SVM Model Selection Using PSO for Learning Handwritten Arabic Characters

Mamouni El Mamoun[1, *], Zennaki Mahmoud[1] and Sadouni Kaddour[1]

Abstract: Using Support Vector Machine (SVM) requires the selection of several parameters such as multi-class strategy type (one-against-all or one-against-one), the regularization parameter C, kernel function and their parameters. The choice of these parameters has a great influence on the performance of the final classifier. This paper considers the grid search method and the particle swarm optimization (PSO) technique that have allowed to quickly select and scan a large space of SVM parameters. A comparative study of the SVM models is also presented to examine the convergence speed and the results of each model. SVM is applied to handwritten Arabic characters learning, with a database containing 4840 Arabic characters in their different positions (isolated, beginning, middle and end). Some very promising results have been achieved.

Keywords: SVM, PSO, handwritten Arabic, grid search, character recognition.

1 Introduction

Research on Arabic characters recognition reveals a rapidly expanding field and is now a concern whose relevance is undisputed by the research community, which has devoted its efforts to reducing constraints and expanding the field of Arabic character recognition.

Among the techniques used for Arabic handwriting recognition is the SVM introduced in the early 1990s by Boser et al. [Boser, Guyon and Vapnik (1992); Cortes and Vapnik (1995)], which has been very successful in many areas of machine learning. Today, it can be said without exaggeration that these machines have replaced neural networks and other learning techniques.

The adjustment of the hyper-parameters of the SVM classifier is a crucial step in building an effective recognition system. For a long time, the model selection was carried out by a "grid search" method, where a systematic search is implemented by discretizing the parameter space using a fixed step [Xiao, Ren, Lei et al. (2014); Wojciech, Sabina and Andrzej (2015)].

More recently, model selection has been considered as an optimization task. In this context, an optimization algorithm is implemented in order to find all the hyper-parameters that achieve the best classification performance. Among the existing

[1] Département Informatique Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf USTO-MB, BP 1505 El M'naoeur, 31000, Oran, Algérie.

[*] Corresponding Author: Mamouni El Mamoun. Email: elmamoun.mamouni@univ-usto.dz.

CMC. doi:10.32604/cmc.2019.08081 www.techscience.com/cmc

optimization algorithms, the gradient descent method has often been used for SVM model selection [Ayat, Cheriet and Suen (2005); Jiang and Siddiqui (2019)].

Metaheuristic techniques were also used for SVM model selection. Genetic algorithms [Sun, Guo, Wang et al. (2017); Phan, Nguyen and Bui (2017)], evolutionary strategies [Liu, Liu, Yang et al. (2006); Phienthrakul and Kijsirikul (2010)] and taboo search metaheuristic [Zennaki, Mamouni and Sadouni (2013); Corazza, Di Martino, Ferrucci et al. (2013)] were used to find the best configuration of SVM parameters.

In this work, the PSO technique was adapted for parameter selection in order to maximize the cross validation accuracy and a comparative study between different SVM models is presented.

The rest of the article is organized as follows. Section 2 presents the SVM and the two multi-class approaches one-against-all and one-against-one. The PSO method is described in Section 3. Section 4 provides a brief description of the proposed recognition system. Section 5 describes the experimental results. Finally, Section 6 draws the conclusions of this study.

2 Support vector machine

Originally, SVM processes the binary classification (two classes). Considering the learning base S composed of input vectors x_i, the classification of these vectors is known in advance. It is represented by the output vector: $y_i = \{-1,1\}$. It is therefore sufficient to know the sign of the classifier to determine the class of the example. If S is of dimension *m*, then the output value of binary classifier is given by:

$$h(x) = sign(\sum_{i=1}^{m} \alpha_i y_i K(x, x_i) + b) \qquad (1)$$

and $\forall (x_i, y_i) \in S, 0 \leq \alpha_i \leq C \; et \; \sum_i y_i \alpha_i = 0$

where, *K* is the kernel function, C the coefficient of regularization and α_i the coefficients of Lagrange.

The learning algorithm for SVM aims to find the hyperplane of maximum geometric margin that separates the data in variables space as shown in Fig. 1. Vapnik [Vapnik (1998)] was the first to introduce hyperplane concepts into support vector algorithms.

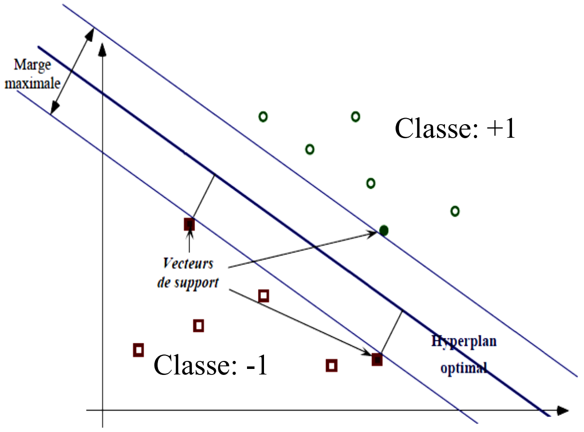


Figure 1: Representation of the hyperplane separating the data in the variables space

To determine the hyperplane equation, the problem is modeled as a mathematical program that maximizes the geometric margin between the data, taking into account the correct classification of the training set.

The effectiveness of SVM algorithm is because it combines two relevant ideas. The first is the change of landmark and input variables to another feature space. This double change simplifies the construction of nonlinear classifiers by using only the hyperplanes in the feature space.

The second consists in constructing separating hyperplanes in the feature space with the widest geometric margin possible [Vapnik (1998)]. On the other hand, the SVM approach is based on a statistical foundation, a theory that easily justifies its statements.

The choice of the kernel function k is very important; it must respect certain conditions and correspond to a scalar product in a high dimensional space. The conditions that *K* must satisfy to be a kernel function are as follows: It must be symmetric and positive-semi definite.

The simplest kernel function is the linear kernel:

$$K\left(x_i, x_j\right) = x_i . x_j \quad (2)$$

Thus, in this case the linear classifier is used without changing the space. The kernel approach generalizes the linear approach. The linear kernel is sometimes used to evaluate the difficulty of a problem.

The kernels commonly used with SVM are expressed as follows:

Polynomial

$$K\left(x_i, x_j\right) = (\gamma\, x_i . x_j + coef)^d \quad (3)$$

Gaussian (RBF)

$$K\left(x_i, x_j\right) = exp^{-\gamma \|x_i - x_j\|^2} \quad (4)$$

Laplacian

$$K\left(x_i, x_j\right) = exp^{-\sqrt{\gamma}\|x_i - x_j\|} \quad (5)$$

Sigmoid

$$K\left(x_i, x_j\right) = tanh(\gamma(x_i . x_j) + coef) \quad (6)$$

Multi-class extensions

SVM is binary in their origin. However, real-world problems are in most cases multiclass. Therefore, multiclass SVM reduce the problem to a composition of several two-class hyperplanes to draw the decision boundaries between the different classes.

The principle is to decompose the examples into several subsets; each of them represents a binary classification problem. A separating hyperplane is determined for each problem by the binary SVM classifier. There are several decomposition methods in the literature, the most commonly used are:

2.1 One-against-all

This is the simplest and oldest method. According to Vapnik's formulation [Vapnik (1998)], it is a question of determining for each class k a hyperplane $H_k\ (w_k, b_k)$ separating it from all other classes. This class k is considered as the positive class (+1) and the other classes as the negative class (-1) so for a problem of K classes, K SVM binary is obtained. Fig. 2 shows a case of separation of three classes.

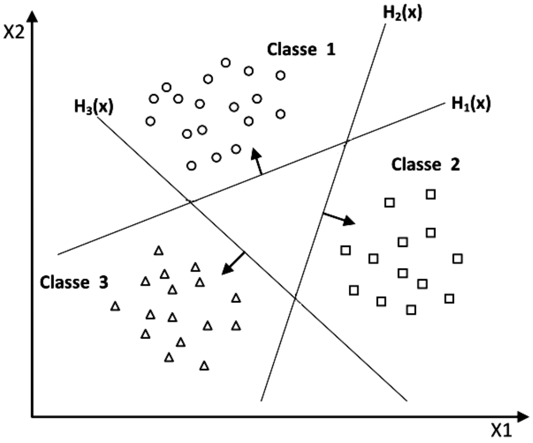


Figure 2: Approach One-against-all

2.2 One-against-one

This approach consists in using a classifier for each two classes. This method discriminates each class of every other class, thus K(K-1)/2 decision functions are learned.

For each pair of classes (k, s), this method defines a binary decision function. The assignment of a new example is done by voting list. An example is tested by calculating its decision function for each hyperplane. For each test, there is a vote for the class to which the example belongs (winning class).

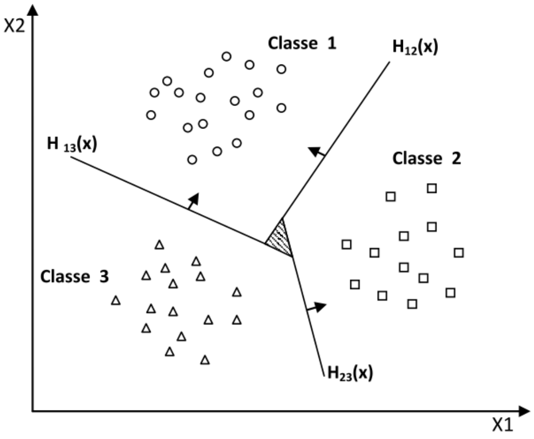


Figure 3: Approach One-against-one

3 Particle swarm optimization (PSO)

PSO is a stochastic optimization method developed by Eberhart and Kennedy in 1995 [Kennedy and Eberhart (1995)]. Originally inspired by the world of life, more specifically the social behavior of swarming animals, such as schools of fish and flocks of birds [Li and Li (2014)]. This method is based on a set of individuals called particles, originally arranged randomly, which can move in the search space. Each particle represents a solution to the problem and has a position X_i and velocity V_i. In addition, each particle has a memory that contains its best position visited $\overrightarrow{P_i}$ and the best G position among the positions of all particles. The evolution of the algorithm equations is given as follows:

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{7}$$

$$V_i(t+1) = \omega\, V_i(t) + C_1R_1(P_i - X_i) + C_2R_2(G - X_i) \tag{8}$$

where, ω represents the coefficient of inertia, the coefficients $C1$ and $C2$ are constants defined empirically according to the relation ship $C_1 + C_2 \leq 4$ and finally, $R1$ and $R2$ are random positive numbers that follow a uniform distribution over [0,1] [Kennedy and Eberhart (1995)].

The strategy of moving a particle is influenced by the following three components:

1. Inertia ($\omega\, V_i(t)$): the particle tends to follow its current direction;

2. Cognitive component ($C_1R_1(P_i - X_i)$): The particle tends to move towards the best position already visited;

3. Social component ($C_2R_2(G - X_i)$): The particle tends to move towards the best position already reached by all the particles of swarm.

PSO for SVM model selection

The number of SVM model parameters depends on the type of kernel and their parameters. For example, the polynomial kernel has three parameters ($\gamma, coef\ and\ d$), therefore, the SVM model has four parameters if the regulation parameter C is added. In this case the search space is of dimension D=4.

The PSO algorithm maintains a set of particles. Each particle represents a candidate solution to the studied problem and considered as an object that has the following characteristics.

- The current position is a vector $X = (x_1, .., x_D)$ where, $x_1, .., x_D$ represents the values of SVM model parameters and D the dimension of the search space.
- The current velocity of the particle $V = (v_1, .., v_D)$ is used to gradually modify the values of SVM model parameters according to Eqs. (7) and (8).
- The best position visited of the particle $P = (p_1, .., p_D)$ is used to store the best parameter values found by the particle (a previous value of X).
- The best position among all particle positions $G = (g_1, .., g_D)$ represents the best values of SVM parameters found so far.

The algorithm is expressed as follows:

Initialize the number of particles N.
Initialize velocity V_i and randomly initialize the position X_i of each particle.
Set the maximum number of iterations T_{max}.

```
t = 0
Repeat
  For i = 1 To N
    Calculate the cross validation accuracy of SVM with parameters $x_i$
    If accuracy of SVM with parameters $x_i$ greater then accuracy of SVM with parameters $p_i$ Then
    Update $p_i = x_i$
    If accuracy of SVM with parameters $p_i$ greater then accuracy of SVM with parameters $g$ Then
    Update $g = p_i$
  End for
    For i = 1 To N
      Update $x_i$ using Eqs. (7) and (8)
    End for
  t = t + 1
Until ( $t > T_{max}$)
Return $g$
```

4 Recognition system of handwritten Arabic characters

In the context of this study, a character recognition system has been developed. This section provides a brief description of the database and the technique used to extract the features.

A consistent database was built in SIMPA laboratory with the contribution of several researchers and students, containing 4840 examples of handwritten Arabic characters, as shown in Fig. 4. The letters are in different positions (isolated, beginning, middle and end), for isolated letters there are 100×28=2800 images and for others there are 30×68=2040 images. The number of classes is 28+68=96.

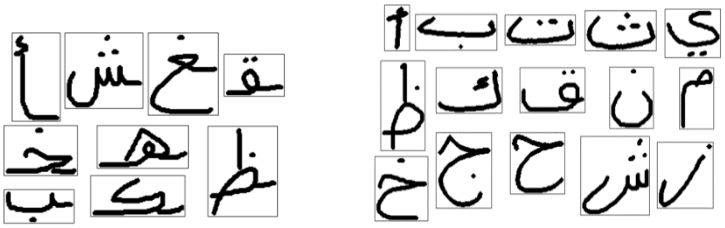
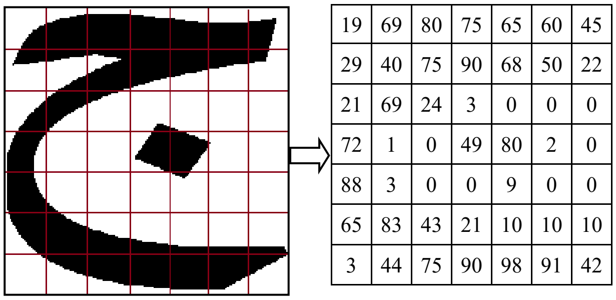


Figure 4: Samples from our database

For feature extraction, the image size was normalized to 70×70 and then the zoning technique [Anitha Mary and Dhanya (2015); Dinesh and Sabenian (2017)] was implemented. The principle is to divide the image into 49 zones or 10×10 pixel size blocks, and then count the number of black pixels in each zone. Fig. 5 shows, for example, the letter 'jim', which is written in Arabic 'ج' in its isolated form.



19	69	80	75	65	60	45
29	40	75	90	68	50	22
21	69	24	3	0	0	0
72	1	0	49	80	2	0
88	3	0	0	9	0	0
65	83	43	21	10	10	10
3	44	75	90	98	91	42

Figure 5: Feature extraction of letter "jim"

5 Implementation and results

In the framework of this research, an open-source SVM engine developed by Thorsten Joachims in 2008 was used. It is available at Joachims [Joachims (2008)], with documentation, examples and bibliographic references.

The selection of parameters is typically performed by minimizing the generalization error, so we used cross-validation (k-fold), this method consists of dividing the learning set into k disjoint subsets of the same size, then learn about the k-1 subsets, and test on the kth part this process is repeated k times, the cross validation accuracy is obtained by calculating the average of k previous accuracy. In these experiments we used k=5 (5-fold).

First, the Grid Search method for SVM model selection was used to compare the results obtained with the PSO method. Since the Grid search is only used for selection of models that use few parameters (≤ 2), it was implemented with RBF and Laplacian kernel. There are two parameters, C and γ; the range of C used is $[10^0, 10^1, \dots, 10^4, 10^{15}]$ and the range of γ is $[10^{-15}, 10^{-14}, \dots, 10^1, 10^2]$, for 16x18=288 iterations.

The results obtained for one-against-one approach are illustrated as follows:

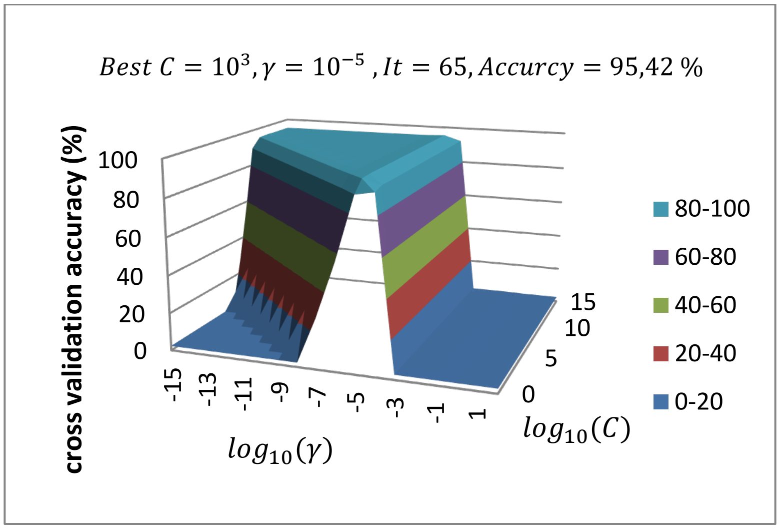


Figure 6: Grid search results for RBF kernel for one-against-one approach

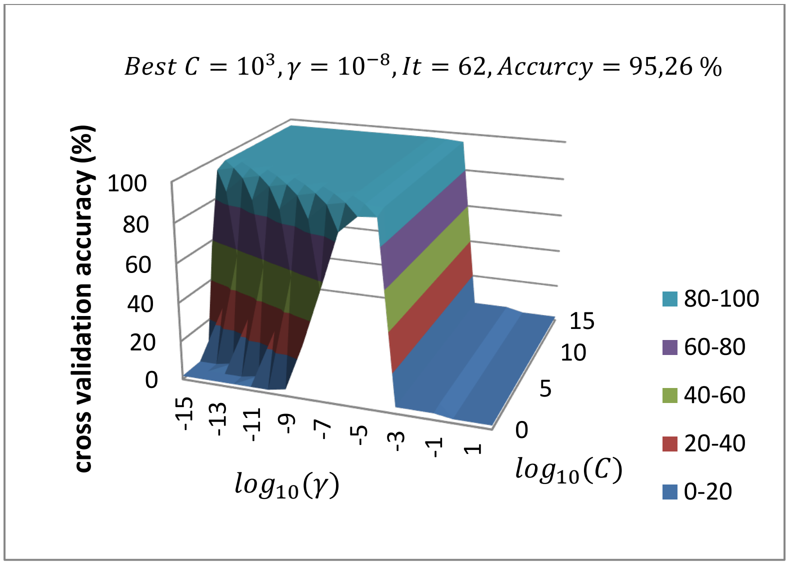


Figure 7: Grid search results for Laplacian kernel for one-against-one approach

For one-against-all approach, the results obtained are illustrated as follows:

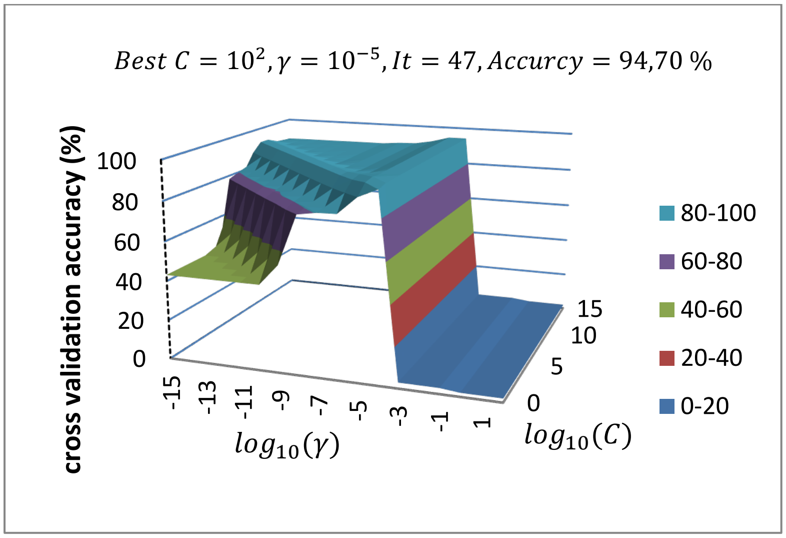


Figure 8: Grid search results for RBF kernel for one-against-all approach

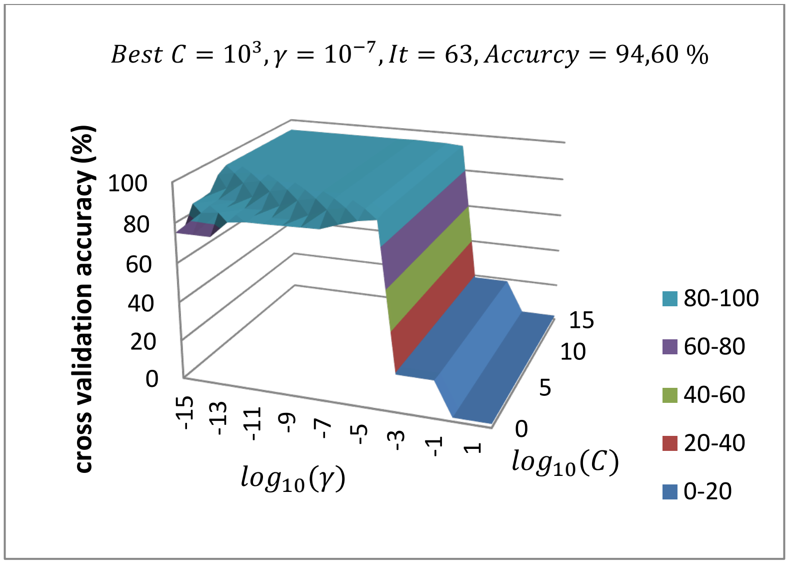


Figure 9: Grid search results for Laplacian kernel for one-against-all approach

According to these results, the surface area of the good results (Accuracy >90%) of Laplacian kernel is large compared to the RBF kernel, and the latter gave the best results. The same can be said about the two SVM approaches used, the surface of the good results of the one-against-all approach is large compared to the one-against-one, and the results of the latter are the best.

During these experiments, the CPU time (duration) was measured at each iteration, the results obtained are presented in Fig. 10. The results show that the one-against-one approach (1×1) is faster than the other approach (1×N) and the Laplacian kernel is faster than the RBF. The one-against-all approach becomes very slow when the γ parameter takes large values such as 1, 10 and 100. This is evident in Fig. 10 the iteration 18 and their multiples.

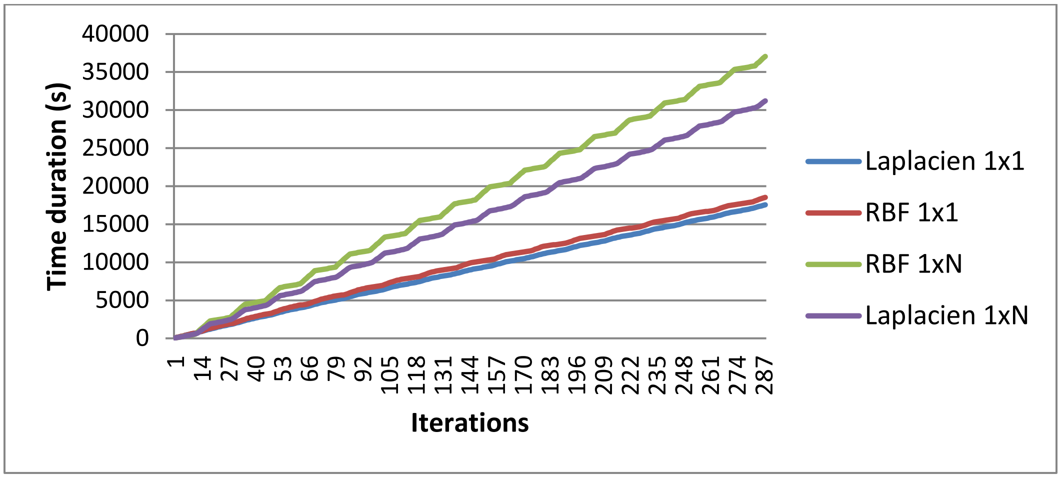


Figure 10: Time duration in seconds of each SVM model

Then, for the selection of SVM model parameters, the PSO method was used, in particular the gbest model [Frans (2006)]. For these experiments, 5 particles with coefficient values were used: $\omega = 0.7298$ and $C_1 = C_2 = 1.4962$, to ensure convergence [Dang and Luong (2011)].

Each particle encodes the SVM model parameters; the number of these parameters is varied according to the type of kernel. The values considered are $C \in [10^0, 10^{15}]$, $\gamma \in [10^{-15}, 10^2]$, $coef \in [-10, 10^2]$ $and\ d \in [1,10]$. If a particle exceeds the search space boundary, this particle will be repositioned at the boundary and its velocity is set to zero. The results obtained for one-against-one approach are expressed as follows:

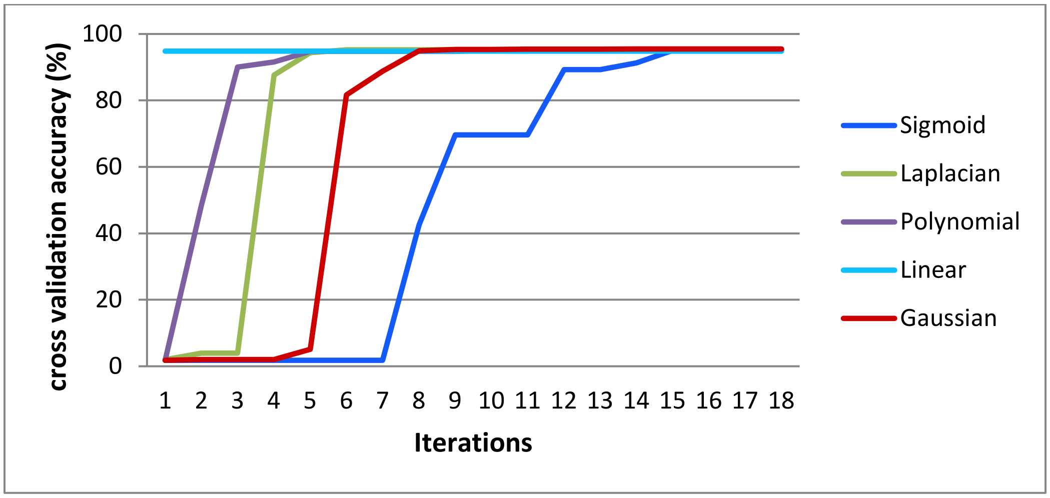


Figure 11: Comparison results of different kernels for one-against-one approach

Table 1: Best results obtained for each kernel for one-against-one approach

	Iterations	C	γ	D	Coef	Accuracy
Gaussian (RBF)	14	117405	1,75E-05	/	/	95,49
Sigmoid	17	65975	9,08E-08	/	-1,97	95,18
Laplacian	12	82771	1,10E-11	/	/	95,26
Polynomial	12	172320	1,02E-10	2	11	95,20
Linear	1	13692	/	/	/	94,83

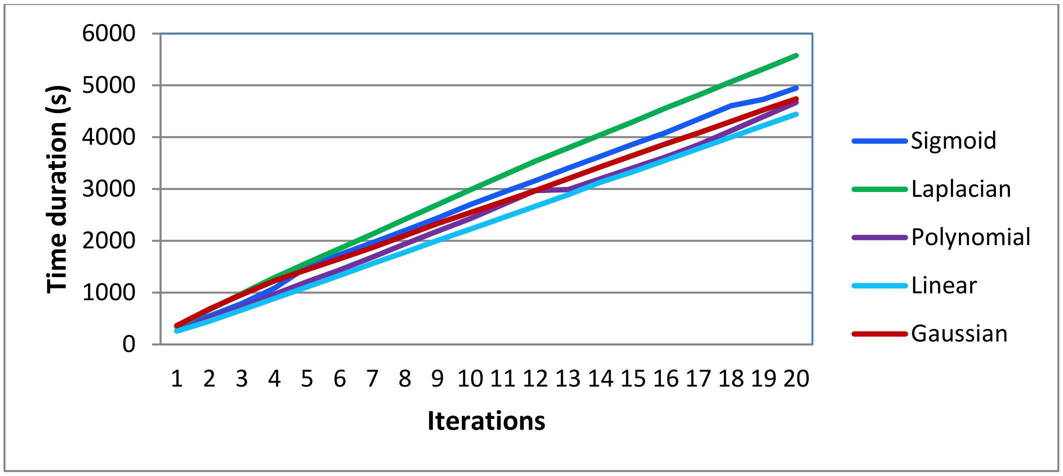


Figure 12: Time duration in seconds of each kernel for one-against-one approach

For one-against-all approach, the results obtained are expressed as follows:

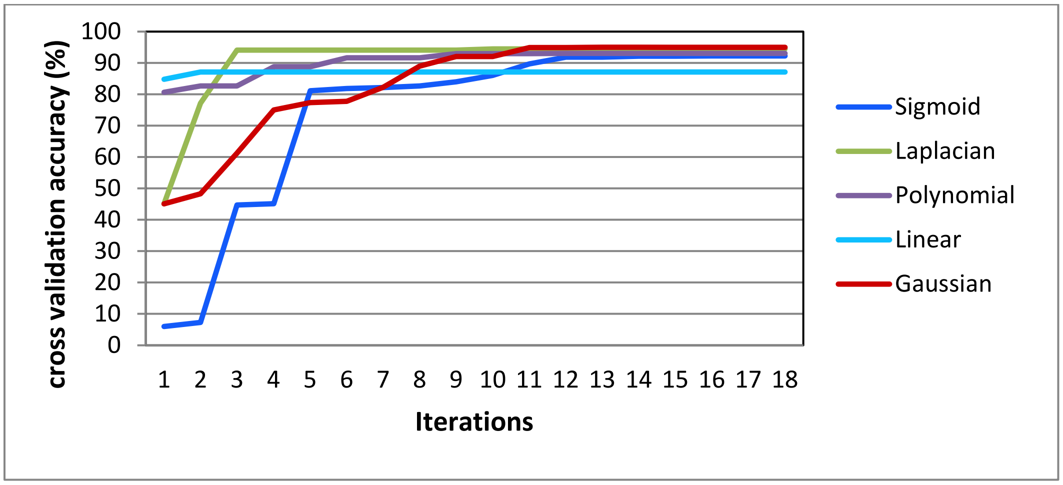


Figure 13: Comparison results of different kernels for one-against-all approach

Table 2: The best results obtained for each kernel in one-against-all approach

	Iterations	C	γ	D	Coef	Accuracy
Gaussian (RBF)	15	46651	4,27E-05	/	/	94,99
Sigmoid	16	85617	1,20E-07	/	-2,03	92,24
Laplacian	10	67577	3,40E-08	/	/	94,48
Polynomial	12	13863	4,70E-03	2	1	93,06
Linear	2	2310	/	/	/	87,12

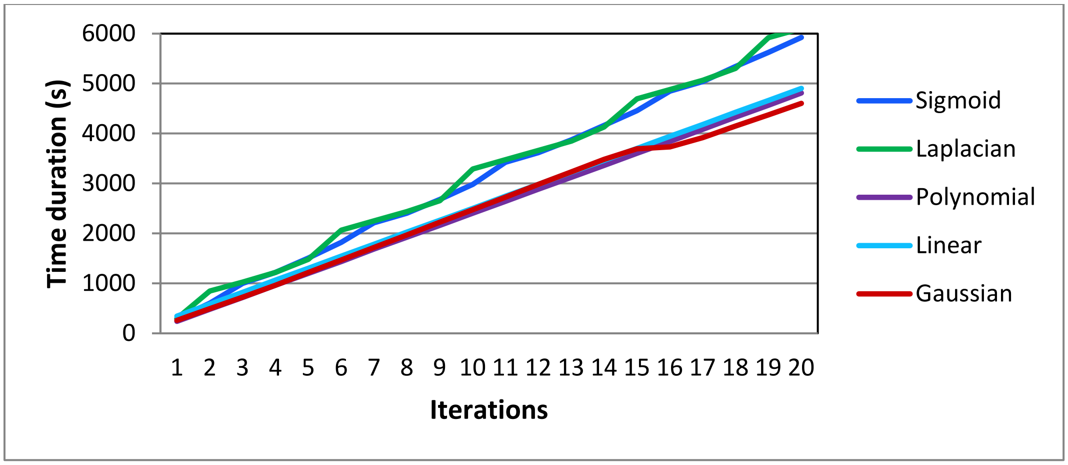


Figure 14: Time duration in seconds of each kernel for one-against-all approach

According to the results obtained, the linear kernel converges rapidly towards the optimum, because the SVM model has only one parameter (C), and the value of this parameter does not significantly affect the model's behavior. On the other hand, the sigmoid kernel is the slowest because in this case, there are three parameters and the model is very sensitive to the change of these parameters. The best result in terms of recognition rate is for the RBF kernel in one-against-one approach.

Finally, it can be noted that Grid Search is an effective technique to provide an overview of the search space (allows extracting promising regions), so to find good results it is necessary to use a local search or to launch the Grid Search a second time in the promising area. This method requires a lot of time to explore the search space. On the other hand, the PSO produces good results quickly.

6 Conclusions

Based on the experimental results obtained, the use of SVM approach for Arabic character recognition is strongly recommended due to its superior generalization ability to classify high-dimensional data, even when there is a large number of classes (in this case study: 96 classes).

It is also important to note that the RBF kernel is the most suitable for the recognition of Arabic handwritten characters. Indeed this kernel has better results than the other kernels with a cross validation accuracy of 95, 49% (SVM one-against-one, $\gamma = 1{,}75\mathrm{E} - 05$). As a result, the PSO is more effective than Grid search in selecting SVM models.

References

Anitha Mary, M. O. C.; Dhanya, P. M. (2015): A comparative study of different feature extraction techniques for offline malayalam character recognition. *Computational Intelligence in Data Mining*, vol. 2, pp. 9-18.

Ayat, N. E.; Cheriet, M.; Suen, C. Y. (2005): Automatic model selection for the optimization of SVM kernels. *Pattern Recognition*, vol. 38, no. 10, pp. 1733-1745.

Boser, B.; Guyon, I.; Vapnik, V. (1992): A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144-152.

Corazza, A.; Di Martino, S.; Ferrucci, F.; Gravino, C.; Sarro, F. et al. (2013): Using tabu search to configure support vector regression for effort estimation. *Empirical Software Engineering*, vol. 18, no. 3, pp. 506-546.

Cortes, C.; Vapnik, V. (1995): Support-vector networks. *Machine Learning*, vol. 20, no. 3, pp. 273-297.

Dang, H. N.; Luong, C. M. (2011): A new model of particle swarm optimization for model selection of support vector machine. *New Challenges for Intelligent Information and Database Systems*, pp. 167-173.

Dinesh, P. M.; Sabenian, R. S. (2017): Comparative analysis of zoning approaches for recognition of Indo Aryan language using SVM classifier. *Cluster Computing*, pp. 1-8.

Frans, V. (2006): *An Analysis of Particle Swarm Optimizers (Ph.D. Thesis)*. University of Pretoria, Afrique du Sud.

Jiang, W.; Siddiqui, S. (2019): Hyper-parameter optimization for support vector machines using stochastic gradient descent and dual coordinate descent. *EURO Journal on Computational Optimization*, vol. 7, no. 1, pp. 1-17.

Joachims, T. (2008): *SVMLight: Support Vector Machine*. Cornell University.

http://svmlight.joachims.org/.

Kennedy, J.; Eberhart, R. (1995): Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942-1948.

Li, J.; Li, B. (2014): Parameters selection for support vector machine based on particle swarm optimization. *International Conference on Intelligent Computing*, pp. 41-47.

Liu, R.; Liu, E.; Yang, J.; Li, M.; Wang, F. (2006): Optimizing the hyper-parameters for SVM by combining evolution strategies with a grid search. *Intelligent Control and Automation*, pp. 712-721.

Phan, A. V.; Nguyen, M. L.; Bui, L. T. (2017): Feature weighting and SVM parameters optimization based on genetic algorithms for classification problems. *Applied Intelligence*, vol. 46, no. 2, pp. 455-469.

Phienthrakul, T.; Kijsirikul, B. (2010): Evolutionary strategies for hyperparameters of support vector machines based on multi-scale radial basis function kernels. *Soft Computing*, vol. 14, no. 7, pp. 681-699.

Sun, Y.; Guo, L.; Wang, Y.; Ma, Z.; Jin, S. (2017): The comparison of optimizing SVM by GA and grid search. *Proceedings of the IEEE International Conference on Electronic Measurement & Instruments*, pp. 354-360.

Vapnik, V. (1998): *Statistical Learning Theory*. John Wiley & Sons Inc.

Wojciech, M. C.; Sabina, P.; Andrzej, J. B. (2015): Robust optimization of SVM hyperparameters in the classification of bioactive compounds. *Journal of Cheminformatics*, vol. 7, no. 38, pp. 1-15.

Xiao, T.; Ren, D.; Lei, S.; Zhang, J.; Liu, X. (2014): Based on grid-search and PSO parameter optimization for support vector machine. *Proceedings of the IEEE World Congress on Intelligent Control and Automation*, pp. 1529-1533.

Zennaki, M.; Mamouni, E. M.; Sadouni, K. (2013): A comparative study of SVM models for learning handwritten Arabic characters. *WSEAS Transactions on Advances in Engineering Education*, vol. 10, no. 1, pp. 32-43.