ISSN 2007-9737

Evolutionary Instance Selection Based on Preservation of the Data Probability Density Function

Samuel Omar Tovias-Alanis, Wilfrido Gómez-Flores, Gregorio Toscano-Pulido

Instituto Politécnico Nacional,
Centro de Investigación y de Estudios Avanzados,
Mexico

{samuel.tovias, wgomez, gtoscano}@cinvestav.mx

Abstract. The generation of massive amounts of data has motivated the use of machine learning models to perform predictive analysis. However, the computational complexity of these algorithms depends mainly on the number of training samples. Thus, training predictive models with high generalization performance within a reasonable computing time is a challenging problem. Instance selection (IS) can be applied to remove unnecessary points based on a specific criterion to reduce the training time of predictive models. This paper introduces an evolutionary IS algorithm that employs a novel fitness function to maximize the similarity of the probability density function (PDF) between the original dataset and the selected subset, and to minimize the number of samples chosen. This method is compared against six other IS algorithms using four performance measures relating to the accuracy, reduction rate, PDF preservation, and efficiency (which combines the first three indices using a geometric mean). Experiments with 40 datasets show that the proposed approach outperforms its counterparts. The selected instances are also used to train seven classifiers, in order to evaluate the generalization and reusability of this approach. Finally, the accuracy results show that the proposed approach is competitive with other methods and that the selected instances have adequate capabilities for reuse in different classifiers.

Keywords. Instance selection, probability density function, evolutionary algorithm.

1 Introduction

Nowadays, ubiquitous computing and the Internet of Things are generating massive amount of multivariate data. As a result, researchers in many scientific and engineering fields have applied machine learning (ML) techniques to take advantage of this information for data modeling and decision making [5].

In supervised learning, ML algorithms are used to create prediction models based on a set of labeled training data. A dataset is typically represented by a matrix $X \in \mathbb{R}^{n \times d}$ composed of n instances and d predictor variables. Furthermore, the ith instance is represented by the vector $\mathbf{x}_i = [x_{i1}, \ldots, x_{id}]$, associated with an actual class label $y_i \in \Omega = \{\omega_1, \ldots, \omega_c\}$, where c is the total number of classes. Thus, training of the model implies the supervised learning of a mapping function $g : \mathbb{R}^{n \times d} \rightarrow \hat{y}$, where $\hat{y} \in \Omega$ is a predicted class label [11].

The computational complexity of ML algorithms depends mainly on the number of training instances and prediction variables. Thus, for extensive datasets, building a classification model within a reasonable computing time is a challenging problem. In addition, the model evaluation stage and hyperparameter tuning increase the training time [3]. Instance selection (IS) algorithms remove unnecessary patterns based on some elimination criterion. The aim is to select a representative subset (denoted by X_S) from the original dataset (denoted by X_O) to reduce the training time when building prediction models.

IS algorithms can be divided into wrapper and filter methods; the former uses a classification rate from a supervised learning algorithm, whereas the latter uses statistical information from data [13].

In addition, metaheuristic-based IS techniques generally use evolutionary algorithms (EAs), in which each member of the population represents a subset of selected instances. EAs encode the individuals as a fixed-length binary vector of size n_O, corresponding to the number of instances in X_O. In this encoding scheme, a value of 1 means that the corresponding instance is selected, whereas a value of 0 indicates the opposite. Optimization is commonly performed using a wrapper scheme that maximizes both the classification accuracy and the reduction rate [9].

According to Reeves and Bush [17], the training set should be an accurate representation of the actual probability distribution over the input space. However, selecting instances using an EA-based wrapper scheme may lead to solutions that fulfill the classifier's criteria but cannot preserve the original probability distribution. For instance, if a support vector machine (SVM) is used, the selected instances may be biased towards the local distribution of the support vectors. We can therefore state that the selection of instances should be made only once, and the resulting data subset can then be used to train different classifiers without loss of generalization, thus avoiding the need to repeat the selection process for each type of classifier.

This work presents an evolutionary IS method based on a filter approach, in which the fitness function incorporates both preservation of the probability density function (PDF) and a reduction rate. The underlying concept is to preserve the original data distribution by maximizing the similarity between the X_O and X_S PDFs, while reducing the number of instances in X_S. Both objectives are combined through the use of a weighted sum, and a global optimization scheme based on a genetic algorithm (GA) with binary encoding is used to carry out instance selection.

2 Related Work

2.1 Classical IS Methods

Classical IS methods generally reduce the number of instances using the nearest neighbor rule. These approaches can be divided into condensation, edition, and hybrid methods. The condensed nearest neighbor (CNN) method retains points closer to the decision boundaries, while internal points are removed, since they do not affect the decision boundaries [14]. The edited nearest neighbor (ENN) approach preserves internal samples while removing points closer to the decision boundaries with class labels that are different from their neighboring points (i.e., noisy points) [20]. The decremental reduction optimization procedure (DROP3) is a hybrid method that removes border points by first applying ENN to filter noisy instances, and then removes internal instances far from the decision boundaries [21]. Finally, the iterative case filtering algorithm (ICF) produces data clusters based on reachable and coverage sets, where points with a reachable set size greater than the coverage set size are removed [4].

2.2 Evolutionary IS Methods

An IS based on an EA is generally classed as a wrapper scheme. In this context, Kuncheva [15] proposed a GA with binary encoding, where the fitness function measured the error rate of the k-nearest neighbors (kNN) classifier.

In another work, Cano et al. [6] analyzed the performance of four binary-based representation EAs. The objective function adopted in this case was a weighted sum of the classification error and the reduction rate, with the same relative importance. Likewise, Garcia et al. [12] proposed a memetic algorithm that combined the heuristic approach of population-based algorithms with local search methods. The fitness function in this approach maximized both the accuracy and the reduction rate.

Aldana et al. [2] introduced a method based on an eclectic GA (EGA). This approach adopted a binary string encode in which two positive integers represented the number of randomly sampled instances. The EGAs objective function evaluated the reduction rate, and two constraints were considered: the error between the original and selected sets, and the proportion of elements between the quantiles of both sets.

Rosales-Perez et al. [18] used a multiobjective EA to solve the IS problem. Their solution encoded the reduction technique and the hyperparameters of an SVM, and the two objective functions were the classification rate and the reduction rate.

3 Kernel Density Estimation

Kernel density estimation (KDE) is a non-parametric method for estimating the PDF of a random variable, and can handle an arbitrary distribution without requiring any assumptions about the form of its underlying density [11].

Let $x_1, x_2, \ldots, x_n$ be independent and identically distributed samples, taken randomly from a distribution with unknown density $p(x)$. KDE in a region $\mathcal{R}$ centered at $\hat{x}$ is given by:

$$\hat{p}_h(\hat{x}) = \frac{1}{nh} \sum_{i=1}^{n} \phi_{\mathcal{N}} \left(\frac{\|\hat{x} - x_i\|_2}{h} \right), \tag{1}$$

where $\| \cdot \|_2$ denotes the Euclidean distance, $\phi_{\mathcal{N}}(\cdot)$ is the Gaussian kernel function with zero mean and unit variance, expressed as:

$$\phi_{\mathcal{N}}(u) = \frac{1}{2\pi^{(1/2)}} \exp\left(-\frac{u^2}{2} \right), \tag{2}$$

and $h > 0$ is a smoothing parameter, also known as the bandwidth. This parameter must be fine-tuned, since it has a strong influence on the result of the density estimation. When $h \to 0$, the shape of the estimated PDF is noisy and may include spurious peaks; conversely, if $h \to \infty$, the shape of the estimated PDF is over-smoothed.

The optimality criterion that is typically applied to select h is the expected L_2-risk function, also known as the mean integrated squared error (MISE). In this work, we use the direct plug-in rule (DPI), which is a method for automatically selecting a near-optimal h value by minimizing the MISE quality estimates (ψ). The following steps are used to calculate h based on the DPI rule [19]:

1. Estimate ψ_8 using an estimator of dispersion $\hat{\sigma}$, such as the median absolute deviation:

$$\hat{\psi}_8^{\hat{\sigma}} = \frac{105}{32\pi^{1/2}\hat{\sigma}(x)^9}, \tag{3}$$

2. Estimate ψ_6 using the estimator $\hat{\psi}_6(g_1)$, where:

$$g_1 = \left(\frac{11.9683}{\hat{\psi}_8^{\hat{\sigma}} n} \right)^{1/9}, \tag{4}$$

3. Estimate ψ_4 using the estimator $\hat{\psi}_4(g_2)$, where:

$$g_2 = \left(-\frac{2.3937}{\hat{\psi}_6(g_1) n} \right)^{1/7}, \tag{5}$$

4. The value of the bandwidth h is then calculated as:

$$h = \left(\frac{0.2821}{\hat{\psi}_4(g_2) n} \right)^{1/5}. \tag{6}$$

In Steps 2 and 3, the estimator $\hat{\psi}_r(g)$ is:

$$\hat{\psi}_r(g) = \frac{g^{(-r-1)}}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} \phi_{\mathcal{N}}^{(r)} \left(\frac{x_i - x_j}{g} \right), \tag{7}$$

where $\phi_{\mathcal{N}}^{(r)}$ is the rth derivative of the kernel $\phi_{\mathcal{N}}$.

4 Proposed Approach

Finding the optimal subset of instances in the IS task implies exploring a search space of size $2^{n_O} - 1$. This number reflects the possible subsets X_S, with cardinality $n_S = 1, \ldots, n_O - 1$, chosen from X_O with n_O instances. Although the search space is finite, it grows exponentially, making an exhaustive exploration intractable, and IS is therefore generally addressed as an optimization problem, using metaheuristics to find a sub-optimal solution within a reasonable computing time. In this following, we describe the proposed evolutionary IS method based on a PDF preservation approach.

4.1 Fitness Function

We propose a novel fitness function for use in an evolutionary IS algorithm. It has two components: (i) maximizing the similarity between the X_O and X_S PDFs; and (ii) minimizing the number of instances in X_S.

This first component uses the Hellinger distance to measure the distributional divergence. The Hellinger distance between two densities p and q is defined as [8]:

$$\mathcal{H}(p,q) = \left(\frac{1}{2}\int\left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx\right)^{1/2}. \quad (8)$$

and satisfies the property $0 \leq \mathcal{H}(p,q) \leq 1$, where a value closer to zero indicates a higher similarity between the densities p and q. Hence, maximizing the similarity between p and q implies minimizing the Hellinger distance.

It is worth noting that p and q are univariate density functions. In contrast, the dataset X_O usually contains instances in $\mathbb{R}^d$. To handle multivariate data, the Hellinger distance is calculated for each predictor variable for each class, to form the following matrix:

$$H = \begin{pmatrix} \mathcal{H}_{1,1} & \cdots & \mathcal{H}_{1,d} \\ \vdots & \ddots & \vdots \\ \mathcal{H}_{c,1} & \cdots & \mathcal{H}_{c,d} \end{pmatrix}, \quad (9)$$

where $\mathcal{H}_{i,j} \equiv \mathcal{H}(p_{i,j}, q_{i,j})$ is the Hellinger distance between the original, $p_{i,j}$, and approximated, $q_{i,j}$, densities of the jth variable in the ith class. The PDF of a predictor variable is obtained using KDE, as described in Section 3. It is worth mentioning that the DPI rule is calculated on the jth dimension of the ith class of X_O (for $i = 1, \ldots, c$ and $j = 1, \ldots, d$), and the resulting bandwidth $h_{i,j}$ is used to estimate the densities $p_{i,j}$ and $q_{i,j}$.

The second objective is addressed by measuring the fraction of selected instances as:

$$s_f = \frac{n_S}{n_O}, \quad (10)$$

where n_S and n_O denote the number of instances in X_S and X_O, respectively. This objective varies in the range $[0, 1]$, and a value close to zero indicates an adequate reduction rate.

Finally, the fitness function is used to combine the two criteria in (9) and (10) through a weighted sum, as follows:

$$f = \frac{w}{c \cdot d}\sum_{i=1}^{c}\sum_{j=1}^{d}\mathcal{H}_{i,j} + (1-w)\cdot s_f, \quad (11)$$

where $w \in (0, 1)$ is a weight coefficient that expresses the importance of each objective. This weighted fitness function varies in the range $[0, 1]$, where an aptitude value close to zero indicates that X_S achieves a high reduction rate and preserves the probability distribution of X_O. We evaluated the impact of w by varying its value in the range $[0.50, 0.95]$ with steps of 0.05. Algorithm 1 presents a pseudocode for the evaluation of the fitness function.

4.2 Evolutionary IS

In this work, we use a GA to minimize the objective function in (11). Each individual in the population is a fixed-length binary vector of size n_O, where a value of 1 means that the corresponding instance in X_O is selected, and 0 indicates the opposite.

Solutions are randomly initialized with a discrete uniform distribution in the range $[0, 1]$. The parent selection strategy uses a two-way tournament approach. Next, a two-point crossover is performed in the recombination step to exchange the selected parents' genetic information to generate new offspring. Then, based on the mutation probability factor, the mutation operator performs a bit flip in random positions of each offspring vector. An elitist strategy is also applied to ensure that the quality of the solution does not decrease over the generations. Finally, the GA returns the best individual in the last generation.

In this case, a population of 100 individuals was evolved over 2000 generations. The crossover and mutation probabilities were set to 0.9 and $1/n_O$, respectively.

5 Experimental Setup

5.1 Datasets

The datasets used in the experiments were obtained from the KEEL repository [1] and the UCI Machine Learning Database [10]. Table 1 summarizes the characteristics of 40 small datasets (with no more than $5,456$ instances). In order to test the performance of the proposed method with large datasets, two medium-sized datasets were considered: Magic Gamma Telescope (MGT) with

Algorithm 1 Fitness function evaluation

Input: Individual $\mathbf{q} \in \{0,1\}^{n_O}$, original dataset normalized in the range $[-1,1]$: $\hat{X}_O \in \mathbb{R}^{n_O \times d}$, set of density estimates of $\hat{X}_O$: $\{p_{1,1},\ldots,p_{c,d}\}$, set of bandwidths: $\{h_{1,1},\ldots,h_{c,d}\}$, $\mathcal{R} = 100$ equidistant points in the range $[-1.5,1.5]$: $\hat{\mathbf{x}} = [\hat{x}_1,\ldots,\hat{x}_{\mathcal{R}}]$ and the weight of the fitness function: w
Output: Fitness value: f

Decode $\mathbf{q}$ to obtain the selected subset with n_S instances from $\hat{X}_O$: X_S
Get the number of classes of $\hat{X}_O$: c
Get the number of classes of X_S: c_S
if $c = c_S$ **then**
  Initialize the cumulative sum of the values of H (9): $\Sigma_{\mathcal{H}} = 0$
  for $i = 1$ to c **do**
    for $j = 1$ to d **do**
      Get the values of the jth variable from the ith class of X_S: $\mathbf{x}_{i,j} = [x_1,\ldots,x_n]$
      Compute KDE (1) for each point in $\hat{\mathbf{x}}$ using $\mathbf{x}_{i,j}$ and $h_{i,j}$: $q_{i,j}$
      Compute the Hellinger distance (8): $\mathcal{H}_{i,j} \equiv \mathcal{H}(p_{i,j}.q_{i,j})$
      Update the cumulative sum of the elements of H: $\Sigma_{\mathcal{H}} = \Sigma_{\mathcal{H}} + \mathcal{H}_{i,j}$
    end for
  end for
  Compute the average of H: $\mu_{\mathcal{H}} = \Sigma_{\mathcal{H}}/(c \cdot d)$
  Compute the second objective: $s_f = n_S/n_O$
  Compute the fitness function (11): $f = w \cdot \mu_{\mathcal{H}} + (1-w) \cdot s_f$
else
  Penalize solution if one or more classes are eliminated: $f = 1$
end if
return f

$n = 19,020$, $d = 10$, and $c = 2$, and Letter Recognition (LT) with $n = 20,000$, $d = 16$, and $c = 26$.

5.2 Instance Selection Methods

The proposed approach, denoted as F_w (i.e., the proposed filter method with a specific weight value w), was compared against six other IS methods.

The first alternative approach was an evolutionary IS algorithm based on a wrapper scheme. This method used the same GA as the proposed approach but applied a fitness function that is commonly adopted in the literature, in which a weighted sum is used to combine the classification accuracy and the reduction rate with the same relative importance [17, 6, 12]. When evaluating the fitness function, the selected subset X_S, given by a potential solution, is used to train the classification model, whereas the validation set is obtained as $X_V = X_O - X_S$, and is used for measuring the classification accuracy. Two classifiers are considered, SVM and kNN, and the two variants of this approach are denoted as W_{SVM} and W_{kNN} (i.e., the wrapper method with SVM and kNN, respectively).

In addition, the soft margin parameter (C) and the bandwidth of the Gaussian kernel (γ) used for the W_{SVM} algorithm were found using the grid search method in the ranges $C = [2^{-5}, 2^{-3}, \ldots, 2^{15}]$ and $\gamma = [2^{-15}, 2^{-13}, \ldots, 2^{3}]$, with 5-fold cross-validation [7]. Appendix 7 lists the C and γ hyperparameters found.

The four remaining IS methods are CNN, ENN, DROP3, and ICF, as depicted in Section 2.1. The number of nearest-neighbors for W_{kNN} and the classical methods are set to $k = 3$.

5.3 Performance Assessment

The performance of the IS methods was measured using four indices:

Table 1. Characteristics of the datasets: n is the number of instances, d is the dimensionality, and c is the number of classes

ID	Dataset	n	d	c	ID	Dataset	n	d	c
1	Appendicitis	106	7	2	21	Ionosphere	351	33	2
2	Australian	690	14	2	22	Iris	150	4	3
3	Balance	625	4	3	23	Led7digit	500	7	10
4	Banana	5,300	2	2	24	Mammographic	830	5	2
5	Bands	365	19	2	25	Monk-2	432	6	2
6	Breast	277	9	2	26	LIBRAS	360	90	15
7	Bupa	345	6	2	27	New Thyroid	215	5	3
8	Car	1,728	6	4	28	Pima	768	8	2
9	Cleveland	297	13	5	29	Saheart	462	9	2
10	Contraceptive	1,473	9	3	30	Sonar	208	60	2
11	Crx	653	15	2	31	Spectfheart	267	44	2
12	Dermatology	358	34	6	32	Tae	151	5	3
13	Flare	1,066	11	8	33	Tic-Tac-Toe	958	9	2
14	German	1,000	20	2	34	Vehicle	846	18	4
15	Glass	214	9	6	35	Vowel	990	13	11
16	Haberman	306	3	2	36	Wall Following	5,456	2	4
17	Hayes-Roth	160	4	3	37	WDBC	569	30	2
18	Heart	270	13	2	38	Wine	178	13	3
19	Hepatitis	80	19	2	39	Wisconsin	683	9	2
20	Housevotes	232	16	2	40	Yeast	1,484	8	10

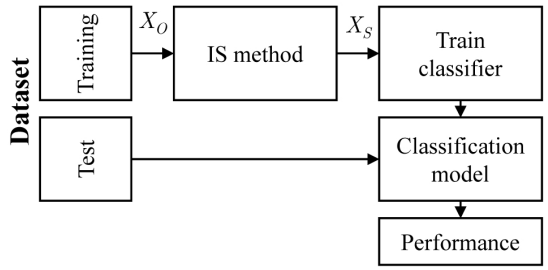

Fig. 1. Evaluation framework for IS methods

— Accuracy (ACC): A classifier was trained using the selected instances, and the accuracy (success rate) was measured on an independent test set.

— Reduction rate (RR): The fraction of removed instances was calculated as $1 - s_f$, where s_f is given by (10).

— Hellinger distance complement (HDC): The similarity between the densities X_O and X_S was calculated using the mean Hellinger distance (HD) for the matrix in (9). From a maximization perspective, HDC $= 1 -$ HD.

— Efficiency (E): The geometric mean $\sqrt[3]{\text{ACC} \times \text{RR} \times \text{HDC}}$ was used to calculate the tradeoff between accuracy, reduction rate, and PDF preservation.

A t-times k-fold cross-validation method (where $t = 10$ and $k = 5$) was used to split the small datasets into training and test sets. This resampling process reduced the influence of randomness introduced by data splitting [22]. To divide the medium-sized datasets, a 10-fold cross-validation technique was applied. The procedure illustrated in Fig. 1 was then performed on each fold.

The reusability of the selected instances is related to the ability to train different classifiers without losing generalization. In this sense, in wrapper techniques, the subset X_S could fulfill the classifier's criteria to increase the accuracy, but loses similarity with the X_O distribution when the number of instances is reduced; thus, X_S may be useless for training other types of classifiers. To measure the reusability of X_S, we use two types of accuracy:

— Type 1, which measures the classification performance on the test set using only the classifier within the wrapper method.

— Type 2, which measures the classification performance on the test set using different classifiers that are not used by the wrapper method.

Seven classifiers were considered when measuring the classification accuracy: classification and regression tree (CART), linear discriminant analysis (LDA), quadric discriminant analysis (QDA), naïve Bayes classifier (NB), radial basis function network (RBFN), SVM, and kNN. Furthermore, max$(3, \sqrt{n})$ hidden nodes were used for the RBFN architecture (where n is the number of the training instances), the number of nearest neighbors in kNN was set to k=3, and the hyperparameters of the SVM classifier were tuned for each subset using the grid search method depicted in Section 5.2 [11, 3].

The non-parametric Kruskal-Wallis test, followed by a Bonferroni correction ($\alpha = 0.05$), was performed for multiple comparisons to determine the statistical significance between the proposed

approach and the six IS methods in terms of the four indices listed above.

Additionally, the McNemar test ($\alpha = 0.05$) was used to statistically assess the accuracy of two classification models trained with X_O and X_S against the actual labels. It detects whether the difference between the misclassification rates is statistically significant. The null hypothesis establishes that the two predicted class labels, $\hat{y}_1$ and $\hat{y}_2$, have equal accuracy when predicting the actual class labels, y.

The testing platform used a computer with four cores at 3.5 GHz (Intel i7 4770k) and 32 GB of RAM. All the algorithms were developed in MATLAB 2018b [16], and the source codes are available upon request to the authors.

6 Results

6.1 Instance Selection Performance on Small Datasets

Fig. 2 shows the average performance results on the 40 small datasets, for all of the classifiers specified in Section 5.3. For the proposed method, the F_w performance changed according to w, as expected. As w increases, the values of ACC and HDC also increase, while the values of E and RR decrease. However, the first five values of w (i.e., 0.50 to 0.70) produced the same efficiency (E=0.81), which was the highest for all of the IS methods. Furthermore, F_{65} and F_{70} yielded a value of ACC=0.70 and a fairly similar PDF preservation (HDC=0.93). F_{65} gave the best tradeoff, as it achieved a higher reduction rate (RR=0.85).

Moreover, the F_w variants gave better PDF preservation than the wrapper and classical IS algorithms. Notably, only ENN achieved the same Hellinger distance complement as F_{50} (HDC=0.90), the F_w variant with the lowest PDF preservation.

Of the classical IS methods, ENN obtained the best accuracy (ACC=0.76) and PDF preservation (HDC=0.90), although it had the worst efficiency of all of the methods (E=0.51) due to the low reduction rate (RR=0.24). CNN obtained the second-best value of accuracy (ACC=0.71) and PDF preservation (HDC=0.89) of the classical techniques, although it outperformed ENN in terms of efficiency (E=0.71) and reduction rate (RR=0.59). The hybrid methods yielded better reduction rates and efficiency but lower accuracies and PDF preservation than CNN and ENN. For instance, DROP3 had a better reduction rate (RR=0.83) and efficiency (E=0.74) than ICF, but the lowest accuracy (ACC=0.67) and PDF preservation (HDC=0.76) of all the methods. In contrast, ICF achieved a better accuracy (ACC=0.70) and PDF preservation (HDC=0.81), but a lower reduction rate (RR=0.72) and efficiency (E=0.73) than DROP3.

The wrapper methods achieved better efficiency and reduction rate than the classical techniques. Specifically, W_{kNN} obtained the highest efficiency (E=0.76), and W_{SVM} the best reduction rate (RR=0.87). However, in terms of the accuracy index, the wrapper methods did not outperform the classical ones. For example, W_{kNN} had the highest classification performance of the wrapper methods, but this was the same as for DROP3 (ACC=0.67), which was the worst of the classical algorithms in terms of accuracy. Regarding PDF preservation, the wrapper methods were again surpassed by CNN and ENN; however, W_{kNN} slightly outperformed ICF, while W_{SVM} had the second-worst HDC index, only outperforming DROP3.

Fig. 3 shows a detailed comparison of the results from F_{65} against all the IS methods. Tables 2 and 3 show the resulting p-value of the Kruskal-Wallis test for multiple comparisons of each performance index.

Regarding accuracy, F_{65} attained the third-best result, and was only outperformed by CNN and ENN; still, according to Table 2, there are no statistical differences from IS methods. Although F_{65} obtained the second-best value in terms of reduction rate, there was no significant difference from W_{SVM}, which achieved the highest performance for this index.

These methods also showed statistical differences from CNN, ENN, and ICF. Despite the high reduction rate of F_{65}, it achieved the highest PDF preservation and showed statistical differences from W_{kNN}, W_{SVM}, DROP3, and ICF on the HDC index.

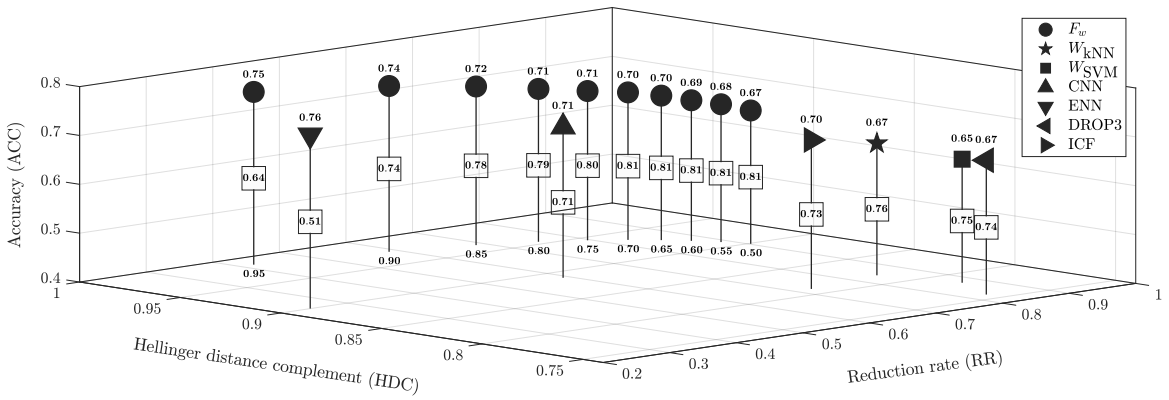

Fig. 2. Average performance results. They were calculated over the seven classifiers on the 40 datasets. Above each marker is shown the accuracy (ACC). The squared label shows the efficiency (E). The weight values (w) for the proposed approach F_w are shown at the bottom

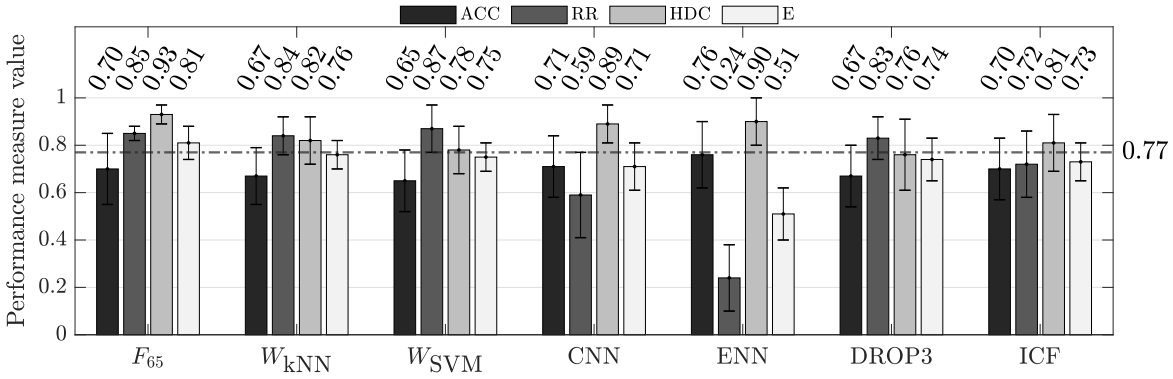

Fig. 3. Average performance results over the seven classifiers on the 40 datasets. Above each bar, the value of the corresponding measure. The dashed line marks the average accuracy obtained by the seven classifiers trained on the original datasets

For the efficiency, F_{65} attained the highest value, and the results in Table 3 show that this method gave statistical differences from all of the IS methods except W_{kNN}.

These results suggest that for a specific value of w, the proposed approach selects a subset of instances that preserve the probability distribution of the original dataset with a high reduction rate. The selected subset is also useful for training distinct classification models with good generalization performance.

6.2 Instance Selection Performance on Medium-Size Datasets

The medium-sized datasets described in Section 5.1 were used to test the performance of the F_w method, with $w = 0.50$ to give both objectives in the fitness function the same relative importance. The accuracy was calculated as the average over the results from the seven classifiers described in Section 5.3.

Table 2. Bonferroni correction results. The upper triangular matrix shows the p-values for ACC, and the lower triangular matrix shows the p-values for RR. In bold, $p < 0.05$

	F_{65}	W_{kNN}	W_{SVM}	CNN	ENN	DROP3	ICF
F_{65}	-	1.000	1.000	1.000	1.000	1.000	1.000
W_{kNN}	1.000	-	1.000	1.000	0.158	1.000	1.000
W_{SVM}	1.000	1.000	-	1.000	**0.022**	1.000	1.000
CNN	**0.000**	**0.000**	**0.000**	-	1.000	1.000	1.000
ENN	**0.000**	**0.000**	**0.000**	**0.010**	-	0.355	1.000
DROP3	1.000	1.000	1.000	**0.000**	**0.000**	-	1.000
ICF	**0.007**	**0.016**	**0.000**	1.000	**0.000**	0.086	-

Table 3. Bonferroni correction results. The upper triangular matrix shows the p-values for E, and the lower triangular matrix shows the p-values for HDC. In bold, $p < 0.05$

	F_{65}	W_{kNN}	W_{SVM}	CNN	ENN	DROP3	ICF
F_{65}	-	0.297	**0.057**	**0.000**	**0.000**	**0.023**	**0.005**
W_{kNN}	**0.000**	-	1.000	0.437	**0.000**	1.000	1.000
W_{SVM}	**0.000**	1.000	-	1.000	**0.000**	1.000	1.000
CNN	1.000	**0.005**	**0.000**	-	**0.000**	1.000	1.000
ENN	1.000	**0.000**	**0.000**	1.000	-	**0.000**	**0.000**
DROP3	**0.000**	1.000	1.000	**0.000**	**0.000**	-	1.000
ICF	**0.000**	1.000	1.000	**0.012**	**0.000**	1.000	-

On the MGT dataset, the accuracy of the subsets was slightly better (ACC=0.81) than that obtained on the original dataset (ACC=0.80). The F_{50} method attained a regular efficiency (E=0.74) due to the poor reduction rate (RR=0.52) and a high PDF preservation (HDC=0.98).

Regarding the LT dataset, the accuracy on the original set (ACC=0.81) was slightly higher that attained on the subsets (ACC=0.79).

However, similarly to the MGT dataset, the efficiency was regular (E=0.73) due to the low reduction rate (RR=0.52) and the high PDF preservation (HDC=0.96).

6.3 Reusability of Selected Instances

The reusability results are shown in Fig. 4, where the Type 1 and 2 accuracies are displayed as pairs of box plots. The upper graphic relates to W_{kNN}, in which the Type 1 accuracy was measured only using the kNN classification, while the Type 2 accuracy was measured using the remaining six classifiers.

Likewise, the lower graphic considers W_{SVM}, in which the Type 1 accuracy was measured only using SVM classification, while the Type 2 accuracy was measured using the remaining six classifiers.

For the proposed approach F_w, the accuracy improved with the values of the weights, i.e., the higher the weight, the better the accuracy. For each weight value, the median values for the two types of accuracy were quite similar. This suggests that the proposed method can give similar Type 1 and 2 accuracies, independently of the classifier, due to the criterion used to preserve the PDF of the original data.

The classical methods gave slightly better performance for Type 1 than Type 2, and ENN attained the highest accuracy due to its lower reduction rate. On the other hand, W_{kNN} and W_{SVM} obtained consistently better accuracy for Type 1 than Type 2. These results confirm that X_S is biased towards the classifier characteristics in the wrapper method, limiting the reusability of the selected data for training other classifiers.

Table 4. $\log_2 C$ and $\log_2 \gamma$ are the logarithms of the soft margin parameter and the Gaussian kernel's bandwidth, respectively, corresponding to the SVM classifier used in the W_{SVM} method

ID	Dataset	$\log_2 C$	$\log_2 \gamma$	ID	Dataset	$\log_2 C$	$\log_2 \gamma$
1	Appendicitis	11	−5	21	Ionosphere	5	−1
2	Australian	1	−15	22	Iris	5	−7
3	Balance	13	−7	23	Led7digit	15	−15
4	Banana	3	−1	24	Mammographic	13	−11
5	Bands	1	−9	25	Monk-2	1	−1
6	Breast	1	−3	26	LIBRAS	5	−1
7	Bupa	9	−15	27	New Thyroid	9	−15
8	Car	3	−1	28	Pima	1	−15
9	Cleveland	9	−15	29	Saheart	11	−15
10	Contraceptive	11	−9	30	Sonar	3	−1
11	Crx	1	−15	31	Spectfheart	3	−15
12	Dermatology	13	−15	32	Tae	11	−13
13	Flare	1	−3	33	Tic-Tac-Toe	11	−7
14	German	−5	−7	34	Vehicle	9	−15
15	Glass	5	−5	35	Vowel	7	−3
16	Haberman	13	−13	36	Wall Following	15	1
17	Hayes-Roth	5	−5	37	WDBC	5	−15
18	Heart	9	−15	38	Wine	11	−15
19	Hepatitis	13	−15	39	Wisconsin	3	−13
20	Housevotes	3	−7	40	Yeast	5	1

Finally, Fig. 5 shows the number of datasets for which there was no rejection of the null hypothesis

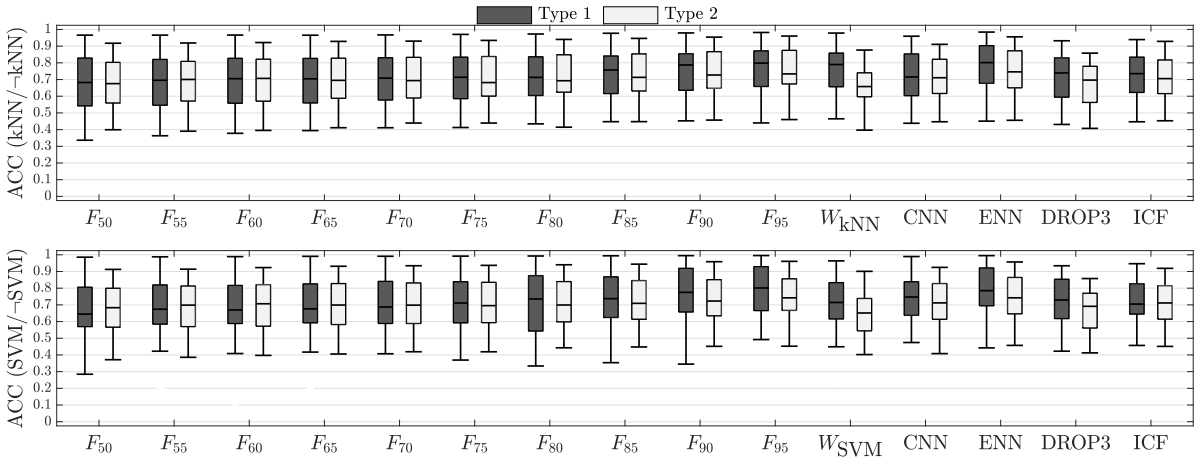

Fig. 4. Two types of accuracy results on the 40 datasets. Type 1: accuracy considering a single classifier, i.e., kNN (upper) and SVM (lower). Type 2: accuracy of six classifiers disregarding kNN (upper) and SVM (lower)

in the McNemar test for each classifier. The higher the count, the more similar the classification accuracy between models trained with X_O and X_S.

In the proposed approach F_w, the higher the values of the weights, the lower the number of rejections. Thus, F_{95} obtained the highest number of selected subsets that did not show a statistical difference from X_O for all of the supervised learning algorithms. Of the wrapper methods, W_{kNN} had a higher count of no rejections than W_{SVM}.

For the kNN classifier, W_{kNN} attained a significantly higher count than any other classifier, which was as expected since the selection criterion involved maximizing the accuracy of that specific classifier. W_{SVM} attained inferior results, even for the SVM classifier, where the count was not notably higher for the different classifiers, unlike the behavior of W_{kNN} for the kNN classifier.

Of the classical IS methods, ENN achieved a higher count in most cases, but gave a similar count to CNN for the kNN classifier, and was slightly outperformed by the same condensation method for RBFN. In terms of the number of counts for each classifier, the NB attained the highest value, and the QDA obtained the lowest counts for most IS methods.

These results reveal that the proposed method produces subsets of instances that can be reused to train different classifiers with a similar classification performance to that achieved from training with the original dataset.

6.4 Case Study on the Banana Dataset

Fig. 6 shows a case study carried out on the Banana dataset to compare the performance indices obtained by X_O and X_S. In terms of accuracy, F_{65} achieved a competitive result (ACC=0.74) for X_O and obtained better performance than its counterparts except for ENN. For the reduction rate, F_{65} removed more than 85% of the samples (RR=0.86) and surpassed W_{SVM}, CNN, and ENN. It also yielded a competitive reduction rate with regard to W_{kNN}, DROP3, and ICF.

For PDF preservation, F_{65} gave the highest performance (HDC=0.98), and the data points in the feature space show that the selected subset follows the distribution shape of the original dataset, despite the high reduction rate. In contrast, the selected subset generated by ICF had holes and clumps, and produced the worst PDF preservation of all the methods compared here (HDC=0.79).

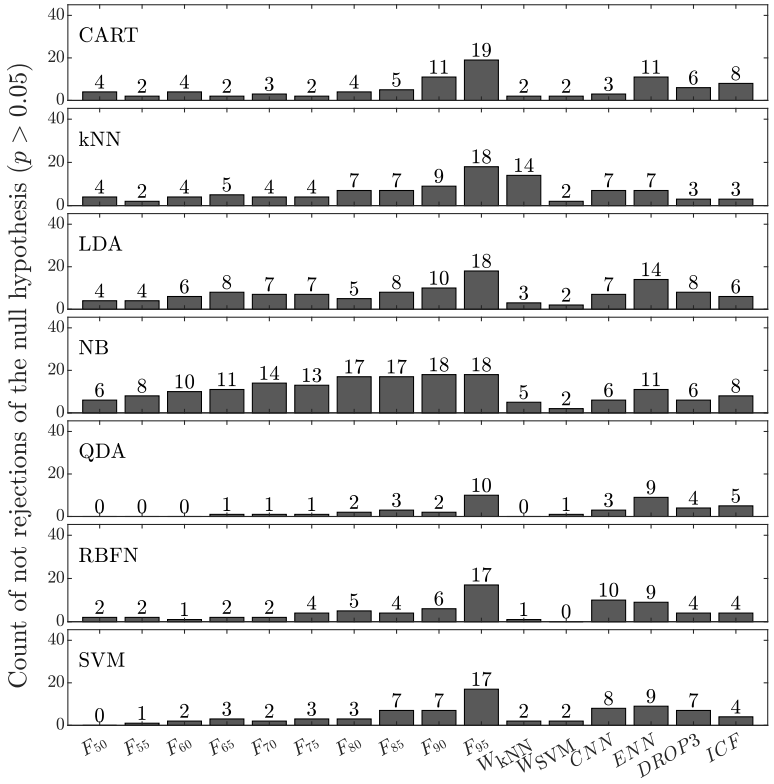

Fig. 5. Counts of no rejections of the null hypothesis of the McNemar test

Finally, the F_{65} method gave the best efficiency (E=0.85) due to the high accuracy, reduction rate, and highest PDF preservation. ENN had the worst efficiency (E=0.44) as it gave the lowest value of the reduction rate (RR=0.12), despite its high accuracy (ACC=0.77) and PDF preservation (HDC=0.96).

Fig. 7 shows a comparison between the X_O and X_S PDFs for each class and dimension of the Banana dataset. The three IS methods yielded the highest HDC values in this case study, namely F_{65}, W_{SVM}, and ENN.

The results show that the proposed approach produced a selected subset that correctly matched the probability distribution of X_O. In contrast, W_{SVM} and ENN gave a set of instances with slightly different distributions regarding X_O, even when these methods obtained a lower reduction rate than F_{65}.

7 Hyperparameters for the W_{SVM} Method

Table 4 shows the hyperparameters used by the W_{SVM} method.

8 Conclusions

This paper has presented an evolutionary IS method called F_w, which maximizes the PDF similarity between the original dataset and the selected subset and minimizes the number of selected instances. In this method, we consider the

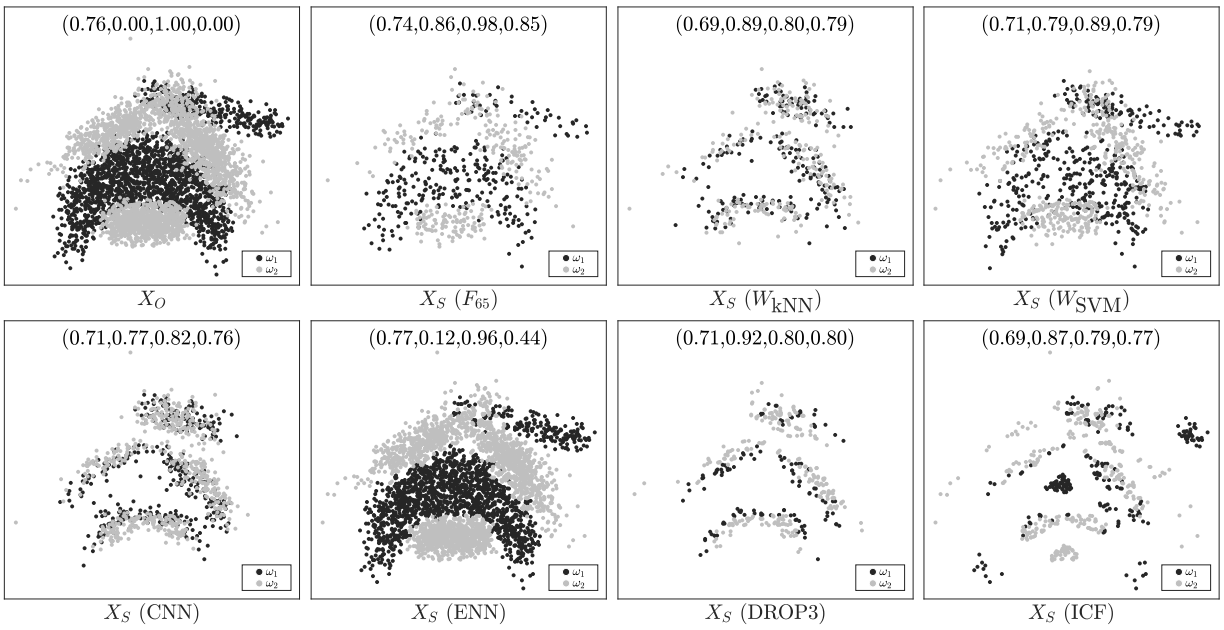

Fig. 6. Instance selection results on the Banana dataset. The average (ACC, RR, HDC, E) measures are shown in parenthesis inside each plot

IS task as an optimization problem, and aim to find subsets of instances that appropriately represent the original samples in the feature space.

We used the Hellinger distance to compare the similarity between two PDFs, since this is a measure of distributional divergence. Thus, we introduced a fitness function that calculates a weighted sum of the Hellinger distance between the original and selected subsets and the reduction rate of the selected subset. To examine the influence of the weight values on the performance, 10 different values were evaluated in the range $w = [0.50, 0.95]$ with steps of 0.05.

The results revealed that these two objectives conflict, i.e., the higher the weight value, the better the PDF preservation but the lower the reduction rate.

Unlike the EA wrapper methods in the literature, the proposed technique does not use a classifier to bias the search process towards samples that maximize the classification accuracy, but instead uses a PDF preservation approach as a novel heuristic. We evaluated the reusability of the obtained subsets to train different classifiers, comparing them against six IS methods: two EA wrappers (W_{kNN} and W_{SVM}) and four classic IS techniques (CNN, ENN, DROP3, and ICF). We also used four performance indices (the average accuracy, reduction rate, PDF preservation, and efficiency) to evaluate and compare the IS methods.

The results revealed that depending on the weight value, the proposed approach was able to outperform the alternative methods.

For instance, F_{95} attained the highest accuracy, while F_{50} obtained the best reduction rate. However, for $0.50 \leq w \leq 0.70$, F_w obtained the higher efficiency, and for all $w > 0.50$, the proposed approach outperformed all the EA wrappers and classical techniques in terms of PDF preservation.

It is worth mentioning that the results in Fig. 4 show that the proposed approach yields better

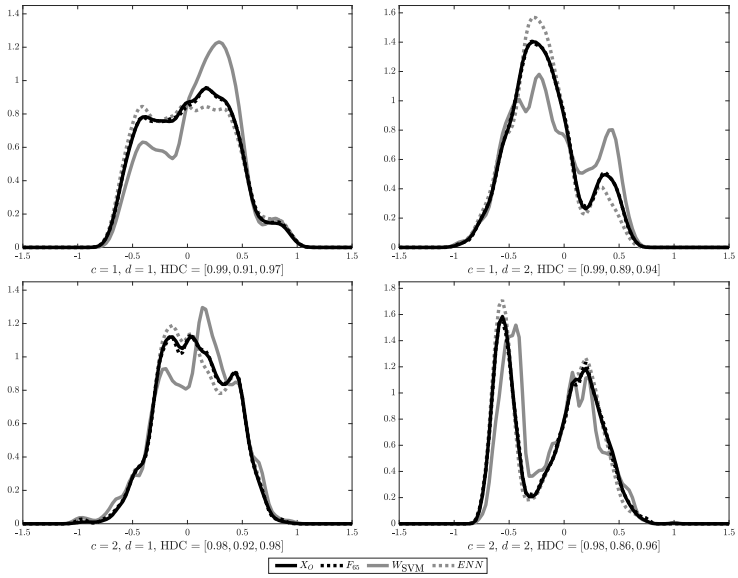

Fig. 7. KDE results by class and predictor variable on the Banana dataset for X_O and X_S subsets obtained with F_{65}, W_{SVM}, and ENN, respectively

generalization of the classification performance of different supervised learning algorithms than the EA wrappers; that is, F_w produces selected instances with good reusability capabilities in terms of training different classifiers.

The results of a McNemar test showed that the F_{95} method gave more subsets that did not have statistical differences from the original datasets than any alternative algorithm; this was due to the weight value in the fitness function ($w = 0.95$), which produced a high PDF preservation but poor performance in terms of the reduction rate.

The F_{50} method attained a regular efficiency on the medium-sized datasets due to its high PDF preservation and low reduction rate. Given the numbers of instances in these larger datasets, the proposed method may require more generations to explore the vast search space, which grows exponentially due to the original patterns.

It will therefore be necessary to investigate new representation schemes for evolutionary IS algorithms that do not explicitly encode all the original dataset instances and reduce the search space size for the IS problem.

Future work could focus on using multiobjective optimization to maximize the PDF preservation and minimize the reduction rate, so that non-dominated Pareto front solutions are obtained to make decisions and choices from among the different possible selected subsets.

Acknowledgments

Samuel Omar Tovias-Alanis thanks the National Council of Science and Technology (CONACyT, Mexico) for the doctoral scholarship.

References

1. **Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S. (2011).** Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. Mult-Valued Log. S., Vol. 17, pp. 255–287.
2. **Aldana-Bobadilla, E., Lopez-Arevalo, I., Molina-Villegas, A. (2017).** A novel data reduction method based on information theory and the eclectic genetic algorithm. Intell. Data Anal., Vol. 21, No. 4, pp. 803–826.

3. **Bishop, C. M. (2006).** Pattern Recognition and Machine Learning. Springer-Verlag, Berlin, Heidelberg.
4. **Brighton, H., Mellish, C. (2002).** Advances in instance selection for instance-based learning algorithms. Data Min. Knowl. Discov., Vol. 6, pp. 153–172.
5. **Cady, F. (2017).** The Data Science Handbook. Wiley.
6. **Cano, J. R., Herrera, F., Lozano, M. (2003).** Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study. IEEE Trans. Evol. Comput., Vol. 7, No. 6, pp. 561–575.
7. **Chang, C.-C., Lin, C.-J. (2011).** LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, Vol. 2, pp. 27:1–27:27.
8. **Cutler, A., Cordero, O. I. (1996).** Minimum Hellinger distance estimation for finite mixture models. J. Am. Stat. Assoc., Vol. 91, No. 436, pp. 1716–1723.
9. **Derrac, J., García, S., Herrera, F. (2010).** A survey on evolutionary instance selection and generation. Int. J. Appl. Metaheuristic Comput., Vol. 1, No. 1, pp. 60–92.
10. **Dua, D., Graff, C. (2017).** UCI machine learning repository.
11. **Duda, R. O., Hart, P. E., Stork, D. G. (2000).** Pattern Classification (2nd Edition). Wiley-Interscience, USA.
12. **García, S., Cano, J. R., Herrera, F. (2008).** A memetic algorithm for evolutionary prototype selection: A scaling up approach. Pattern Recogn., Vol. 41, No. 8, pp. 2693–2709.
13. **Garcia, S., Derrac, J., Cano, J., Herrera, F. (2012).** Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Trans. Pattern Anal. Mach. Intell., Vol. 34, No. 3, pp. 417–435.
14. **Hart, P. (1968).** The condensed nearest neighbor rule. IEEE Trans. Inf. Theory, Vol. 14, No. 3, pp. 515–516.
15. **Kuncheva, L. I. (1995).** Editing for the k-nearest neighbors rule by a genetic algorithm. Pattern Recognit. Lett., Vol. 16, No. 8, pp. 809–814.
16. **MATLAB (2018).** 9.5.0.944444 (R2018b). The MathWorks Inc., Natick, Massachusetts.
17. **Reeves, C. R., Bush, D. R. (2001).** Using Genetic Algorithms for Training Data Selection in RBF Networks. Springer US, Boston, MA, pp. 339–356.
18. **Rosales-Perez, A., Garcia, S., Gonzalez, J. A., Coello, C. A. C., Herrera, F. (2017).** An evolutionary multiobjective model and instance selection for support vector machines with pareto-based ensembles. IEEE Trans. Evol. Comput., Vol. 21, No. 6, pp. 863–877.
19. **Wand, M. P. (1995).** Kernel smoothing. Chapman & Hall, London New York.
20. **Wilson, D. L. (1972).** Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybern. Syst., Vol. SMC-2, No. 3, pp. 408–421.
21. **Wilson, D. R., Martinez, T. R. (2000).** Reduction techniques for instance-based learning algorithms. Mach. Learn., Vol. 38, No. 3, pp. 257–286.
22. **Zhou, Z.-H. (2012).** Ensemble methods: foundations and algorithms. CRC Press.

Article received on 30/09/2021; accepted on 16/12/2021.
Corresponding author is Samuel Omar Tovias-Alanis.