0, 1], and j_{rand} is a randomly chosen integer number in the range [1, D_m] where D_m is the dimensionality of the individual. The condition $j = j_{rand}$ ensures that at least one position of the offspring has the mutation information.

The exponential crossover (*exp_cross*) splits the mutant and parent vectors in a random position and interchanges its parts to create a new vector, as shown in Eq. (23):

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & \text{if} \quad j = \langle n \rangle_{D_m}, \langle n+1 \rangle_{D_m}, \ldots, \langle n+L-1 \rangle_{D_m} \\ x_{ij}^{G}, & \text{otherwise,} \end{cases} \tag{23}$$

where $\langle\rangle_D$ denotes the modulus operator with modulus D_m. The number L is an integer drawn from [1, D_m], and n is a random number.

Equations (22) and (23) are applied over disjunctive subsets randomly chosen from the population of *NP* individuals. Subsets have the same cardinality initially and are updated at every five iterations based on Eq. (12).

5. *Evaluation*: Individuals represent the hyperparameters required to train an SVM/SVR. The training of an SVM consists of solving an optimization problem (Eq. (2)) for a given training set (see the detail of this step in Fig. 1). The model is generated by the LibSVM library [28], being evaluated with a validation data set. The 15% of the remaining data (the test data set) are reserved for evaluating the models belonging to PF. Therefore, this process allows calculating the $\mathbf{F}(\omega)$, as shown in Eq. (5). Additionally, in APMT-MODE and AP-MODE, the user can employ the *n-fold-cross-validation* technique to calculate the same objective function values, being user responsible for defining the parameter n in the cross-validation.
6. *Selection*: The offspring joins the population, and so the non-dominated sorting is applied in [15]. This way, only non-dominated individuals are carried on to the next generations. This iterative process repeats from step 2 until it achieves the maximum number of generations $G_{\max}$ (see Fig. 1).

A contribution of APMT-MODE with regards to [1] is the replacement of the mutation strategy shown in Eq. (24) by the tournament mutation (see Eq. (21)), which allows individuals, not necessarily the best ones, to have the opportunity to guide the mutation process.

$$\mathbf{v}_i = \mathbf{x}_i + F_i(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) + F_i(\mathbf{x}_{c\_best} - \mathbf{x}_i) \tag{24}$$

Another relevant contribution is the adaptation of the Single-Objective algorithm proposed by Fan and Zhang in [1] to address the problem of the selection of SVM hyperparameters as a MOOP where all objective functions have the same importance. However, to chose the best individual in APMT-MODE, one of the objective functions is randomly selected to be the decision criteria. This approach is called *Criticism of Lexicographic Ordering* (CLO) [2]. The strategy proposed by [1] initializes the adaptive process after $G = 0.2 \times G_{\max}$ first generations, and afterwards, the parameters are updated at each generation. In our proposal, the adaptive process updates the parameters every five iterations.

The values of σ in Eqs. (7) and (8) were tuned in order to improve our results as discussed in Section 5, having different values than those suggested by [1]. In our work, the equation that controls the behavior of sigma (σ) generates values close to its maximum point in the initial generations reaching half of the maximum value in 80% of all generations. This behavior increases the probability that the values of F and Cr, obtained by the distribution $N(\mu, \sigma)$, are far from the average μ, increasing the duration of the exploration phase of the developed APMT-MODE and AP-MODE.

Another difference is that in our metaheuristic, the CLO changes from one objective function to another, giving more chances of individuals and their parameters to improve a specific objective function. The main strength of CLO is that it may be able to depict a concave Pareto Front, although this depends on the distribution of the population and the problem itself. Otherwise, the CLO's main weakness is that it tends to favor specific objective functions when many of them are present in the problem because of the randomness behavior of the process [2]. However, the PSP, as approached in this work, has only two criteria, which minimizes this risk. An additional contribution is that the used SLHD technique proved to be more capable of increasing the diversity of the initial population [19].

5. Results and discussion

5.1. Results in classification benchmarks

Our goal of using classification benchmarks detailed in Table 3 is to demonstrate that an adequate choice of hyperparameters can produce better solutions using classical and simpler kernels than the ones presented in [24]. Each benchmark problem was solved using both the Gaussian and the polynomial kernels with the AP-MODE and APMT-MODE metaheuristics. For each benchmark and kernel, a PF was generated with a set of equivalent solutions, from its respective Pareto Set (PS). Table 4 illustrates the Pareto Set (*C, Degree*) with its respective PF (E, C_{SV}) for the polynomial kernel on the Ecoli benchmark. To better compare these results with the ones in [24], for each benchmark a solution with similar error and a lower #*SV* was chosen from each PF. For instance, in the particular case of Table 4, the first result is selected (in boldface font). For a fair comparison with the results in [24], the 10-fold-cross-validation strategy was adopted to generate the classifier models from both AP-MODE and APMT-MODE optimization process.

We chose the Hermite, H-C, and H-G kernels to be compared since they show the best performances over the remaining kernels presented in [24]. In this case, the comparison was made against

Table 3
Classification benchmarks characteristics.

Benchmark	Class	Data	Feat.
Ecoli	8	336	7
Glass	7	214	9
Haberman	2	306	3
Heart	2	267	44
Ionosphere	2	351	34
Iris	3	150	4
Liver	2	345	6
Pendigits	10	3498	36
Satimage	6	4435	36
Sonar	2	208	60
Wdbc	2	569	30

Table 4
The Ecoli with polynomial kernel PS and PF example.

C	Deg.	Error	#SV
2.50	**1**	**12.4**	**126**
10.0	1	14.8	99
1.75	2	12.7	112
−1.25	7	13.3	104
7.00	1	14.5	101
6.25	1	14.2	102
5.50	1	13.9	103
−2.75	6	13.9	103

the Gaussian and polynomial kernels using our two metaheuristics (APMT-MODE and AP-MODE). The best results are depicted in boldface in Table 5 where it can be observed that APMT-MODE has more than twice the best results. It can be observed that in most benchmarks, the APMT-MODE error results are lower or at most 19% better than previous results. The only exception was for the Satimage benchmark. On the other hand, most APMT-MODE results present a reduction of 15% to 85% in the *#SVs*, the only exception being the Sonar benchmark, which gives equivalent results.

5.1.1. Statistical analysis

For an overall comparison, Table 5 presents SVM models obtained by different combinations of kernels and hyperparameter adjustment techniques. In this case, there are seven possible combinations of *algorithms* × *kernels*, namely: Hermite, H-C, H-G, AP-G, AP-Poly, APMT-G, and APMT-Poly, where the prefixes AP and APMT stand for the AP-MODE and APMT-MODE followed by a polynomial (Poly) and Gaussian (G) kernels. Initially, we applied the Friedman's test [29] with significance $\alpha = 0.05$ for the error and obtained the Friedman statistic $F_s = 9.83$ and $p\text{-}value = 0.1316$. Since the *p-value* is greater than α, the null hypothesis is not rejected and, therefore, all methods are equivalent in terms of the error metric (accuracy).

The same procedure was applied for the *#SV*s, obtaining $F_s = 45.08$ and $p\text{-}value = 4.5E{-}8$, with the *p-value* being less than α. Therefore, Friedman's test null hypothesis was rejected and, thus, there is at least one of the seven combinations that generate solutions with different means than others. In the *post-hoc* assessment (everyone against all) the APMT-Poly performed better than Hermite, H-C, H-G, and G-AP with a *p-value* less than α. The comparison of APMT-Poly with the other methods has confidence lower than 95%. Friedman's test, in its ordering strategy, points to APMT-Poly as the best method, when considering the *#SV* metric.

By comparing *AP-MODE vs. APMT-MODE*, we identify the actual contribution of the APMT-MODE algorithm and, more specifically, of the tournament mutation. In this case, we applied the Wilcoxon's test [29] for the metaheuristics with a combination of *kernel* × *benchmark*. For instance, the Gaussian and polynomial kernels with the Ecoli benchmark generate two problems: Ecoli-Poly and Ecoli-G.

The Wilcoxon's test was applied separately for the error and *#SV*. APMT-MODE and AP-MODE do not reject the null hypothesis for the error metric, reaffirming the methodology for choosing models from PF, as described. The results show that the APMT-MODE is better than the AP-MODE with significance higher than 95% for *#SV* - i.e., for models with errors of equal means, the models generated by the APMT-MODE have a smaller cardinality in terms of *SV*s than the models generated by the AP-MODE.

5.2. Computational effort comparison

Computational effort is a metric with high importance for real-time performance and power consumption, especially on embedded systems. In this case, each SVM and kernel configuration was implemented in the C language, and run on an ARM processor (Raspberry Pi 3). Fig. 2 shows the results for each configuration in Table 5, where the total computing time for each kernel is represented as relative to the highest time of each benchmark (100%). From Fig. 2, it can be observed that both the polynomial and Gaussian kernels with parameters defined by APMT-MODE represent a fraction of the total time when compared to the kernels presented in [24]. This time reduction is due to their lower complexity and also lower *#SV*.

5.3. Results in SVR benchmarks

The developed APMT-MODE was also applied for models that have continuous outputs called regressors (SVRs). In this case, apart from both the regularization parameter and the kernel parameters, the ϵ-insensitive parameter must be set, all of them being the decision variables. To assess APMT-MODE's ability to obtain regression models, the benchmarks described in Table 6 were used, and their data were divided into 70% for the training set, 15% for the validation set and 15% for the test.

APMT-MODE generated 32 Pareto Fronts for each of the benchmarks and each kernel type. To define which model achieved the highest generalization capacity of each PF, the model with the lowest MSE in the test set was chosen. The *#SV*s of these models was also submitted to statistical analysis. The statistical test was used in both the *#SV*s and the MSE metrics. Table 7 presents the medians of the MSE and *#SV* of each benchmark for each of the kernels, where P, G, and H are polynomial, Gaussian and Hermitian kernels, respectively.

Friedman's test assumes as null hypothesis that the polynomial, Gaussian, and Hermitian kernels produce results that come from the same population, that is, that there is no statistical difference between the results produced by these kernels. The Friedman test kernels ranking for *#SV* and MSE with p-value $\alpha_{\#SV} = 0.349$ and $\alpha_{MSE} = 0.165$ are shown in Table 8. Although Gaussian and polynomial kernels have the best ranking for both the MSE and *#SV* metrics, there is no significance in rejecting the null hypothesis. Therefore, for the case study in the next section (*weld bead depth and width SVR models*), the polynomial, Gaussian, and Hermitian kernels will be used to identify which one best fits the problem in question.

5.4. Case study - SVR models for weld bead width and depth estimation

For final APMT-MODE validation, a case study of the weld bead geometry estimator presented in [4] was selected. The measure of weld bead geometry parameters is essential in Gas Metal Arc Welding (GMAW) technique for controlling the welding process to obtain a good quality weld bead (see Fig. 3 for a weld bead example). In our work, a similar methodology and the same two experimental datasets from [4] were used to obtain the SVR. Experiment 1 was used to model the weld bead depth, and Experiment 2 was used to model the weld bead width. Besides, the data set from Experiment 2 was useful for testing the weld bead depth model, and the data set from Experiment 1 was helpful for testing the weld bead width model.

For real-time estimation, the SVR models of the weld bead depth and width were generated. In both cases, the input vector is composed of eight decision variables: thermographic peak (T_p), thermographic base (T_b), thermographic area (T_a), thermographic volume (T_v), thermographic width (T_w), measurement of welding current/voltage (iu) in time (nT), previous sample of current/voltage in time ($nT - T$), and previous estimated value

Table 5
Error and *#SVs* of Hermite, H-C and H-G, where the *#SV* are the cardinality of support vectors set [24]. Boldface enters means the overall best results.

Benchmark	Hermite Kernels [24]						AP-MODE				APMT-MODE			
	Hermite		H-C		H-G		Gaussian		Polynomial		Gaussian		Polynomial	
	#SV	Error	#SV	Error	#SV	Error	#SV	Error	#SV	Error	#SV	Error	#SV	Error
Ecoli	278	11.89	273	13.10	161	**10.09**	154	12.00	122	12.72	**117**	11.81	126	12.42
Glass	153	28.30	148	30.10	151	26.10	112	**24.00**	115	26.19	102	28.09	**95**	26.19
Haberman	163	24.09	167	27.16	161	24.09	144	26.66	138	24.00	142	25.33	**137**	**24.00**
Heart	135	18.27	131	19.47	132	17.07	92	16.92	**89**	17.69	94	17.69	**89**	18.46
Ionosphere	104	6.21	92	4.14	107	**4.01**	4.57	6.00	92	7.14	**79**	4.57	**79**	7.28
Iris	75	3.40	107	4.10	101	3.10	29	2.66	23	**2.00**	12	2.66	**11**	2.00
Liver	293	26.11	287	**24.28**	291	24.90	145	27.50	**119**	27.08	139	26.66	133	26.66
Pendigits	2543	24.18	2876	27.10	2760	25.72	495	28.00	453	31.00	332	0.74	**287**	0.28
Satimage	3711	8.26	3432	**5.17**	3510	9.42	1334	7.85	1081	9.02	1256	7.83	**1062**	9.09
Sonar	75	14.09	73	16.19	79	13.19	80	**11.50**	73	15.50	80	12.50	**73**	15.50
Wdbc	75	3.69	70	4.12	79	**2.18**	74	2.80	53	2.50	**48**	2.60	54	2.79

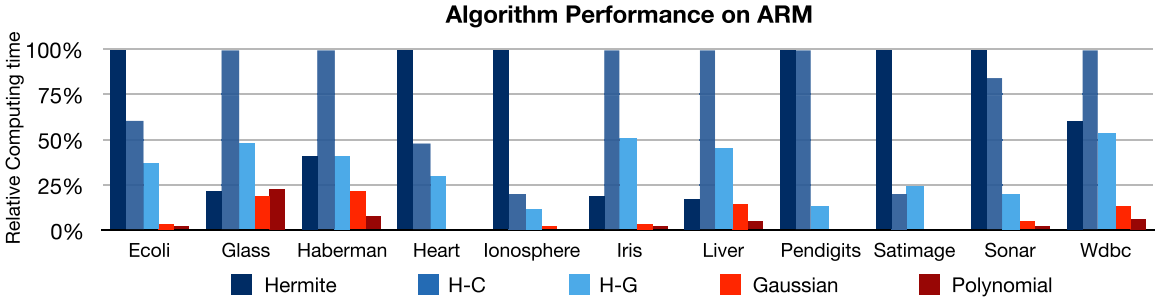


Fig. 2. Computing effort comparison among different kernels for each benchmark.

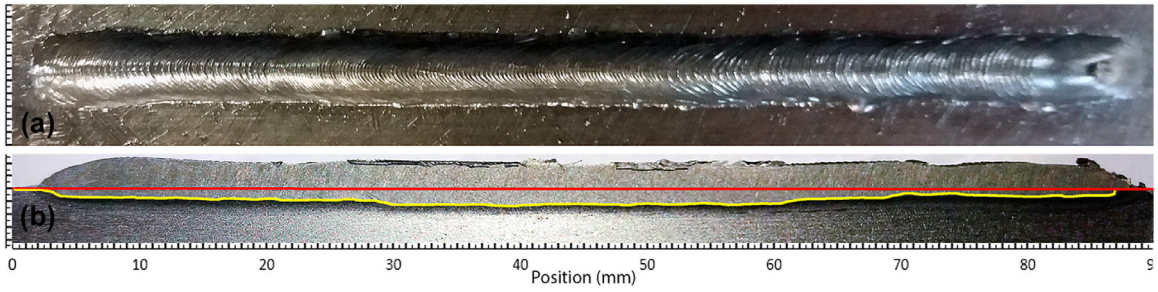


Fig. 3. Weld bead used to obtain the data set for modeling the weld bead depth and validate the weld bead width model[4].

Table 6
Regression benchmarks description.

Benchmark	Features	Number of instances			
		Total	Train	Validation	Test
1027 ESL	4	488	342	73	73
Auto MPG	7	398	278	60	60
Concrete	8	1030	720	155	155
Yacht	6	308	216	46	46
Daily	12	60	42	9	9

Table 7
Regression benchmarks results.

Benchmark	MSE			#SV		
	P	G	H	P	G	H
1027 ESL	0.3383	0.3408	0.3404	37	11	15
Auto MPG	6.1796	6.1560	6.5162	103	100	100.5
Concrete	32.2274	39.7414	41.7207	477	446	498
Yacht	0.9564	0.5295	5.7285	75.5	161.5	76
Daily	0	0.1250	0	39	38	38

$\hat{DW}(nT - T)$. The T symbol is the sample time and n is the sample number [4]. Both weld bead depth and width SVR models were generated for the polynomial, Gaussian, and Hermitian kernels. APMT-MODE was used to configure the SVR hyperparameters, being the results shown in Table 9.

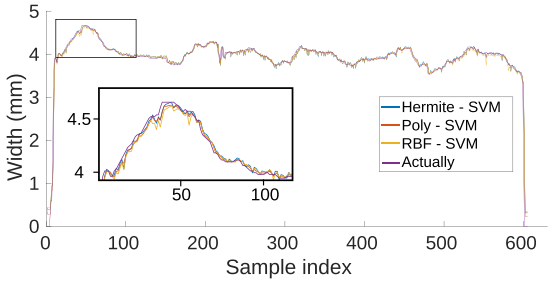
Models with polynomial (P), Gaussian (G), and Hermitian (H) kernels achieved an MSE for the closed-loop of 3.3, 3.5, and 3.8 times smaller than the ANN used in [4] respectively. In the open-loop, the MSE of models generated by APMT-MODE was approximately 4 times lower than the ANN. These results highlight the efficiency of the APMT-MODE in finding quality SVR models for complex practical problems. We can also highlight the flexibility of APMT-MODE to get good models for different kernel types. In this

Table 8
Average Rankings of the kernels.

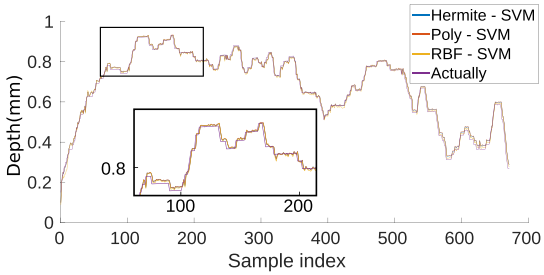
Kernel	Ranking(#SV)	Ranking(MSE)
Gaussian	1.5	2.0
Hermite	2.1	2.6
Polynomial	2.4	1.4

Table 9
The SVR weld bead estimation results and ANN gathered from [4].

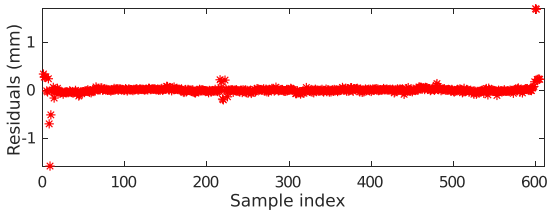
Metric	Depth				Width			
	P	G	H	ANN	P	G	H	ANN
Fit R^2	9.96E-1	9.95E-1	9.96E-1	9.90E-1	9.76E-1	9.75E-1	9.74E-1	8.51E-1
Open-loop	2.73E-4	2.53E-4	2.60E-4	8.07E-4	1.72E-2	1.87E-2	1.91E-2	1.28E-1
Closed-loop	6.67E-2	6.36E-2	5.75E-2	2.24E-1	2.38E-1	2.27E-1	2.18E-1	6.70E-1



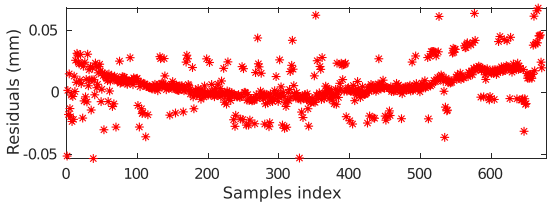
(a) Width weld models predictions.



(b) Depth weld models predictions.



(c) Residuals for width using poly-kernel.



(d) Residuals for depth using poly-kernel.

Fig. 4. Plots of measurements and estimations obtained by SVR models of the width and depth of the weld bead.

specific application, there is no evidence of a statistical difference between kernels. However, in a hardware architecture, the tuned kernels may perform differently due to their specific mathematical characteristics.

Fig. 4 plots the weld bead depth and width estimations for the polynomial, Gaussian, and Hermite kernels. It is also observed that even for the test set, the residuals plot for the width estimation does not show any pattern for the three kernels (e.g., Fig. 4(c)). Whereas for the estimation of weld bead depth, there is still a parabolic pattern, repeating the same pattern for all kernels (e.g., Figs. 4(d)).

6. Conclusions

The APMT-MODE metaheuristic has being evaluated using UCI benchmarks for classification and regression problems. Classification results show that, in most cases, the solutions had lower errors and a smaller number of support vectors while compared to kernels employed in Moghaddam and Hamidzadeh work [24]. Furthermore, all solutions presented have at least four times less computational effort.

We can conclude that although statistically, the average errors of the models generated by the APMT-MODE are equal, they have a higher generalization capability. Finally, a case study of a welding bead geometry predictor based on an SVR is developed and compared to a reference design that uses a Perceptron NN presented by Bestard and Sampaio in [4]. The SVR models achieved by APMT-MODE yield better performance for all chosen kernels. Future works include developing a metric that allows comparing the complexity of models with different kernels and adjusting the metaheuristic algorithm to find not only the hyperparameters but also to indicate the best kernel type for each problem.

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] Q. Fan, Y. Zhang, Self-adaptive differential evolution algorithm with crossover strategies adaptation and its application in parameter estimation, Chemometrics and Intelligent Laboratory Systems 151 (2016) 164–171.
[2] C.A.C. Coello, G.B. Lamont, D.A.V. Veldhuizen, Evolutionary Algorithms for Solving Multi-objective Problems, Springer Science, USA, 2007.
[3] J.E. Volder, The CORDIC trigonometric computing technique, IRE Trans. Electron. Comput. EC-8 (3) (1959) 330–334.
[4] G.A. Bestard, R.C. Sampaio, J.A.R. Vargas, S.C.A. Alfaro, Sensor fusion to estimate the depth and width of the weld bead in real time in GMAW processes, Sensors 18 (4) (2018) 962.
[5] G.A. Bestard, S.C.A. Alfaro, Measurement and estimation of the weld bead geometry in arc welding processes: the last fifty years of development, Journal of the Brazilian Society of Mechanical Sciences and Engineering 40 (4) (2018).
[6] W. Zhang, J. Wang, J. Wang, Z. Zhao, M. Tian, Short-term wind speed forecasting based on a hybrid model, Applied Soft Computing Journal 13 (7) (2017) 3225–3233.
[7] A. Tharwat, Y.S. Moemen, A.E. Hassanien, Classification of toxicity effects of biotransformed hepatic drugs using whale optimized support vector machines, J Biomed Inform 68 (2017) 132–149.
[8] F. Ye, X.Y. Lou, L.F. Sun, An improved chaotic fruit fly optimization based on a mutation strategy for simultaneous feature selection and parameter optimization for SVM and its applications, PLoS ONE 12 (4) (2017) 1–36.
[9] A. Subasi, J. Kevric, M.A. Canbaz, Epileptic seizure detection using hybrid machine learning methods, Neural Computing and Applications 31 (1) (2019) 317–325.
[10] H. Faris, M.A. Hassonah, A.-Z. AlaM, S. Mirjalili, I. Aljarah, A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture, Neural Computing and Applications 30 (8) (2018) 2355–2369.
[11] A. Al Shorman, H. Faris, I. Aljarah, Unsupervised intelligent system based on one class support vector machine and grey wolf optimization for IoT botnet detection, J Ambient Intell Humaniz Comput (2019) 1–17.

[12] D.J. Kalita, S. Singh, SVM Hyper-parameters optimization using quantized multi-PSO in dynamic environment, Soft comput 24 (2) (2020) 1225–1241.
[13] G.N. Kouziokas, A new W-SVM Kernel combining PSO-neural network transformed vector and bayesian optimized SVM in GDP forecasting, Eng Appl Artif Intell 92 (2020) 103650.
[14] R. Laref, E. Losson, A. Sava, M. Siadat, On the optimization of the support vector machine regression hyperparameters setting for gas sensors array applications, Chemometrics and Intelligent Laboratory Systems 184 (2019) 22–27.
[15] C. Igel, Multi-objective model selection for support vector machines, Lect. Notes Comput. Sci. 3410 (2005) 534–546.
[16] J. Salmen, T. Suttorp, J. Edelbrunner, C. Igel, Evolutionary Optimization of Wavelet Feature Sets for Real-Time Pedestrian Classification, in: 7th International Conference on Hybrid Intelligent Systems (HIS 2007), IEEE, Kaiserlautern, Germany, 2007, pp. 222–227.
[17] C. Chatelain, S. Adam, Y. Lecourtier, L. Heutte, T. Paquet, Multi-Objective Optimization for SVM Model Selection, in: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 1, 2007, pp. 427–431. Parana, Brazil
[18] W. Li, L. Liu, W. Gong, Multi-objective uniform design as a SVM model selection tool for face recognition, Expert Syst Appl 38 (6) (2011) 6689–6695.
[19] Z. Zhao, J. Yang, Z. Hu, H. Che, A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric latin hypercube design for unconstrained optimization problems, Eur J Oper Res 250 (1) (2016) 30–45.
[20] P.B. Miranda, R.B. Prudencio, A.C.P. De Carvalho, C. Soares, Multi-objective Optimization and Meta-learning for SVM Parameter Selection, in: The 2012 International Joint Conference on Neural Networks (IJCNN), IEEE, Brisbane, Australia, 2012, pp. 1–8.
[21] S. Xiong, C. Wang, Z. Fang, D. Ma, Multi-Step-Ahead carbon price forecasting based on variational mode decomposition and fast multi-Output relevance vector regression optimized by the multi-Objective whale optimization algorithm, Energies 12 (2019) 147.
[22] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Min Knowl Discov 2 (2) (1998) 121–167.
[23] H. Drucker, C.J. Burges, L. Kaufman, A. Smola, V. Vapnik, et al., Support vector regression machines, Adv Neural Inf Process Syst 9 (1997) 155–161.
[24] V.H. Moghaddam, J. Hamidzadeh, New hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier, Pattern Recognit 60 (2016) 921–935.
[25] P.B. Miranda, R.B. Prudêncio, Active testing for svm parameter selection, in: The 2013 International Joint Conference on Neural Networks (IJCNN), IEEE, Dallas, Texas, USA, 2013, pp. 1–8.
[26] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[27] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P.N. Suganthan, C.A.C. Coello, F. Herrera, Bio-inspired computation: where we stand and what's next, Swarm Evol Comput 48 (2019) 220–250.
[28] C.-C. Chang, C.-J. Lin, LIBSVM: A Library For support vector machines, ACM Trans Intell Syst Technol 2 (2011) 27:1–27:27.
[29] P. Mittal, Knowledge extraction based on evolutionary learning (KEEL): analysis of development method, genetic fuzzy system, International Journal of Computer Applications and Information Technology (2012) 1–4.

Carlos Eduardo Santos is a Professor in the Federal Institute of Science, Education and Technology of Tocantins (IFTO), Brazil. His primary research interest is in the application of machine learning to system recognition, selection parameter problem of support verctor machines, metaheuristics algorithms and multiobjective optimization problem. He received his BSc in mathematics from the University of Rio Verde, Rio Verde, Brazil (2001), MSc in electrical engineering from the Unesp, São Paulo and Ph.D. degree in Mechatronics Systems from Universidad de Brasília (UnB), Brazil, 2019.

Renato Sampaio is a professor in Faculdade do Gama (FGA) at Universidade de Brasilia (UnB), Brazil. His research interests include digital design, reconfigurable hardware architectures, embedded systems engineering, applied robotics and predictive control. He has a Ph.D. degree in Mechatronic Systems from University of Brasilia, Brazil, 2018.

Leandro dos Santos Coelho is a professor of Pontifícia Universidade Catlica do Paraná (PUCPR) and Universidade Federal de Parana (UFPR). His research interests include soft-computing, Optimizaton, System Identification, Control, and Machine Learning. He received his Ph.D. degree in Electrical Engineering from Universidade Federal de Santa Catarina, Brazil, 2000. He is a Member of the IEEE.

Guillermo Alvarez Bestarsd is a professor in Faculdade do Gama (FGA) at Universidade de Brasilia (UnB), Brazil. His research interests include automation and control system design, welding process, and digital design. He has a PhD degree in Mechatronic Systems from University of Brasilia, Brazil, 2018.

Carlos Humberto Llanos is a professor in the Mechanical Department, University of Brasilia (UnB), Brazil. His research interests include Reconfigurable Computing, Embedded Systems Design, Machine Learning, Optimization, and Automation/Control Design in Embedded System Platforms. He received his Ph.D. degree in Electrical Engineering from Escola Politécnica da Universidade de São Paulo (EPUSP), Brazil, 1998. He is a Member of the IEEE.