www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 512-528. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2025.v16i3.856

Feature Selection through Filtering with Mono and Multi-Objective Memetic Algorithms Using Correlation

Daniel E. Zamarron-Escobar[1], Jesús D. Teran-Villanueva[1], Salvador Ibarra-Martinez[1] and Aurelio A. Santiago-Pineda[1]
[1] Universidad Autónoma de Tamaulipas, Facultad de Ingeniería Tampico
a2223338026@alumnos.uat.edu.mx, jdteran@docentes.uat.edu.mx, sibarram@docentes.uat.edu.mx, aurelio.santiago@uat.edu.mx

Abstract. Feature selection is the process of extracting the most relevant features from a dataset, helping to reduce its dimensionality by eliminating non-essential features. This leads to simpler, faster models and optimises training efficiency. This paper presents two memetic algorithms: one employs a mono-objective filter method as a fitness function, while the other adopts a multi-objective approach. The latter uses the number of attributes in the dataset as the first objective, and the sum of Pearson's correlations for the selected attributes as the second. Additionally, we apply a novel approach to the use of correlation for attribute selection within the aforementioned memetic algorithms. Both proposals aim to identify the most relevant attributes to reduce the dimensionality of twelve test datasets. The performance of the selected features was evaluated using a J48 decision tree. The results showed a reduction in the number of attributes ranging from 14% down to 5%, while accuracy varied from −5% up to 11%, with an average improvement of over 4% (considering only those datasets where accuracy changed).
Keywords: Feature Selection, Memetic Algorithm, NSGA-II.

Article Info
Received February 01, 2025.
Accepted April 02, 2025

1 Introduction

Nowadays, we are required to process information daily. This information processing is evident with increased data science and analysis usage. In data science, we find data preprocessing to be an essential element for information extraction.

Furthermore, we find multiple techniques to enhance data in data preprocessing by cleaning, normalizing, carrying out imputation, attribute selection, and attribute engineering, among others.

Datasets are expected to have high dimensionality, which causes high computational cost. These datasets come from real-world applications such as face recognition (Kim et al., 2002), speech recognition (Lima et al., 2003), DNA microarray analysis (Yu & Liu, 2004), content-based image retrieval (Dy & Brodley, 2001), among other areas.

After applying all the needed data preprocessing, we use Data Mining, defined as analyzing data and extracting information by using statistical models, artificial intelligence, machine learning, and analytical tools to condense that information into more manageable form (Han et al., 2011; Pal, 2011; Algarni, 2016).

Data Mining collects many features related to their field of study to create an abstract model. A feature is an individual measurable property associated with its corresponding model (Chandrashekar & Sahin, 2014).

Despite having a very detailed model, we try to find a subset of features of the model. This subset helps reduce the computational cost, taking irrelevant and/or noisy information out, resulting in easier processing to other tasks (Lu et al., 2007). Finding a subset that could minimize the number of features needed to extract the most relevant information is known as the curse of dimensionality (Verleysen & François, 2005). There are different techniques to reduce the dimensionality, called Attribute Selection (also known as Feature Selection) (Kumar & Minz, 2014).

One of the classical techniques for extracting the most relevant feature is Principal Component Analysis – PCA (Pearson, 1901). PCA is a linear method for dimensionality reduction, simplifying data into the most essential features as the main advantage. One of the drawbacks of PCA is the high computational cost for large datasets because calculating the covariance matrix is directly proportional to the number of features (Van Der Maaten et al., 2009).

Another approach to solve the curse of dimensionality is using evolutionary algorithms like genetic, ant colony optimization, and particle swarm optimization, among others (Abd-Alsabour, 2014). Evolutionary algorithms are metaheuristics applied to solve optimization problems by simulating natural biological evolution. Their key feature is producing multiple solutions, allowing them to explore many solutions.

These evolutionary algorithms need a mechanism called fitness function to evaluate their solutions. This function could use two of the following evaluation methods: Filter and wrapper methods (Talavera, 2005; Nnamoko et al., 2014).

Filter methods use an evaluation function to measure solutions, employing the dataset's attributes to give a score to the solutions. The advantages of filter methods are easy to implement, saving computational cost (Talavera, 2005).

Wrapper methods employs multiple subsets of features from the dataset to train and calculate classification accuracy until it finds the best subset. While its reductions solve feature selection, this method has a high computational cost, does not scale with a high number of features, and is tied to model information (Nnamoko et al., 2014).

Hybrid approaches employ filtering and wrapping methods (Kundu & Mallipeddi, 2022), combining their best advantages to find better solutions.

2 State of the Art

Deb et al. (2000) presents a fast, non-dominated sorting for multi-objective problems called Non-Dominated Sort Genetic Algorithm II (NSGA-II); this is used by some other authors that study attribute selection with multi-objective approaches. NSGA-II improves its previous iteration, which had higher computational complexity from $O(MN^3)$ to $O(MN^2)$, lack of elitism, and the need to define a parameter for diversity purposes. This improved algorithm approach categorizes all the possible solutions of the population into fronts based on their dominance relationships, where higher Pareto fronts dominate the solutions in lower Pareto fronts. NSGA-II also includes crowding distance as a diversity mechanism to encourage a better solution space in the Pareto front. Additionally, with these improvements, NSGA-II, at its core, has all the benefits of evolutionary algorithms, in this case, Genetic Algorithm, with selection, crossover, and mutation operators to evolve through generations.

Kannan and Ramaraj (2010) explored a new approach in featured selection using a memetic algorithm with their Local Search using Symmetrical Uncertainty (SU) as correlation-based filter ranking. Their proposed Local Search calculates SU for every feature and ranks them in descending order. From this ranking, it selects features essential for classification without overlapping with other relevant features and removes them from the ranking up to having no more features excluded. Their proposed local search allows them to explore attribute subsets by adding or removing attributes to ensure finding the best individual attributes. Additionally, their algorithm uses Subset Size-Oriented Common Feature for the cross-over operator and Bit-Flip for the mutation operator in their evolutionary algorithm in conjunction with their proposed Local Search.

Yildirim et al. (2021) tested two alternatives for reducing features of high-dimensional space applied for speech emotion recognition with two metaheuristics approaches, an NSGA-II and a Cuckoo Search. The authors use IEMOCAP and EMO-DB databases to transform audio clips into a dataset representation of those two databases. For each metaheuristic, they generate their initial population through random generation and Relieff generation. Afterward, Cuckoo Search tries to maximize classification accuracy using three criteria: K-nearest neighborhood, bagged decision tree, and support vector machine. NSGA-II minimizes the number of features while maximizes accuracy. The results showed that their proposed methods had comparable classification accuracy while significantly reducing the number of features.

Kundu and Mallipeddi (2022) proposed a hybrid filter multi-objective evolutionary algorithm (HFMOFEA) where the two main objectives are minimizing the number of features selected and maximizing the classification accuracy calculated by Support Vector Machine (SVM). The authors' approach to feature selection is as follows: they first divide the target dataset into training and testing sets. Later, they initialize the population with solutions taken from the training set, where each solution's chromosome

is a binary chromosome corresponding to one of the ten proposed filter methods. Furthermore, the authors select a defined number of best-ranking methods into the chromosome as 1. After producing the algorithm population, they start with the multi-objective evolutionary part based on the NSGA-II framework from Deb et al. (2000) until reaching the termination criterion. It finalizes selecting the optimal Pareto front.

Benito et al. (2023) presents a comparative study on feature selection techniques applied to five datasets and proposes a Tabu Search metaheuristic using a wrapper. Traditional techniques used for this comparison are Entropy, Correlation, and Principal Component Analysis (PCA). For entropy feature selection, they calculated entropy and sorted features in descending order, removing features below a certain defined threshold. Using the correlation method, they calculated the correlation matrix, following a procedure similar to entropy, sorting in descending order and removing features below a threshold. For PCA, they calculated the first two principal components. Then, they evaluated the Euclidean distance between those two components to create two subsets grouped by quartiles, one with features included in quartile 1 only and the other features in quartiles 1 and 2. Their proposed method is as follows: they start by defining a binary array to represent the feature that will be used; they use a J48 classification tree to calculate the accuracy and use it as their objective function. Additionally, they use exchange movements to navigate through the neighborhood and look for the best possible solution. Once a solution is located, the algorithm will lock that move in the Tabu list for a specific time. Their results showed that the proposed methodology outperformed all other techniques. However, they mentioned that the correlation method is faster to implement, has a low computational cost, and has an acceptable performance.

This paper proposes two approaches for selecting the most important attributes: a mono-objective memetic algorithm with a filter method using Pearson's correlation matrix and a multi-objective NSGA-II memetic algorithm with a filter method. Our multi-objective proposal searches for the following two objectives: minimize the most relevant attributes of the dataset and the remaining attribute's correlation sum.

3 Proposed Methodology

We define the datasets used to test out all our memetic algorithms. Table 1 shows the twelve dataset names, number of attributes, and number of records. UCI Machine Learning Repository (Kelly et al., 2023) provided all the datasets used in this paper.

Table 1. List of Datasets

Dataset	Number of Attributes	Number of Records
Australian Credit	15	690
Balance Scale	4	625
Breast Cancer Wisconsin	9	683
Ecoli	8	336
Glass	9	214
Ionosphere	34	351
Iris	4	150
SoyBean Small	35	45
Teaching Assistant	5	151
Thyroid (Training)	22	3772
TicTacToe	9	958
Wine	13	178

Algorithm 1 presents our proposed mono-objective memetic algorithm. The objective is to find a combination of attributes that provides the minimum sum of their correlation. The process starts by calculating the correlation matrix for the current dataset in the test in Line 1 To calculate the correlation matrix, we remove all id-related alphanumeric attributes and other attributes with the same value on all records because in both cases are entirely useless for classification purposes.

In the case of Breast Cancer Wisconsin's correlation matrix, we removed 16 of 699 records with missing values on the Barei Nuclei attribute. These records correspond to 2.3% of total records counts, opting to apply Listwise imputation to those records. Even though we are losing information removing those records, Graham (2009) recommends it is less than 5% of total missing values to apply Listwise imputation.

Algorithm 1: Proposed Mono-Objective Memetic Algorithm

```
1  CalculateCorrelationMatrix(Dataset)
2  Population ← GenerateInitialPopulation(PopulationSize)
3  Population ← CalculateFitnessValue(Population)
4  BestSol ← CalculateBestSolution(Population)
5  While(IterationsWithoutImprove < MaxIterations)
6     While(CalculatePopulationAlive < PopulationSize)
7        ParentA, ParentB ← SelectParentsFromPopulation(Population)
8        If(CalculateCrossOver() < MutationPercentage)
9           NewSol ← CrossOver(ParentA, ParentB)
10       Else
11          NewSol ← SelectRandomParent(ParentA, ParentB)
12          NewSol ← ApplyLocalSearch(NewSol)
13       NewSol ← CalculateFitnessValue(NewSol)
14       InsertNewSolutionIntoPopulation(NewSol)
15    FitMeanValue ← CalculateFitnessMeanValue(Population)
16    Population ← RemoveSolutionsFromPopulation(FitMeanValue)
17    BestSol ← CalculateBestSol(Population)
18    IterationsWithoutImpove ← BestSolChanged(BestSol)
```

Before generating the population, we establish the representation of a valid solution. We are defining a binary vector of n length as a solution. Where n is the number of attributes, a value of 1 on the i position of the binary vector means that attribute i will remain in the dataset for classification, whereas a value of 0 means otherwise.

Figure 1 shows an example of a solution dataset. Using *Iris* dataset, this dataset has four attributes: *SepalLength*, *SepalWidth*, *PetalLength*, and *PetalWidth*; our binary vector representing the solution, has 1 on the first and second positions of the vector, and 0 on the remaining attributes. Therefore, *SepalWidth* and *PetalLength* remain for calculations while *SepalLength* and *PetalWidth* are not considered.

Solution Representation

Iris Attributes	Sepal Length	Sepal Width	Petal Length	Petal Width
Position	*1*	*2*	*3*	*4*
Sol	0	1	1	**0**

Fig. 1. Example of a solution vector.

We generated a random solution by creating a binary vector and setting 0 or 1 randomly to each position. The process ensures that solutions contain at least 25% of the vector length with the value of 1. This process continues until the solutions fulfill the previously stated condition.

With a solution created, we calculate it's the fitness value in Equation 1:

$$Min(FV) = \sum_{a_i \in A} \sum_{a_j \in A} Corr(a_i, a_j) \mid s_i = 1 \wedge s_j = 1 \wedge i \neq j \quad (1)$$

Where $A = \{a_1, a_2, \ldots, a_n\}$ is the set of attributes, s_i and s_j are binary values on i and j position of $Sol = \{s_1, s_2, \ldots, s_n\}$, such that if $s_k = 1$ it means that the attribute will be considered to remain in the dataset and $s_k = 0$ otherwise.

We have two operators to create solutions: *CrossOver* and *Mutation* Operators, in which either operator has a random percentage chance of being selected. Both operators select two parents randomly and differently.

Once the selection process ends, the algorithm randomly chooses between the *CrossOver* or the *Mutation* operator, seen in Line 8. If the algorithm selects the *CrossOver* operator, then select a random position in the solution vector to make a single point cross-over, where the first parent will copy from the beginning of the solution to that cross-over position to the offspring. The second parent will continue to copy from that position onwards to the end of the parent. Figure 2 illustrates an example of a single point cross-over. We define a single point cross-over in position 3, and from Sol_1 the algorithm copies its values to the offspring up to the single point cross-over, then it will resume copying starting up from Sol_2.

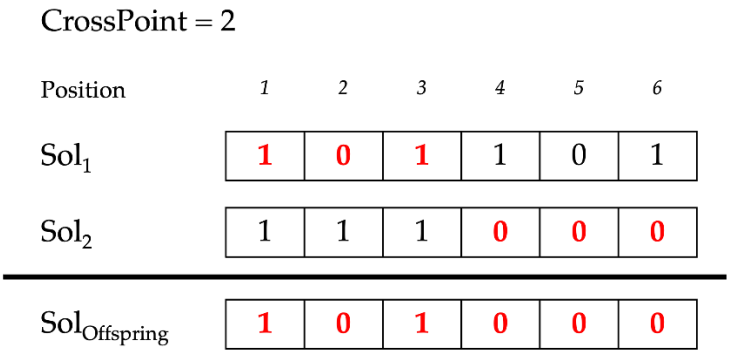

Fig. 2. Single point cross-over operator example.

The *Mutation* operator applies a local search to the offspring from a random parent. Then, copy this parent entirely to the offspring and apply local search. This local search will try to select one of the already selected attributes in the solution vector and exchange its value with positions where the attributes in the vector are not considered. This exchange applies to all attributes taken, the process ends after finding the first position that improves the fitness value or visiting every possible neighbor. Figure 3 provides a visual example of this operator.

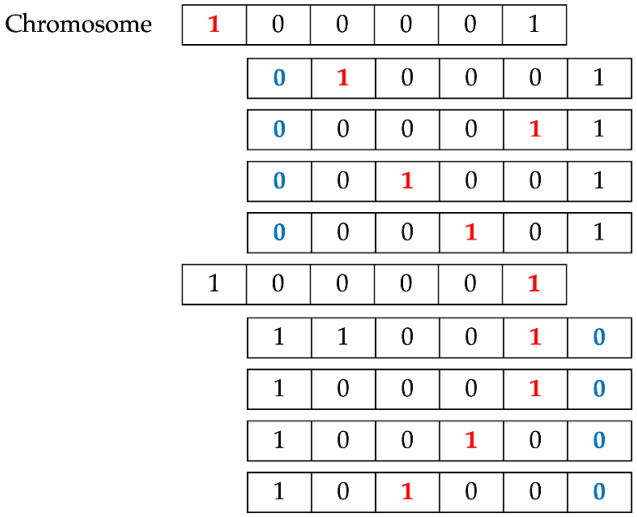

Fig. 3. Mutation operator example.

After completing the offspring generation to fill the population, we calculate the mean fitness value of the population as seen in Line 15 and proceed to remove solutions above the mean fitness value up to a limit to avoid removing too many solutions of the population. Afterwards it will update the best solution known if there is a better solution found up to this point.

Algorithm 1 will continue until we find a new best solution for a defined number of iterations without improvement.

One of the downsides of the mono-objective memetic algorithm is forcing solutions with at least 25% of the total number of attributes.

After producing our memetic proposal for reducing the dimensionality of datasets, we explained that all our solutions created should have at least 25% of the total number of attributes. This prerequisite ensures that multiple attributes are needed for calculating the fitness value; otherwise, it would calculate incorrect values to solutions.

From this issue, we came up with another alternative that uses a multi-objective approach instead of limiting the number of attributes removed. This new approach is a memetic algorithm based on (Deb et al., 2000) NSGA-II that uses two objectives: minimizing the dataset's number of removed while minimizing the sum of the correlation matrix for the remaining attributes.

When discussing multi-objective approaches, it is a requirement that objectives be against each other's objective. In this case, if we want to minimize the sum of the correlation matrix for the remaining attributes, the optimal case will be when there are no remaining attributes. In contrast, the other objective pushes in the opposite direction, trying to minimize the removed attributes.

The main differences between the mono-objective memetic algorithm and this multi-objective version are in Lines 3, 15 to 17 for Algorithm 2.

Algorithm 2: Proposed Multi-Objective Memetic Algorithm

```
1  CalculateCorrelationMatrix(Dataset)
2  Population ← GenerateInitialPopulation(PopulationSize)
3  Population ← CalculateNormalizedFitnessValue(Population)
4  BestSol ← CalculateBestSolution(Population)
5  While(IterationsWithoutImprove < MaxIterations)
6     While(CalculatePopulationAlive < PopulationSize)
7        ParentA, ParentB ← SelectParentsFromPopulation(Population)
8        If(CalculateCrossOver() < MutationPercentage)
9           NewSol ← CrossOver(ParentA, ParentB)
10       Else
11          NewSol ← SelectRandomParent(ParentA, ParentB)
12          NewSol ← ApplyLocalSearch(NewSol)
13       NewSol ← CalculateFitnessValue(NewSol)
14       InsertNewSolutionIntoPopulation(NewSol)
15    ParetoElements ← NSGAIISelection (Population)
16    Population ← RemoveSolutionsFromPopulation(ParetoElements)
17    BestSol ← CalculateBestSol(Population)
18    IterationsWithoutImpove ← BestSolChanged(BestSol)
```

For this version, Equation 2 shows the new fitness value function for evaluating solutions:

$$Min(FV) = \left(\sum_{a_i \in A} \sum_{a_j \in A} Corr(a_i, a_j) \ / \ SumCorr\right) * \ |A| \ | \ s_i \ = \ 1 \ \wedge s_j \ = \ 1 \ \wedge i \ \neq j \tag{2}$$

Where $A = \{a_1, a_2, \ldots, a_n\}$ is the set of attributes, $SumCorr$ is the sum of correlation of all attributes, s_i and s_j are binary values on i and j position of $Sol = \{s_1, s_2, \ldots, s_n\}$, such that if $s_k = 1$ it means that the attribute will be considered to remain in the dataset and $s_k = 0$ otherwise.

Additionally, we calculate the score of the J48 Classifier Algorithm using SciKit Learn implementation for all solutions to track their classification accuracy.

The function *NSGAIISelection* in Line 15 based on (Deb et al., 2000), is described as follows: for every solution in Population, we evaluate the solution against the other solutions in the Population through dominated Pareto fronts. We define two ways to assess which solution dominates the other or is equally dominated. The first identifies the solution with the lowest fitness value of the two solutions, and the second evaluates the selected number of attributes in the solution. A solution is dominated if both evaluations of a solution dominate the others; therefore, the first solution dominates the second, and the second solution is dominated by the first. Solutions that dominate each other by one evaluation are considered non-dominated and equivalent, meaning they have the same rank.

After we evaluated all solutions with each other, we sorted the solutions by Pareto fronts. We defined a Pareto front as groups where solutions are non-dominated by each other. The first Pareto front constitutes the best-found solutions, and they dominate the rest of the solutions in the remaining Pareto fronts; the second Pareto front dominates Pareto fronts below it while being dominated by the first front. This is true for all Pareto fronts.

Once we have all the front calculated, in Line 16, we select half of the population for the next iteration and the other for elimination; the process is as follows: adding the number of solutions from the first to last Pareto front while not exceeding our threshold. When we reach a Pareto front where the sum of fronts exceeds the threshold, we calculate their crowding distance for all the solutions, which is the Euclidean distance between two neighboring solutions. The algorithm sorts the crowding distance in descending order, taking solutions up to filling, reaching a threshold of 50% of the population.

This can be visually illustrated in Figure 4. Where P_t are solutions carried over from the previous generation, Q_t are newly created solutions for this generation, $P_t \cup Q_t$ is the population, F_t are the Pareto fronts, and the threshold is defined as the half of the population.

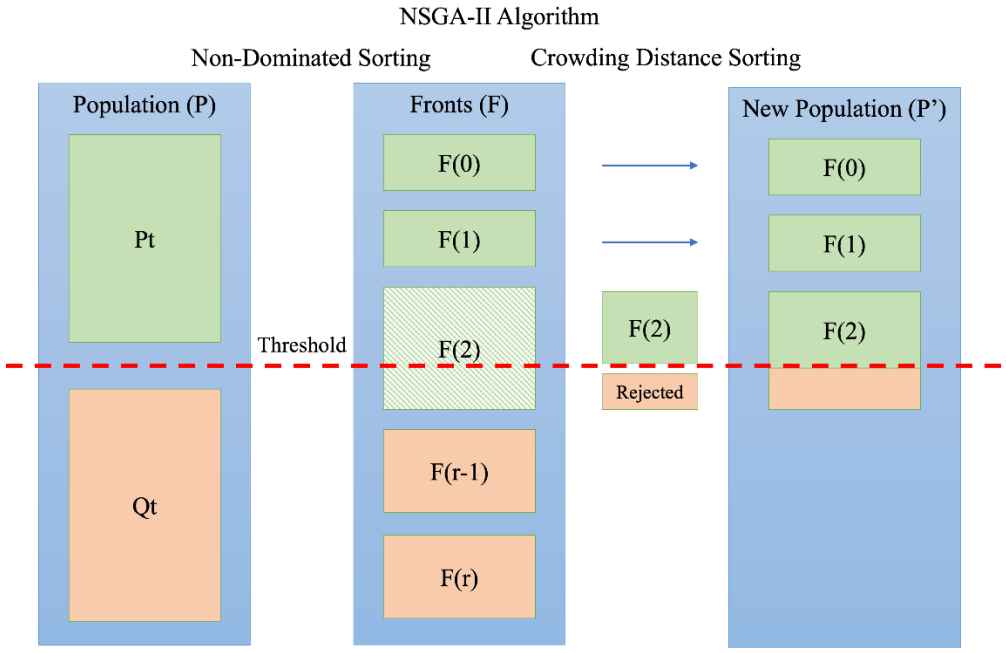

Fig. 4. NSGA-II Sort.

Finally, selects the best solution and updates it if improves the best known one in Line 17. We use the J48 classification scores to select the highest score as the best solution. If a solution ties in with the best-known solution's score, the one with the least number of attributes is selected.

Algorithm 2 will continue until we find a new best solution for a defined number of iterations without improvement.

4 Results

Before going into the results, it is essential to explain the selection of the attributes where $s_i = 1$. On the one hand, keeping the attributes with a low correlation value $SumCorr \approx 0$ implies that the attributes do not contain duplicate information. On the other hand, we are keeping attributes where its $SumCorr \approx 1$ suggests that those attributes with high correlation values do contain duplicate information. Ultimately, logic would indicate using attributes that do not contain or have low overlapped information where $SumCorr \approx 0$.

Nevertheless, to avoid assumptions, we will test subsets of attributes Z and Z'. Z subset contains the attributes selected by our methodology ($SumCorr \approx 0$) and Z' subset is the complement of Z, such that $Z \cup Z' = A$. For our experimentation, we used Python Version 3.11.5 with Pandas Version 2.1.0 and SciKit Learn Version 1.3.1's J48 Classifier implementation to calculate their classification scores. Table 1 presents the datasets employed for our tests.

Table 2 shows the number of attributes and J48 value for each Z and Z'. We can appreciate that Z'J48 score has equal or higher precision over Z for all datasets. Most of the dataset's number of attributes of Z' have a higher count over Z.Nowadays, we are required to process information daily. This information processing is evident with increased data science and analysis usage. In data science, we find data preprocessing to be an essential element for information extraction.

Table 2. Dataset's accuracy and number of attributes

Dataset	Z		Z'	
	Number of Attributes	J48 Classification Accuracy	Number of Attributes	J48 Classification Accuracy
Australian Credit	4	74.64	10	79.71
Balance Scale	2	60	2	69.6
Breast Cancer Wisconsin	5	92.7	4	96.16
Ecoli	3	67.65	4	80.88
Glass	2	65.12	7	81.4
Ionosphere	11	85.92	22	91.55
Iris	2	56.67	2	100
SoyBean Small	14	100	7	100
Teaching Assistant	2	51.61	3	54.84
Thyroid (Training)	7	91.13	14	99.87
TicTacToe	2	61.46	7	85.42
Wine	4	80.56	9	97.22

Considering the results of Table 2, the best option is using Z' where $SumCorr \approx 1$. Table 3 shows the results of our methodology. First, we define *Unmodified* as the number of attributes found in the instance before any reduction is applied. *Mono-Objective* shows the results using the Filter Method Mono-Objective Memetic Algorithm. *NSGA-II* shows the results obtained by the Filter Method Multi-Objective NSGA-II Memetic Algorithm. The parameters used for the Memetic Algorithm in both of our proposals are as follows: population size is set to 100 elements, Cross-Over and Mutation are set to 90% and 10% respectively and it ends until 25 generation pass without improvement.

Table 3. Results of our proposed methods

Dataset	Unmodified		Mono-Objective		NSGA-II	
	No of Attributes	J48 Accuracy	No of Attributes	J48 Accuracy	No of Attributes	J48 Accuracy
Australian Credit	14	82%	11	81%	6	91%
Balance Scale	4	74%	2	67%	2	70%
Breast Cancer Wisconsin	9	96%	7	91%	5	98%
Ecoli	7	83%	5	79%	4	85%
Glass	10	74%	7	84%	7	84%
Ionosphere	34	85%	26	87%	20	96%
Iris	4	100%	2	100%	2	100%
SoyBean Small	36	100%	30	90%	19	100%
Teaching Assistant	5	61%	3	55%	3	68%
Thyroid (Training)	21	100%	16	91%	19	100%
TicTacToe	9	96%	7	76%	7	91%
Wine	13	92%	10	92%	10	100%

The results show reductions in both of our proposed methods, showing that *NSGA-II* has equal to better reductions in most datasets than the *Mono-Objective*. Figures 5 and 6 show the dataset's reduction represented as percentage values. *Unmodified* columns are the dataset's total number of attributes, thereby showing 100%, showing the total usage of their attributes.

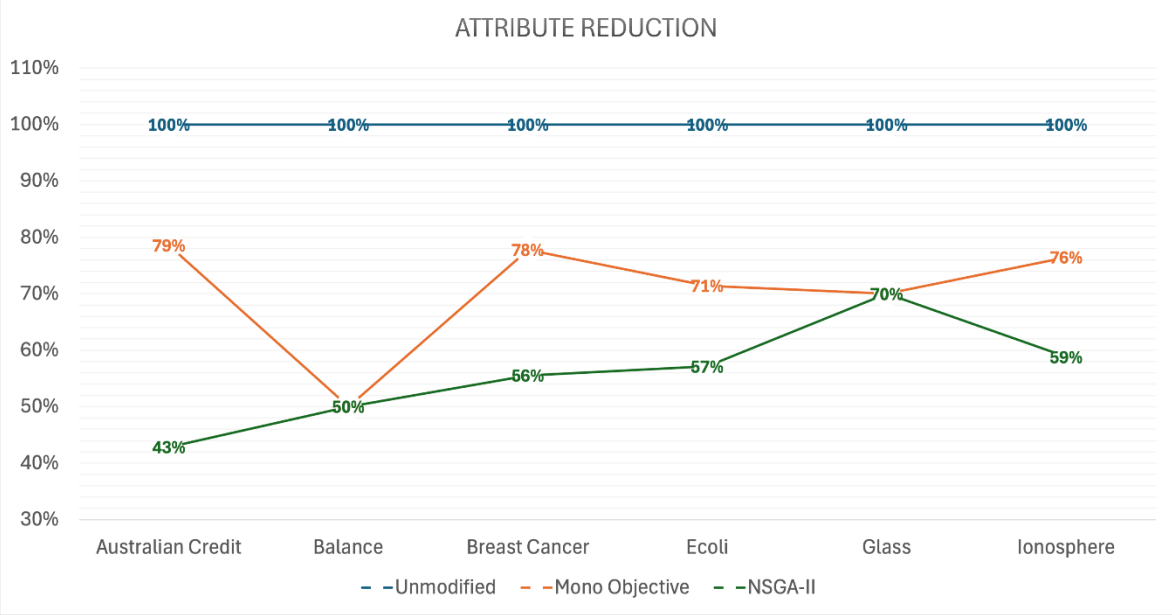

Fig. 5. Attribute reduction results – Part 1.

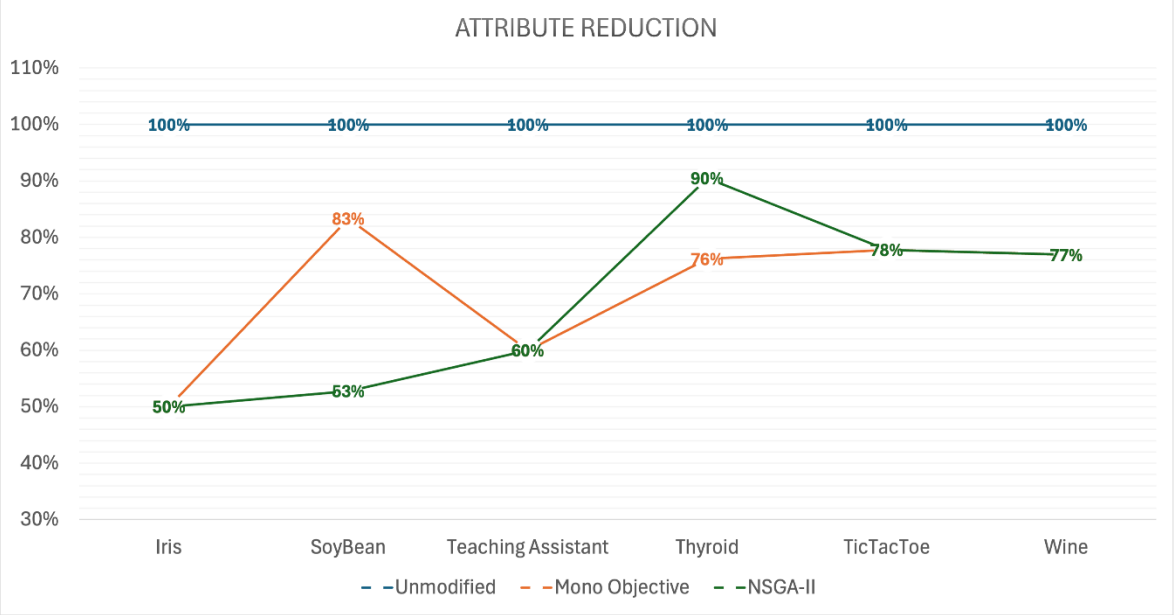

Fig 6. Attribute reduction results – Part 2.

A critical difference between our *Mono-Objective* and *NSGA-II* proposal is using classification accuracy to select the best solutions instead of just the correlation value. The first one uses correlation as the only value to reduce attributes, and the latter one uses the *NSGA-II* using number of attributes combined with the correlation normalized value to select the best solutions. This difference will be shown when we analyze the classification accuracy of these two proposals, as the *Mono-Objective* is using correlation alone as mentioned before and we are limiting the minimum amount of attributes it can have an impact on what attributes are selected; In contrast, the *NSGA-II* using both a normalized correlation and number of attributes, could produce better selections of attributes as both objectives are trying to balance each other. We can view the behavior of the classification accuracy in the next figures.

Figures 7 and 8 show the dataset's classification accuracy as percentage values. *NSGA-II* performance in most datasets ties or exceeds accuracy on *Unmodified* and *Mono-Objective*. Finally, considering both attribute reduction and classification accuracy results, we can conclude that *NSGA-II* outperforms the other two methods.

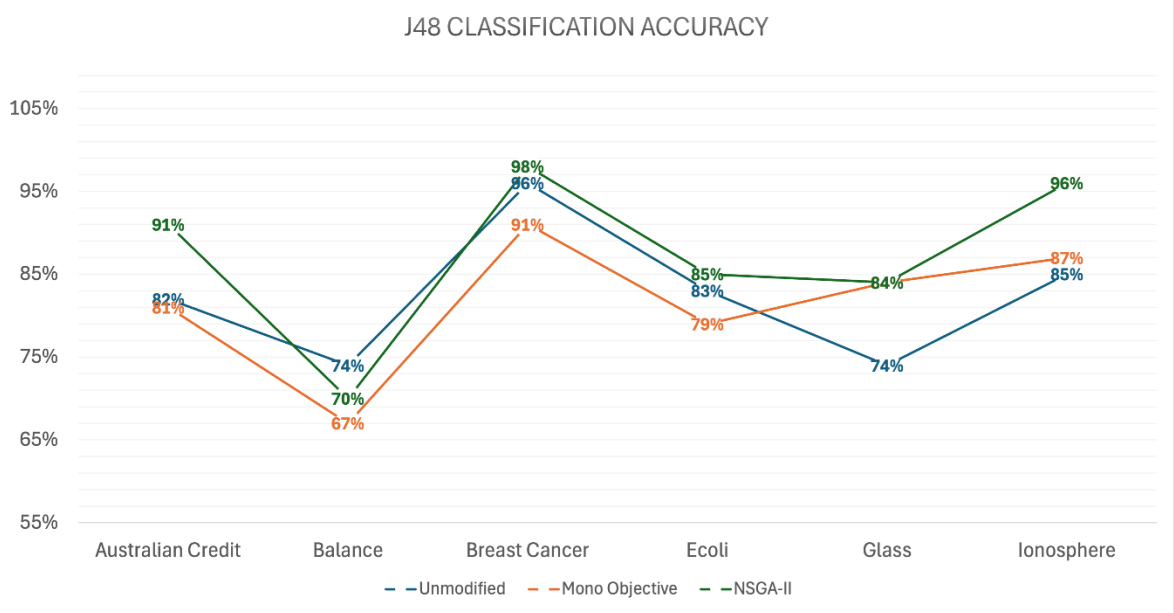

Fig. 7. Classification accuracy results – Part 1.

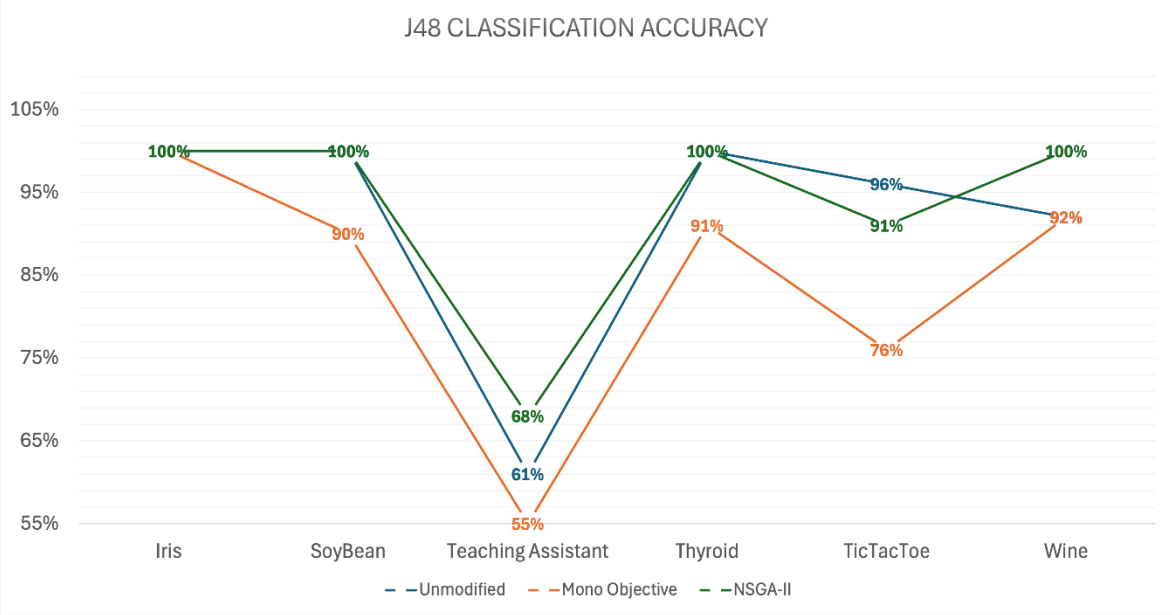

Fig. 8. Classification accuracy results – Part 2.

To corroborate the statistical difference among all our proposals, we used two different nonparametric statistical tests (Rainio et al., 2024). For our case study, we chose to run Friedman's and Wilcoxon's tests to verify our results to our attribute reduction and classification results. The first statistical test assesses if there is a difference between all proposals, and the latter evaluates if there is a difference between any of the pairing proposals. We define the null Hypothesis as having no real difference between our proposals. To reject or accept this Hypothesis, we look for the p-value or probability value calculated from the tests and compare it to α, the level of significance. If the p-value is smaller α, there is a statistical difference in the test, rejecting the null Hypothesis; on the other hand, a higher value means accepting the null Hypothesis. We set α to 95% equivalent to 0.05 p-value.

Table 4 and 5 present the results for Friedman and Wilcoxon tests, respectively applied to the attribute reduction values from Table 3.

Table 4. Friedman's test result on attributes

Friedman's Test	
	Ranks
Methodology	**Mean Ranks**
Unmodified	3.00
Mono-Objective	1.67
NSGA-II	1.33
P-Value	0.000

Our focus is to minimize the number of attributes in the dataset; therefore, we are looking for the lowest Mean Rank in the test. According to the Friedman Results in the attribute reduction, the Mean Rank of *NSGA-II* is 1.33, close to *Mono-Objective* with 1.67. Both proposals have a lower Mean Rank than the *Unmodified* version. Furthermore, the $P - Value \approx 0$ means a statistical difference among the three tests.

Table 5. Wilcoxon's test result on attributes

Wilcoxon's Test						
	Ranks				Test Statistics	
		N	Mean Rank	Sum Rank	Z	P-Value
Unmodified - NSGA-II	Negative Rank	0	-	-	-3.09	0.002
	Positive Rank	12	6.50	78.00		
	Ties	0	-	-		
	Total	12	-	-		
Mono-Objective - NSGA-II	Negative Rank	1	3.00	3.00	-1.57	0.116
	Positive Rank	5	3.60	18.00		
	Ties	6	-	-		
	Total	12	-	-		

The results from Wilcoxon show a statistical difference between *Unmodified* and *NSGA-II* with a $P - Value = 0.002$. On the other hand, while comparing the *Mono-Objective* and *NSGA-II*, there is no clear statistical difference between both proposals with a $P - Value = 0.116$, meaning that both proposals are statistically equivalent. However, *NSGA-II* tends to produce better results, as shown in the Positive Sum of Rank with a value of 18. Therefore, adding more datasets for testing may reduce the $P - Value$ to show a significant statistical difference.

Additionally, we show the Friedman and Wilcoxon nonparametric tests for the classification accuracy in Table 6 and 7, respectively; from the results presented in Table 3.

Table 6. Friedman's test result on classification accuracy

Friedman's Test	
	Ranks
Methodology	Mean Ranks
Unmodified	2.04
Mono-Objective	1.33
NSGA-II	2.63
P-Value	0.002

Here, we focused on maximizing the classification accuracy; hence, we aim for higher Mean Ranks. The Friedman test shows *NSGA-II* as the top Mean Rank with a value of 2.63; furthermore, the *Unmodified* version outperformed the *Mono-Objective*. Regarding this last result, the *Mono-Objective* excessively reduces the number of attributes, negatively impacting classification accuracy.

The Wilcoxon test regarding the classification accuracy shows that *NSGA-II* outperformed the *Mono-Objective* alternative with a significant statistical difference with a $P - Value = 0.005$. On the other hand, the *Unmodified* version barely reached statistical equivalence with a $P - Value = 0.066$, as we can see in the Negative Sum of Rank of 38, *NSGA-II* has better classification accuracy performance; hence, we believe that by using more datasets *NSGA-II* might reach statistical difference.

Table 7. Wilcoxon's test result on attributes

Wilcoxon's Test						
	Ranks				Test Statistics	
		N	Mean Rank	Sum Rank	Z	P-Value
Unmodified - NSGA-II	Negative Rank	7	5.43	38.00	-1.84	0.066
	Positive Rank	2	3.50	7.00		
	Ties	3	-	-		
	Total	12	-	-		
Mono-Objective - NSGA-II	Negative Rank	10	5.50	55.00	-2.81	0.005
	Positive Rank	0	-	-		
	Ties	2	-	-		
	Total	12	-	-		

Therefore, the *NSGA-II* proposal has a better balance between minimizing the number of attributes and maximizing classification accuracy than the *Mono-Objective* and the Unmodified alternatives.

Finally, we carried out the Principal Component Analysis (*PCA*) technique using SciKit Learn's *PCA* implementation to extract the PC1 (Principal Component 1)

Table 8 presents the PC1 weighting attributes from Iris Dataset in descending order. In this case, we only selected *PetalLength* attribute because the following attribute *SepalLength* has a noticeable weight difference when comparing to *PetalLength*. The remaining attributes are not considered because they had even lower weighting values. From this extraction we calculate the attribute reduction percentage and the classification accuracy for all datasets using the most significant attributes selected from PC1. The final results of this extraction for our datasets are in Table 9.

Table 8. Iris Principal Component 1

Dataset	Weight
PetalLength	**0.857**
SepalLength	0.361
PetalWidth	0.358
SepalWidth	0.085

PCA has a high attribute reduction level on most datasets due to the selection of the best attributes according to the value of their principal components; this can be a drawback as we remove information that could help for classification purposes.

Table 9. PCA results on datasets

Dataset	Number of Attributes	Principal Component 1 (PC 1)	PC 1 Number of Attributes	PC1 J48 Classification Accuracy
Australian Credit	14	99.89	1	70%
Balance Scale	4	25.00	1	56%
Breast Cancer Wisconsin	9	69.05	7	96%
Ecoli	7	51.62	4	75%
Glass	10	47.62	2	72%
Ionosphere	34	31.34	12	85%
Iris	4	92.46	3	100%
SoyBean Small	36	47.93	4	70%
Teaching Assistant	5	63.40	1	48%
Thyroid (Training)	21	32.37	2	91%

TicTacToe	9	15.18	4	67%
Wine	13	99.81	1	67%

Now, we can compare our *NSGA-II* results from Table 3 with PCA results from Table 8. Figures 9 and 10 show the difference between reduction in the number of attributes.

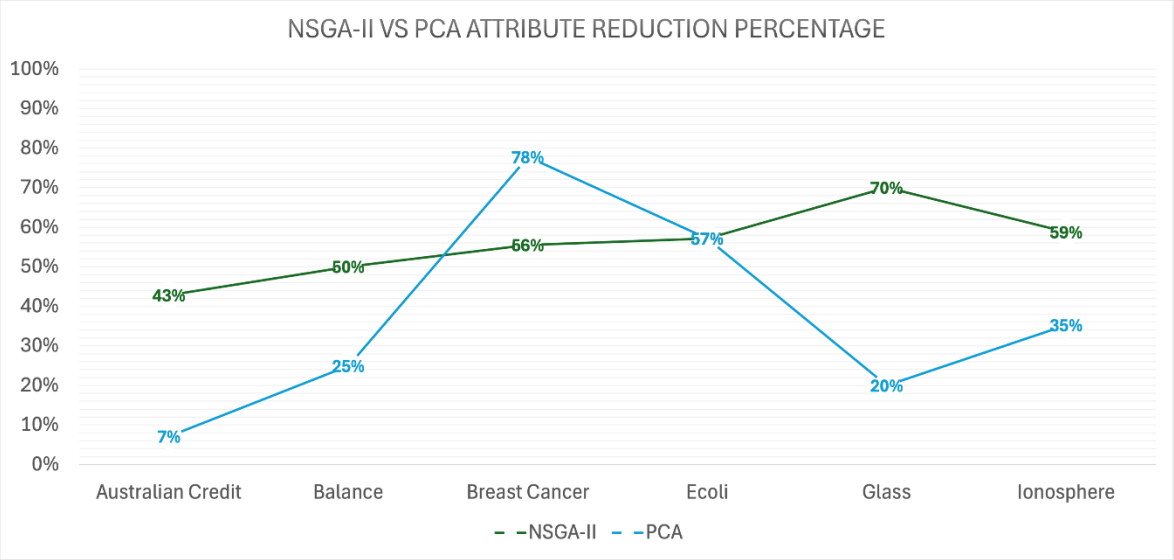

Fig. 9. NSGA-II vs PCA attribute reduction comparison – Part 1.

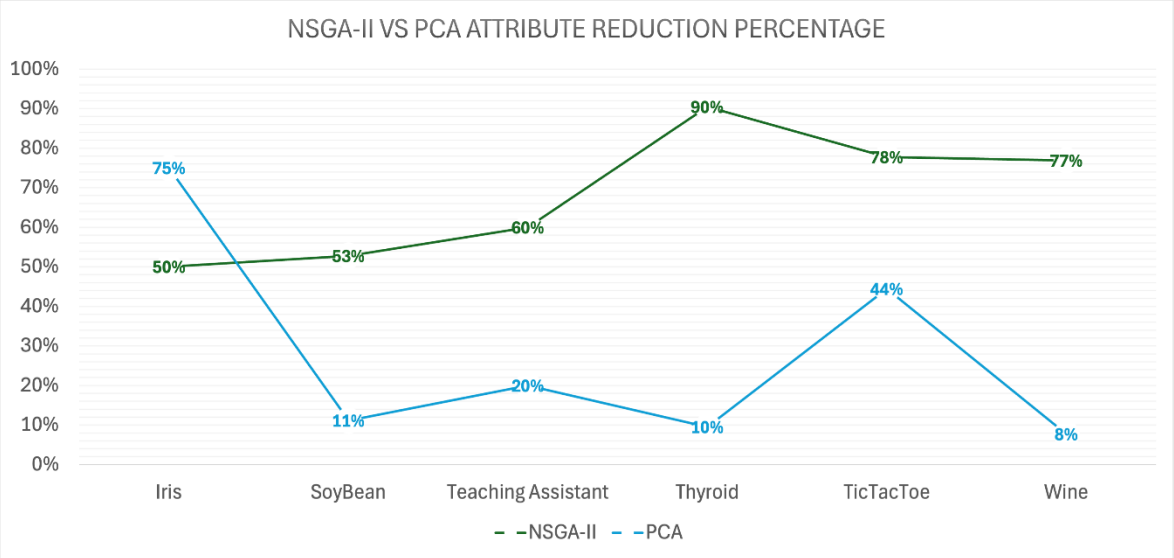

Fig. 10. NSGA-II vs PCA attribute reduction comparison – Part 2.

Figures 11 and 12 show the difference between their classification accuracy using a J48 decision tree.

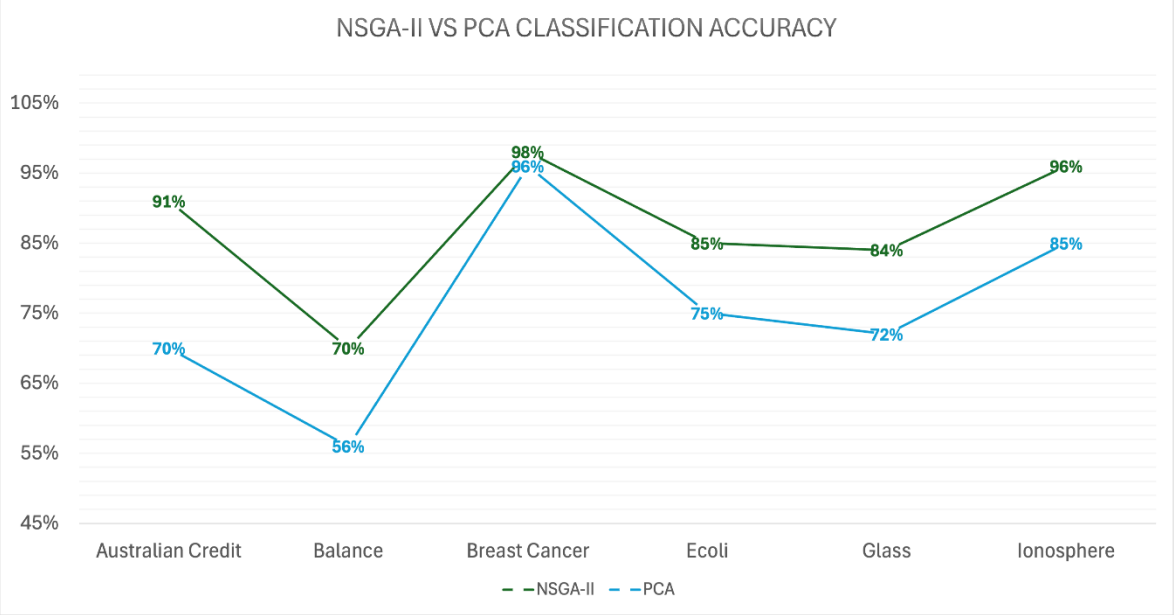

Fig. 11. NSGA-II vs PCA classification accuracy comparison – Part 1.

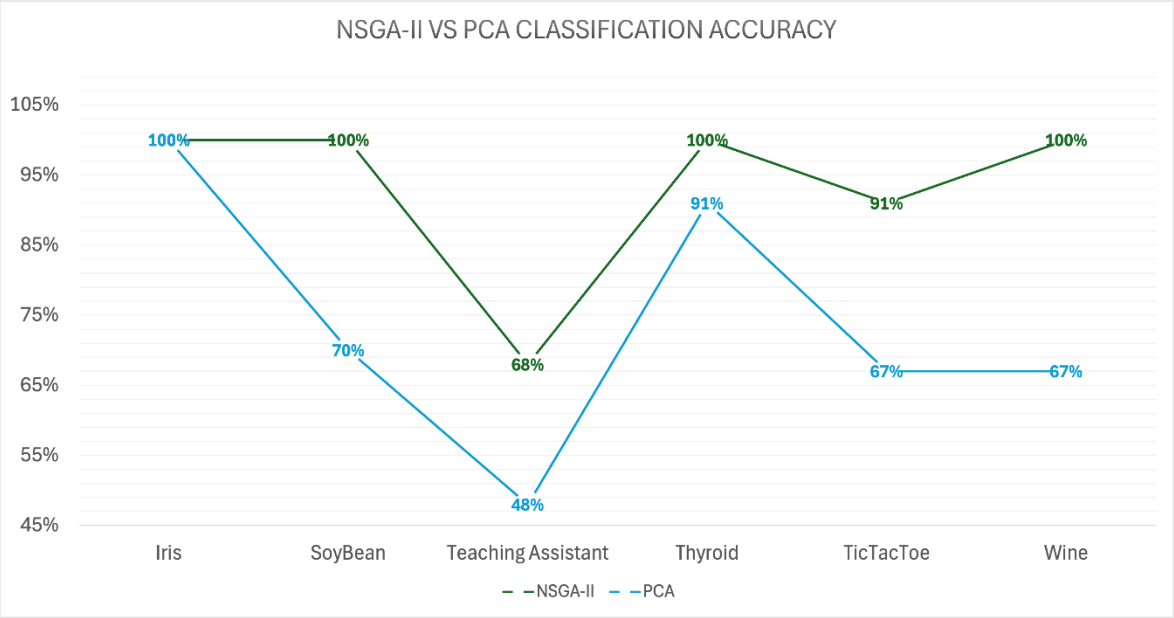

Fig. 12. NSGA-II vs PCA classification accuracy comparison – Part 2.

We can see that in most test cases, *PCA* uses fewer attributes than *NSGA-II*. Nevertheless, as expected, *PCA* classification scores are lower than the *NSGA-II*; this indicates that the attribute selection done by the *PCA* is inefficient. We believe that *PCA* exerts this behavior because it has several disadvantages, such as: features being determined by linear correlations, principal components are based on estimations of means and covariance from variables from the original dataset (Palo et al. 2021). Additionally, the main purpose of *PCA* is to produce new attributes that can explain the behavior of the dataset.

5 Conclusions and Discussions

In this paper, we propose two attribute selection techniques, thus reducing dataset dimensionality. The proposals are a mono-objective memetic algorithm and a multi-objective memetic non-dominated sorting genetic algorithm (NSGA-II). Both proposed algorithms employ a different approach, using correlation to select the most relevant attributes in datasets to solve the curse of dimensionality.

At first, we supposed that keeping attributes with low correlation hinted that we get more information from the dataset as greater entropy because attributes *show* us distinct significance. Nevertheless, when selecting Z' and determining what produced better results, we realized conserving attributes sharing similar likelihood was appropriate. This selection could mean the other attributes may contain noise or trash values. Keeping the characteristics with a high correlation value implies that those attributes "tell" the same story. There is a pattern that describes the class of each dataset's record.

It is essential to reaffirm that when calculating the correlation matrix, we set every value to its absolute value, meaning that a strong negative correlation is still a high correlation between two attributes; the negative or positive indicates a direction between that correlation, going from attribute one to attribute two or otherwise. We chose to research correlation as an absolute positive or negative value.

The results show a noticeable improvement in the dimensionality reduction of the reference dataset for both techniques. Furthermore, NSGA-II gained the edge over alternative accuracy and attribute reduction techniques. Regarding attribute selection, NSGA-II statistically outperformed the memetic algorithm, while regarding accuracy, NSGA-II improved the memetic algorithm statistically and nearly obtained a statistical difference against the reference dataset, missing only 1.6%.

As we stated in our results sections, NSGA-II performance was achieved due to both objectives, the number of attributes, and the normalized correlation value, balancing each other and selecting the best solution by calculating the accuracy with a J48 Classifier, in contrast to the Mono-Objective, which uses only the correlation values to pick out the best solution with also restrictions such as limiting the number of attributes selected in solutions, that restriction is not used in NSGA-II.

We also suspect that the different ways of picking up the best solution in each proposal are a possible reason why NSGA-II outperforms the Mono-Objective proposal. The solution selection with the best classification accuracy inside the Pareto front for the NSGA-II gives an edge regarding the Mono-Objective selection, even with the same number of attributes.

Therefore, in future work, we propose using more test datasets with a higher number of attributes or a classification accuracy between 60% to 80% to evaluate the performance of NSGA-II and to identify if it can reach a statistical difference compared to the reference dataset; that is because a higher number of attributes could show a diverse range of results as each method selects what it considers the best attributes and the suggested accuracy classification seems to be inclined to higher variability when selecting attributes.

Additionally, enhancing or re-designing the fitness function in the NSGA-II variant might improve the results so far, as it can help improve the solution selection and increase accuracy.

Finally, PCA attribute reduction outperformed all other proposals; however, this extremely high level of reduction had a great negative impact on the classification accuracy, which was seriously compromised. Therefore, producing worse results than NSGA-II in all the datasets except the Iris dataset, which is extremely easy to classify.

References

Abd-Alsabour, N. (2014). A review on evolutionary feature selection. In *2014 European Modelling Symposium* (pp. 20–26). IEEE.

Algarni, A. (2016). Data mining in education. *International Journal of Advanced Computer Science and Applications, 7*(6), 456–461.

Benito-Epigmenio, L., Ibarra-Martínez, S., Ponce-Flores, M., & Castán-Rocha, J. A. (2023). Feature selection: Traditional and wrapping techniques with tabu search. In *Innovations in Machine and Deep Learning: Case Studies and Applications* (pp. 21–38). Springer Nature Switzerland.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering, 40*(1), 16–28.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings* (Vol. 6, pp. 849–858). Springer.

Dy, J. G. (2001). *Feature selection for unsupervised learning applied to content-based image retrieval* (Doctoral dissertation, Purdue University).

Graham, J. W. (2009). Missing data analysis: Making it work in the real world. *Annual Review of Psychology, 60*(1), 549–576.

Han, J., Pei, J., & Tong, H. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.

Kannan, S. S., & Ramaraj, N. (2010). A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm. *Knowledge-Based Systems, 23*(6), 580–585.

Kelly, M., Longjohn, R., & Nottingham, K. (2023). *The UCI Machine Learning Repository*. Retrieved December 31, 2024, from https://archive.ics.uci.edu/

Kim, K. I., Jung, K., & Kim, H. J. (2002). Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters, 9*(2), 40–42.

Kumar, V., & Minz, S. (2014). Feature selection. *SmartCR, 4*(3), 211–229.

Kundu, R., & Mallipeddi, R. (2022). HFMOEA: A hybrid framework for multi-objective feature selection. *Journal of Computational Design and Engineering, 9*(3), 949–965.

Lima, A., Zen, H., Nankaku, Y., Miyajima, C., Tokuda, K., & Kitamura, T. (2004). On the use of kernel PCA for feature extraction in speech recognition. *IEICE Transactions on Information and Systems, 87*(12), 2802–2811.

Lu, Y., Cohen, I., Zhou, X. S., & Tian, Q. (2007). Feature selection using principal feature analysis. In *Proceedings of the 15th ACM International Conference on Multimedia* (pp. 301–304).

Nnamoko, N., Arshad, F., England, D., Vora, J., & Norman, J. (2014). Evaluation of filter and wrapper methods for feature selection in supervised machine learning. *Age, 21*(81), 33–2.

Pal, J. K. (2011). Usefulness and applications of data mining in extracting information from different perspectives. *Annals of Library and Information Studies, 58*(1), 7.

Palo, H. K., Sahoo, S., & Subudhi, A. K. (2021). Dimensionality reduction techniques: Principles, benefits, and limitations. In *Data Analytics in Bioinformatics* (pp. 77–107). Wiley.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2*(11), 559–572.

Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports, 14*(1), 6086.

Talavera, L. (2005). An evaluation of filter and wrapper methods for feature selection in categorical clustering. In *International Symposium on Intelligent Data Analysis* (pp. 440–451). Springer.

Van Der Maaten, L., Postma, E. O., & Van Den Herik, H. J. (2009). Dimensionality reduction: A comparative review. *Journal of Machine Learning Research, 10*(66–71), 13.

Verleysen, M., & François, D. (2005). The curse of dimensionality in data mining and time series prediction. In *International Work-Conference on Artificial Neural Networks* (pp. 758–770). Springer.

Yildirim, S., Kaya, Y., & Kılıç, F. (2021). A modified feature selection method based on metaheuristic algorithms for speech emotion recognition. *Applied Acoustics, 173*, 107721.

Yu, L., & Liu, H. (2004, August). Redundancy-based feature selection for microarray data. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 737–742).