

Article

Assessing Performance and Crossover Operators on Differential Evolution for Attribute Weighting

Andrea Ortega-Guzmán , Salvador Ibarra-Martínez , José Antonio Castan-Rocha * ,
J. David Teran-Villanueva * , Mayra Guadalupe Treviño-Berrones , Aurelio Alejandro Santiago-Pineda 

Facultad de Ingeniería Tampico, Universidad Autónoma de Tamaulipas, Centro Universitario Sur, Tampico 89109, Tamaulipas, Mexico; aoguzman@uat.edu.mx (A.O.-G.); sibarram@docentes.uat.edu.mx (S.I.-M.); mgtrevino@docentes.uat.edu.mx (M.G.T.-B.); aurelio.santiago@uat.edu.mx (A.A.S.-P.)

* Correspondence: jacastan@docentes.uat.edu.mx (J.A.C.-R.); jdteran@docentes.uat.edu.mx (J.D.T.-V.)

Abstract

Machine learning has a wide range of applications, including classification, which categorizes elements based on their characteristics. This paper addresses the challenge of optimizing attribute weighting while assessing two crossover operators on differential evolution optimization and increasing the performance of the k-nearest neighbors classification algorithm (KNN). We use a differential evolution optimization method and assess the performance of both the differential evolution and the harmony crossover operators. Finally, the optimization method uses the accuracy of a KNN classification algorithm as a fitness function. The results show that the proposed method significantly enhances the KNN performance while proposing an alternative for other classification models such as neural networks and Random Forest.

Keywords: differential evolution; crossover operator; KNN; classification; data extraction; heuristics

1. Introduction

Nowadays, machine learning is an essential part of many different applications and technologies. This subfield of Artificial Intelligence involves supervised learning, including techniques such as classification and regression; unsupervised learning, including clustering and dimensionality reduction; and reinforcement learning, applied to medical treatment, navigation systems, and customized recommendations.

Classification can be defined as the logical function of organizing elements into categories or classes according to their characteristics. Classification tasks remain relevant in the field of machine learning, as they have wide applications in different sectors, such as the medical health industry [1–3], the work industry [4,5], the educational field [6–8], the musical industry [9,10], and the fashion field [11,12], among others.

We can categorize classification techniques into complex and simple. In the simple technique subset, we can find logistic regression, k-nearest neighbors (KNN), decision trees, and Naive Bayes. These techniques are easy to implement, have a low computational cost, and are deterministic. Additionally, KNN is a simple distance-based classification algorithm that lacks attribute weighting and commonly does not reach the performance of complex approaches.

In the complex technique subset, we can find Random Forest (RF), support vector machines, and artificial neural networks (ANN) in all their variants—long short-term memory, recurrent neural networks, convolutional neural networks, and adversarial generative



Academic Editor: Tin-Chih Toly Chen

Received: 7 February 2026

Revised: 22 April 2026

Accepted: 2 May 2026

Published: 7 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

networks—which can handle complex and high-dimensional data. Additionally, ANNs modify the attribute values by applying weightings to activate follow-up neurons until they reach the output neurons. These weightings adjust the importance of the attributes, highlighting the most important while diminishing the impact of the less important ones.

These weightings can be applied to other algorithms, e.g., KNN, which is based on distances; the attribute weightings can be of great importance.

The following paragraphs review work on weighing attributes and attribute selection for datasets used with classification or regression techniques.

In 2012, Cong et al. [13] proposed a new attribute importance measure and selection methods based on attribute ranking. For the attribute selection method, the authors applied input–output correlation (IOC) to calculate the attribute importance and then sorted them according to descending order; also, the authors proposed a hybrid of the Back Propagation Neural Network (BPNN) and Particle Swarm Optimization (PSO) to optimize the weights and thresholds of the BPNN for overcoming the inherent shortcoming of the BPNN. Their results showed that the proposed attribute selection method is an effective preprocessing technology.

Jianhua Dai and Qing Xu, in 2013 [14], proposed an attribute selection method based on fuzzy gain ratio under the fuzzy rough set theory framework. The authors compare their approach with other works on three real-world tumor datasets in gene expression. In the comparison among the fuzzy rough set algorithms, the authors found that the proposed algorithm GAIN Ratio Autocorrelation (AR) feature range stretching (FRS) had better classification accuracies than the algorithm GAIN AR FRS using the probabilistic neural network, support vector machines, and the C4.5 decision tree as classifiers. Results showed that fuzzy rough set algorithms perform better than crisp rough set algorithms.

Cheng et al. in 2018 [15] proposed a weighted KNN using weights obtained from Particle Swarm Optimization (PSO) and compared it with attribute selection methods, such as Information Gain and Relief Attribute Evaluation, on regression problems. Results from two datasets, the Physicochemical Properties of Protein Tertiary Structure dataset and the Airfoil Self-Noise dataset, show that PSO performs better than the tested attribute selection methods.

Al-Yaseen et al., in 2021 [16], presented a new breast cancer prediction model based on a modified K-means algorithm and a support vector machine classification algorithm. The model uses a modified K-means algorithm to create a new training dataset of breast cancer to reduce the training time of classifiers and improve the performance of support vector machine classifier. They used two datasets, Wisconsin Breast Cancer and Wisconsin Diagnostic Breast Cancer, to examine and appraise the performance of the proposed model. Results showed that the proposed model performs better than other works and has an accuracy of 98%.

In 2024, Alghamdi et al. [17] proposed an automated ML model for various kinds of cardiovascular disease (CVD) prediction and classification, implementing an arithmetic optimization algorithm as an attribute selection technique to select the best subset of attributes that influence the accuracy of the prediction model, and for the classification task, the authors used a multilayer perceptron neural network to classify the instances of the dataset into two class labels, determining whether they have a CVD or not. Results showed that the proposed model achieved 88.89% prediction accuracy.

Recent studies highlight the continued relevance of improving simple distance-based classifiers through weighting and optimization strategies. In 2025, Abdalla and Amer [18] proposed a locally informed weighted k-nearest neighbors algorithm that integrates neighborhood label information into the weighting process. The proposed approach reduces the influence of outliers and class imbalance by adapting the relevance of the attributes lo-

cally, achieving a superior classification accuracy compared to standard KNN and globally weighted variants.

Amer et al. [19] presented effective KNN-based classification models designed to address class overlap and noisy instances. Their methodology incorporated refined neighbor selection and weighting mechanisms, demonstrating consistent performance improvements across multiple benchmark datasets when compared to traditional KNN and baseline distance-based classifiers.

Sujdha et al. [20] developed an optimized weighted KNN approach for clinical data classification. Their model combined attribute and distance weighting with Bayesian refinement to enhance diagnostic accuracy. Experimental results showed that the optimized weighted KNN outperformed conventional KNN and several existing weighted variants in disease classification tasks.

Lidya et al. [21] introduced a triple-weighted KNN algorithm that simultaneously considers distance, neighbor contribution, and class distribution. The proposed method demonstrated improved robustness and classification accuracy in image-based agricultural datasets when compared to classical KNN and single-weighted approaches.

Additionally, Vega-Huerta et al. [22] applied KNN optimization techniques to environmental data classification. Their study showed that appropriately optimized KNN models can achieve high predictive performance in real-world scenarios, reinforcing the relevance of enhancing simple classifiers through weighting and optimization mechanisms.

In this paper, we propose an optimization methodology to tackle the problem of finding the best weights of a given dataset. This methodology uses a differential evolution heuristic to find the best-known weight for each attribute, with KNN accuracy as the fitness function. This approach integrates a Harmony-based crossover mechanism into a differential evolution framework for continuous feature weights. We position this contribution by directly comparing it with standard differential evolution and representative weighting baselines. This enables us to evaluate the practical impact of the proposed crossover design. We further examine the method's stability across 20 heterogeneous datasets, each with diverse characteristics and classification challenges. In addition, we study the learned weights beyond KNN and treat them as a complementary mechanism for other classifiers.

To clearly position our contribution, this work introduces a hybrid optimization strategy in which the classical crossover operator of differential evolution is replaced by a Harmony-based candidate generation mechanism for continuous attribute weighting.

The main contributions of this work are as follows: (i) the design of a modified crossover mechanism within differential evolution based on Harmony Search principles; (ii) the application of this hybrid strategy to attribute weighting in distance-based classification; (iii) a systematic evaluation across 20 heterogeneous datasets to assess the robustness of the proposed method; and (iv) an analysis of the transferability of the learned weights to other classifiers; however, their performance is dataset-dependent and therefore we treat this approach as an alternative.

The main contribution of this work lies in the design of a hybrid optimization strategy that replaces the classical crossover operator of differential evolution with a Harmony-based candidate generation mechanism. Unlike standard DE, in which crossover combines target and mutant vectors, the proposed approach introduces a memory-driven, dimension-wise sampling process controlled by $HMCR = 0.85$ and $PAR = 0.75$ parameters. This modification changes the exploration–exploitation balance of the algorithm and enables a more adaptive construction of candidate solutions.

In contrast to previous studies that apply weighting strategies or optimization techniques independently, this work focuses on modifying the internal search dynamics of DE for attribute weighting. Additionally, we provide a systematic evaluation across

multiple heterogeneous datasets and compare the proposed method against representative weighting baselines, allowing us to assess its practical impact in distance-based classification scenarios.

The remainder of the paper is as follows: Section 2 presents the datasets used in this work and the preprocessing performed on each one. Section 3 shows our approach for this study. Section 4 contains the results reached for this study, and Section 5 shows our conclusions about this work.

2. Datasets and Preprocessing

We used twenty datasets obtained from publicly available repositories, Kaggle [23] and UCI Machine Learning Repository [24]. The datasets have different sizes, numbers of attributes, and data types. Additionally, all preprocessing steps, including normalization, encoding of categorical variables, and dataset transformations, were applied consistently across all datasets. Table 1 shows the details of the datasets, where the first column presents the dataset name, the second shows the number of numeric attributes, the third column contains the number of nominal attributes, the fourth column indicates the number of nominals with one-hot encoding, the fifth column shows the total number of attributes in the dataset, the sixth and seventh show the total number of instances and the number of classes in each dataset, and the last column presents the class balance entropy for each dataset, calculated using

$$H_k = - \sum_{i=1}^k p_i \log_k(p_i), \quad (1)$$

where p_i is the proportion of instances belonging to class i , and k is the total number of classes.

Table 1. Datasets: NA = numeric attributes, NtO = nominals to ordinal, NtOH = nominals to one-hot encoding, TA = total attributes, NR = number of records, C = classes, Entropy = class balance normalized entropy k .

Dataset Name	NA	NtO	NtOH	TA	NR	C	Entropy
Body Performance	10	0	1	11	13,393	4	1.0000
Caffeine Content of Drink	3	0	0	3	610	6	0.8649
Cirrhosis Patient Survival	11	1	6	18	418	5	0.7949
Cryotherapy Dataset	4	1	1	6	90	2	0.9968
Customer Behavior	7	4	0	11	1125	2	0.7219
Diabetes Multiclass	10	1	0	11	264	3	0.9146
Drivers Licence	10	3	2	15	500	2	1.0000
Drug 200	2	2	1	5	200	5	0.8479
Flavor Sense	1	1	3	5	10,000	4	0.8172
Gender Classification	2	5	0	7	5001	2	1.0000
Heart Failure Clinical	7	5	0	12	918	2	0.9918
Iris	4	0	0	4	150	3	1.0000
ME/CFS vs. Depression	9	4	2	15	1000	3	0.9615
Migraine Classification	1	22	0	23	400	7	0.6639
Online Gaming Behavior	6	14	0	20	40,034	3	0.9561
Personality	5	4	0	9	2900	2	0.9994
Shopping Behavior	4	4	8	16	3900	7	0.9998
Spotify Recommendations	10	3	0	13	195	2	0.9995
Water Potability	9	0	0	9	3276	2	0.9649
Wine	11	0	0	11	1143	6	0.6585

To preprocess the datasets, we analyzed the categorical attributes and selected an encoding strategy based on whether the categories were continuous. In Table 1, the NtO column reports the number of categorical attributes encoded as ordinal variables. We

used ordinal encoding when the categories followed a natural progression or could be interpreted as ordered values, including binary variables (e.g., Yes/No and Male/Female) and variables coded as Good = 1, Regular = 2, and Bad = 3. In contrast, the NtOH column reports the number of categorical attributes encoded with one-hot encoding. We used one-hot encoding when the categories did not preserve continuity and represented distinct groups without a defensible ordinal relationship. We applied these rules to all datasets without further optimization.

To ensure full transparency of the preprocessing stage, we provide the original and preprocessed versions of all datasets in the public repository associated with this work in Section 5. These resources include the categorical variables encoded using ordinal schemes, the resulting numerical representations, and the applied transformations for each dataset.

We used preprocessed datasets obtained with the normalization Equation (2), which normalize values from 1 to 10. We use this scale to avoid dealing with very small floating-point numbers while debugging.

$$Norm(x_i) = \frac{x_i - x_{min}}{x_{max} - x_{min}} * 9 + 1 \quad (2)$$

where $x_i \in X$ is the value i , for attribute X , $x_{min} \in X$ is the minimum value of the X attribute, and $x_{max} \in X$ is the maximum value of the X attribute.

3. Weighting Through Differential Evolution

This study proposes adding the attribute-weighting attribute to the KNN classification algorithm using the differential evolution heuristic optimization technique. The objective is to assign $w_i \in W$ for each $a_i \in A$ in the dataset, where each weight w_i is the weight of every a_i attribute. Additionally, w_i represents the attribute's importance; i.e., larger absolute weights relate to more important attributes.

Figures 1–4 illustrate the attribute-weighting mechanism and the optimization workflow at a conceptual level. Their purpose is to support the methodological explanation of the proposed approach rather than to provide empirical diagnostic visualizations.

Figure 1 shows the process of converting the attributes by adjusting its importance w by multiplying it with a_i to produce a new value aw_i .

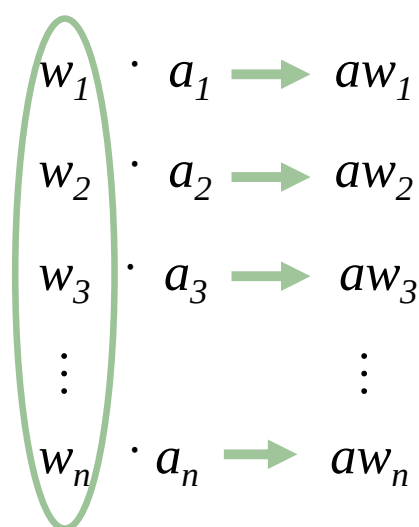


Figure 1. Weighting attributes.

3.1. KNN

The KNN algorithm is a supervised learning algorithm that classifies an element of the dataset based on its distance to other elements of the dataset. Figure 2 shows the data distribution and the way that KNN identifies if a specific element of unknown classification belongs to the category “A” or “B”.

To use this model, we need to calculate the distance among the new element with all the elements in the dataset, select the class of the k nearest elements, and assign the label of the majority vote to the new element.

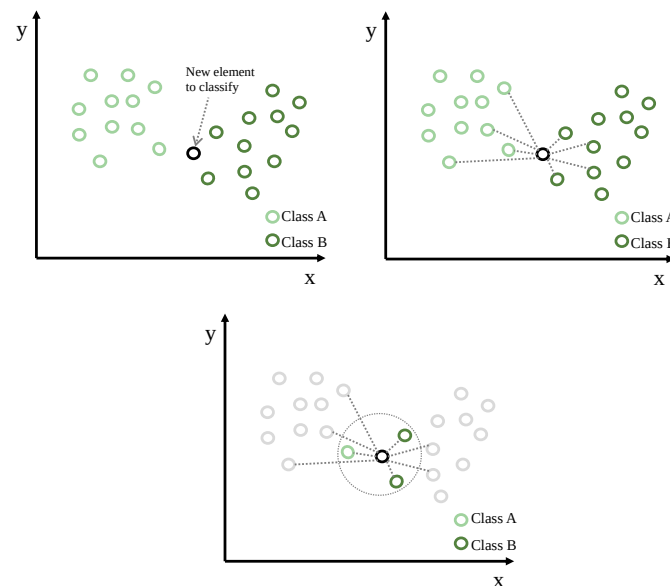


Figure 2. KNN process.

Our proposal is based on the concept of rescaling the attributes, trying to find a scale that enhances the distance classification. Figure 3 shows an example of the rescaling reducing the y-dimensionality.

The adjustment of the attributes weights provides a better representation of the space, visibly distinguishing the elements of each class, producing a more reliable distance-based classification.

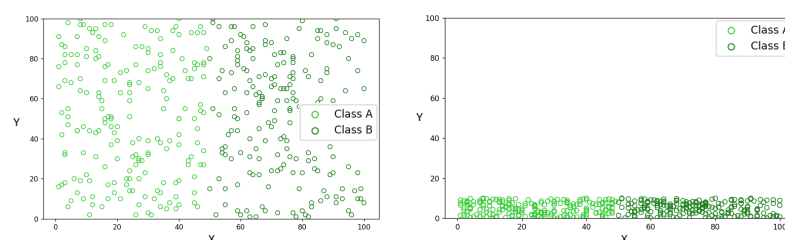


Figure 3. KNN plane distribution.

3.2. Differential Evolution Algorithm

Differential evolution (DE) is a population-based metaheuristic optimization algorithm designed for solving continuous optimization problems. Storn and Price originally proposed DE as a simple yet efficient evolutionary strategy that relies on vector differences to guide the search process [25].

DE operates on a population of candidate solutions, where each individual is represented as a real-valued vector. In this study, each individual corresponds to a candidate weight vector $W = [w_1, w_2, \dots, w_n]$, where each component represents the weight assigned

to a dataset attribute. The optimization process iteratively improves the population through mutation, crossover, and selection operators.

At each generation g , DE generates a mutant vector v_i^g for each target vector x_i^g by combining three distinct individuals randomly selected from the population, r_1, r_2 and r_3 .

$$v_i^g = x_{r_1}^g + F \cdot (x_{r_2}^g - x_{r_3}^g), \quad (3)$$

where $F \in (0, 2)$ is the mutation scaling factor.

Subsequently, the algorithm applies a binomial crossover operation to generate a trial vector u_i^g by combining the mutant vector with the target vector:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & \text{if } \text{rand}_j \leq CR \text{ or } j = j_{rand}, \\ x_{i,j}^g, & \text{otherwise,} \end{cases}$$

where $CR \in [0, 1]$ is the crossover rate and j_{rand} is a randomly selected index that guarantees that at least one component of the trial vector is inherited from the mutant vector.

Finally, the selection step compares the trial vector with the target vector using a predefined fitness function. The vector that achieves better fitness survives to the next generation:

$$x_i^{g+1} = \begin{cases} u_i^g, & \text{if } f(u_i^g) \geq f(x_i^g), \\ x_i^g, & \text{otherwise.} \end{cases}$$

Algorithm 1 summarizes the classical differential evolution procedure.

Algorithm 1 Classical differential evolution algorithm

```

1: Input: Population size  $NP$ , scaling factor  $F$ , crossover rate  $CR$ , stagnation limit  $S$ 
2: Output: Best solution vector  $x_{best}$ 
3: Initialize population  $\{x_i\}_{i=1}^{NP}$  randomly
4: Evaluate each solution using the KNN-based fitness function
5:  $x_{best} \leftarrow$  best solution in the population
6:  $stagnation \leftarrow 0$ 
7: while  $stagnation < S$  do
8:    $improved \leftarrow \text{false}$ 
9:   for  $i = 1$  to  $NP$  do
10:    Select distinct indices  $r_1, r_2, r_3 \neq i$ 
11:    Generate mutant vector  $v_i$ 
12:    Apply binomial crossover with mandatory index  $j_{rand}$  to generate trial vector  $u_i$ 
13:    Evaluate  $u_i$  using the KNN-based fitness function
14:    if  $f(u_i) > f(x_i)$  then
15:       $x_i \leftarrow u_i$ 
16:       $improved \leftarrow \text{true}$ 
17:      if  $f(x_i) > f(x_{best})$  then
18:         $x_{best} \leftarrow x_i$ 
19:      end if
20:    end if
21:  end for
22:  if  $improved$  then
23:     $stagnation \leftarrow 0$ 
24:  else
25:     $stagnation \leftarrow stagnation + 1$ 
26:  end if
27: end while
28: return  $x_{best}$ 

```

3.3. Differential Evolution with Harmony-Based Crossover (DE-HX)

Although the classical DE procedure uses binomial crossover with a mandatory inherited component, the proposed DE-HX replaces that stage with a Harmony-based candidate generation mechanism. The DE-HX approach differs from classical differential evolution by modifying the crossover mechanism, introducing a Harmony-inspired candidate generation process that alters the construction of new solutions.

In this section, we define DE-HX as a hybrid optimization approach that integrates differential evolution (DE) with a Harmony-inspired candidate generation mechanism for attribute weighting.

DE-HX replaces the classical crossover stage of DE. It does not introduce an additional recombination operator after crossover, nor does it maintain a separate Harmony Memory structure in parallel with the DE population. Instead, the current DE population serves as the source memory from which candidate components are sampled.

Each candidate solution represents a vector of continuous attribute weights. For each current solution W_i , DE-HX generates a new candidate vector W_{new} in a dimension-wise manner. For each component j , the algorithm selects a value from another solution W_r in the current population, with $r \neq i$, with probability HMCR (Harmony Memory Considering Rate); otherwise, it generates a random value within the predefined search bounds $[L_b, U_b]$. Additionally, with probability PAR (Pitch Adjusting Rate), the selected component is locally adjusted to promote exploration. After each update, the component is clipped to remain within the valid bounds.

Once W_{new} is generated, the algorithm evaluates it using the KNN-based fitness function. If $fitness(W_{new}) > fitness(W_i)$, then the current solution W_i is replaced by W_{new} . If the updated solution also improves the best solution found so far, the algorithm updates W_{best} accordingly.

This procedure preserves the population-based iterative structure of DE while replacing the standard crossover stage with a Harmony-inspired mechanism for constructing candidate solutions.

To make the optimization process fully reproducible, Algorithm 2 summarizes the complete DE-HX procedure, including population initialization, Harmony-based candidate generation, KNN-based fitness evaluation, local replacement of solutions, and the stagnation-based stopping criterion. In this formulation, the current DE population is the only memory structure used during the search.

Figure 4 shows the evaluation process for the initial weightings; it begins by multiplying the dataset by the proposed weights, yielding a new weighted dataset, and then the KNN algorithm obtains its accuracy. This process continues with the following weights.

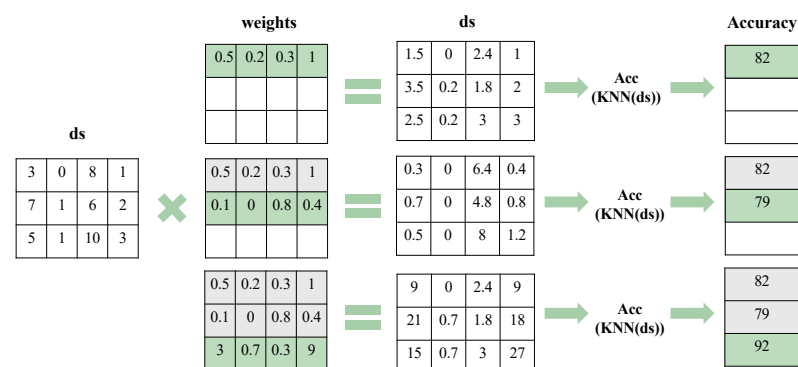


Figure 4. Evaluation of initial weights.

Algorithm 2 Differential evolution with Harmony-based crossover (DE-HX)

```

1: Input: Population size  $NP$ , lower bound  $L_b$ , upper bound  $U_b$ , HMCR, PAR, stagnation
   limit  $S$ 
2: Output: Best weight vector  $W_{best}$ 
3: Initialize population  $W = \{W_1, W_2, \dots, W_i\}$  randomly in  $[L_b, U_b]$ 
4: Evaluate each solution using the KNN-based fitness function
5:  $W_{best} \leftarrow$  best solution in  $W$ 
6:  $stagnation \leftarrow 0$ 
7: while  $stagnation < S$  do
8:    $improved \leftarrow \text{false}$ 
9:   for each solution  $W_i \in W$  do
10:    Generate empty candidate vector  $W_{new}$ 
11:    for each dimension  $j$  do
12:      if  $rand() \leq HMCR$  then
13:        Select another solution  $W_r$  from the current population, with  $r \neq i$ 
14:         $W_{new}[j] \leftarrow W_r[j]$ 
15:        if  $rand() \leq PAR$  then
16:          Apply local adjustment to  $W_{new}[j]$ 
17:        end if
18:      else
19:         $W_{new}[j] \leftarrow$  random value in  $[L_b, U_b]$ 
20:      end if
21:      Clip  $W_{new}[j]$  to  $[L_b, U_b]$ 
22:    end for
23:    Evaluate  $W_{new}$  using the KNN-based fitness function
24:    if  $fitness(W_{new}) > fitness(W_i)$  then
25:      Replace  $W_i \leftarrow W_{new}$ 
26:       $improved \leftarrow \text{true}$ 
27:      if  $fitness(W_i) > fitness(W_{best})$  then
28:         $W_{best} \leftarrow W_i$ 
29:      end if
30:    end if
31:  end for
32:  if  $improved$  then
33:     $stagnation \leftarrow 0$ 
34:  else
35:     $stagnation \leftarrow stagnation + 1$ 
36:  end if
37: end while
38: return  $W_{best}$ 

```

3.4. Reproducibility and Code Availability

To ensure transparency and reproducibility, the implementation of the proposed DE-HX algorithm is made available through a public repository presented in Section 5.

All experiments were conducted using Python 3.13.4 and standard machine learning libraries, including scikit-learn. The main hyperparameters used in the experiments are explicitly defined in the manuscript, including the number of neighbors in KNN, the distance metric, and the parameters of the DE-HX algorithm.

The datasets were obtained from Kaggle and the UCI Machine Learning Repository, and correspond to the versions available during the period January–March 2026. Table 1 provides the list of datasets used in this study.

The link to the repository, for reproducibility, is shown in Section 5.

4. Results

In this section, we show the results and statistical tests for our proposal. The computational experimentation was carried out on a computer with an Intel(R) Core(TM) i5-8265U CPU@1.60GHz 1.80 GHz processor, 8.00 GB of RAM, an operational system of 64 bits, an x64 processor, Windows 11 Home Edition, 23H2 Version, 22631.3447 OS Compilation, and a Windows attribute Experience Pack 1000.22688.1000.0.

4.1. Experimental Protocol

To ensure a consistent and reproducible evaluation, all experiments were conducted using a k-fold cross-validation scheme with $k = 5$. The dataset is randomly shuffled before splitting, using a fixed random seed to guarantee reproducibility across runs.

The DE and DE-HX configurations use fixed hyperparameters across all experiments. For classical DE, we used population size $NP = 12$, scaling factor $F = 1$, and weight initialization within the interval $[0, 1)$. For DE-HX, we used the same population size and initialization range, together with $HMCR = 0.85$ and $PAR = 0.75$. In both methods, the candidate weights were constrained to the interval $[0, 1)$, and all experiments used random seed = 42.

During the optimization process, each candidate solution, representing a set of attribute weights, is evaluated using the k-nearest neighbors (KNN) classifier, using $k = 5$ neighbors and Euclidean as the distance metric. The fitness function corresponds to the average classification accuracy obtained across the cross-validation folds.

For the final evaluation of the best solution, we generate out-of-fold predictions using cross-validation. This approach allows us to compute multiple evaluation metrics, including accuracy and Macro-F1, providing a balanced evaluation across all classes, and a more robust assessment of classification performance.

Both differential evolution (DE) and differential evolution with harmony crossover (DE-HX) use the same initialization scheme, fitness function, random seed policy, and stopping criterion, providing a consistent comparison framework across methods. All experiments were conducted using a fixed random seed. The stopping criterion for both DE and DE-HX is based on stagnation detection. Specifically, the optimization process stops when no improvement in the fitness function is observed for a predefined number of consecutive iterations. Therefore, the comparison focuses on the quality of the obtained solutions under a common optimization protocol rather than on runtime or a strictly fixed computational budget. Although multiple independent runs and nested cross-validation were not performed due to computational limitations, the use of k-fold cross-validation with fixed seeds provides a consistent evaluation framework across all methods.

4.2. Differential Evolution (DE) vs. Differential Evolution with Harmony Crossover (DE-HX)

Here, we compare our proposal's methodology with the standard KNN and other complex classification methods, such as ANN and Random Forests.

Table 2 compares the classification accuracy, and the Macro-F1 obtained by differential evolution using the harmony crossover operator (DE-HX) against standard differential evolution (DE) across twenty datasets. The results indicate that, in general, DE-HX outperforms the standard DE approach in the majority of cases; notable performance gains are observed on datasets such as Cryotherapy, Diabetes Multiclass, Drug 200, ME/CFS vs. Depression and Migraine Classification, where DE-HX achieves substantially higher accuracy values. We highlighted the best-performing accuracy algorithm in green and the best-performing Macro-F1 in yellow.

Table 2. Comparison between differential evolution with harmony crossover, and differential evolution.

Dataset	DE-HX		DE	
	Accuracy	Macro-F1	Accuracy	Macro-F1
Body Performance	68.44	68.37	69.67	67.37
Caffeine Content of Drink	69.67	67.54	69.67	67.37
Cirrhosis Patient Survival	61.28	39.37	55.76	34.67
Cryotherapy Dataset	97.78	97.76	92.22	92.14
Customer Behavior	80.89	54.02	88.98	58.77
Diabetes Multiclass	98.87	99.09	96.23	95.3
Drivers Licence	88	87.95	86.2	86.14
Drug 200	99	98.6	97.5	97.29
Flavor Sense	91.53	89.75	90.61	88.26
Gender Classification	97.7	97.7	97.6	97.56
Heart Failure Clinical	90.3	90.12	88.12	87.93
Iris	98	98	98	98
ME/CFS vs. Depression	99.6	99.5	90.8	87.36
Migraine Classification	91.5	78.17	86.25	62.54
Online Gaming Behavior	91.92	91.44	88.29	87.78
Personality	93.38	93.24	93.41	93.24
Shopping Behavior	20.08	19.59	17.23	16.74
Spotify Recommendations	96.41	96.4	93.85	93.84
Water Potability	68.16	64.61	65.96	61.92
Wine	63.51	30.43	60.89	28.27

We use gradients to highlight large values, Green for Accuracy and Yellow for Macro-F1.

The differential evolution approach enhanced with the harmony crossover (DE-HX) consistently achieves higher classification accuracy than standard differential evolution, demonstrating the effectiveness of the proposed crossover strategy for attribute-weighting optimization.

4.3. WeightedKNN and ReliefF+KNN vs. Differential Evolution with Harmony Crossover (DE-HX)

To provide a stronger baseline for our results, we include widely used weighting techniques in the comparison. Tables 3 and 4 compare accuracy and Macro-F1 performance, respectively, for our proposal (DE-HX) with a WeightedKNN implementation of the sklearn.neighbors library in Python; we used KNeighborsClassifier with $k = 5$ and distance-based voting (weights = “distance”) and the ReliefF+KNN baseline implementation with the skrebate library, and we computed ReliefF weights with 5 n_neighbors and n_features_to_select equal to the full number of input features. We clipped negative weights to zero, applied the resulting weights to the feature space, and trained a KNN classifier with $k = 5$; both techniques use a 5-fold cross-validation. We highlighted the best-performing accuracy in green and the best Macro-F1 in yellow for each dataset.

We selected these baselines because they represent commonly used and well-established attribute-weighting strategies in distance-based classification, allowing a meaningful comparison with the proposed DE-HX approach.

In general, Tables 3 and 4 reinforce that DE-HX delivers strong overall performance across most datasets among the weighting strategies evaluated and provides a stronger feature-weighting scheme for distance-based classification than distance-weighted KNN and ReliefF+KNN in the same experimental setting, showing the best performance against weighted methods, in datasets like Cirrhosis Patient Survival, Diabetes Multiclass, Heart Failure Clinical, Migraine Classification, and Spotify Recommendations.

Table 3. Accuracy comparison among differential evolution with harmony crossover, WeightedKNN and ReliefF+KNN.

Dataset	DE-HX	Accuracy	
		WeightedKNN	ReliefF+KNN
Body Performance	68.44	56.19	65.86
Caffeine Content of Drink	69.67	69.02	67.54
Cirrhosis Patient Survival	61.28	48.09	48.56
Cryotherapy Dataset	97.78	87.78	82.22
Customer Behavior	80.89	77.33	77.69
Diabetes Multiclass	98.87	85.98	92.42
Drivers Licence	88	82	83
Drug 200	99	84.5	96.5
Flavor Sense	91.53	88.57	90.18
Gender Classification	97.7	96.72	96.8
Heart Failure Clinical	90.3	84.75	83.44
Iris	98	95.33	95.33
ME/CFS vs. Depression	99.6	65.9	96
Migraine Classification	91.5	79.5	81
Online Gaming Behavior	91.92	83.4	91.25
Personality	93.38	91.76	92.79
Shopping Behavior	20.08	15.56	14.13
Spotify Recommendations	96.41	90.26	91.79
Water Potability	68.16	62.55	64.96
Wine	63.51	62.12	55.29

We use Green gradients to highlight large values for Accuracy.

Table 4. Macro-F1 comparison among differential evolution with harmony crossover, WeightedKNN and ReliefF+KNN.

Dataset	DE-HX	Macro-F1	
		WeightedKNN	ReliefF+KNN
Body Performance	68.37	56.43	65.82
Caffeine Content of Drink	67.54	66.17	61.49
Cirrhosis Patient Survival	39.37	29.12	27.88
Cryotherapy Dataset	97.76	87.74	82.21
Customer Behavior	54.02	51.1	48.01
Diabetes Multiclass	99.09	80.31	89.35
Drivers Licence	87.95	81.96	82.91
Drug 200	98.6	78.5	94.35
Flavor Sense	89.75	86.3	88.37
Gender Classification	97.7	96.72	96.8
Heart Failure Clinical	90.12	84.5	83.21
Iris	98	95.33	95.33
ME/CFS vs. Depression	99.5	60.15	94.85
Migraine Classification	78.17	46.23	45.97
Online Gaming Behavior	91.44	82.82	90.78
Personality	93.24	91.75	92.79
Shopping Behavior	19.59	15.54	13.76
Spotify Recommendations	96.4	90.23	91.79
Water Potability	64.61	57.76	61.44
Wine	30.43	34.03	26.06

We use Yellow gradients to highlight large values for Macro-F1.

4.4. Classification Algorithms vs. Differential Evolution with Harmony Crossover (DE-HX) and Non-Parametric Tests

For more sustainable results, we compare our approach with three classification algorithms: KNN, ANN, and Random Forest.

For the standard KNN, we used the KNeighborsClassifier with $k = 5$ and the default Euclidean distance. For the ANN, we used scikit-learn's MLPClassifier with two hidden layers of 10 neurons each, ReLU activation, the Adam optimizer, a maximum of 1000 iterations, and `random_state = 42`. Unless otherwise stated, the remaining parameters were kept at their default values. For the RF, we used the RandomForestClassifier with 100 trees and `random_state = 42`, while the remaining parameters were kept at their default values. These three techniques were evaluated using 5-fold cross-validation.

Tables 5 and 6 show the accuracy and Macro-F1 performance comparison evaluation of the DE-HX against KNN, ANN, and Random Forest. The first column presents each dataset used in this study; in the second column, we show the performance of the KNN algorithm, which serves as a baseline for classification performance, followed by our proposal DE-HX, ANN, and RF, each of which is considered complex. In particular, DE-HX achieves competitive performance across many datasets when compared with the evaluated classifiers. For easy visualization, we highlight the highest accuracies obtained with these datasets using a green gradient, and the highest Macro-F1 performance using a yellow gradient.

In several cases, such as Cirrhosis Patient Survival, Cryotherapy Dataset, Diabetes Multiclass, Drivers Licence, Shopping Behavior, and Spotify Recommendations, the DE-HX approach achieves the highest classification accuracy among all evaluated methods.

Table 5. Accuracy comparison among k-nearest neighbors, differential evolution with harmony crossover, artificial neural network and Random Forest.

Dataset	Accuracy			
	KNN	DE-HX	ANN	RF
Body Performance	55.69	68.44	65.53	73.75
Caffeine Content of Drink	67.21	69.67	65.9	72.62
Cirrhosis Patient Survival	50.24	61.28	47.61	52.39
Cryotherapy Dataset	86.67	97.78	85.56	92.22
Customer Behavior	77.33	80.89	79.38	78.31
Diabetes Multiclass	80.88	98.87	80.3	96.97
Drivers Licence	82	88	81	81
Drug 200	80	99	92.5	99
Flavor Sense	87.51	91.53	91.19	89.92
Gender Classification	96.88	97.7	96.84	96.74
Heart Failure Clinical	84.53	90.3	85.51	86.38
Iris	96	98	92	94.67
ME/CFS vs. Depression	65.6	99.6	92.5	99.2
Migraine Classification	78.5	91.5	89.5	87.5
Online Gaming Behavior	83.4	91.92	91.78	90.17
Personality	92.79	93.38	93.41	91.93
Shopping Behavior	15.28	20.08	14.69	14.44
Spotify Recommendations	90.26	96.41	90.26	92.82
Water Potability	62.55	68.16	66	66.91
Wine	55.82	63.51	62.64	66.67

We use Green gradients to highlight large values for Accuracy.

Non-parametric tests analyze data that do not have a specific distribution and are suitable for small datasets. The Friedman test analyzes multiple samples, and the Wilcoxon signed-rank test compares the mean rank of two related samples to determine if there are statistical differences between them.

Table 6. Macro-F1 comparison among k-nearest neighbors, differential evolution with harmony crossover, artificial neural network and Random Forest.

Dataset	Macro-F1			
	KNN	DE-HX	ANN	RF
Body Performance	55.67	68.37	65.32	73.72
Caffeine Content of Drink	62.57	67.54	62.21	65.95
Cirrhosis Patient Survival	27.23	39.37	28.02	29.67
Cryotherapy Dataset	86.66	97.76	85.51	92.21
Customer Behavior	47.83	54.02	44.25	48.65
Diabetes Multiclass	80.88	99.09	69.07	96.59
Drivers Licence	81.96	87.95	80.97	80.96
Drug 200	71.47	98.6	89.36	98.63
Flavor Sense	85.67	89.75	89.05	87.53
Gender Classification	96.88	97.7	96.84	96.74
Heart Failure Clinical	84.27	90.12	85.29	86.16
Iris	96	98	92.04	94.66
ME/CFS vs. Depression	60.34	99.5	90.12	99.01
Migraine Classification	40.96	78.17	75.57	68.13
Online Gaming Behavior	82.76	91.44	91.34	89.65
Personality	92.79	93.24	93.41	91.92
Shopping Behavior	14.98	19.59	14.61	14.36
Spotify Recommendations	90.23	96.4	90.25	92.81
Water Potability	57.76	64.61	58.58	60.5
Wine	26.37	30.43	29.64	35.89

We use Yellow gradients to highlight large values for Macro-F1.

We use the Friedman test to compare the general performance of the techniques used for this study. Table 7 presents the results of the Friedman non-parametric statistical test, ranking seven evaluated techniques according to their overall performance across the 20 datasets used. This study revealed statistically significant differences among the seven evaluated techniques ($N = 20$, $k = 7$, $\chi^2(6) = 61.870$, $p < 0.001$). DE-HX achieved the highest average rank of 6.58, followed by DE and RF with 5.25 and 4.58, respectively.

Table 7. Comparison ranking of the seven techniques applied in this study evaluated with the Friedman test.

Order	Model	Ranking
1	DE-HX	6.58
2	DE	5.25
3	RF	4.58
4	ANN	3.50
5	ReliefF+KNN	3.20
6	WeightedKNN	2.53
7	KNN	2.38

This ranking indicates that the proposed DE-HX achieves the best overall average performance across the evaluated datasets.

To control the family-wise error rate across multiple pairwise comparisons, we applied the Holm post hoc correction using DE-HX as the control method. Table 8 shows the Holm post hoc comparisons using DE-HX as the control method. After controlling the family-wise error rate, DE-HX significantly outperformed RF, ANN, ReliefF+KNN, WeightedKNN, and KNN. The difference between DE-HX and standard DE barely loses significance after Holm adjustment with $p = 0.0515$.

Table 8. Post hoc comparisons with differential evolution with harmony crossover against techniques applied in this study.

Comparison	z	Raw <i>p</i> -Value	Holm-Adjusted <i>p</i> -Value
DE-HX vs. DE	1.947	0.0515	0.0515
DE-HX vs. RF	2.928	0.0034	0.0068
DE-HX vs. ANN	4.509	0.00000652	0.0000196
DE-HX vs. ReliefF+KNN	4.948	6.45×10^{-7}	0.00000258
DE-HX vs. WeightedKNN	5.929	3.06×10^{-9}	1.53×10^{-8}
DE-HX vs. KNN	6.148	7.84×10^{-10}	4.70×10^{-9}

Table 9 reports the results of the Wilcoxon signed-rank test comparing DE-HX with the standard differential evolution and the Random Forest. The table includes the Wilcoxon statistic (W), the standardized test statistic (Z), the two-tailed *p*-value, the effect size (r), the number of wins (W), ties (T), and losses (L) across datasets, and the median performance difference.

Table 9. Wilcoxon signed-rank test comparison between differential evolution with harmony crossover against differential evolution and Random Forest.

Comparison	W	Z	<i>p</i> -Value	Effect Size (r)	W/T/L	Median Difference
DE-HX vs. DE	22	−2.765	0.006	0.652	15/2/3	2.19
DE-HX vs. RF	34	−2.455	0.014	0.563	16/1/3	1.83

The direct Wilcoxon signed-rank comparison between DE-HX and standard DE still showed a statistically significant difference in favor of DE-HX ($Z = -2.765$, $p = 0.006$, $r = 0.652$), which suggests that the Harmony-based crossover improves DE in the primary pairwise comparison, although this effect got reduced under the Holm statistical test. However, we believe this difference may exceed the $p = 0.05$ threshold with additional datasets.

The reported effect size (r) quantifies the magnitude of the difference between methods. Values above 0.5 indicate a large effect, suggesting that the observed improvements are not only statistically significant but also practically meaningful.

4.5. Differential Evolution with Harmony Crossover (DE-HX) Extraction Dataset vs. Preprocessed Dataset in Classification Algorithms

Tables 10 and 11 present the improvement in accuracy and Macro-F1 achieved by KNN after applying the DE-HX attribute-weighting strategy. These tables report the performance obtained by KNN and DE-HX-weighted, along with the corresponding improvement percentage for each dataset. We highlight in a green gradient the highest accuracy and in a yellow gradient the highest Macro-F1 across the 20 datasets.

When comparing DE-HX with the baseline classifiers, the results indicate that k-nearest neighbors is the technique that benefits most clearly from the optimized weighting scheme. As shown in Tables 10 and 11, the weighted representation produced by DE-HX improves KNN performance across most datasets, with larger gains in datasets where attribute relevance varies more strongly.

Tables 12 and 13 report the impact of the DE-HX on artificial neural network performance by comparing baseline ANN accuracy with the accuracy obtained after applying the weighted dataset, in the second and third columns, respectively. Finally, the fourth column includes the differences between DE-HX → ANN and ANN. We highlight in green the best accuracy performance obtained with these datasets and indicate in yellow color the best Macro-F1 performance.

Table 10. Accuracy comparison between weighted k-nearest neighbors, differential evolution with harmony crossover, and the improvement percentage.

Dataset	KNN	Accuracy DE-HX	Imp Percent
Body Performance	55.69	68.44	12.75
Caffeine Content of Drink	67.21	69.67	2.46
Cirrhosis Patient Survival	50.24	61.28	11.04
Cryotherapy Dataset	86.67	97.78	11.11
Customer Behavior	77.33	80.89	3.56
Diabetes Multiclass	80.88	98.87	17.99
Drivers Licence	82	88	6
Drug 200	80	99	19
Flavor Sense	87.51	91.53	4.02
Gender Classification	96.88	97.7	0.82
Heart Failure Clinical	84.53	90.3	5.77
Iris	96	98	2
ME/CFS vs. Depression	65.6	99.6	34
Migraine Classification	78.5	91.5	13
Online Gaming Behavior	83.4	91.92	8.52
Personality	92.79	93.38	0.59
Shopping Behavior	15.28	20.08	4.8
Spotify Recommendations	90.26	96.41	6.15
Water Potability	62.55	68.16	5.61
Wine	55.82	63.51	7.69

We use Green gradients to highlight large values for Accuracy.

Table 11. Macro-F1 comparison between weighted k-nearest neighbors, differential evolution with harmony crossover, and the improvement percentage.

Dataset	KNN	Macro-F1 DE-HX	Imp Percent
Body Performance	55.67	68.37	12.7
Caffeine Content of Drink	62.57	67.54	4.97
Cirrhosis Patient Survival	27.23	39.37	12.14
Cryotherapy Dataset	86.66	97.76	11.1
Customer Behavior	47.83	54.02	6.19
Diabetes Multiclass	80.88	99.09	18.21
Drivers Licence	81.96	87.95	5.99
Drug 200	71.47	98.6	27.13
Flavor Sense	85.67	89.75	4.08
Gender Classification	96.88	97.7	0.82
Heart Failure Clinical	84.27	90.12	5.85
Iris	96	98	2
ME/CFS vs. Depression	60.34	99.5	39.16
Migraine Classification	40.96	78.17	37.21
Online Gaming Behavior	82.76	91.44	8.68
Personality	92.79	93.24	0.45
Shopping Behavior	14.98	19.59	4.61
Spotify Recommendations	90.23	96.4	6.17
Water Potability	57.76	64.61	6.85
Wine	26.37	30.43	4.06

We use Yellow gradients to highlight large values for Macro-F1.

Artificial neural networks exhibit a more heterogeneous response to the proposed weighting strategy. As reported in Tables 12 and 13, ANN performance improves notably in specific datasets, including Body Performance, Cryotherapy, Drug 200 and Me/cfs vs. Depression. In several other datasets, the performance of the ANN shows only minor

reductions, indicating that the benefits of attribute weighting depend on the interaction between the optimized data learning dynamics of the neural model.

Table 12. Accuracy comparison between artificial neural network, DE-HX-weighted dataset, and the improvement percentage.

Dataset	Accuracy		Imp Percent
	ANN	DE-HX → ANN	
Body Performance	65.53	69.03	3.5
Caffeine Content of Drink	65.9	67.16	1.26
Cirrhosis Patient Survival	47.61	45.32	−2.29
Cryotherapy Dataset	85.56	93.26	7.7
Customer Behavior	79.38	79.54	0.16
Diabetes Multiclass	80.3	86.31	6.01
Drivers Licence	81	81.56	0.56
Drug 200	92.5	94.47	1.97
Flavor Sense	91.19	91.19	0
Gender Classification	96.84	96.92	0.08
Heart Failure Clinical	85.51	85.61	0.1
Iris	92	93.96	1.96
ME/CFS vs. Depression	92.5	98.7	6.2
Migraine Classification	89.5	82.46	−7.04
Online Gaming Behavior	91.78	91.47	−0.31
Personality	93.41	93.45	0.04
Shopping Behavior	14.69	13.95	−0.74
Spotify Recommendations	90.26	88.66	−1.6
Water Potability	66	64.76	−1.24
Wine	62.64	61.03	−1.61

We use Green gradients to highlight large values for Accuracy.

Table 13. Macro-F1 comparison between artificial neural network, DE-HX-weighted dataset, and the improvement percentage.

Dataset	Macro-F1		Imp Percent
	ANN	DE-HX → ANN	
Body Performance	65.32	68.98	3.66
Caffeine Content of Drink	62.21	60.76	−1.45
Cirrhosis Patient Survival	28.02	29.28	1.26
Cryotherapy Dataset	85.51	93.22	7.71
Customer Behavior	44.25	44.73	0.48
Diabetes Multiclass	69.07	77.04	7.97
Drivers Licence	80.97	81.54	0.57
Drug 200	89.36	91.25	1.89
Flavor Sense	89.05	88	−1.05
Gender Classification	96.84	96.92	0.08
Heart Failure Clinical	85.29	85.36	0.07
Iris	92.04	94	1.96
ME/CFS vs. Depression	90.12	98.39	8.27
Migraine Classification	75.57	66.27	−9.3
Online Gaming Behavior	91.34	91	−0.34
Personality	93.41	93.44	0.03
Shopping Behavior	14.61	13.8	−0.81
Spotify Recommendations	90.25	88.63	−1.62
Water Potability	58.58	57.28	−1.3
Wine	29.64	29.38	−0.26

We use Yellow gradients to highlight large values for Macro-F1.

In this transferability analysis, the learned DE-HX weights are applied to ANN and RF without an additional model-specific re-tuning stage. Therefore, these results should be interpreted as an exploratory assessment of transferability rather than as a fully optimized comparison for those classifiers.

Tables 14 and 15 analyze the effect of the DE-HX-weighted dataset on Random Forest classification accuracy and Macro-F1 performance. The results demonstrate that RF achieves modest improvements across a limited number of datasets. In many datasets, performance remains unchanged or slightly decreases after weighting. We highlight in a green gradient the highest accuracies achieved with these techniques, and use yellow color to indicate the Macro-F1 improvement.

Table 14. Accuracy comparison between Random Forest, DE-HX-weighted dataset, and the improvement percentage.

Dataset	Accuracy		Imp Percent
	RF	DE-HX → RF	
Body Performance	73.75	73.22	−0.53
Caffeine Content of Drink	72.62	72.74	0.12
Cirrhosis Patient Survival	52.39	50.36	−2.03
Cryotherapy Dataset	92.22	92.13	−0.09
Customer Behavior	78.31	79.72	1.41
Diabetes Multiclass	96.97	97.72	0.75
Drivers Licence	81	81.76	0.76
Drug 200	99	98.99	−0.01
Flavor Sense	89.92	90.12	0.2
Gender Classification	96.74	96.8	0.06
Heart Failure Clinical	86.38	87.79	1.41
Iris	94.67	95.97	1.3
ME/CFS vs. Depression	99.2	99.2	0
Migraine Classification	87.5	88.72	1.22
Online Gaming Behavior	90.17	90	−0.17
Personality	91.93	91.93	0
Shopping Behavior	14.44	14.41	−0.03
Spotify Recommendations	92.82	92.27	−0.55
Water Potability	66.91	67.45	0.54
Wine	66.67	65.67	−1

We use Green gradients to highlight large values for Accuracy.

These results align with the internal mechanisms of Random Forest, which already incorporate random attribute selection and bagging, reducing the marginal benefit of prior attribute weighting.

This behavior suggests that externally learned attribute weights do not necessarily align with the internal feature selection and partitioning mechanisms of Random Forest, particularly when the model is evaluated without an additional re-tuning stage after weighting.

Beyond the differences observed in accuracy and Macro-F1, the results show that the proposed DE-HX attribute-weighting strategy can also work as a complementary element in ensemble-oriented schemes. Instead of relying only on the output of an artificial neural network, the optimized attribute weights provide an explicit representation of attribute relevance that may support the final decision in hybrid models. This perspective is particularly relevant for datasets in which ANN shows only limited sensitivity to the weighted representation, since externally optimized weights may still provide useful information when combined with neural predictions in decision fusion strategies.

Table 15. Macro-F1 comparison between Random Forest, DE-HX-weighted dataset, and the improvement percentage.

Dataset	Macro-F1		Imp Percent
	RF	DE-HX → RF	
Body Performance	73.72	73.19	−0.53
Caffeine Content of Drink	65.95	68.52	2.57
Cirrhosis Patient Survival	29.67	29.01	−0.66
Cryotherapy Dataset	92.21	92.12	−0.09
Customer Behavior	48.65	51.39	2.74
Diabetes Multiclass	96.59	97.45	0.86
Drivers Licence	80.96	81.71	0.75
Drug 200	98.63	98.63	0
Flavor Sense	87.53	87.7	0.17
Gender Classification	96.74	96.8	0.06
Heart Failure Clinical	86.16	87.58	1.42
Iris	94.66	96	1.34
ME/CFS vs. Depression	99.01	99.01	0
Migraine Classification	68.13	72.06	3.93
Online Gaming Behavior	89.65	89.49	−0.16
Personality	91.92	91.92	0
Shopping Behavior	14.36	14.33	−0.03
Spotify Recommendations	92.81	92.25	−0.56
Water Potability	60.5	61.01	0.51
Wine	35.89	35.7	−0.19

We use Yellow gradients to highlight large values for Macro-F1.

A similar interpretation applies to Random Forest. Although RF already includes internal mechanisms such as random attribute selection and bagging, the attribute weights generated by DE-HX may still provide complementary information in ensemble settings that combine different types of learners. In these cases, the optimized weights can support the forest-based decision process as an additional source of attribute-level guidance, especially in stacking or voting schemes. These results reinforce the idea that the proposed weighting methodology can play a flexible role within broader ensemble frameworks, even when its direct impact on Random Forest remains limited.

The results presented in Tables 2–15 provide a broader view of the effect of the proposed attribute-weighting methodology across different classifiers. In general, the method produces its strongest impact on classifiers that depend directly on the geometry of the attribute space, while ANN shows a more heterogeneous response, and Random Forest shows only modest gains. These results indicate that the proposed weighting strategy fits distance-based classifiers more naturally, while in other models it works better as a complementary component than as a universal improvement mechanism.

5. Conclusions

This study introduced an attribute-weighting methodology based on differential evolution, enhanced with a Harmony-based crossover mechanism (DE-HX), to improve classification performance through optimized data representation.

From a methodological perspective, the main contribution of this work is the modification of the crossover stage within differential evolution, replacing it with a Harmony-based candidate generation mechanism. This approach provides an alternative search strategy for attribute-weighting problems and alters the exploration–exploitation balance of the optimization process.

Furthermore, from a computational perspective, the proposed approach relies on repeated evaluations of candidate solutions using cross-validation, which represent a signifi-

cant computational cost. However, the optimization process is guided by a stagnation-based stopping criterion, allowing each method to reach its maximum performance rather than being constrained by a fixed runtime. Future work will incorporate detailed runtime analysis and convergence profiling to better characterize the efficiency of the proposed method.

The experimental results show that the proposed approach achieves strong overall performance and frequently improves classification accuracy when compared to standard differential evolution.

The analysis indicates that k-nearest neighbors benefits most clearly from the optimized weighting scheme. By refining the attribute-space representation, the proposed methodology improves distance-based discrimination, leading to consistent performance gains across most of the evaluated datasets. These results confirm that attribute weighting plays a critical role in models that rely directly on distance computations.

Artificial neural networks exhibit a dataset-dependent response when trained on weighted data representations. In several datasets, the optimized weights lead to noticeable improvements; however, in other cases, performance remains stable or slightly decreases. This behavior suggests that the effectiveness of attribute weighting depends on the interaction between the transformed feature space and the internal learning dynamics of the neural model.

Random Forest shows limited sensitivity to the proposed weighting strategy. The observed variations in performance are generally modest and dataset-specific, with several cases showing negligible or slightly negative changes. This behavior aligns with the internal structure of Random Forest, where random feature selection and ensemble learning reduce the influence of externally imposed attribute weights.

Overall, the proposed DE-HX methodology proves most effective for classifiers that depend directly on the geometry of the feature space, particularly distance-based methods such as KNN. In contrast, its impact on models with internal feature selection or representation learning mechanisms is less consistent and should be interpreted with caution.

The present study focuses on comparative predictive performance and statistical validation. It does not include convergence plots, parameter sensitivity analyses, or stability analyses of the learned attribute weights. These elements would provide a deeper understanding of the optimization dynamics and the interpretability of the learned solutions, and therefore constitute an important direction for future work.

These findings highlight the importance of aligning attribute-weighting strategies with the characteristics of the learning model. Future work will explore the integration of the proposed approach into hybrid and ensemble frameworks, the incorporation of local search strategies to refine the optimization process, and the use of near-zero weights as a basis for attribute selection.

Author Contributions: Conceptualization, J.D.T.-V., A.O.-G. and S.I.-M.; methodology, S.I.-M. and J.A.C.-R.; software, J.D.T.-V. and A.A.S.-P.; validation, M.G.T.-B., S.I.-M. and J.A.C.-R.; formal analysis, A.O.-G. and J.D.T.-V.; investigation, S.I.-M., A.O.-G. and M.G.T.-B.; resources, A.O.-G. and A.A.S.-P.; data curation, A.O.-G., S.I.-M. and J.D.T.-V.; writing—original draft preparation, A.O.-G., J.D.T.-V. and M.G.T.-B.; writing—review and editing, S.I.-M., A.O.-G. and M.G.T.-B.; visualization, A.O.-G. and M.G.T.-B.; supervision, J.D.T.-V., A.A.S.-P. and J.A.C.-R.; project administration, A.A.S.-P., J.D.T.-V. and S.I.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The experimentation code, as well as the original and preprocessed datasets, is available for review in https://alumnosuat-edu-my.sharepoint.com/:f:/r/personal/aoguzman_uat_edu_mx/Documents/DE-HX?csf=1&web=1&e=04IyoT (accessed on 20 April 2026).

Conflicts of Interest: The authors declare no conflicts of interest. The funder had no role in the design of the study, in the collection, analysis, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

1. Harb, H.; Desuky, A.S. Feature Selection on Classification of Medical Datasets based on Particle Swarm Optimization. *Int. J. Comput. Appl.* **2014**, *104*, 14–17. [\[CrossRef\]](#)
2. Patel, H.; Shah, H.; Patel, G.; Patel, A. Hematologic cancer diagnosis and classification using machine and deep learning: State-of-the-art techniques and emerging research directives. *Artif. Intell. Med.* **2024**, *152*, 102883. [\[CrossRef\]](#)
3. Bai, Y.; Zhang, H. The cluster analysis of traditional Chinese medicine authenticity identification technique assisted by chemometrics. *Heliyon* **2024**, *10*, e37479. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Chen, Y.; Annebicque, D.; Carre-menetrier, V.; Philippot, A.; Daneau, T. Data Model Classification for Interoperability in the Industry. *IFAC-PapersOnLine* **2022**, *55*, 2354–2359. [\[CrossRef\]](#)
5. Nawaz, M.S.; Nawaz, M.Z.; Fournier-Viger, P.; Luna, J.M. Analysis and classification of employee attrition and absenteeism in industry: A sequential pattern mining-based methodology. *Comput. Ind.* **2024**, *159–160*, 104106. [\[CrossRef\]](#)
6. Luo, X. Efficient English text classification using selected Machine Learning Techniques. *Alex. Eng. J.* **2021**, *60*, 3401–3409. [\[CrossRef\]](#)
7. Dol, S.M.; Jawandhiya, P.M. Classification Technique and its Combination with Clustering and Association Rule Mining in Educational Data Mining—A survey. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106071. [\[CrossRef\]](#)
8. Lefebvre, G.; Elghazel, H.; Guillet, T.; Aussem, A.; Sonnat, M. A new sentence embedding framework for the education and professional training domain with application to hierarchical multi-label text classification. *Data Knowl. Eng.* **2024**, *150*, 102281. [\[CrossRef\]](#)
9. Wang, H.; Siti, S.; Chen, Z.; Shan, Q.; Ren, L. An intelligent music genre analysis using feature extraction and classification using deep learning techniques. *Comput. Electr. Eng.* **2022**, *100*, 107978. [\[CrossRef\]](#)
10. Liuwanyue, S. Course genres classification of music e-learning platform based on deep learning big data intelligent processing algorithm. *Entertain. Comput.* **2024**, *50*, 100704. [\[CrossRef\]](#)
11. Balim, C.; Özkan, K. Diagnosing fashion outfit compatibility with deep learning techniques. *Expert Syst. Appl.* **2023**, *215*, 119305. [\[CrossRef\]](#)
12. Balloni, E.; Pietrini, R.; Fabiani, M.; Frontoni, E.; Mancini, A.; Paolanti, M. Social4Fashion: An intelligent expert system for forecasting fashion trends from social media contents. *Expert Syst. Appl.* **2024**, *252*, 124018. [\[CrossRef\]](#)
13. Jin, C.; Jin, S.W.; Qin, L.N. Attribute selection method based on a hybrid BPNN and PSO algorithms. *Appl. Soft Comput.* **2012**, *12*, 2147–2155. [\[CrossRef\]](#)
14. Dai, J.; Xu, Q. Attribute selection based on information gain ratio in fuzzy rough set theory with application to tumor classification. *Appl. Soft Comput. J.* **2013**, *13*, 211–221. [\[CrossRef\]](#)
15. Chen, Z.; Li, B.; Han, B. Improve regression accuracy by using an attribute weighted KNN approach. In Proceedings of the ICNC-FSKD 2017—13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, Guilin, China, 29–31 July 2017; pp. 1838–1843. [\[CrossRef\]](#)
16. Al-Yaseen, W.L.; Jehad, A.; Abed, Q.A.; Idrees, A.K. The Use of Modified K-Means Algorithm to Enhance the Performance of Support Vector Machine in Classifying Breast Cancer. *Int. J. Intell. Eng. Syst.* **2021**, *14*, 190. [\[CrossRef\]](#)
17. Alghamdi, F.A.; Almanaseer, H.; Jaradat, G.; Jaradat, A.; Alsmadi, M.K.; Jawarneh, S.; Almurayh, A.S.; Alqurni, J.; Alfagham, H. Multilayer Perceptron Neural Network with Arithmetic Optimization Algorithm-Based Feature Selection for Cardiovascular Disease Prediction. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 987–1008. [\[CrossRef\]](#)
18. Abdalla, H.I.; Amer, A.A. Enhancing data classification using locally informed weighted k-nearest neighbor algorithm. *Expert Syst. Appl.* **2025**, *276*, 126942. [\[CrossRef\]](#)
19. Amer, A.A.; Ravana, S.D.; Habeeb, R.A.A. Effective k-nearest neighbor models for data classification enhancement. *J. Big Data* **2025**, *12*, 86. [\[CrossRef\]](#)
20. Sujdha, C.; Thriumalai Selvi, R.; Bharathidasan, S. Improving the classification performance for the effective diagnosis of diseases using optimized weighted KNN on clinical dataset. *Indian J. Sci. Technol.* **2025**, *18*, 215–222. [\[CrossRef\]](#)
21. Ningsih, L.; Priyatno, A.M.; Yusmar, A. A new triple-weighted k-nearest neighbor algorithm for tomato maturity classification. *J. RESTI (Rekayasa Sist. Teknol. Inf.)* **2025**, *9*, 955–962. [\[CrossRef\]](#)

22. Vega-Huerta, H.; Pajuelo-León, J.; De-la Cruz, P.; Calderón, D.; Maquen-Niño, G.L.E.; Rios-Castillo, M.E.; Camara-Figueroa, A.; Gil-Calvo, R.; Guerra-Grados, L.; Benito-Pacheco, O. K-Nearest Neighbors Model to Optimize Data Classification According to the Water Quality Index of the Upper Basin of the City of Huarmey. *Appl. Sci.* **2025**, *15*, 10202. [[CrossRef](#)]
23. Kaggle. Kaggle. 2019. Available online: <https://www.kaggle.com/datasets> (accessed on 5 August 2025).
24. Dua Dheeru, G.C. UCI Machine Learning Repository. 2019. Available online: <https://archive.ics.uci.edu/datasets> (accessed on 5 August 2025).
25. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.