# Hybrid Intelligent Parsimony Search in Small High-dimensional Datasets

Jose Divasón<sup>®</sup>, Alpha Pernia-Espinoza<sup>®</sup>, Ana Romero<sup>®</sup>, and Francisco Javier Martinez-de-Pison<sup>®</sup>

University of La Rioja, Spain {jose.divason,alpha.pernia,ana.romero,fjmartin}@unirioja.es

**Abstract.** The search for machine learning models that generalize well with small high-dimensional datasets is a current challenge. This paper shows a specific hybrid methodology for this kind of problems combining HYB-PARSIMONY and Bayesian Optimization. The methodology proposes to use HYB-PARSIMONY with different random seeds and select those features that had the highest mean probability. Subsequently, with these features, a hyperparameter adjustment is performed with Bayesian Optimization. The results show that the methodology substantially improves the degree of generalization and parsimony of the obtained models compared to previous methods.

Keywords: HYB-PARSIMONY  $\cdot$  small high-dimensional datasets  $\cdot$  parsimonious modeling  $\cdot$  auto machine learning  $\cdot$  PSO-PARSIMONY  $\cdot$  GAPARSIMONY

### 1 Introduction

Obtaining models that generalize well with small high-dimensional datasets (SHDD) is not an easy task. The curse of dimensionality coupled with the low number of instances causes many machine learning algorithms to have trouble describing the underlying structure of the data. A common way to deal with such problems is to use robust validation methods and algorithms that perform well with high-dimensional datasets, such as trees and neural networks. One of the AutoML libraries that gives the best results in this kind of problems is **Autogluon** on tabular data which constructs an ensemble with artificial neural networks and tree-based algorithms such LightGBM, XGBoost, CatBoost, RandomForest, and so on.

However, complex ensemble models that make use of advanced methods may contain biases that are difficult to detect. This is why companies are increasingly demanding explainable models with a small number of input variables, even if their accuracy is not as good as with ensemble models. Thus, a linear model or a decision tree with a few rules can be more useful in many decision-making processes. Even a black box model that has been created with a reduced selection of the original features can be more easily analyzed with current techniques such as ELI5 and SHAP.

This paper presents a new hybrid methodology that combines HYB-PARSIMONY (presented at HAIS 2022) and Bayesian Optimization (BO) for the search of accurate low complexity (parsimonious) models on datasets of small size (hundreds or few thousands of instances) and high dimensionality (tens or hundreds of features).

# 2 Related work

Hyperparameter optimization (HO) and feature selection (FS) are important techniques in machine learning because they can improve the accuracy of predictive models. However, determining the right hyperparameters and the most relevant subset of features can be a complex problem, especially when dealing with high-dimensional datasets.

Current approaches to solving combinatorial problems in machine learning often draw inspiration from nature, particularly from biological systems such as animal herding, bacterial growth, and other natural phenomena. These methods usually involve a population of simple individuals that interact both locally and globally with each other according to simple rules. For example, one such meta-heuristic approach is the Grey Wolf Optimizer (GWO), which was proposed by Mirjalili *et al.* [10] and was inspired by the behavior of grey wolves. The Salp Swarm Algorithm, also proposed by Mirjalili *et al.* in [9], was inspired by the swarming behavior of salps when navigating and foraging in oceans. Other techniques inspired by animals include bats [13], glowworm [8], and bee colony [6] optimization.

Particle Swarm Optimization (PSO) is one of the most commonly used optimization technique. Originally proposed by Kennedy and Eberhart [7], PSO has been the subject of much research, with numerous improvements proposed in terms of topology, parameter selection, and other technical modifications. For example, there are hybridizations of PSO with other meta-heuristic methods, such as the improved binary particle swarm optimization proposed by Chuang *et al.* [3], which uses the 'catfish effect' to introduce new particles into the search space if the best solution does not improve in a certain number of consecutive iterations.

Despite the success of these approaches, there are challenges associated with using meta-heuristic methods to solve combinatorial problems in machine learning. For example, GA-PARSIMONY was proposed in [12,11] to search for parsimonious solutions with genetic algorithms (GA) by performing HO and FS, and was successfully applied in many fields [1,5]. However, in this kind of problems where each solution has a high computational cost, it is not possible to evaluate a large number of individuals in each iteration. This makes GA not as efficient as other optimization techniques where hundreds or thousands of individuals are evaluated. As a continuation of this methodology, the authors used PSO combined with a parsimony criterion to find parsimonious and accurate machine learning models. The main novelty in the PSO-PARSIMONY methodology [2] was that it included a strategy in which the best position of each particle was Algorithm 1 Pseudo-code of the HYB-PARSIMONY algorithm [4]

1:	Initialization of positions $\mathbf{X}^{0}$ using a random and uniformly distributed Latin hy-
	percube within the ranges of feasible values for each input parameter
2:	Initialization of velocities according to $\mathbf{V}^{0} = \frac{random_{LHS}(s, D) - \mathbf{X}^{0}}{2}$
3:	for $t = 1$ to $T$ do
4:	Train each particle $\mathbf{X}_{i}^{t}$ and validate with cross-validation
5:	Fitness evaluation and complexity evaluation of each particle
6:	Update individual best $\hat{\mathbf{X}}_i$ , individual parsimonious best $\hat{\mathbf{X}}_i^p$ and global best $\hat{\hat{\mathbf{X}}}$
7:	if early stopping is satisfied then
8:	return $\hat{\hat{\mathbf{X}}}$
9:	end if
10:	Generation of new neighborhoods if $\hat{\mathbf{X}}$ did not improve
11:	Update each neighborhoods best $\hat{\mathbf{L}}_i$
12:	Select elitist population $P_e$ from for reproduction
13:	Obtain a pcrossover % of worst individuals $P_w$ to be substituted with crossover
14:	Crossover $P_e$ to substitute $P_w$ with new individuals
15:	Update positions and velocities of $P_e$ following the PSO formulas
16:	Mutation of $\%$ of hyperparameters
17:	Mutation of $\%$ of features
18:	Limitation of velocities and out-of-range positions
19:	end for
20:	return global best X

computed considering not only the goodness-of-fit but also the principle of parsimony. The comparison between both methods was performed on 13 public datasets, and the results showed that PSO always improved accuracy over GA, but GA found solutions approximately 10% less complex on datasets with a low number of features.

# 3 The HYB-PARSIMONY method

To combine the strengths of GA-PARSIMONY and PSO-PARSIMONY, the algorithm HYB-PARSIMONY was proposed in [4] by Divasón *et al.* as a hybrid combination that incorporates GA operations (selection, crossover and mutation) and PSO optimization<sup>1</sup> (see Algorithm 1). The methodology improved the search of parsimonious ML models against the other methodologies.

In HYB-PARSIMONY, the following equation was proposed to calculate the percentage of particles to be substituted by GA crossover in each iteration t:

$$pcrossover = max(0.80 \cdot e^{(-\Gamma \cdot t)}, 0.10) \tag{1}$$

<sup>&</sup>lt;sup>1</sup> HYB-PARSIMONY is available for Python at https://github.com/jodivaso/ HYBparsimony



Fig. 1: Example of thirteen curves created with different  $\Gamma$  values to establish the percentage of individuals to be replaced by crossover in each iteration.

Figure 1 shows thirteen curves obtained with different  $\Gamma$  values. In the first iterations, the hybrid method performs the substitution by crossing a high percentage of particles. As the optimization process progresses, the number of substituted particles is reduced exponentially until it ends up fixed at a percentage of 10%. Thus, the hybrid method begins by facilitating the search for parsimonious models using GA-based mechanisms and ends up using more PSO optimization.

#### 3.1 Performance of feature selection in high-dimensional datasets

To analyze the behavior of the hybrid method in high-dimensional datasets as a function of  $\Gamma$ , and with different dimensions and populations, a methodology was implemented with the following experiment's parameters:

- *method*: HYB vs previous methods (PSO or GA).
- nruns: number of runs with different random seeds. Value: 10.
- $\Gamma$  (only for the hybrid method). Values: 0.005, 0.007, 0.010, 0.015, 0.020, 0.030, 0.050, 0.080, 0.130, 0.210, 0.350, 0.560, 1.100, 1.170.
- P: population size. Values:  $[5, 5+1 \cdot 5, 5+2 \cdot 5, ..., 40]$ .
- #feats: dimension of the hypothetical data set. Values: 50, 150, 250, 350.
- $i_{dim}$ : intrinsic dimension that refers to the features,  $F_{selec}$ , with relevant information present in a dataset. That is, the number of input features of the hypothetical model that explains an hypothetical target. Values: 5, 5+1.20, 5+2.20, ...,  $|0.90 \cdot \#feats|$ .
- $-\beta$ : value which balances the weight between recall and precision in the  $F_{beta}$  score used to evaluate each individual (see below). Values: [0.20, 0.20+1.0.06, 0.20+2.0.06, ..., 1.68].

For each combination of experiment's parameters,  $F_{selec}$  were randomly selected according to  $i_{dim}$ . In particular,  $F_{selec}$  corresponded to  $i_{dim}$  random feature positions selected within the range [0, #feats - 1].

To evaluate each solution,  $F_{beta}$  score was used. Based on the F1 score,  $F_{beta}$  is the weighted harmonic mean of precision and recall where  $\beta$  determines the weight between recall and precision in the combined score.  $\beta < 1$  gives more weight to precision, while  $\beta > 1$  favors recall.  $F_{beta}$  is equal to F1 score with  $\beta = 1.0$  and to precision with  $\beta = 0.0$ .

It is defined as:

$$F_{beta} = \frac{(1+\beta^2)(precision \cdot recall)}{\beta^2 \cdot precision + recall}$$
(2)

and:

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

where TP are the correctly chosen features belonging to  $F_{selec}$ , TN the features not chosen and not belonging to  $F_{selec}$ , FP the features chosen but not belonging to  $F_{selec}$ , and FN the features not chosen but belonging to  $F_{selec}$ .



Fig. 2: Distribution of the best  $\beta$  that successfully met the objectives of overcoming a minimum precision and recall defined by  $thr_{pr}$  (values=0.80, 0.85, 0.90, 0.95).

Each combination of [method with  $\Gamma$ , P, #feats,  $i_{dim}$  and  $\beta$ ] was run 10 times with different randoms seeds, a maximum number of iterations of T = 300,  $tol = 10^{-9}$ , and an early stopping of 35.

All experiments<sup>2</sup> were implemented in 2 separately 40-core composed, respectively, of Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz with 128 GB of RAM memory, and Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz with 192 GB of RAM memory.



Fig. 3: Mean of the  $F_{beta}$  (left) and last iteration ( $last_{iter}$ ) (right) achieved with  $\beta = 1.34$  and for each method and P.

Figure 2 shows the number of experiments that successfully met the objectives of overcoming a minimum precision and recall defined by a threshold,  $thr_{pr}$ , and by each  $\beta$  and P values. The distribution of the best  $\beta$  is presented for each experiment and  $thr_{pr}$ . At low  $thr_{pr}$  values, the median of the best  $\beta$  for each combination of [method,  $\Gamma$ , P, #feats,  $i_{dim}$ ] is about 1.3. This indicates that precision tends to be prioritized over recall. Only at  $thr_{pr} = 0.95$  it is observed that the median of the best  $\beta$  is close to 1.0, so the relationship between precision and recall is balanced when the level of demand is very high.

Figure 3 shows respectively the mean of the last iteration  $(last_{iter})$  and the average of  $F_{beta}$  with  $\beta = 1.34$  and for each method and P. GA and HYB with low  $\Gamma$  converge, on average, faster than PSO and HYB with high  $\Gamma$  values, as they reach twice the number of final iterations. With respect to  $F_{beta}$ , the highest averages are obtained with PSO and HYB with  $\Gamma$  values greater than 0.08, but GA has similar performance to HYB 0.08.

However, these results are average values that may be different, for each method, depending on #feats and  $i_{dim}$ . Figure 4 shows the distribution of

 $<sup>^2</sup>$  The total number of experiments was 115170, resulting from all combinations.

7



Fig. 4: Violin plot of  $F_{beta}$  with  $\beta = 1.34$ , #feats = 150 and for each method and four different  $i_{dim}$  values.

 $F_{beta}$  for each method with #feats = 150,  $\beta = 1.34$  and four different  $i_{dim}$  values: 5, 45, 85 and 125. At very low values of  $i_{dim}$ , GA is competitive with the hybrid and PSO methods. However, as the intrinsic dimension is closer to the real dimension of the dataset, PSO and hybrid models with high  $\Gamma$  obtain better accuracy. The problem is that for a particular dataset the intrinsic dimension of the data will be unknown, so it will be necessary to realize an estimation of  $i_{dim}$  in order to select an appropriate method.

In order to have a quick estimate of the hybrid model for  $F_{beta}$  and  $last_{iter}$ , linear *Ridge* models were trained with the previously obtained dataset but eliminating instances corresponding to GA and PSO, and selecting only those cases with a  $\beta$  within the range [0.92, 1.64] where the methodology was most successful. Equations 5 and 6 correspond to the best Ridge models selected with a 10-fold cross-validation RMSE error of 0.0815 and 57.36 with values of the *alpha* Ridge's hyperparameter equal to 4.0 and 2.0, respectively.

 $\hat{F_{beta}} = -0.0462 \cdot \Gamma - 0.0027 \cdot P + 0.0012 \cdot \# feats - 0.0011 \cdot i_{dim} - 0.0108 \cdot \beta - 0.88$ (5)

 $las\hat{t}_{iter} = 28.391 \cdot \Gamma - 0.8883 \cdot P + 0.2963 \cdot \# feats - 0.38 \cdot i_{dim} + 36.517 \cdot \beta + 72.71$ (6)



Fig. 5: Box plots of  $F_{beta}$  (left) and  $last_{iter}$  (right) for  $\Gamma \in [0.0, 0.0 + 0.1, ..., 1.1]$ and four #feats values obtained by simulation with P = 15, and different  $i_{dim}$ and  $\beta$  values.

Using these two models, it was possible to approximately predict the values of  $F_{beta}$  and  $last_{iter}$  for a data set dimension, #feats, and a fixed value of P. Figure 5 shows the box plots for  $F_{beta}$  and  $last_{iter}$  obtained from a simulation performed with multiple combinations of the input values for four highdimensional data sets: ailerons (#feats = 40), crime (#feats = 127), blog (#feats = 276) and slice (#feats = 378). Each simulation was performed by fixing P = 15 and for each #ncols, with  $\Gamma \in [0.0, 0.0 + 0.1, ..., 1.1]$ ,  $i_{dim} \in [0.10 \cdot \#feats, 0.10 \cdot \#feats + 5, ..., \#feats]$ , and  $\beta \in [0.92, 0.92 + 0.1, ..., 1.64]$ . The graphs show the expected reduction that can be obtained in  $last_{iter}$  vs.  $F_{beta}$  depending on the  $\Gamma$  used. However, estimates will be approximate and may vary greatly depending on the dataset (type and size<sup>3</sup>) and the machine learning algorithm used for modeling the problem.

### 4 Strategy for working with SHDD

Creating accurate models with SHDD is a current challenge. If the dataset has hundreds or a few thousand instances, and the dimension is high (several tens or hundreds of features), the search for models that correctly generalize the problem will face two fundamental problems: the curse of dimensionality and an excessive over-fitting in the optimization process.

Although there are algorithms, such as trees and neural networks, that may be less affected by the curse of dimensionality, in these cases it is highly recommended to use feature selection or dimensional reduction. In addition, the regularization included in machine learning algorithms helps penalize models that are too complex and with high probability of overfitting. The proposed

 $<sup>^{3}</sup>$  As can be seen in Figure 5.

	Data	set		BO	HYB-PARSIMONY and BO				
name	$train_{size}$	$test_{size}$	# feats	$J_{BO}$	$last_{iter}$	J0.5	Fs0.5		
slice	2000	23000	378	.1414	165.8(42.4)	.1449 (.0022)	148.0 (3.74)		
blog	2000	50397	276	.8216	178.6 (45.6)	.8154 $(.0170)$	67.0(17.42)		
crime	1107	1108	127	.6373	199.0(0.0)	.6379 $(.0076)$	28.8(4.32)		
ailerons	2000	11750	40	.3984	131.2 (17.9)	.3982 (.0016)	13.2(2.05)		
bank	2000	6192	32	.6745	160.0 (32.1)	.6726 (.0023)	15.6(1.52)		
puma	2000	6192	32	.8762	106.2 (16.3)	.2006 (.0250)	3.6(0.89)		
pol	2000	13000	26	.3165	148.0(32.2)	$.2413\ (.0034)$	7.2(0.45)		

Table 1: Results for 7 datasets obtained with BO (J) versus HYB-PARSIMONY with  $thr_{fs} = 0.50$  followed by BO.

hybrid methodology greatly facilitates both aspects since it seeks to reduce as much as possible the number of features of the selected model, as well as its internal complexity.

However, HYB-PARSIMONY is such an intensive search method that when working with SHDD the method may find parsimonious solutions that are too specific to that set of instances. Thus, the chosen hyperparameters and the selected features may be the most appropriate for that sample but not be sufficient to create a model that will generalize correctly in the future. To reduce this overfitting and to find a feature selection that can be used to create a robust model that generalizes correctly in this kind of problems, we propose the following methodology:

- 1. Repeat *n* runs with different random seeds the search for the best model with HYB-PARSIMONY and hold-out validation. In each repetition, extract the feature probability vector of the best individual.
- 2. Average the probabilities for each feature and select those that have a value greater than a given threshold,  $thr_{fs}$ .
- 3. Performs hyperparameter tuning with BO and the features selected in the previous point.
- 4. Repeat points 2 and 3 with different  $thr_{fs}$ .
- 5. Select the model that obtains the best error J with another test dataset.

Table 1 shows the results with 7 high-dimensional datasets of using the described methodology versus using BO with all features (#feats). In these experiments, 2000 rows were selected for training/validation (except crime where half of them were used) and the rest were utilized as a test dataset to verify the degree of generalization of the models.  $J_{BO}$  corresponds to the testing RMSE error obtained from a model that used all the input features and whose hyperparameters were adjusted by BO. The last three columns corresponds to the new proposal. First, 25 runs of HYB-PARSIMONY were performed with  $\Gamma = 0.50$ , nruns = 200, P = 15, early\_stopping = 35, hold-out validation with a 20%, and KernelRidge as ML algorithm. Finally, ReRank was set to 0.001 which corresponds to the maximum difference between the J of two models to

be considered equal. A high value of this parameter facilitates the search for parsimony in HYB-PARSIMONY because between two models with a similar J the less complex model is selected. Next, hyperparameter tuning with BO was done of a *KernelRidge* algorithm with the features whose probabilities were greater o equal than 0.50 ( $thr_{fs} = 0.50$ ). Columns in table indicate: the last iteration ( $last_{iter}$ ) of HYB-PARSIMONY, RMSE (J0.5) and the number of features used (Fs0.5) in the final model. The results correspond to the average values and the standard deviation (in parentheses) of 5 runs of the whole methodology with different random seeds.

Table 2: Proposed methodology with different  $thr_{fs}$  vs. BO with all features  $(J_{BO})$ .

dataset	$ J_{BO} $	J0.1	Fs0.1	J0.2	Fs0.2	J0.3	Fs0.3	J0.4	Fs0.4	J0.5	Fs0.5	J0.6	Fs0.6	J0.7	Fs0.7
slice	.1414	.1413	369.2	.1392	346.4	.1374	294.2	.1370	228.2	.1449	148.0	.1583	85.8	.2267	40.2
blog	.8216	.8222	270.0	.8265	247.0	.8288	196.8	.8234	128.8	.8154	67.0	1.014	29.4	.9462	10.8
crime	.6373	.6373	123.2	.6367	108.8	.6371	85.0	.6386	56.4	.6379	28.8	.6461	12.8	.7867	3.8
ailerons	.3984	.3983	36.6	.3986	33.2	.3994	25.4	.3993	17.8	.3982	13.2	.4338	9.8	.5043	5.6
bank	.6745	.6746	31.6	.6729	29.4	.6756	24.6	.6744	20.4	.6726	15.6	.6791	11.4	.6889	9.0
puma	.8762	.8383	24.8	.3030	12.2	.2308	7.6	.2096	4.6	.2006	<b>3.6</b>	.2007	3.6	.2230	2.8
pol	.3165	.3066	20.6	.2736	13.2	.2584	10.4	.2461	8.2	.2413	7.2	.2413	7.2	.2387	6.4

As can be observed in Table 1, the proposed methodology obtained more accurate models in 5 of the 7 databases, in addition to the fact that in the remaining ones the differences between  $J_{BO}$  and J0.5 were not excessive. However, the most outstanding results were observed in the significant reduction of the number of average features. For example, in **slice** the number of features was reduced to 39.1%, in **blog** to 24.2%, in **crime** to 22.7%, in **ailerons** to 33%, in **puma** to 11.3%, or in **pol** to 27.7%. In conclusion, the methodology helped to find more accurate models with a significant reduction of features. However, these results could be improved by using different  $thr_{fs}$  as shown in Table 2.

Finally, Table 3 shows a comparative analysis of the HYB-PARSIMONY methodology versus the previous method, PSO-PARSIMONY. The results show that the new methodology improved J in the four higher dimensionality datasets in conjunction with a considerable reduction in the number of features. However, PSO-PARSIMONY obtained better J in bank, puma and pol, but with worse parsimony in the first two.

# 5 Conclusions

GA-PARSIMONY, PSO-PARSIMONY and HYB-PARSIMONY are methodologies that have been developed for the search of accurate but low complexity ML models. However, an intensive search with these methods in SHDD can lead to overfitted models.

	H	YB-I	PARSIMONY		PSO-PARSIMONY					
dataset	last <sub>iter</sub> i	$thr_{fs}$	J	Fs	$last_{ite}$	$r thr_{fs}$	J	Fs		
slice	165.8(42.4)	.4	.1370(.003)	228.2(6.0)	182.8(21.73	) .5	.1372(.002)	239.0(6.9)		
blog	178.6(45.6)	.5	.8154(.017)	67.0(17.4)	191.6(16.55	0. (	.8215(.000)	276.0(0.0)		
crime	199.0(0.0)	.2	.6367(.002)	108.8(1.3)	175.4(32.94)	) .1	.6371(.000)	124.8(1.3)		
ailerons	131.2(17.9)	.5	.3982(.002)	13.2(2.1)	154.8(34.15	) .5	.3984(.002)	16.0(3.5)		
$\operatorname{bank}$	160.0(32.1)	.5	.6725(.002)	15.6(1.5)	149.8(48.84	) .5	.6724(.002)	16.8(1.6)		
puma	106.2(16.27)	.5	.2006(.025)	3.6(0.9)	104.4(13.81)	).4	.1894(.000)	4.0(0.0)		
pol	148.0(32.2)	.7	.2387(.004)	6.4(0.6)	127.2(15.50)	) .7	.2374(.003)	6.2(0.5)		

Table 3: HYB-PARSIMONY vs. PSO-PARSIMONY.

The proposed methodology is based on repeating HYB-PARSIMONY with different random seeds and by using hold-out validation. In this way, at each run the search for the best model is validated with a different part of the dataset. Averaging the feature probability vectors allows one to make a more robust selection of the final features. Once these are selected, with different thresholds, BO is used to fit the hyperparameters of the model.

Results demonstrated that it is possible to obtain more accurate models with a significant reduction in the number of features.

Acknowledgements The work is supported by grant PID2020-116641GB-I00 and the European Regional Development Fund under Project PID2021-123219OB-I00 funded by MCIN / AEI/ 10.13039 501100011033 FEDER, UE. We are also greatly indebted to Banco Santander for the REGI2020/41 and REGI2022/60 fellowships.

# References

- Antonanzas-Torres, F., Urraca, R., Antonanzas, J., Fernandez-Ceniceros, J., Martinez-de Pison, F.J.: Generation of daily global solar irradiation with support vector machines for regression. Energy Conversion and Management 96, 277–286 (2015). https://doi.org/https://doi.org/10.1016/j.enconman.2015.02.086
- Ceniceros, J.F., Sanz-Garcia, A., Pernia-Espinoza, A., Martinez-de Pison, F.J.: PSO-PARSIMONY: A new methodology for searching for accurate and parsimonious models with particle swarm optimization. application for predicting the force-displacement curve in t-stub steel connections. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) Hybrid Artificial Intelligent Systems. pp. 15–26. Springer, Cham (2021)
- Chuang, L.Y., Tsai, S.W., Yang, C.H.: Improved binary particle swarm optimization using catfish effect for feature selection. Expert Syst. Appl. 38(10), 12699– 12707 (2011). https://doi.org/10.1016/j.eswa.2011.04.057
- 4. Divasón, J., Pernia-Espinoza, A., Martinez-de Pison, F.J.: New hybrid methodology based on particle swarm optimization with genetic algorithms to improve

the search of parsimonious models in high-dimensional databases. In: Hybrid Artificial Intelligent Systems. pp. 335–347. Springer International Publishing, Cham (2022)

- Dulce-Chamorro, E., de Pison, F.J.M.: An advanced methodology to enhance energy efficiency in a hospital cooling-water system. Journal of Building Engineering 43, 102839 (2021). https://doi.org/https://doi.org/10.1016/j.jobe.2021.102839
- Karaboga, D., Basturk, B.: Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) Foundations of Fuzzy Logic and Soft Computing. pp. 789–798. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
- Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. vol. 4, pp. 1942–1948 vol.4 (1995). https://doi.org/10.1109/ICNN.1995.488968
- Marinaki, M., Marinakis, Y.: A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands. Expert Systems with Applications 46, 145–163 (2016). https://doi.org/10.1016/j.eswa.2015.10.012
- Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software 114, 163–191 (2017). https://doi.org/10.1016/ j.advengsoft.2017.07.002
- Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in Engineering Software 69, 46–61 (2014). https://doi.org/10.1016/j.advengsoft.2013.12.007
- Martinez-de Pison, F.J., Ferreiro, J., Fraile, E., Pernia-Espinoza, A.: A comparative study of six model complexity metrics to search for parsimonious models with GAparsimony R Package. Neurocomputing 452, 317–332 (2021). https://doi.org/ 10.1016/j.neucom.2020.02.135
- 12. Martinez-de Pison, F.J., Gonzalez-Sendino, R., Aldama, A., Ferreiro-Cabello, J., Fraile-Garcia, E.: Hybrid methodology based on bayesian optimization and gaparsimony to search for parsimony models by combining hyperparameter optimization and feature selection. Neurocomputing **354**, 20–26 (2019). https://doi. org/https://doi.org/10.1016/j.neucom.2018.05.136, recent Advancements in Hybrid Artificial Intelligence Systems
- 13. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010), pp. 65–74. Springer (2010)