



Article

# Simultaneous Instance and Attribute Selection for Noise Filtering

Yenny Villuendas-Rey <sup>1,\*</sup>, Claudia C. Tusell-Rey <sup>2</sup> and Oscar Camacho-Nieto <sup>1,\*</sup>

<sup>1</sup> Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Juan de Dios Bátiz s/n, Mexico City 07700, Mexico

<sup>2</sup> Instituto Politécnico Nacional, Centro de Investigación en Computación, Juan de Dios Bátiz s/n, Mexico City 07738, Mexico; clautusellrey2024@gmail.com

\* Correspondence: yvilluendasr@ipn.mx (Y.V.-R.); ocamacho@ipn.mx (O.C.-N.)

**Featured Application:** The application of this paper lies in improving the preprocessing of hybrid and incomplete data for supervised classifiers.

**Abstract:** The existence of noise is inherent to most real data that are collected. Removing or reducing noise can help classification algorithms focus on relevant patterns, preventing them from being affected by irrelevant or incorrect information. This can result in more accurate and reliable models, improving their ability to generalize and make accurate predictions on new data. For example, among the main disadvantages of the nearest neighbor classifier are its noise sensitivity and its high computational cost (for classification and storage). Thus, noise filtering is essential to ensure data quality and the effectiveness of supervised classification models. The simultaneous selection of attributes and instances for supervised classifiers was introduced in the last decade. However, the proposed solutions present several drawbacks because some are either stochastic or do not handle noisy domains, and the neighborhood selection of some algorithms allows very dissimilar objects to be considered as neighbors. In addition, the design of some methods is just for specific classifiers without generalization possibilities. This article introduces an instance and attribute selection model, which seeks to detect and eliminate existing noise while reducing the feature space. In addition, the proposal is deterministic and does not predefine any supervised classifier. The experiments allow us to establish the viability of the proposal and its effectiveness in eliminating noise.

**Keywords:** noise filtering; instance selection; feature selection; nearest neighbor



**Citation:** Villuendas-Rey, Y.; Tusell-Rey, C.C.; Camacho-Nieto, O. Simultaneous Instance and Attribute Selection for Noise Filtering. *Appl. Sci.* **2024**, *14*, 8459. <https://doi.org/10.3390/app14188459>

Academic Editor: Lamberto Tronchin

Received: 24 July 2024

Revised: 13 September 2024

Accepted: 16 September 2024

Published: 19 September 2024



**Copyright:** © 2024 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) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The k nearest neighbor (NN) classifier [1] is currently still a good option when looking for a simple and effective classification model, especially in problems where interpretability and flexibility are important. Its logic is easy to explain to non-technical people, making it easy to communicate results, and it does not assume any specific distribution of data, making it flexible for different types of data. This classification model is easy to understand and implement; it does not require a training process as it is a lazy classifier, and it works well in situations where local similarity is relevant, allowing the use of different distance or dissimilarity functions according to the needs of the user.

However, one of the major disadvantages of NN classifiers is their computational cost, which grows as the training data increase. In addition, the existence of mislabeled objects or instances can affect the quality of classification. The existence of noise is inherent to most real data that are collected. There are several well-known sources of noise, such as people giving false information in some scenarios (i.e., increasing salary, decreasing age or weight, lying about health), data having transmission errors for some sensors and leading to outlier values, human errors in manual annotation, and many others. Removing or reducing noise

can help classification algorithms focus on relevant patterns, preventing them from being affected by irrelevant or incorrect information. This can result in more accurate and reliable models, which in turn improves their ability to generalize and make accurate predictions on new data. Thus, noise filtering is essential to ensure data quality and the effectiveness of supervised classification models. Several approaches have been followed for noise removal by instance selection in the context of imbalanced data [2], deep neural networks [3], and regression [4].

Data reduction and noise filtering for instance-based classifiers have been addressed in the literature by selecting features [5], selecting objects or instances [6], and also by selecting both features and instances [7–10]. This last approach can lead to better results than the sequential selection of features and instances [7] because, in the sequential selection, the first applied method can access the whole training data, but the second can only access the results of the application of the first method. On the other hand, the simultaneous (often called “dual”) approach can access all training data.

Several algorithms for selecting both instances and features have been proposed, mostly following an evolutive approach [10–12]. However, such methods are focused on reducing the data cardinality and not on noise filtering, sometimes resulting in huge drops in classifier performance. In addition, most methods are specific to certain classifiers and are not directly applicable to others. To address such issues, we use the experimental method, and the main objective of this paper is to propose a novel algorithm for simultaneous instance and feature selection that is able to detect and filter noise and reduce the training data without sacrificing classifier performance.

The contributions of this paper are as follows:

- We introduce ROFS, a deterministic method for simultaneously selecting features and instances in hybrid and incomplete data.
- The proposed model does not predefine any supervised classifier and, therefore, can be applied to different supervised classifiers.
- We analyze the performance of the compared algorithms under noisy environments and can ensure our proposal overpasses others in recognizing and deleting noisy instances.
- The statistical analysis concludes our proposal obtained significantly more accurate results while using a fraction of instances and attributes.

The remainder of this paper is as follows: Section 2 presents the datasets and the algorithms used in the experimental analysis, as well as the performance measures. Section 3 introduces the novel Robust Objective and Filtering Selection (ROFS) algorithm for simultaneously selecting attributes and instances. Section 4 discusses the experimental results and provides the corresponding statistical analysis, and Section 5 presents the conclusions and avenues of future research. The Abbreviations provides a list of all mathematical symbols used in the paper.

## 2. Materials and Methods

### 2.1. Algorithms for Simultaneous Instance and Feature Selection

There are three major approaches for simultaneous features and instances selection: evolutionary, embedding editing, and fusion. In the following, we detailed their main characteristics.

An evolutionary approach to instance and feature selection consists of the application of evolutionary methods, such as Genetic Algorithms, to select simultaneously both features and objects. This is the oldest yet still used approach for simultaneous data processing, dating back to 1994 when Skalak proposed the first method in this strategy, the RMHC-FP1 [13]. It used a Random Mutation Hill-Climbing strategy to choose the best subset of features and instances by generating a binary string of length number of features + number of objects, which represents the inclusion/exclusion of each feature and instance in the result. Then, this initial solution evolves until a termination criterion is fulfilled. With similar representation, several authors used Genetic Algorithms for simultaneously selecting

features and instances. These GAs have an encoding strategy similar to the RMHC-FP1 algorithm, an elitist selection strategy, and they usually differ in fitness functions and parameters [7,8,11,12,14,15].

The embedded editing strategy consists of embedding the instance selection process into a feature selection method. It applies an error-based editing method and a condensation method to each candidate feature set into a wrapper feature selection method. Dasarathy's method [16] (here referred to as DS) combines the Sequential Backward Search (SBS) [17] method with the application of two instance selection methods: the Proximity Graph-based Editing using Relative Neighborhood Graphs (RNG-Edt) [18] and the Minimal Consistent Subset method (MCS) [19]. DS uses as a fitness function for the SBS method a combined measure of the 1-NN accuracy with respect to a validation set and the amount of instance reduction achieved by the sequential application of the RNG-Edt + MCS methods. To combine the accuracy and reduction, he uses the Euclidean distance between both measures. The sequential application of two instance selection methods allows important instance reduction as well as atypical and mislabeled object filtering. This strategy is time-consuming but has good experimental results.

Fusion is a deterministic strategy and is based on obtaining small sets of candidate features and instances. Then, it uses different approaches for merging those candidate sets in order to obtain an optimal set of features and instances. The first realization of this strategy is the SOFSA method [20]. Similar to TCCS [21], SOFSA uses the Compact Set Editing [22] and the LEX [23] algorithms for obtaining the small candidate sets of instances and features (called submatrices). It uses a merging strategy based on the individual quality of the submatrices. Other methods of fusion are SHERA [9], designed for educational classification, and AFIS [24], focused on imbalanced data.

In this paper, we experimentally analyze the performance of different methods for simultaneously selecting features and instances in noisy environments. We choose RMHC-FP1 [13], AKH-GA [12], KJ-GA [7], IN-GA [11] (evolutionary approach), DS [16] (embedded editing approach), and SOFSA [20] and TCCS [21] (fusion approach) to ensure the representativity of the different simultaneous attribute and instance selection strategies.

## 2.2. Datasets

We selected 15 well-known hybrid and incomplete datasets (Table 1) from the Machine Learning Repository of the University of California at Irvine (UCI) [25], having multiple decision classes, to show that the proposed method can handle diverse datasets, such as hybrid, incomplete, and multiclass.

**Table 1.** Description of the datasets.

Datasets	Instances	Numerical Features	Categorical Features	Classes	Missing Values
autos	205	15	10	7	x
breast-w	699	9	0	2	
credit-a	690	6	9	2	x
diabetes	768	8	0	2	
heart-c	303	6	7	5	
hepatitis	155	6	13	2	x
iris	150	4	0	3	
labor	57	8	8	2	x
lymph	148	3	15	4	
post-operative	90	0	8	3	x
primary-tumor	339	1	16	22	x
vehicle	946	18	0	4	
vote	435	0	16	2	x
wine	178	13	0	3	
zoo	101	1	16	7	

We use 10-fold cross-validation and inject mislabeled instances into each dataset by randomly changing the class of a percent of objects: 5%, 10%, and 15% in the training set. By using cross-validation, we ensure that the training and testing sets are disjoint sets and that no testing data were used for training during the experiments.

### 2.3. Performance Measures

We analyze the performance of the different methods using three magnitudes: instance retention ratio, feature retention ratio, and classifier error. Let  $T$  be the instances in the training data described by a set of features  $A = \{A_1, \dots, A_n\}$ , and let  $P$  be the set of instances returned by the simultaneous instance and feature selection algorithm, described by a set of features  $B \subseteq A$ . The instance retention ratio is given by:

$$\text{InstanceRetention} = \frac{|P|}{|T|} \quad (1)$$

The feature retention ratio is given by:

$$\text{FeatureRetention} = \frac{|B|}{|A|} \quad (2)$$

Finally, the classifier error is computed as the ratio of incorrectly classified testing instances. Let  $X$  be the testing set, where each instance  $x \in X$  has a true class label  $\alpha(x)$ . The classifier error is computed as:

$$\text{Error} = \frac{|\{x \in X \mid \alpha(x) \neq \text{clasif}(x)\}|}{|X|} \quad (3)$$

where  $\text{clasif}(x)$  is the class label assigned by the corresponding classifier.

The main problem we are addressing is to obtain a reduced dataset by selecting a relevant set of features and instances without sacrificing classifier accuracy.

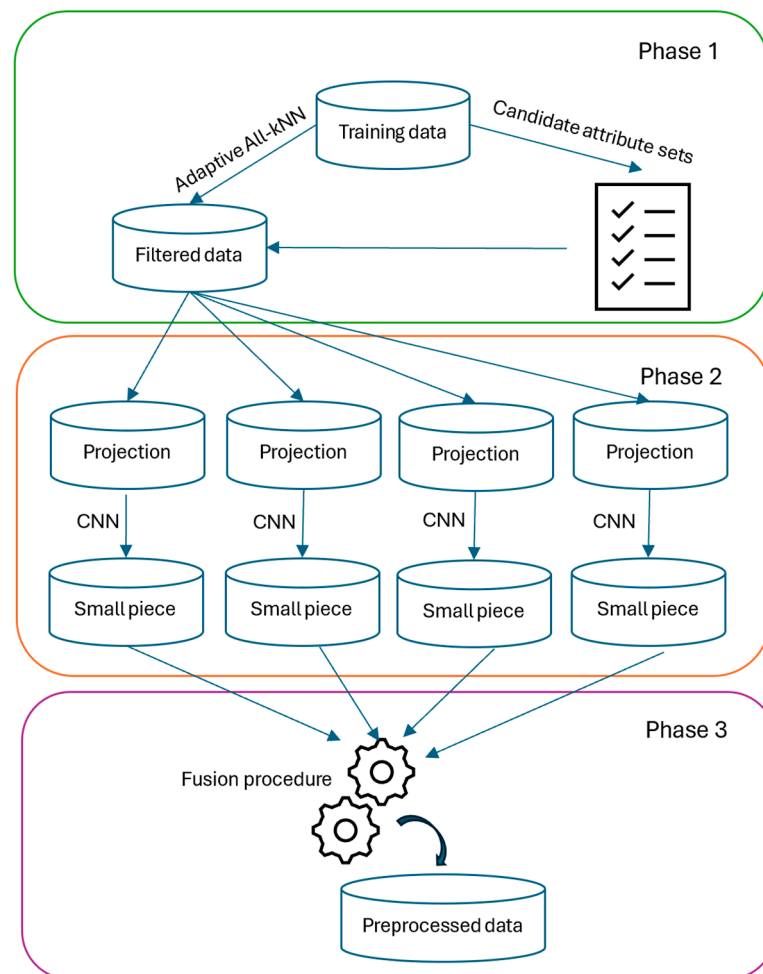
Considering these three objectives, we treat the methods' performance results as a multi-objective optimization problem with three functions to minimize instance retention, feature retention, and classifier error.

We execute the experiments on a laptop with a Windows 8 operating system, an AMD Sempron SI-42 processor at 2.1 GHz, and usable RAM of 2.75 GB, and under low priority. This is why we cannot compute the time used by the compared algorithms.

## 3. Results

This section introduces the proposed Robust Objective Filtering Selection (ROFS) algorithm. ROFS has three main phases. The first phase considers the parallel computation of candidate feature sets and adaptative noise filtering; the second phase deals with obtaining small pieces of information in the form of attributes and instances; and the third phase comprises the final integration of the small pieces. Therefore, ROFS belongs to the fusion approach to instance and feature selection. Figure 1 presents the flowchart of the proposed ROFS.

The idea of using candidate feature sets is inspired by Voting Algorithms [26] and the multi-view learning paradigm [27]. Having several possible sets of relevant attributes allows the simultaneous procedure to have a better data representation in terms of descriptive characteristics and helps discriminate between relevant and irrelevant attributes.



**Figure 1.** Flowchart of the proposed ROFS.

To obtain candidate feature sets, we can use deterministic algorithms (e.g., typical testers computation) or stochastic algorithms that return several attribute sets (e.g., evolutionary algorithms). Each possibility has its own advantages and disadvantages. Deterministic algorithms ensure finding all relevant attribute sets, but they can be computationally expensive for some datasets. On the other hand, stochastic algorithms do not always guarantee finding relevant feature sets. We do not impose any strategy for finding candidate feature sets in ROFS; on the contrary, we consider that as an algorithm parameter.

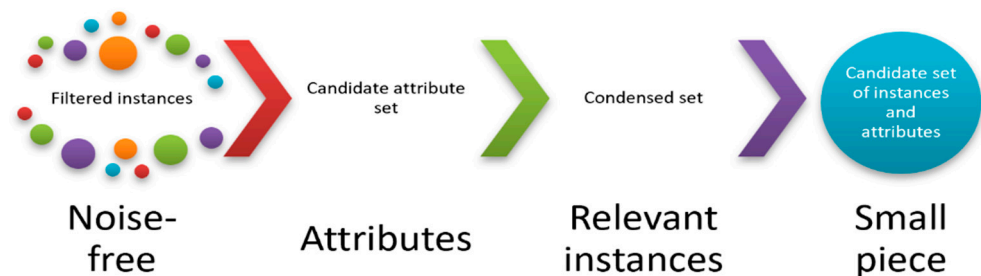
The main issue with existing noise filtering algorithms is that they can delete an entire class if considered noise. This is especially hazardous in imbalanced data with highly overlapped instances or when small disjoints appear. To avoid this situation, we introduce an adaptative noise filtering method inspired by the All-KNN editing method [28].

The All-KNN algorithm deletes the instances misclassified by one of the  $k$ -NN classifiers from  $k = 1$  to  $kMax$ . Our variant (named Adaptive All-KNN) consists of deleting the instances misclassified by one of the  $k$ -NN classifiers, but with a  $k$  value that guarantees no class is entirely deleted (see Algorithm 1). The Adaptive All-kNN considers all classes, guaranteeing that no class will be entirely deleted. So, we stop increasing  $k$  if there is a class with all instances misclassified. By doing this, we avoid an entire class deletion and adjust the algorithm parameter to better fit each dataset's specific conditions.

**Algorithm 1.** Adaptive All-KNN method

1. For  $k = 1$  to  $kMax$ 
  - 1.1. Mark every instance misclassified by a k-NN classifier
  - 1.2. Set flag to false
  - 1.3. For each class  $C_i$ 
    - 1.3.1. If all objects belonging to  $C_i$  are marked, unmark the objects marked by the last k-NN and set flag to true
  - 1.4. If flag is true, break and goto step 2
2. Return the unmarked objects

The second phase of ROFS receives the candidate feature sets and the filtered data as inputs. Then, it first obtains candidate sets of instances and attributes (small pieces of information) by using the filtered instances and attribute sets and condenses them using a condensation algorithm (see Figure 2). As for feature selection, we do not impose any condensation algorithm in ROFS or any supervised classifier. Those are user-defined parameters.



**Figure 2.** Second phase of ROFS. For each feature set, it will obtain a noise-free selected set by projecting the filtered instances using only the current attributes, and then it will obtain relevant instances by applying a condensation method, finally obtaining a small piece.

The third phase of ROFS receives the small pieces of information and fuses them into a single set of relevant features and instances. To do so, it applies a sophisticated fusion strategy. The idea is that to improve the classifier accuracy of the best small piece, it is necessary to correctly classify the instances already misclassified. Therefore, it makes no sense to integrate with another small piece if these misclassified instances will not be well classified after fusion. The ROFS integration process analyzes if there exists a small piece such that it correctly classifies the misclassified instances. If found, it then analyzes if the integration improves accuracy. If not, it does not carry the fusion process. This process avoids unnecessary integrations and contributes to obtaining small sets of instances and features. In addition, this strategy facilitates improving the desired supervised classifier (user-defined parameter) and not only the 1-NN. A complete description of the ROFS process appears in Algorithm 2.

The complexity of the proposed ROFS depends on several factors: (a) the complexity of the method to obtain candidate feature sets ( $CF$ ), (b) the complexity of the condensing method ( $D$ ), (c) the complexity of the supervised classifier used ( $S$ ), and (d) the complexity of the sorting algorithm ( $R$ ). Let  $m$  be the number of instances, and let  $f$  be the number of candidate attribute sets. The complexity of phase 1 comprises the computation of the candidate attribute sets and the computation of the Adaptive All-kNN algorithm and, therefore, is bounded by  $O(CF + m^2)$ . Phase 2 comprises the computation of the small pieces, and its complexity is bounded by the execution of the condensing algorithm over each projection as  $O(f * D)$ . Finally, the third phase consists of the fusion procedure and is bound by the sorting procedure and the complexity of the supervised classifier used over each small piece, being  $O(R + f * S)$ . Finally, the complexity of ROFS is given by  $O(CF + m^2 + f * D + R + f * S)$ .

In addition, its ability to handle diverse types of data (e.g., numeric, categorical, missing, multiclass) will depend on the method used to obtain candidate feature sets, the condensation method used, and the supervised classifier selected.

---

**Algorithm 2.** Robust Objective Filtering Selection
 

---

**Inputs:** Training set:  $T$   
 Method to compute candidate attribute sets:  $CA$   
 Method to condense instances:  $Cond$   
 Supervised classifier:  $classif$

**Output** Selected instances and attributes:  $E$

Initialization

1. Obtain a validation set  $V$  from  $T$
2. Compute the initial classifier accuracy with respect to the validation set as  $ini\_acc = classif(V, T)$

Phase 1

3. Apply the method to compute candidate attribute sets, as  $CAS = CA(T)$
4. Apply the All-KNN method for noise filtering as  $F = AdaAllKNN(T)$

Phase 2

5.  $SP = \emptyset$
6. For each candidate attribute set  $ca_i \in CAS$ 
  - 6.1. Project the instances in  $F$  using the features in to obtain an initial small piece  $ca_i$  to obtain an initial small piece  $sp_i$
  - 6.2. Apply the condensation method as  $csp_i = Cond(sp_i)$
  - 6.3. Add the current condensed small piece to  $SP$  as  $SP = SP \cup \{csp_i\}$

Phase 3

7. Sort the set  $SP$  according to the classifier accuracy with respect to the validation set  $V$  of each of the small pieces
8. Consider the best small piece as the current solution  $E$
9. While the accuracy of the current solution is lower than the initial accuracy and there exist small pieces that can improve the accuracy
  - 9.1. Find the instances misclassified by the current solution ( $Miss$ )
  - 9.2. Find the small pieces  $csp_i \in SP$  that correctly classify the misclassified instances (only consider the small pieces that contribute with at least one attribute)
  - 9.3. Consider the candidate integration  $EI = E \cup \{csp_i\}$
  - 9.4. If the fusion of the current solution with one of the found small pieces leads to better accuracy, consider it as the current solution  $E = EI$ .
10. Return  $E$

---

The advanced nature of the proposed ROFS is supported by the incorporation of the Adaptive All-kNN algorithm for noise filtering, the computation of candidate attribute sets, and the iterative process to fuse small pieces of instances and features. These novel characteristics make ROFS suitable for obtaining high-quality subsets of attributes and instances.

The theoretical justification of the design choices made in ROFS is supported by the following:

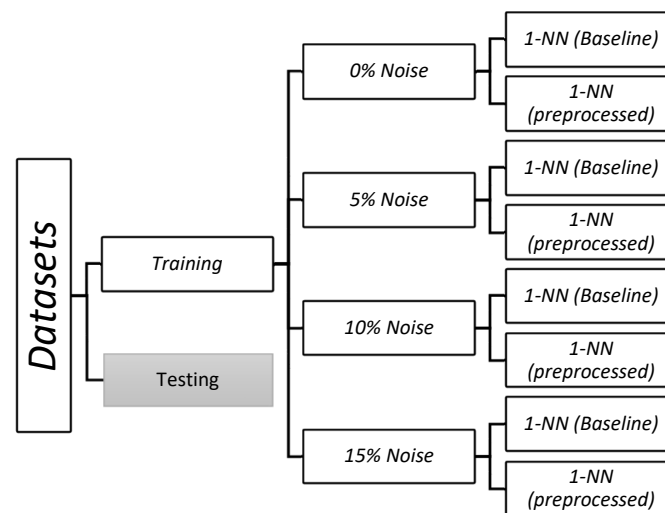
- (a) The use of candidate attribute sets is inspired by Voting Algorithms [26] and has obtained good results by supporting multiple views of the existing data.
- (b) The design of the Adaptive All-kNN maintains the desired characteristic of the previous All-kNN algorithm of reducing the Bayes error and solves the drawback of deleting too many instances.
- (c) The use in the experiments of a baseline algorithm (the Condensed Nearest Neighbor, CNN) as a condensation method preserves the decision boundary of the data and leaves room for further improvement because there has been plenty of research in data condensation techniques since 1968.

- (d) The obtention of candidate small pieces guarantees having multiple views of the same dataset, and the posterior fusion procedure allows selecting relevant attributes and instances with minimum information loss.

To assess the proposed ROFS's performance, we compared it to existing algorithms for simultaneous instance and feature selection of hybrid data. Such results are analyzed in the next section.

#### 4. Discussion

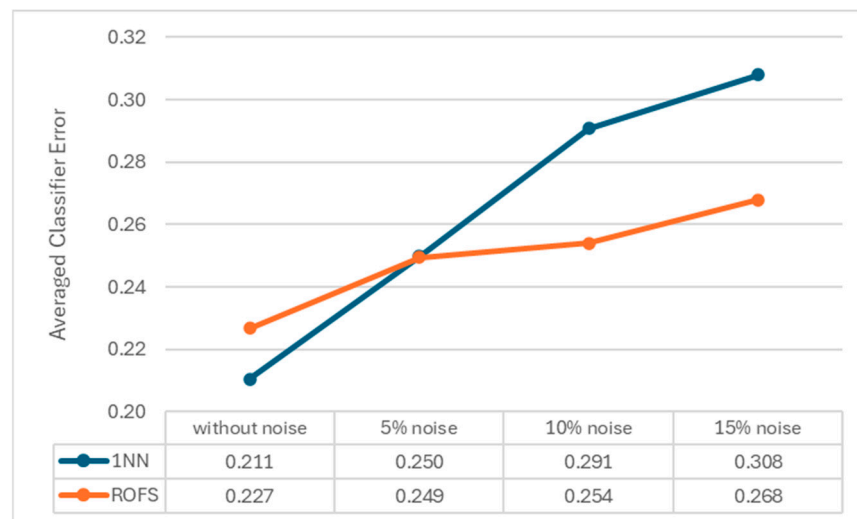
In the experimental analysis, we considered different levels of noise (5%, 10%, 15%, and the original training data). We applied the classifier under each of those conditions (Figure 3) before and after preprocessing the training data with the simultaneous instance and feature selection algorithms. We use the HOEM dissimilarity [29] in all comparisons.



**Figure 3.** Experimental configuration for each dataset. The 1-NN classifier was trained before (baseline) and after each preprocessing algorithm, considering four scenarios: without noise and with noise of 5%, 10%, and 15%. Then, the performance over the testing set was computed.

We show that the 1-NN classifier degrades as noise increases, and using the proposed algorithm contributes to diminishing the classifier's error (Figure 4). Considering the data without noise and with 5% noise, the averaged classifier error with and without preprocessing is similar (0.211 vs. 0.227 and 0.250 vs. 0.249, respectively). However, as the noise increases, the error results with preprocessing are much lower than the results without it (0.291 vs. 0.254 and 0.308 vs. 0.268, respectively).





**Figure 4.** Impact of ROFS preprocessing in the averaged performance of 1-NN, according to classifier error.

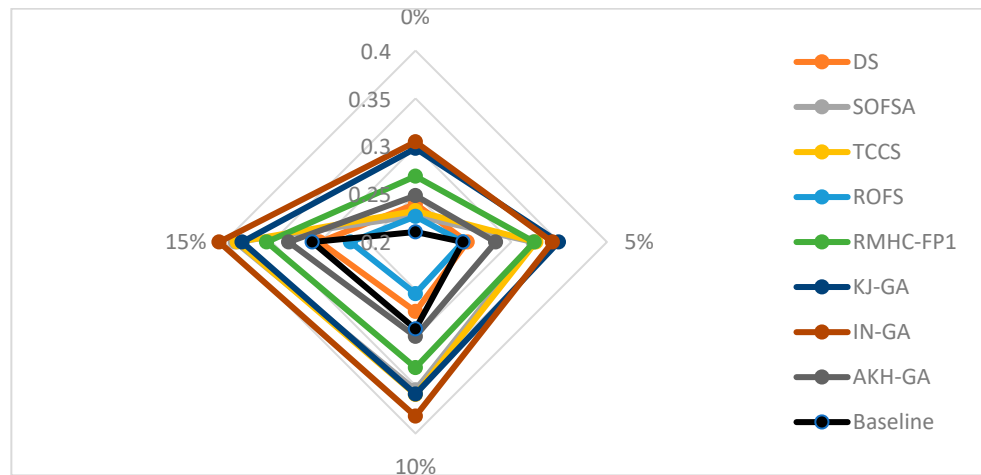
Table 2 shows the detailed classifier error results for the 1-NN classifier after preprocessing with ROFS. The detailed results for the 1-NN without preprocessing and after preprocessing with other compared algorithms can be found in the Supplementary Materials under the folders “summaries” and “results per fold”.

**Table 2.** Detailed classifier error for 1-NN with ROFS preprocessing.

Datasets	0% Noise	5% Noise	10% Noise	15% Noise
autos	0.444	0.509	0.507	0.430
breast-w	0.060	0.054	0.046	0.059
credit-a	0.174	0.201	0.188	0.197
diabetes	0.293	0.299	0.298	0.319
heart-c	0.254	0.284	0.286	0.290
hepatitis	0.226	0.226	0.221	0.273
iris	0.107	0.080	0.040	0.080
labor	0.137	0.233	0.213	0.207
lymph	0.181	0.291	0.270	0.298
post-operative	0.333	0.311	0.422	0.444
primary-tumor	0.623	0.634	0.640	0.655
vehicle	0.336	0.359	0.364	0.348
vote	0.062	0.065	0.090	0.110
wine	0.073	0.068	0.085	0.112
zoo	0.098	0.126	0.138	0.198

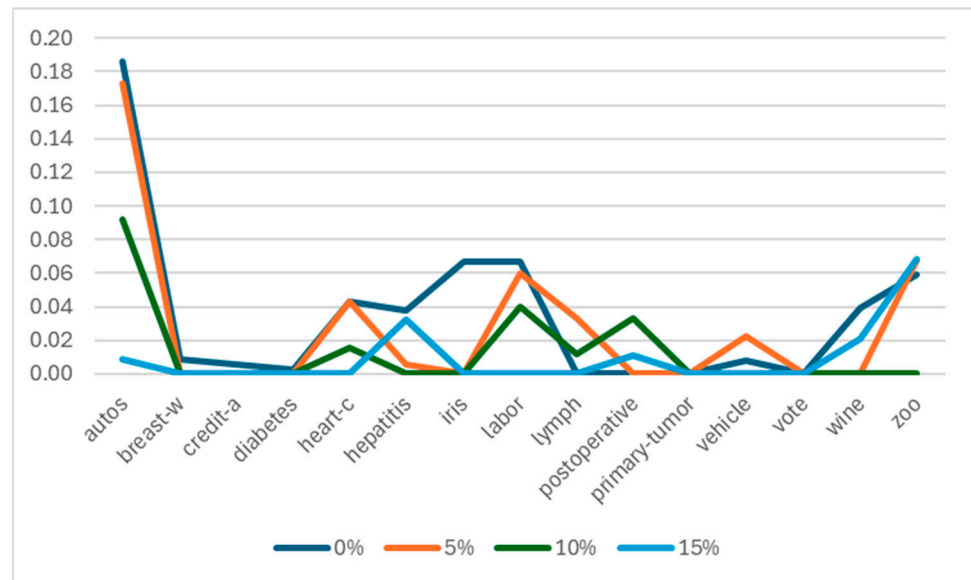
As mentioned in the Materials and Methods section, we compare the results of ROFS with respect to the following existing algorithms for simultaneous instance and feature selection: AKH-GA [12], DS [16], IN-GA [11], KJ-GA [7], RMHC-FP1 [13], SOFSA [20], TCCS [21], and the classifier without preprocessing (baseline). All methods were executed with their default parameters, as in the EPIC software version 1.0.0.7 [30]. We did not fine-tune the parameters of any algorithm (ROFS included).

Figure 5 shows the averaged classifier error of the 1-NN after using the preprocessing algorithms and the ones for the 1-NN classifier without any preprocessing (baseline). As shown, the proposed ROFS beats all other preprocessing algorithms and improves the 1-NN in all noisy scenarios. It is only slightly surpassed by 1-NN in noise-free environments.



**Figure 5.** Comparison of the averaged classifier error obtained by 1-NN with and without preprocessing.

While comparing ROFS with other algorithms according to classifier error, Figure 6 presents the results of ROFS minus the best-performing algorithm for each noise percent analyzed. As shown, the worst results of ROFS were obtained for the autos dataset without noise and with 5% noise, in which ROFS had a drop of 18% in performance. For the remaining datasets, ROFS had very good results with respect to the best-performing method (including the baseline classifier) with less than 7% noise.



**Figure 6.** Subtraction of the classifier error obtained by 1-NN with ROFS preprocessing with respect to the classifier error obtained by the best-performing algorithm (with and without preprocessing) for the analyzed scenarios.

We believe the reasons why ROFS had bad results for the autos dataset and no so good results for iris and labor datasets for noise-free scenarios do not depend on the characteristics of the overlapping region of the data but on the results of the condensing algorithm. We hypothesize that the more representative the results of the condensation method, the more accurate ROFS will be. We will need more datasets and more controlled experiments to test that hypothesis, and we will address that behavior in future works.

In addition, we want to evaluate each algorithm's ability to remove or not remove each of the noisy instances. We then computed the percentage of noise removal for each of the compared algorithms.

In Table 3, we summarize the averaged percent of noisy instances filtered by each method (algorithms appear in alphabetical order).

**Table 3.** Summary of averaged noise detection and filtering for the compared algorithms.

Methods	5% Noise	10% Noise	15% Noise
AKH-GA	0.521	0.514	0.513
DS	0.907	0.911	0.899
IN-GA	0.522	0.500	0.510
KJ-GA	0.503	0.489	0.492
RMHC-FP1	0.513	0.493	0.501
ROFS	0.931	0.927	0.930
SOFSA	0.136	0.169	0.178
TCCS	0.161	0.186	0.183

Both DS and ROFS were able to effectively deal with noisy environments, having noise filtering percents around 90%. This behavior is expected because DS uses RNG-Edt + MCS for instance selection, and RNG-Edt is designed for noise filtering. The evolutionary methods have noise detection rates around 50%, while SOFSA and TCCS almost do not detect noise at all (less than 20% of noise was detected). Evolutionary methods have a combination of objectives in their fitness functions, and they tend to keep half of instances and features. On the other hand, SOFSA and TCCS do not include any mechanism to detect noise.

In selecting both features and instances to improve training data, our first concern is to keep the classifiers' performance as high as possible. However, we also want to minimize the number of features and instances in the training data. Tables 4 and 5 present the averaged results, for instance, retention and feature retention of the compared preprocessing methods, respectively.

**Table 4.** Averaged instance retention of the compared preprocessing methods.

Methods	0% Noise	5% Noise	10% Noise	15% Noise
AKH-GA	0.502	0.500	0.502	0.496
DS	0.130	0.125	0.125	0.130
IN-GA	0.481	0.474	0.478	0.475
KJ-GA	0.489	0.485	0.49	0.486
RMHC-FP1	0.494	0.492	0.499	0.502
ROFS	0.135	0.134	0.134	0.137
SOFSA	0.609	0.592	0.608	0.631
TCCS	0.547	0.545	0.572	0.607

**Table 5.** Averaged feature retention of the compared preprocessing methods.

Methods	0% Noise	5% Noise	10% Noise	15% Noise
AKH-GA	0.509	0.513	0.506	0.511
DS	0.937	0.931	0.931	0.933
IN-GA	0.451	0.451	0.446	0.456
KJ-GA	0.475	0.468	0.465	0.490
RMHC-FP1	0.500	0.497	0.512	0.506
ROFS	0.604	0.613	0.619	0.619
SOFSA	0.809	0.799	0.804	0.813
TCCS	0.758	0.767	0.775	0.791

According to instance retention, the best method is DS (12.5–13% of retained instances), closely followed by ROFS (13.4–13.7% of retained instances). The remaining methods retained around 50% of instances. For feature selection, the methods keeping the lowest number of features are the evolutionary ones, preserving around 50% of them. The proposed ROFS retained around 60% of all features, followed by TCCS and SOFSA (around 80%), and finally, DS (over 90% of retained features).

Despite the promising results obtained by ROFS and outlined before, to determine the best methods in each scenario, we applied the Friedman test followed by the Holm post hoc test at a 95% confidence level. This combination of non-parametric statistical tests is suggested in [31]. We used the KEEL software (v.3.0) [32] for the statistical analysis. In Supplementary Materials, under the folder “statistics”, interested readers can find the full documentation reported by KEEL.

In all cases, we select as null hypothesis H0 that there are no differences in the performance of the compared algorithms and as alternative hypothesis H1 that there are differences in the performance of the compared algorithms. In the comparisons, we include the classifier error with no preprocessing as a baseline.

Table 6 presents the results of the statistical analysis for classifier error, considering the four scenarios. For each scenario, we include Friedman’s ranking as well as the probability values ( $p$ -values) obtained by Holm’s post hoc tests comparing the first-ranked algorithm with respect to all others.

**Table 6.** Friedman + Holm results for classifier error of 1-NN. Rejected hypotheses are in italics.

Methods	0% Noise		5% Noise		10% Noise		15% Noise	
	Friedman’s Ranking	Holm’s $p$ -Value	Friedman’s Ranking	Holm’s $p$ -Value	Friedman’s Ranking	Holm’s $p$ -Value	Friedman’s Ranking	Holm’s $p$ -Value
Baseline	2.467	-	2.900	0.594	3.767	0.067	3.567	0.042
ROFS	3.633	0.243	2.367	-	1.933	-	1.533	-
SOFSA	4.167	0.089	4.633	0.023	4.900	<i>0.003</i>	5.600	<i>0.000</i>
AKH-GA	4.400	0.053	4.833	0.014	4.567	<i>0.008</i>	5.033	<i>0.000</i>
DS	4.433	0.049	3.167	0.424	2.667	0.463	2.900	0.172
TCCS	4.867	0.016	5.067	<i>0.007</i>	5.500	<i>0.000</i>	5.667	<i>0.000</i>
RMHC-FP1	6.333	<i>0.000</i>	6.933	<i>0.000</i>	6.233	<i>0.000</i>	5.867	<i>0.000</i>
KJ-GA	7.067	<i>0.000</i>	7.900	<i>0.000</i>	7.567	<i>0.000</i>	7.133	<i>0.000</i>
IN-GA	7.633	<i>0.000</i>	7.200	<i>0.000</i>	7.867	<i>0.000</i>	7.700	<i>0.000</i>

However, we also want to minimize the number of features and instances in the training data. Tables 7 and 8 present the respective statistical analyses. In Tables 6–8, the compared methods are ordered according to the ranking obtained by Friedman’s test considering the noise-free scenario, and the cases in which the null hypothesis was rejected are highlighted in italics.

For the noise-free scenario, the best result was obtained by the baseline 1-NN (the first in Friedman’s ranking). ROFS, SOFSA, AKH-GA, DS, and TCCS had classifier error results without significant differences. However, RMHC-FP1, KJ-GA, and IN-GA had significantly worse performance than the baseline 1-NN. For 5% of noise, ROFS had the best result, without significant differences with the baseline classifiers SOFSA, AKH-GA, and DS. These results point out that some evolutionary algorithms do not accurately represent the data, therefore increasing classifier error after preprocessing.

The methods TCCS, RMHC-FP1, KJ-GA, and IN-GA had significantly worse performance than the proposed ROFS. As the noise increases to 10% and 15%, ROFS remains the best-performed method, without significant differences in classifier error with DS and the baseline classifier. All remaining preprocessing algorithms obtained significantly higher results for the classifier error measure.

**Table 7.** Friedman + Holm results, for instance, retention of 1-NN. Rejected hypotheses are in italics.

Methods	0% Noise		5% Noise		10% Noise		15% Noise	
	Friedman's Ranking	Holm's <i>p</i> -Value	Friedman's Ranking	Holm's <i>p</i> -Value	Friedman's Ranking	Holm's <i>p</i> -Value	Friedman's Ranking	Holm's <i>p</i> -Value
ROFS	1.467	-	1.667	0.947	1.667	0.947	1.600	-
DS	1.533	0.947	1.600	-	1.600	-	1.667	0.947
IN-GA	4.000	0.011	3.933	0.020	3.533	0.053	3.667	0.039
KJ-GA	4.667	0.001	4.600	0.003	4.400	0.005	4.200	0.009
RMHC-FP1	5.400	0.000	5.400	0.000	5.200	0.000	5.067	0.001
AKH-GA	5.867	0.000	5.933	0.000	5.400	0.000	5.600	0.000
TCCS	6.033	0.000	6.233	0.000	6.933	0.000	6.967	0.000
SOFSA	7.033	0.000	6.633	0.000	7.267	0.000	7.233	0.000
Baseline	9.000	0.000	9.000	0.000	9.000	0.000	9.000	0.000

**Table 8.** Friedman + Holm results for feature retention of 1-NN. Rejected hypotheses are in italics.

Methods	0% Noise		5% Noise		10% Noise		15% Noise	
	Friedman's Ranking	Holm's <i>p</i> -Value	Friedman's Ranking	Holm's <i>p</i> -Value	Friedman's Ranking	Holm's <i>p</i> -Value	Friedman's Ranking	Holm's <i>p</i> -Value
IN-GA	2.633	-	2.567	-	2.267	-	2.167	-
KJ-GA	3.100	0.641	2.767	0.841	2.800	0.594	2.867	0.484
RMHC-FP1	3.400	0.443	3.667	0.271	3.833	0.117	3.533	0.172
AKH-GA	3.667	0.301	3.600	0.301	3.767	0.134	3.700	0.125
ROFS	3.733	0.271	4.167	0.110	4.100	0.067	4.133	0.049
TCCS	5.467	0.005	5.567	0.003	5.633	0.001	6.100	0.000
SOFSA	6.867	0.000	6.600	0.000	6.667	0.000	6.800	0.000
DS	7.333	0.000	7.333	0.000	7.200	0.000	6.967	0.000
Baseline	8.800	0.000	8.733	0.000	8.733	0.000	8.733	0.000

The statistical analysis shows that noise significantly affects preprocessing algorithms but not ROFS and DS, therefore supporting the hypothesis that using an error-based editing approach contributes to maintaining classification performance. In addition, using heuristics does not provide evidence of obtaining highly accurate preprocessed training sets.

Regarding instance retention, the best algorithms were ROFS (first in Friedman's ranking for 0% and 15% and second in the ranking for 5% and 10%) and DS (first in Friedman's ranking for 10% and 5% and second in the ranking for 0% and 15%). There were no statistically significant differences between ROFS and DS in any of the four analyzed scenarios.

For noise-free and 5% noise, all remaining algorithms kept a significantly larger number of instances. However, for the 10% and 15% noise scenarios, the IN-GA algorithm did not show statistically significant differences with respect to the first-ranked algorithm. Unfortunately, it was significantly less accurate, as shown in Table 6. All remaining algorithms kept a significantly larger number of instances than the first-ranked algorithm for 10% and 15% noise.

The statistical analysis regarding instance selection concludes that using a combination of an error-based editing method followed by a condensation method leads to results retaining significantly fewer instances than using evolutive approaches or only condensation approaches.

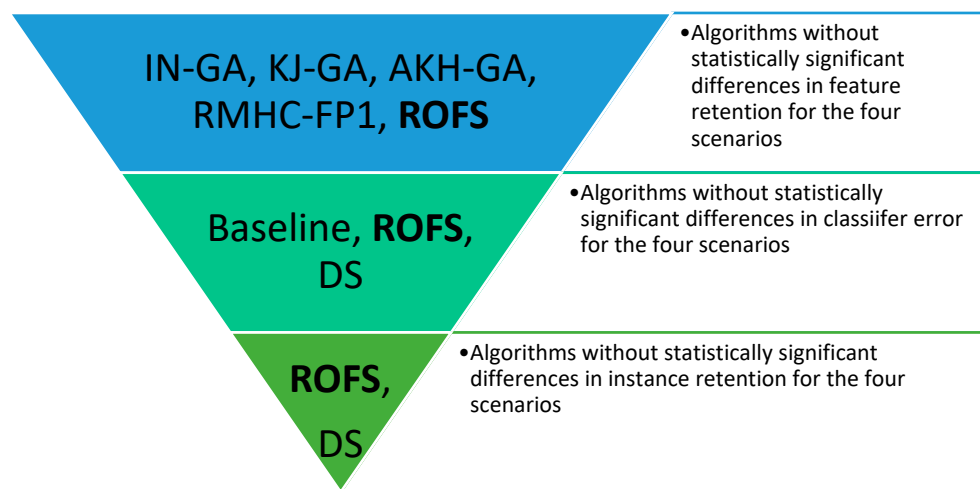
Regarding feature retention, the evolutive algorithms had the best performance, followed by ROFS and without statistically significant differences between them. The remaining algorithms kept a significantly larger number of features in all scenarios. TCCS and SOFSA do not include strategies to reduce features besides using typical testors, and DS uses SBS that starts with all features and then tries to reduce them.

These results show that evolutionary strategies lead to preprocessed data that retain significantly fewer features than using typical testors, wrapper feature selection algorithms,

or computing candidate feature sets. Unfortunately, these results were obtained at the cost of significantly increasing the classification error.

Considering three performance measures leads us to a multi-objective optimization problem. For this task, we employ the Pareto optimality criterion to assess the overall performance of the compared algorithms. The Pareto frontier finds the optimal results in multi-objective problems. A result is Pareto optimal if there is no other result that improves it simultaneously in all objectives. In our case, the three objectives are minimizing the classifier error and the number of features and instances in the training data.

Figure 7 presents the methods without significant differences in performance, according to the analyzed measures. The only algorithm in the Pareto frontier is ROFS because no other algorithm appears in the pool of best-performed methods for the three measures (i.e., it equates or surpasses all other compared algorithms).



**Figure 7.** Comparison of preprocessing algorithms according to their overall performance regarding classifier error, instance selection, and feature selection measures.

In addition, we also applied the Kolmogorov–Smirnov statistical test to compare the distributions of ROFS’s averaged performance with respect to state-of-the-art methods. For this statistical analysis, we used the graphical interface available at <https://www.statskingdom.com/kolmogorov-smirnov-two-calculator.html> (accessed on 13 September 2024), including the computation’s R code. Table 9 presents the results.

**Table 9.** *p*-value results of the Kolmogorov–Smirnov statistical test comparing the averaged performance of the algorithms.

ROFS vs.	Classifier Error	Instance Retention	Feature Retention
AKH-GA	0.2286	<i>0.0286</i>	<i>0.013</i>
DS	0.7714	<i>0.0286</i>	<i>0.013</i>
IN-GA	<i>0.0286</i>	<i>0.0286</i>	<i>0.013</i>
KJ-GA	<i>0.0286</i>	<i>0.0286</i>	<i>0.013</i>
RMHC-FP1	<i>0.0286</i>	<i>0.0286</i>	<i>0.013</i>
SOFSA	0.2286	<i>0.0286</i>	<i>0.013</i>
TCCS	0.2286	<i>0.0286</i>	<i>0.013</i>

The statistically significant results at 95% confidence are highlighted in italics.

Regarding the average performance distributions, the test found significant differences between the distributions of ROFS and all other methods in the Instance Retention and Feature Retention measures. It also found significant differences with IN-GA, KJ-GA, and RMHC-FP1 according to classifier error.

Table 10 summarizes the parameters of the compared algorithms. The values were selected as suggested in the existing papers. We did not fine-tune the parameters of any algorithm (ROFS included).

**Table 10.** Parameters of the compared preprocessing methods.

Methods	Parameters
AKH-GA	population: 200, iterations: 50, crossover probability: 0.7, mutation probability: 0.1
DS	None
IN-GA	population: 50, iterations: 500, crossover probability: 1.0, mutation probability from 0 to 1: 0.1, mutation probability from 1 to 0: 0.01
KJ-GA	population: 10, iterations: 100, crossover probability: 1.0, mutation probability: 0.1
RMHC-FP1	population: 10, iterations: 100
ROFS	CA: RoPM [33], Cond: CNN [34], <i>clasif</i> : 1-NN
SOFSa	None
TCCS	None

The statistical analysis of the compared algorithms under noisy environments concludes our proposal obtained significantly more accurate results while using a fraction of instances and attributes. In addition, our proposal surpasses others in recognizing and deleting noisy instances. As shown, the proposed ROFS always obtained good results according to the classifier error measure while keeping only a fraction of the instances and features in the datasets. As a key contribution, ROFS is deterministic, can handle hybrid and incomplete data, does not predefine any supervised classifier, and, therefore, can be applied to different supervised classifiers.

Regarding scalability, the proposed ROFS depends on the complexity of the computation of the candidate attribute sets and the instance-selection procedures. There are several possibilities for parallel implementation of such algorithms, making them suitable for handling big data.

In addition, regarding the types of real-world problems where ROFS would be most beneficial, we consider that our proposal is particularly useful for social data because it is able to handle hybrid (numeric and categorical) data with multiple decision classes. The main potential limitations it might face in practice are related to handling big data because it will need parallel implementations of the proposal.

## 5. Conclusions

In this paper, we analyzed some of the strategies for simultaneously selecting features and instances and proposed a novel method for this task. We conducted wide numerical experimentation over 15 well-known repository datasets and compared the overall performance of several methods in different scenarios.

Our experimental analysis shows that using error-based editing methods contributes to classifier performance. In addition, using a combination of error-based editing followed by a condensation algorithm leads to highly accurate training sets with fewer instances. However, using evolutionary preprocessing can lead to significant drops in classifier accuracy. Regarding the proposed algorithm, it had the best results by having a good tradeoff between reduction in both features and instances, as well as keeping classifier accuracy as high as possible.

The ROFS method filters noisy instances and focuses on the correct classification of misclassified objects. This method is also deterministic, and it is able to deal with hybrid and incomplete data, as well as using any supervised classifier. Our method outperformed others according to classifier error, retaining a significantly smaller number of instances and features. We believe the main reason ROFS obtained such good results is related to the computation of candidate attribute sets, the use of error-based editing

plus a condensing algorithm, and the intelligent fusion procedure focused on the correct classification of instances.

In future work, we want to explore the use of strategies for making ROFS suitable to deal with big data, such as parallel computation of candidate attribute sets. In addition, we want to test other condensation methods besides CNN to evaluate the impact of specific techniques on the results of the ROFS preprocessing.

**Supplementary Materials:** The supporting information can be downloaded at: <https://github.com/yenny-villuendas/ROFS> (accessed on 15 September 2024).

**Author Contributions:** Conceptualization, Y.V.-R.; methodology, Y.V.-R.; software, Y.V.-R.; validation, O.C.-N., C.C.T.-R. and Y.V.-R.; formal analysis, O.C.-N.; investigation, C.C.T.-R.; data curation, C.C.T.-R. and Y.V.-R.; writing—original draft preparation, Y.V.-R.; writing—review and editing, O.C.-N.; visualization, C.C.T.-R.; supervision, Y.V.-R. and O.C.-N.; project administration, Y.V.-R. 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 data used in this research are publicly available at: <https://archive.ics.uci.edu/datasets> (accessed on 14 April 2024).

**Acknowledgments:** The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, Centro de Innovación y Desarrollo Tecnológico en Cómputo, and Centro de Investigación en Computación), the Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT), and Sistema Nacional de Investigadores (SNII) for their support in developing this work.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

In the following, we list all mathematical symbols used in this paper.

Symbol	Description
T	Training set
X	Testing set
x	Testing instance, $x \in X$
$\alpha(x)$	True label of x
classif	Supervised classifier
clasif(x)	Label assigned to x by the supervised classifier
P	Instance set returning by data preprocessing, $P \subseteq T$
V	Validation set, $V \subseteq T$
F	Instance resulting from applying Adaptive All kNN algorithm, $F \subseteq T$
A	Attribute set describing the instances, $A = \{A_1, \dots, A_n\}$
n	Number of attributes describing the instances
B	Attributes selected by preprocessing algorithms, $B \subseteq A$
k	Number of neighbors
kMa	Maximum number of neighbors
CAS	Candidate Attribute Sets, $CAS = \{ca_1, \dots, ca_p\}$ , where $ca_i \subseteq A$
$Ca_i$	Candidate attribute set, element of CAS
CA	Method to compute Candidate Attribute Sets
Cond	Condensation method, for instance, selection
$C_i$	The i-th decision class
SP	Small pieces of instances and attributes
csp	Result of condensing an SP



Miss	Misclassified instances by the current solution
EI	Candidate integration
E	Set of selected instances and features returned by ROFS algorithm
m	Number of instances
CF	Computational complexity of the procedure to obtain candidate feature sets
D	Computational complexity of the condensing algorithm
S	Computational complexity of the supervised classifier
R	Computational complexity of the sorting algorithm
f	Number of obtained candidate attribute sets, $f =  CAS $

## References

- Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [\[CrossRef\]](#)
- Dixit, A.; Mani, A. Sampling technique for noisy and borderline examples problem in imbalanced classification. *Appl. Soft Comput.* **2023**, *142*, 110361. [\[CrossRef\]](#)
- Song, H.; Kim, M.; Park, D.; Shin, Y.; Lee, J.-G. Learning from noisy labels with deep neural networks: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 8135–8153. [\[CrossRef\]](#) [\[PubMed\]](#)
- Li, C.; Mao, Z. A label noise filtering method for regression based on adaptive threshold and noise score. *Expert Syst. Appl.* **2023**, *228*, 120422. [\[CrossRef\]](#)
- Theng, D.; Bhojar, K.K. Feature selection techniques for machine learning: A survey of more than two decades of research. *Knowl. Inf. Syst.* **2024**, *66*, 1575–1637. [\[CrossRef\]](#)
- Cunha, W.; Viegas, F.; França, C.; Rosa, T.; Rocha, L.; Gonçalves, M.A. A Comparative Survey of Instance Selection Methods applied to Non-Neural and Transformer-Based Text Classification. *ACM Comput. Surv.* **2023**, *55*, 1–52. [\[CrossRef\]](#)
- Kuncheva, L.I.; Jain, L.C. Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognit. Lett.* **1999**, *20*, 1149–1156. [\[CrossRef\]](#)
- Pérez-Rodríguez, J.; Arroyo-Peña, A.G.; García-Pedrajas, N. Simultaneous instance and feature selection and weighting using evolutionary Computation: Proposal and Study. *Appl. Soft Comput.* **2015**, *37*, 416–443. [\[CrossRef\]](#)
- Villuendas-Rey, Y.; Rey-Benguría, C.; Lytras, M.; Yáñez-Márquez, C.; Camacho-Nieto, O. Simultaneous instance and feature selection for improving prediction in special education data. *Program* **2017**, *51*, 278–297. [\[CrossRef\]](#)
- García-Pedrajas, N.; del Castillo, J.A.R.; Cerruela-García, G. SI (FS) 2: Fast simultaneous instance and feature selection for datasets with many features. *Pattern Recognit.* **2021**, *111*, 107723. [\[CrossRef\]](#)
- Ishibuchi, H.; Nakashima, T. Evolution of reference sets in nearest neighbor classification. In *Selected Paper 2, Proceedings of the Simulated Evolution and Learning: Second Asia-Pacific Conference on Simulated Evolution and Learning, SEAL'98, Canberra, Australia, 24–27 November 1998*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 82–89.
- Ahn, H.; Kim, K.-J.; Han, I. A case-based reasoning system with the two-dimensional reduction technique for customer classification. *Expert Syst. Appl.* **2007**, *32*, 1011–1019. [\[CrossRef\]](#)
- Skalak, D.B. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Machine Learning Proceedings 1994*; Elsevier: Amsterdam, The Netherlands, 1994; pp. 293–301.
- Derrac, J.; Cornelis, C.; García, S.; Herrera, F. Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection. *Inf. Sci.* **2012**, *186*, 73–92. [\[CrossRef\]](#)
- García-Pedrajas, N.; De Haro-García, A.; Pérez-Rodríguez, J. A scalable approach to simultaneous evolutionary instance and feature selection. *Inf. Sci.* **2013**, *228*, 150–174. [\[CrossRef\]](#)
- Dasarathy, B.; Sánchez, J. Concurrent feature and prototype selection in the nearest neighbor based decision process. In *Proceedings of the 4th World Multiconference on Systems, Cybernetics and Informatics, Orlando, FL, USA, 23–26 July 2000*; pp. 628–633.
- Kittler, J. Feature set search algorithms. In *Pattern Recognition and Signal Processing*; Chen, C.J., Ed.; Springer: Dordrecht, The Netherlands, 1978; pp. 41–69.
- Toussaint, G.T. Proximity graphs for nearest neighbor decision rules: Recent progress. In *Proceedings of the Interface 2002, 34th Symposium on Computing and Statistics (Theme: Geoscience and Remote Sensing), Montreal, QC, Canada, 17–20 April 2002*.
- Dasarathy, B.V. Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Trans. Syst. Man Cybern.* **1994**, *24*, 511–517. [\[CrossRef\]](#)
- Villuendas-Rey, Y.; García-Borroto, M.; Medina-Pérez, M.A.; Ruiz-Shulcloper, J. Simultaneous features and objects selection for Mixed and Incomplete data. In *Proceedings of the Iberoamerican Congress on Pattern Recognition, Cancun, Mexico, 14–17 November 2006*; pp. 597–605.
- Villuendas-Rey, Y.; García-Borroto, M.; Ruiz-Shulcloper, J. Selecting features and objects for mixed and incomplete data. In *Proceedings of the Progress in Pattern Recognition, Image Analysis and Applications: 13th Iberoamerican Congress on Pattern Recognition, CIARP 2008, Havana, Cuba, 9–12 September 2008*; pp. 381–388.
- García-Borroto, M.; Ruiz-Shulcloper, J. Selecting prototypes in mixed incomplete data. In *Proceedings of the Iberoamerican Congress on Pattern Recognition, Havana, Cuba, 15–18 November 2005*; pp. 450–459.

23. Santiesteban, Y.; Pons-Porrata, A. LEX: A new algorithm for the calculus of typical testors. *Math. Sci. J.* **2003**, *21*, 85–95.
24. Villuendas-Rey, Y.; Yáñez-Márquez, C.; Camacho-Nieto, O. Ant-based feature and instance selection for multiclass imbalanced data. *IEEE Access*, 2024, *Online ahead of print*.
25. Kelly, M.; Longjohn, R.; Nottingham, K. UCI Machine Learning Repository. Available online: <http://archive.ics.uci.edu/ml> (accessed on 14 April 2024).
26. Rodríguez-Salas, D.; Lazo-Cortés, M.S.; Mollineda, R.A.; Olvera-López, J.A.; de la Calleja, J.; Benitez, A. Voting Algorithms Model with a Support Sets System by Class. In Proceedings of the Nature-Inspired Computation and Machine Learning: 13th Mexican International Conference on Artificial Intelligence, MICAI 2014, Tuxtla Gutiérrez, Mexico, 16–22 November 2014; pp. 128–139.
27. Zhao, J.; Xie, X.; Xu, X.; Sun, S. Multi-view learning overview: Recent progress and new challenges. *Inf. Fusion* **2017**, *38*, 43–54. [[CrossRef](#)]
28. Tomek, I. An experiment with the Edited Nearest-Neighbor Rule. *IEEE Trans. Syst. Man Cybern. SMC* **1976**, *6*, 448–452.
29. Wilson, D.R.; Martinez, T.R. Improved heterogeneous distance functions. *J. Artif. Intell. Res.* **1997**, *6*, 1–34. [[CrossRef](#)]
30. Hernández-Castaño, J.A.; Villuendas-Rey, Y.; Camacho-Nieto, O.; Yáñez-Márquez, C. Experimental platform for intelligent computing (EPIC). *Comput. Sist.* **2018**, *22*, 245–253. [[CrossRef](#)]
31. García, S.; Herrera, F. An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *J. Mach. Learn. Res.* **2008**, *9*, 2677–2694.
32. Triguero, I.; González, S.; Moyano, J.M.; García López, S.; Alcalá Fernández, J.; Luengo Martín, J.; Fernández Hilario, A.; Díaz, J.; Sánchez, L.; Herrera Triguero, F. *KEEL*, version 3.0; An Open Source Software for Multi-Stage Analysis in Data Mining; University of Granada: Granada, Spain, 2017.
33. Gómez, J.P.; Montero, F.E.H.; Sotelo, J.C.; Mancilla, J.C.G.; Rey, Y.V. RoPM: An algorithm for computing typical testors based on recursive reductions of the basic matrix. *IEEE Access* **2021**, *9*, 128220–128232. [[CrossRef](#)]
34. Hart, P. The condensed nearest neighbor rule (corresp.). *IEEE Trans. Inf. Theory* **1968**, *14*, 515–516. [[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.