www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 405-419. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2025.v16i3.845

Instance Selection for Hybrid and Incomplete Data based on Clustering

Claudia C. Tusell-Rey[1], Yenny Villuendas-Rey[2], Oscar Camacho-Nieto[2], Cornelio Yáñez-Márquez[1], and Viridiana Salinas-García[2]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, Juan de Dios Bátiz s/n, GAM, CDMX 07738

[2] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Juan de Dios Bátiz s/n, GAM, CDMX 07700

E-mails: clautusellrey2014@gmail.com; {yvillluendasr;ocamacho;coryanez;vsalinasg}@ipn.mx

Abstract. This paper presents the HICCS algorithm, a novel clustering approach that handles hybrid and incomplete data. HICCS improves clustering by using compact sets as initial clusters, employing holotypes to measure intergroup dissimilarity, and merging clusters based on similarity in an order-independent manner. Additionally, it incorporates a user-defined similarity function, making it adaptable to various real-world domains. Furthermore, we introduce the IS-HICCS algorithm for instance selection, which reduces the instance set without compromising classifier accuracy, highlighting clustering's potential to enhance supervised classification models. We evaluate HICCS and IS-HICCS on synthetic and real-life datasets, showing their statistically superior performance compared to other clustering and instance selection methods, respectively.

Keywords: instance selection, hybrid and incomplete data, clustering.

Article Info
Received January 31, 2025
Accepted March 21, 2025

1 Introduction

Unsupervised classification, or clustering, is a powerful method used to analyze, understand, and process data across a wide range of fields (Balcan, Blum, & Vempala, 2008), making it a core technique in data analysis and machine learning. It works by grouping data into clusters, where elements within the same cluster are more similar to each other than to those in other clusters, based on a defined similarity or dissimilarity metric. Clustering is important in various aspects, including:

- **Data exploration**: Clustering helps identify hidden patterns and structures in datasets, aiding in the understanding of the data and revealing previously unknown insights (Fritz, Behringer, Tschechlov, & Schwarz, 2022).
- **Data compression**: By grouping similar data points, clustering can reduce the overall complexity of the data, simplifying models and decreasing computational costs (Lin et al., 2021; Paek & Ko, 2015).
- **Anomaly detection**: Clustering can highlight outliers or anomalies that do not fit well into any of the natural clusters, making it useful for detecting unusual data points (Gunay & Shi, 2020; M. Jain, Kaur, & Saxena, 2022).
- **Data preprocessing**: Clustering is commonly used as an initial step in machine learning workflows, helping to clean and reduce noise in data, which can ultimately improve the accuracy of subsequent models (Askari, 2021; Zhu et al., 2020). In this paper, we will use it for instance selection.

Various clustering methods have been developed and applied across different domains, such as text mining (George & Sumathy, 2023; D. Jain, Borah, & Biswas, 2024), market segmentation (Pradana & Ha, 2021; Sarkar, Puja, & Chowdhury, 2024), and bioinformatics (Zheng He, Shen, Zhou, & Wang, 2024). However, real-world data often involves a combination of numerical and categorical attributes, along with missing information. Therefore, it's essential for clustering algorithms to effectively handle both hybrid data types and missing data.

In this paper, we introduce a new instance selection algorithm named IS-HICCS (Instance Selection based on HICCS), which is based on a novel clustering algorithm named HICCS (Hybrid Instance Clustering based on Compact Sets), which is particularly

effective for clustering hybrid and incomplete datasets. HICCS handles hybrid data by first creating a compact set structure and then performing clustering using an agglomerative approach. Unlike other methods, it utilizes real data instances as cluster representatives, eliminating the need for artificially created instances. We take advantage of such a representative selection strategy to perform instance selection and to improve data preprocessing for supervised classification.

The structure of the paper is as follows: In Section 2, we review related work and discuss the strengths and limitations of existing clustering algorithms as well as their application to instance selection. Section 3 presents the IS-HICCS algorithm, followed in Section 4 by a performance comparison of HICCS and IS-HICCS with other hybrid-data clustering algorithms and instance selection algorithms, respectively. Finally, we conclude the paper in Section 5.

2 Related Works

2.1 Clustering hybrid and incomplete data

Several researchers have worked on the task of unsupervised classification, although the challenge of handling hybrid and incomplete data is often overlooked. Clustering algorithms can generally be categorized based on the heuristic used to form clusters. Below, we discuss various approaches:

K-means-based Algorithms
K-means-based algorithms, often referred to as center-based or partitional clustering methods, are based on the original k-means model (A. K. Jain & Dubes, 1988). This approach involves selecting k cluster centers, assigning instances to those centers, and then updating the centers. For hybrid data, notable K-means-based models include K-Prototypes (Huang, 1998), which separately handle numeric and categorical data; K-means with Similarity Function (KMSF) (García-Serrano & Martínez-Trinidad, 1999), which uses a user-defined similarity to manage hybrid and missing data; and AD, which employs descriptive cluster centers (Ahmad & Dey, 2007). The main limitations of center-based algorithms are their inability to capture non-spherical cluster shapes and their reliance on initial center selection (Ikotun, Ezugwu, Abualigah, Abuhaija, & Heming, 2023).

Hierarchical Algorithms
Hierarchical clustering organizes data into a tree-like structure called a dendrogram. There are two common approaches: bottom-up, where each instance starts as its own cluster and merges with others; and top-down, where all data begins in a single cluster, which is then recursively divided into smaller clusters (Murtagh & Contreras, 2012). The primary drawbacks of hierarchical clustering are its computational complexity (especially for top-down approaches) and its dependence on the order of merging or dividing instances. Algorithms like HIMIC (Ahmed, Borah, Bhattacharyya, & Kalita, 2005) and AERE (Reyes-González & Ruiz-Shulcloper, 1999) have been used for clustering hybrid data.

Model-based Algorithms
Model-based clustering relies on existing models and heuristics to identify clusters, often applying evolutionary and swarm intelligence algorithms. The clustering process is framed as an optimization task aiming to maximize cluster quality, usually measured by an internal validity index. Strategies for model-based clustering of hybrid data include methods like Artificial Bee Colonies and Firefly optimization (Villuendas-Rey, Barroso-Cubas, Camacho-Nieto, & Yáñez-Márquez, 2021), and Genetic Algorithms (Roy & Sharma, 2010).

Ensemble-based Algorithms
Ensemble-based clustering splits the dataset into two subsets—one with numerical data and the other with categorical data—and applies separate clustering algorithms to each. The results are then combined. For instance, the CEBMDC clustering method (Zengyou He, Xu, & Deng, 2005) uses k-means for the numerical subset and Squeezer (Zengyou He, Xu, & Deng, 2002) for the categorical subset, with Squeezer also combining the results. A key limitation of ensemble-based clustering is that by dividing the data, these algorithms fail to account for potential dependencies between numerical and categorical attributes. They also depend heavily on the quality of the individual numerical and categorical clustering methods.

While various clustering techniques have been proposed, no single algorithm addresses all the challenges, and none are universally applicable to every situation. In particular, effectively clustering hybrid and incomplete data remains a persistent challenge for the scientific community.

2.2 Clustering for instance selection

Instance selection is a critical preprocessing step in supervised classification tasks. It involves selecting a subset of relevant instances from a larger dataset to train a classifier more efficiently. The goal is to improve classification accuracy, reduce computational cost, and mitigate issues such as overfitting (Xu & Zhang, 2024). Clustering has been explored as an effective method for instance selection in supervised classification (Tsai, Lin, Hu, & Yao, 2019).

By leveraging clustering for instance selection, we can identify representative examples that capture the essence of each cluster while discarding redundant or outlying instances. After clustering, a selection process is applied to identify a subset of instances from each cluster to be used for training the classifier. The idea is that representative instances from each cluster can effectively capture the underlying patterns in the data, ensuring the classifier learns from the most informative examples.
Several key approaches can be used to implement instance selection through clustering:

1. **Selecting Centroids as Representatives**: One of the most straightforward approaches is to use the centroids (or cluster centers) as representatives for each cluster. In restricted clustering, for example, after the algorithm has partitioned the data into k clusters, the centroid of each cluster is selected as a representative instance. By training the classifier on these centroids, the model can learn the essential characteristics of each group while reducing the number of instances needed (Cohen, Hilario, Sax, Hugonnet, & Geissbuhler, 2006).
2. **Selecting Boundary Instances**: Instead of using the centroids, another approach is to select instances that are on the boundary of each cluster. These boundary instances are typically the data points that are closest to the cluster's centroid or that lie near the decision boundary between clusters. By selecting these instances, the classifier is exposed to diverse examples that are critical for distinguishing between different classes. This method can be particularly useful in situations where the boundary between clusters is important for classification tasks (Saha, Sarker, Al Saud, Shatabda, & Newton, 2022).
3. **Cluster-based Representative Sampling**: In some cases, rather than simply selecting the centroid or border instances, a more sophisticated sampling technique can be employed. For example, is it possible to use hyperrectangle clustering for detecting the desired instances to sample. The idea is to ensure that the selected instances capture the variance within each cluster, improving the diversity of the training data without introducing redundancy (Hamidzadeh, Monsefi, & Yazdi, 2015).
4. **Handling Imbalanced Data with Clustering**: In cases where the data is imbalanced (i.e., some classes are significantly underrepresented), clustering can help identify which classes or clusters are poorly represented. By selectively choosing instances from underrepresented clusters, the instance selection process can ensure that the classifier receives sufficient training examples from all classes. This can help improve classification performance, especially when dealing with skewed class distributions (Camacho-Nieto, Yáñez-Márquez, & Villuendas-Rey, 2020).

However, most of the existing instance selection algorithms based on clustering assume numeric and complete data (Hamidzadeh et al., 2015),(Saha et al., 2022), and others have a high computational cost (Cohen et al., 2006). In the following, we aim to design an instance selection algorithm based on clustering, suitable for hybrid and incomplete data and with a tractable computational complexity.

3 Instance selection and clustering

3.1 Hybrid Instance Clustering Algorithm Based on Compact Sets

In this subsection, we introduce the Hybrid Instance Clustering algorithm based on Compact Sets (HICCS). Let us have a set of instances X described by a set of attributes or features $A = \{A_1, \dots, A_n\}$, and let $x \in X$ be an instance, and $x[i]$ be the value of the i-th feature in the instance x. If $x[i] =?$ then the value of its i-th feature is missing. The objective of a restricted clustering algorithm is to obtain a partition of X into k disjoint subsets, in a way such that it maintains the subsets as compact (similar instances in the same partition) and separated (different instances in different partitions) as possible. Our approach utilizes an agglomerative strategy, halting when the desired number of clusters (k) is reached.

Like many other agglomerative algorithms, our method employs multilayer clustering to construct a hierarchy. As the algorithm progresses through layers, the granularity of the clusters increases— the top layer represents the most general clusters, while the bottom layer (the leaf nodes) represents the most specific. At each level, from root to leaves, the nodes correspond to subsets of

the parent node. At every layer, the process involves calculating the most similar nodes, merging them, and recalculating the node holotypes. Our algorithm considers the instance most similar to all instances in its node as the holotype. The overall process is illustrated in Fig. 1.

Our algorithm requires a similarity measure between objects that can handle hybrid and incomplete data. It begins by calculating the compact sets (Trinidad, Shulcloper, & Cortés, 2000) of the data, with each compact set treated as a separate group. For each group, the algorithm identifies the most similar instance to all other instances in the group. This holotype is selected as the representative of the cluster. If the stopping condition is not met, the algorithm locates the most similar clusters, merges them, and determines the new holotype for the merged group. This process repeats until the stopping condition is satisfied.

Compact sets are the connected components of a maximum similarity graph (Trinidad et al., 2000), and they have been effectively used for handling hybrid and missing data, yielding satisfactory results (Tusell-Rey, Camacho-Nieto, Yáñez-Márquez, & Villuendas-Rey, 2022; Villuendas-Rey, 2022).

To further clarify our approach, consider two-dimensional instances and a similarity function defined as the reciprocal of the Euclidean distance. Fig. 2 illustrates the process of cluster formation (in this case, continuing until all instances are merged into a single cluster, i.e., k=1). Additionally, the full pseudocode for the HICCS algorithm is provided in Fig. 3.

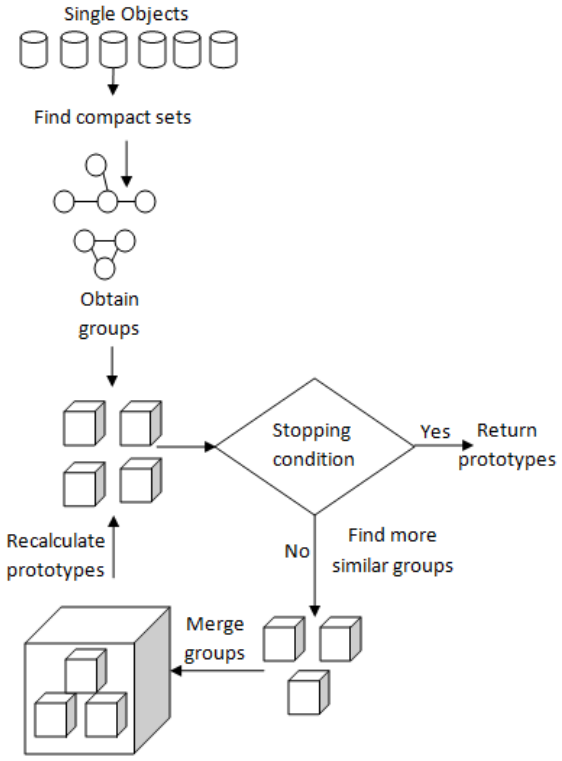

Fig. 1. Overview of the HICCS algorithm. The process begins with the instance similarity matrix, followed by the calculation of compact sets, with each compact set treated as a cluster. If the stopping condition is not met, the algorithm identifies and merges the most similar clusters, continuing this process until the stopping condition is satisfied.

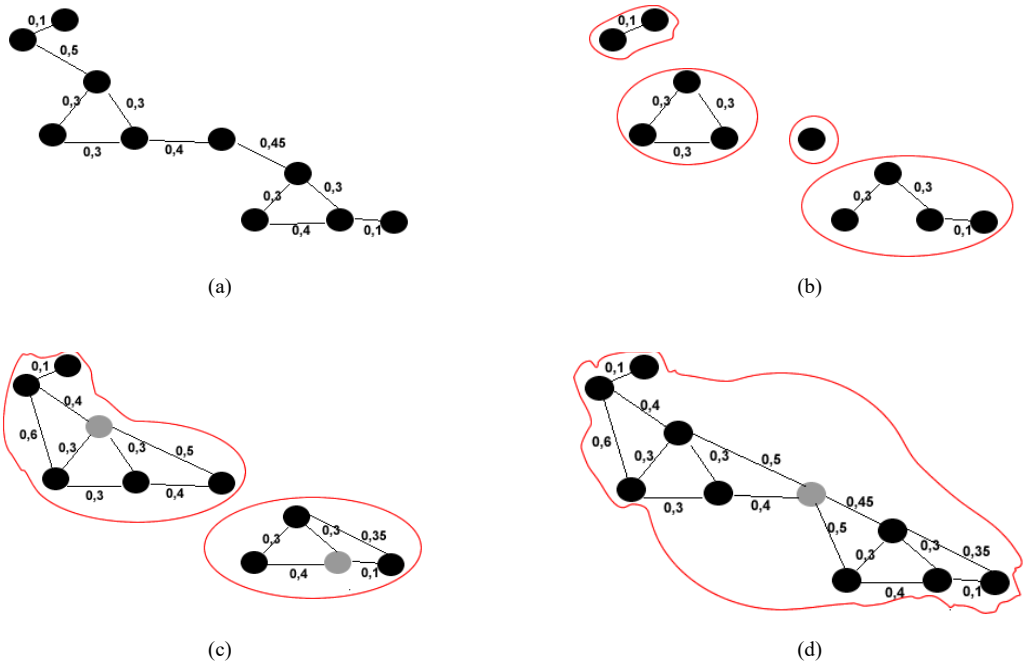

Fig. 2. Example of 2D points and the corresponding HICCS clustering. a) A set of points along with the Euclidean distances between them. b) Initial clustering using compact sets, as described in step 2 of the algorithm (with the cluster holotypes shown in gray). c) and d) Clusters formed by merging the two most similar clusters (based on holotype distances) and the selection of the new holotypes for the merged clusters.

Algorithm # 1 HICCS - Hybrid Instance Clustering algorithm based on Compact Sets
Inputs: MI: matrix of instances k: number of clusters to obtain d: inter instance similarity function D: inter groups dissimilarity function
Output: C: clustering partition
1. C = ϕ 2. Create a maximum similarity graph using the similarity function d. 3. Add to C each connected component of the graph created at step 1. 3.1. Select as cluster center (holotype or prototype) the object that minimizes the overall dissimilarity with respect to every object in the cluster 4. Merge all less dissimilar groups, using the between cluster dissimilarity D 4.1. Recalculate cluster prototypes as in 3.1 5. Repeat step 4, until *k* clusters are obtained. 6. Return C

Fig. 3. Overview of the HICCS algorithm. The process begins by calculating the maximum similarity graph, followed by the computation of compact sets, with each compact set treated as a separate cluster. If the stopping condition is not met, the algorithm identifies and merges the most similar clusters, continuing this process until the desired number of clusters are obtained.

Our approach to clustering differs from previously reported algorithms in the following ways:

- Rather than treating individual instances separately, HICCS uses the connected components of a maximum similarity graph (compact sets) as the initial clustering, which significantly diminishes computational complexity. It also stores the similarity matrix between instances, for further efficient computation.
- It eliminates the need for additional similarity calculations between instances by using holotypes to determine the dissimilarity between groups.

- It merges all selected clusters at each stage based on their similarity, ensuring that the process is independent of the order of operations. For example, if the clustering is C={c1, c2, c3, c4, c5}, and there are three pairs of highly similar groups, such as D(c1, c2) = D(c1, c5) = D(c3, c4) = min{D(ci, cj)}, the HICCS algorithm merges them in a single step. The updated clustering would be C = {{c1 ∪ c2 ∪ c5}, {c3 ∪ c4}}. This speeds up the merging process and removes any dependency on the order of cluster merging.

Clustering-based instance selection provides a powerful method for improving the efficiency and effectiveness of supervised classification. By identifying representative instances through clustering, we can reduce the size of the training set while maintaining the quality of the data used for training. This approach offers several advantages, including improved model generalization, reduced computational cost, and better handling of noisy or imbalanced data. The next subsection presents our proposal for instance selection based on HICCS clustering.

3.2 Instance Selection Based on HICCS

In this subsection, we introduce the proposed algorithm for Instance Selection based on the Hybrid Instance Clustering algorithm based on Compact Sets (IS-HICCS). Again, let us have a set of instances X described by a set of attributes or features $A = \{A_1, \dots, A_n\}$, and let $x \in X$ be an instance, and $x[i]$ be the value of the i-th feature in the instance x. If $x[i] =?$ then the value of its i-th feature is missing. The objective of an instance selection algorithm is to obtain a subset E of X, in a way such that it conserves representative instances. The idea is that if we train a supervised classifier with the selected set E, its performance be equal or better than if we train the classifier with the original training set X.

There are several benefits of instance selection based on clustering. Clustering helps reduce the size of the training set by selecting a representative subset of instances, which in turn reduces the computational burden. With fewer instances, the training process becomes faster, and the classifier can be trained on larger datasets or with limited computational resources. In addition, by selecting representative instances that capture the diversity of the data, clustering-based instance selection can help improve the classifier's ability to generalize to new, unseen data. The classifier is trained on a more diverse and informative set of instances, making it less likely to overfit. With a more concise and representative training set, the classifier is less likely to be overwhelmed by irrelevant or redundant data. This can lead to improved performance, particularly in terms of classification accuracy and robustness. Fig. 4 presents the pseudocode of the proposed IS-HICCS algorithm

Algorithm # 2 IS-HICCS – Instance Selection based on HICCS
Inputs: MI: matrix of instances k: number of instances to select d: inter instance similarity function D: inter groups dissimilarity function
Output: E: selected instances set
1. C = HICCS(MI, k, d, D) 2. E= ϕ 3. For each cluster *c* in C 3.1. Select as cluster center (holotype or prototype) the object that minimizes the overall dissimilarity with respect to every object in the cluster 3.2. Add the cluster center to E 4. Return E

Fig. 4. Overview of the IS-HICCS algorithm. The process begins by computing the desired number of clusters by HICCS. Then, for each cluster, the representative instance is chosen and added to the selected instances set.

It is important to mention that, to increase computational efficiency, it is possible to obtain the matrix similarity of instances, and pass it as a parameter for both HICCS and IS-HICCS algorithms, to avoid duplication of between instances similarity computations. This process can also be performed in parallel, further diminishing computational complexity.

4 Experimental configurations

This section presents the datasets, algorithms, performance measures, and statistical methods used to compare the proposed algorithms with respect to the state-of-the-art. All experiments were run on a personal laptop with Windows 11 Pro, 16GB of RAM, and an Intel Core i7 8th generation processor. As the laptop was not fully dedicated to running the experiments, they were completed with below-normal priority.

4.1 Datasets

For the experimental comparison, we utilize both synthetic and real-world data. We test six 2-D synthetic datasets with varying complexities and shapes (Fig. 5), as well as 10 real-life datasets from the Machine Learning Repository at the University of California, Irvine (UCI) (Kelly, Longjohn, & Nottingham, 2024) (Table 1). All real-world datasets contain hybrid and incomplete data. The datasets vary in size, with attributes ranging from five to 34 and the number of classes spanning from two to seven.

Among synthetic data, the simplest is the Balls database, which has three well-separated spherical clusters (Fig 5a). The Unbalanced dataset has again three well-separated clusters, but with an unbalanced distribution of objects (Fig. 5b). The Banana database has two clusters with banana shapes (Fig. 5c), and the Overlapped database has three spherical clusters, but highly overlapped (Fig. 5d). We also used two complex databases, the Flowers and the Basketball datasets. The first consists of two clusters forming flowers (Fig. 5e), and the latter of three clusters, corresponding to a man playing basketball (Fig. 5f).

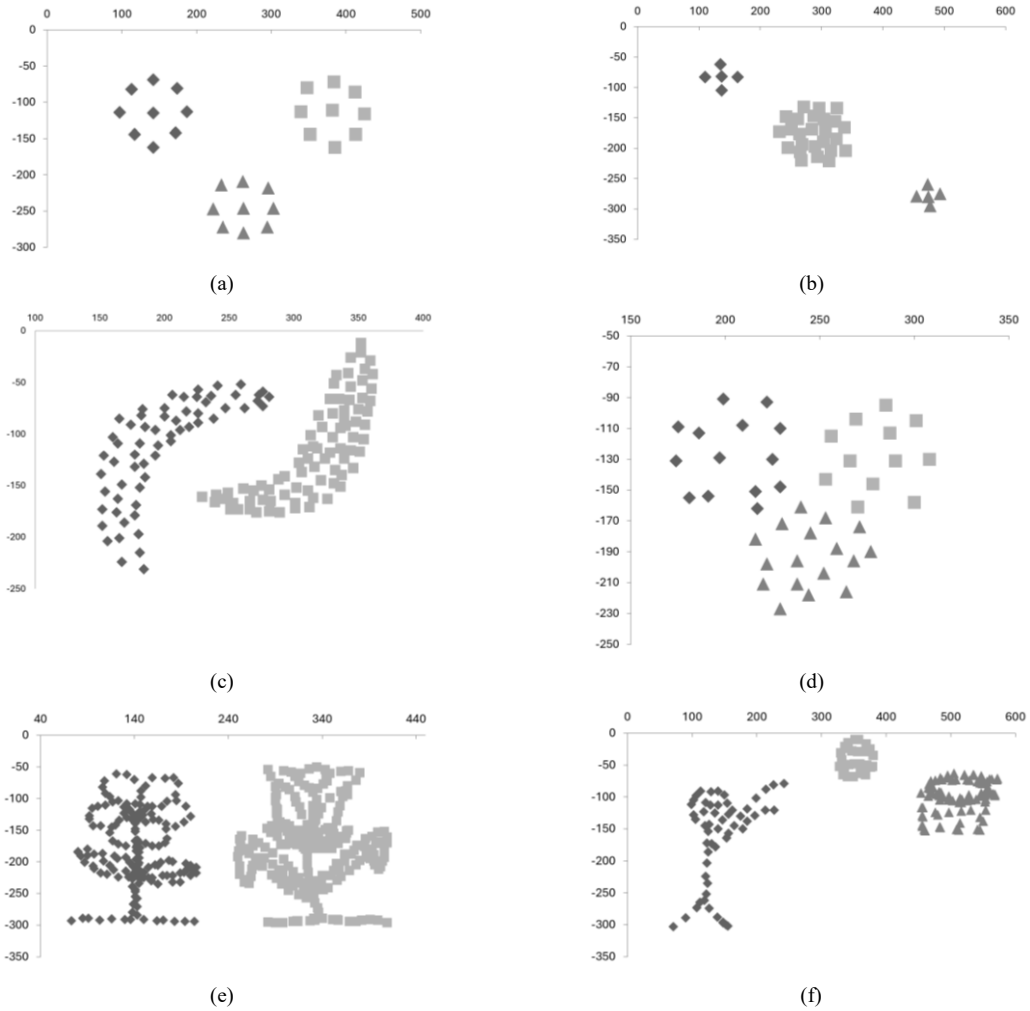

Fig. 4. Synthetic datasets: Ball (a), Unbalanced (b), Banana (c), Overlapped (d), Flowers (e), and Basketball (f).

Table 1. Description of the UCI Machine Learning repository datasets

Datasets	Attributes	Instances	Classes	Hybrid	Missing Values
autos	25	205	6	Yes	Yes
colic	27	368	2	Yes	Yes
credit-a	14	690	2	Yes	Yes
dermatology	34	366	6	Yes	Yes
heart-c	13	303	5	Yes	Yes
hepatitis	19	155	2	Yes	Yes
labor	16	57	2	Yes	No
lymph	19	148	4	Yes	No
tae	5	151	3	Yes	No
zoo	16	101	7	Yes	No

4.2 Algorithms, Performance Measures, and Statistical Methods

For HICCS comparisons, we consider four existing clustering algorithms that represent different approaches to data clustering, as discussed earlier. These include a k-means-based clustering method for hybrid data (AD) (Ahmad & Dey, 2007), HIMIC, a hierarchical agglomerative clustering algorithm (Ahmed et al., 2005), a genetic-based clustering approach (AGKA) (Roy & Sharma, 2010), and an ensemble-based clustering method (CEBMDC) (Zengyou He et al., 2005).

As performance measures, we used two cluster validity indexes: Cluster Error and Entropy. Cluster validation is a critical step in the clustering process. Due to the unsupervised nature of clustering, where multiple solutions can appear plausible, external cluster validity indexes are commonly used to compare different clustering algorithms (Brun et al., 2007). These external measures assess the degree of similarity between the clustering results and the true class labels. Entropy (E) and Cluster Error (CE) are among the most widely used external validity indexes (Brun et al., 2007). Lower values of Entropy and Cluster Error indicate better algorithm performance. Entropy measures the degree of dispersion of classes within the clusters, and Cluster Error counts the number of instances that do not belong to the majority class within each cluster.

Let be C the resulted clustering, C_i is the i-th cluster in C, and n_{ij} the number of instances of the j-th class in the i-th cluster. The Entropy of C with respect to class labels is given by:

$$E(C) = -\sum_i \frac{|C_i|}{|O|} * \sum_j \frac{n_{ij}}{|C_i|} \log\left(\frac{n_{ij}}{|C_i|}\right) \quad (1)$$

The Cluster Error of C with respect to class labels is given by:

$$CE(C) = \sum_i \frac{|C_i| - n_i}{|C_i|} \quad (2)$$

We use as HICCS similarity function the reciprocal of the HEOM dissimilarity proposed by Wilson and Martinez (Wilson & Martinez, 1997). For the AD, we keep the parameters suggested by its authors (S=5, γ=20) and for the AGKA we used population size 100, generations 100, mutation probability 0.05 and crossover probability of 0.9.

Regarding IS-HICCS, we selected four existing instance selection algorithms, including Reduced Nearest Neighbor (RNN) (Gates, 1972), Minimal Consistent Set (MCS) (Dasarathy, 1994), Prototype Selection by Relevance (PSR) (Olvera-López, Carrasco-Ochoa, & Martínez-Trinidad, 2008), and Mutiedit (Devijver, 1980). Such algorithms are from the error-based editing and condensing paradigms for instance selection. For performance measures, we consider the Nearest Neighbor classifier error and the instance retention ratio. We used five-fold cross-validation to compute the supervised performance measures. We used HEOM dissimilarity (Wilson & Martinez, 1997) again for instance selection algorithms.

Let E be the selected set of instances from a training set T, NN_E be the Nearest Neighbor classifier trained by the set E, and P be the testing set, with $p \in P$ and $class(p)$ the true class of p. The classifier error is given by:

$$Error = \frac{|\{p \in P|\ class(p) \neq NN_E(p)\}|}{|P|} \quad (3)$$

The instance retention ratio is given by:

$$Retention = \frac{|E|}{|T|} \quad (4)$$

We used the Friedman test for all statistical comparisons, followed by Holm's post hoc test. This combination of tests was suggested by (Garcia & Herrera, 2008). We set a significant value of 0.05, for 95% confidence.

5 Results and Discussion

This section comprises two main subsections. Subsection 5.1 presents the results of the proposed HICCS clustering and its comparison with state-of-the-art clustering algorithms. Then, subsection 5.2 compares IS-HICCS with respect to other instance selection algorithms for hybrid and incomplete data.

5.1 Results of HICCS algorithm

We compare the performance of the methods according to Entropy (Table 2) and Cluster Error (Table 3). The best (lower) results for each dataset are highlighted in bold. In the database Ball, our HICCS method, as well as the HIMIC and AD methods, achieve perfect clustering. This database has three compact, well-separated, and balanced spherical clusters. On eight real-life datasets, our method always performed best according to entropy. In dataset credit-a, it was outperformed by CEBMDC, and in lymph by HIMIC.

In dataset Unbalanced, the best methods were HICCS and HIMIC, and the second best was AD. However, AD was unable to find the clusters. In dataset Banana, the best method was HICCS, and the second best CEBMDC. However, none of them achieve a perfect clustering. That's due to the elongated nature of the dataset, in which some points in a cluster are very close to the other clusters. Our HICCS method only fails in assigning eight points out of a total of 141 points. Similarly, in the Overlapping dataset, no method achieves perfect clustering. Again, the best was HICCS, and the second best was the HIMIC method. In this dataset, our HICCS method only fails in three points of 43 possible.

Table 2. Results of the compared clustering algorithms according to Entropy

Datasets	AD	AGKA	CEBMDC	HIMIC	HICCS
Balls	**0.0000**	**1.3480**	**1.3878**	**0.0000**	**0.0000**
Unbalanced	**0.5981**	**1.1699**	**0.8541**	**0.0000**	**0.0000**
Banana	**0.8048**	**0.9919**	**0.5589**	**0.5788**	**0.2743**
Overlapping	**0.5739**	**1.5152**	**0.7034**	**0.4795**	**0.2836**
Flowers	**0.8495**	**0.9562**	**0.6419**	**0.0000**	**0.0000**
Basketball	**0.2780**	**1.3971**	**0.5336**	**0.0000**	**0.0000**
Autos	2.2725	2.1314	2.1742	2.0286	**1.8877**
Colic	0.9503	0.9525	0.9490	0.9498	**0.9413**
credit-a	0.9912	0.9927	**0.7316**	1.0146	0.9900
dermatology	2.4326	2.3793	2.0797	2.1326	**0.9157**
heart-c	0.9943	0.9956	0.9942	0.9914	**0.9885**
hepatitis	0.7346	0.6663	0.7317	0.7334	**0.6102**
Labor	0.9348	0.9311	0.9269	0.9208	**0.8891**
Lymph	1.2277	1.0914	1.2033	**0.8553**	1.0519
Tae	1.5845	1.5593	1.5225	1.5325	**1.4712**
Zoo	2.3906	1.9988	0.5228	0.5228	**0.3116**

The datasets Flowers and Basketball, although they have a non-overlapping cluster, have arbitrary shapes. The first consists of two flowers (a violet and a tulip, respectively), and the latter is formed by a man playing basketball (three clusters consisting of a

man, a ball, and a basket). Our HICCS method was able to detect the actual structure of data ideally, and in both cases, so was the HIMIC method. The second-best method was the CEBMDC method on the Flowers dataset and the AD on the Basketball database. Results shown on six artificial and ten real-life data sets show that HICCS performs the best in detecting the appropriate partitioning in most of the cases. It can also be seen from the above results that the proposed HICCS can find out the proper clustering other methods fail. The results on Unbalanced, Flowers, and Basketball show that HICCS can detect clusters irrespective of their densities or shapes. The superiority of HICCS was also established on the ten real-life data sets. The results on 16 synthetic and real-life data sets show that HICCS is well-suited to detect clusters of widely varying characteristics in hybrid and incomplete datasets.

Table 3. Results of the compared clustering algorithms according to Cluster Error

Datasets	AD	AGKA	CEBMDC	HIMIC	HICCS
Balls	**0.0000**	0.5200	0.5556	**0.0000**	**0.0000**
Unbalanced	0.2857	0.3429	0.2857	**0.0000**	**0.0000**
Banana	0.2553	0.4649	0.1418	0.2057	**0.0567**
Overlapping	0.1395	0.5882	0.1860	0.1163	**0.0698**
Flowers	0.2965	0.3814	0.1683	**0.0000**	**0.0000**
Basketball	0.0536	0.4604	0.1607	**0.0000**	**0.0000**
Autos	0.6732	0.6651	0.6732	0.6244	**0.5659**
Colic	**0.3696**	0.3725	**0.3696**	**0.3696**	**0.3696**
credit-a	0.4449	0.4499	**0.2072**	0.4503	0.4420
dermatology	0.6940	0.6911	0.5738	0.6585	**0.3169**
heart-c	**0.4554**	0.4615	**0.4554**	**0.4554**	**0.4554**
hepatitis	0.2065	0.1803	0.2065	0.2065	**0.1613**
Labor	**0.3509**	0.3881	**0.3509**	**0.3509**	**0.3509**
Lymph	0.4527	0.3933	0.4527	**0.2365**	0.4122
Tae	0.6556	0.5959	0.5894	0.5960	**0.5298**
Zoo	0.5941	0.5842	0.5941	0.1386	**0.0891**

Results show that HICCS can detect clusters that are well-separated or hyperspherically shaped. It fails for overlapping-shaped clusters, but only in a few points. Results also show that while AD is only able to detect appropriate partitioning from data sets having hyperspherical-shaped clusters, HIMIC agglomerative clustering for hybrid data can do so for well-separated clusters. Thus, AD performs well for data sets like Balls and Basketball (has two hyperspherical clusters, the ball and the basket, and also the head of the man) but fails for data sets having clusters of non-hyper spherical shapes (e.g. Banana, Flowers), and also in clusters having different densities or highly overlapped (e.g. Unbalanced and Overlapped).

Table 4. Friedman's rankings for Entropy and Cluster Error

Algorithms	Entropy ranking	Cluster Error ranking
HICCS	1.2812	1.6250
HIMIC	2.4375	2.5000
CEBMDC	3.0312	3.2188
AD	4.0625	3.5938
AGKA	4.1875	4.0625

The statistical analysis according to Friedman and Holm's tests for Entropy and Cluster Error are presented in Tables 4 and 5. For both measures, the Friedman test rejects the null hypothesis, with a p-value of 0.0. Holm's tests also reject all hypotheses, showing that the proposed HICCs outperformed all compared algorithms.

Table 5. Holms's post hoc results for Entropy and Cluster Error

Measure	i	Algorithm	z	P	Holm
Entropy	4	AGKA	5.198858	0.000000	0.012500
	3	AD	4.975251	0.000001	0.016667

Cluster Error	2	CEBMDC	3.130495	0.001745	0.025000
	1	HIMIC	2.068363	0.038606	0.050000
	4	AGKA	4.360333	0.000013	0.012500
	3	AD	3.521807	0.000429	0.016667
	2	CEBMDC	2.850987	0.004358	0.025000
	1	HIMIC	1.565248	0.117525	0.050000

In addition, because several intergroup dissimilarities can be used, we compare the HICCS performance using single-linkage (Min), average-linkage (Mean), complete-linkage (Max), and holotype-linkage (Holo) as intergroup dissimilarities. We show the results in Table 6. In such table, Avg, Holo, Max, and Min indicate, respectively, HICCS with average-linkage dissimilarity, with holotype-linkage dissimilarity, with complete-linkage dissimilarity, and with single-linkage dissimilarity. For each datasets the best results are highlighted in bold. The Friedman test comparing the different linkages obtained a p-value of 0.594749 for the Entropy measure and a p-value of 0.406307 for Cluster Error, therefore not rejecting the null hypothesis in either case. That is, we did not find significant differences in performance while using different linkage strategies.

Table 6. Results of the compared clustering algorithms according to Cluster Error

Datasets	Entropy				Cluster Error			
	Avg	Holo	Max	Min	Avg	Holo	Max	Min
Balls	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Unbalanced	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Banana	**0.0000**	0.2743	0.5597	0.4653	**0.0000**	0.0567	0.1631	0.0993
Overlapping	0.6464	0.2836	**0.1183**	0.8782	0.1628	0.0698	**0.0233**	0.3256
Flowers	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Basketball	**0.0000**	**0.0000**	0.1344	0.0370	**0.0000**	**0.0000**	0.0357	0.0060
autos	2.0447	**1.9204**	2.0330	1.9718	0.6049	**0.5805**	0.5951	0.5951
colic	0.9431	0.9467	0.9502	**0.9357**	**0.3696**	**0.3696**	**0.3696**	**0.3696**
credit-a	0.9900	0.9878	**0.9646**	0.9907	**0.4420**	**0.4420**	0.4449	0.4435
dermatology	1.0380	1.0380	**0.7581**	1.2825	0.3443	0.3443	**0.2814**	0.4016
heart-c	**0.7349**	0.7445	0.9073	0.9184	**0.2079**	0.2145	0.3366	0.3432
hepatitis	0.7281	0.5968	**0.5706**	0.6530	**0.2065**	**0.2065**	**0.2065**	**0.2065**
labor	0.7243	0.9125	**0.7076**	0.9113	0.2105	0.3509	**0.1930**	0.3509
lymph	1.0734	1.1029	**0.9815**	0.9925	**0.4189**	**0.4189**	0.4392	0.4257
tae	3.4594	3.4594	**3.3491**	3.4336	0.9091	0.9091	**0.8848**	0.8909
zoo	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**

5.2 Results of IS-HICCS algorithm

Regarding IS-HICCS, we selected four existing instance selection algorithms, including Reduced Nearest Neighbor (RNN) (Gates, 1972), Minimal Consistent Set (MCS) (Dasarathy, 1994), Prototype Selection by Relevance (PSR) (Olvera-López et al., 2008), and Mutiedit (Devijver, 1980). Such algorithms are from the error-based editing and condensing paradigms for instance selection. In addition, they are all available in the EPIC software (Hernández-Castaño, Villuendas-Rey, Camacho-Nieto, & Yáñez-Márquez, 2018; Hernández-Castaño, Villuendas-Rey, Nieto, & Rey-Benguría, 2018), which facilitates numerical comparisons. Table 7 presents the results according to the classifier error measure, while Table 8 shows the instance retention obtained by the compared algorithms.

For instance, selection, we only considered the ten real-world datasets, because we wanted to see the applicability of the clustering procedure for selecting instances over real-world data. We set the maximum number of clusters (instances to select) to 50 and use holotype-linkage for HICCS algorithms. The best results for each dataset are in bold.

Table 7. Classifier error of the compared instance selection algorithms for five-fold cross-validation.

Datasets	MCS	Multiedit	PSR	RNN	IS-HICCS
autos	0.2979	0.7024	0.3312	**0.2976**	0.4000
colic	0.2338	0.2282	**0.2176**	0.2499	0.2179
credit-a	0.2478	**0.1710**	0.2275	0.2188	0.2130
dermatology	0.1146	0.2185	0.0874	0.1336	**0.0983**
heart-c	0.2541	**0.1724**	0.2313	0.2480	0.2151
hepatitis	0.2392	0.2079	0.2325	0.2321	**0.1800**
labor	**0.1033**	0.4800	0.2067	0.2067	0.2300
lymph	0.2229	0.2424	0.2462	0.2433	**0.2262**
tae	**0.3771**	0.6029	0.5692	0.3975	0.4367
zoo	0.0500	0.2664	0.0491	**0.0300**	0.0991

Table 8. Instance retention of the compared instance selection algorithms for five-fold cross-validation.

Datasets	MCS	Multiedit	PSR	RNN	IS-HICCS
autos	0.4347	0.1371	0.4856	0.4417	**0.2710**
colic	0.2801	0.5146	0.3702	0.3863	**0.0296**
credit-a	0.3108	0.6093	0.3535	0.3119	**0.0805**
dermatology	**0.1402**	0.6503	0.5395	0.1715	0.1518
heart-c	0.3480	0.6270	0.3487	0.3528	**0.1834**
hepatitis	0.2961	0.6731	0.3606	0.3563	**0.2007**
labor	**0.1676**	0.3294	0.3392	0.2552	0.1735
lymph	0.3604	0.4054	0.4369	0.3439	**0.1051**
tae	0.5011	**0.1236**	0.3642	0.6041	0.3252
zoo	**0.1177**	0.7348	0.4873	0.1287	0.1232

The statistical analysis showed no significant differences in the performance of IS-HICCS regarding classifier error, with Friedman's p-value of 0.79797. However, IS-HICCS was the first algorithm in Friedman's ranking (Table 9). Regarding instance retention, the Friedman test obtained a p-value of 0.000334, thus rejecting the null hypothesis. Holm's post hoc test (Table 10) did not reject the hypothesis comparing IS-HICCS and MCS according to instance retention and rejected all other hypotheses. It is important to note that, although we set the maximum number of clusters to 50, due to the merging strategy of HICCS, in several datasets, fewer clusters were obtained, thus increasing data reduction.

Table 9. Friedman's rankings for Classifier error and Instance retention

Algorithms	Classifier error ranking	Instance retention ranking
IS-HICCS	2.6000	1.5000
PRS	2.9500	4.0000
RNN	2.9500	3.4000
MCS	3.0000	2.1000
Multiedit	3.5000	4.0000

Table 10. Holms's post hoc results for Instance retention

i	Algorithm	z	p	Holm
4	Multiedit	3.535534	0.000407	0.012500
3	PRS	3.535534	0.000407	0.016667
2	RNN	2.687006	0.007210	0.025000
1	MCS	0.848528	0.396144	0.050000

The statistical analysis shows that the proposed algorithm is able to achieve high instance retention without sacrificing classifier performance. However, while clustering-based instance selection can offer significant benefits, it is not without its challenges. The effectiveness of instance selection through clustering depends on the quality of the clusters formed. If the clustering algorithm fails to partition the data correctly, the selected instances may not adequately represent the true structure of the data, leading to poor model performance. In addition, in some cases, outliers or noise can still be present in the selected instances, even after

clustering. Despite these challenges, our experiments show that clustering remains a valuable tool for enhancing the performance of supervised classification models.

6 Conclusions

Handling hybrid and incomplete data presents a significant challenge in clustering tasks. The HICCS algorithm introduced in this paper brings several innovative features. It begins by using compact sets for initial clustering rather than treating individual instances separately. It also eliminates the need for additional similarity computations by employing holotypes to assess intergroup dissimilarity. Moreover, it merges selected clusters at each stage based on their similarity, ensuring the process is independent of the order of operations. The algorithm allows for a user-defined similarity function, making it adaptable to various real-world applications. Selecting cluster representatives (holotypes) instead of constructing fictional cluster centers guarantees that each cluster is represented by an actual instance.

We evaluated the effectiveness of this algorithm on six synthetic and ten real-world datasets. In our experiments, we compared HICCS to other clustering methods from different approaches and analyzed the statistical significance of the performance. Our method outperformed others regarding Entropy and Cluster Error measures, demonstrating its ability to handle hybrid and incomplete datasets while producing high-quality partitions.

Additionally, we introduced a new instance selection method based on HICCS, called the IS-HICCS algorithm. We assessed its ability to reduce the instance set without compromising classifier accuracy. Statistical analysis confirms that clustering remains a valuable tool for improving the performance of supervised classification models.

In future works, we plan to test our approach on larger datasets, both in terms of instance count and attribute size. We also aim to evaluate the performance of HICCS and IS-HICCS in the presence of noisy data.

References

Ahmad, A., & Dey, L. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering, 63*(2), 503-527.

Ahmed, R., Borah, B., Bhattacharyya, D., & Kalita, J. (2005). HIMIC: A Hierarchical Mixed Type Data Clustering Algorithm. *Department of Computer Science and Information Technology*.

Askari, S. (2021). Fuzzy C-Means clustering algorithm for data with unequal cluster sizes and contaminated with noise and outliers: Review and development. *Expert Systems with Applications, 165*, 113856.

Balcan, M.-F., Blum, A., & Vempala, S. (2008). *Clustering via similarity functions: theoretical foundations and algorithms.* Paper presented at the Proceedings of the 40 th ACM Symposium on Theory of Computing (STOC).

Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., & Dougherty, E. R. (2007). Model-based evaluation of clustering validation measures. *Pattern Recognition, 40*(3), 807-824.

Camacho-Nieto, O., Yáñez-Márquez, C., & Villuendas-Rey, Y. (2020). Undersampling Instance Selection for Hybrid and Incomplete Imbalanced Data. *Journal of Universal Computer Science*(6), 698-720.

Cohen, G., Hilario, M., Sax, H., Hugonnet, S., & Geissbuhler, A. (2006). Learning from imbalanced data in surveillance of nosocomial infection. *Artificial intelligence in medicine, 37*(1), 7-18.

Dasarathy, B. V. (1994). Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Transactions on Systems, Man, and Cybernetics, 24*(3), 511-517.

Devijver, P. A. (1980). *On the edited nearest neighbor rule.* Paper presented at the Proc. 5th Int. Conf. Pattern Recognition, 1980.

Fritz, M., Behringer, M., Tschechlov, D., & Schwarz, H. (2022). Efficient exploratory clustering analyses in large-scale exploration processes. *The VLDB Journal, 31*(4), 711-732.

García-Serrano, J. R., & Martínez-Trinidad, J. F. (1999). *Extension to c-means algorithm for the use of similarity functions.* Paper presented at the European Conference on Principles of Data Mining and Knowledge Discovery.

Garcia, S., & Herrera, F. (2008). An Extension on" Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *Journal of machine learning research, 9*(12).

Gates, G. (1972). The reduced nearest neighbor rule (corresp.). *IEEE transactions on information theory, 18*(3), 431-433.

George, L., & Sumathy, P. (2023). An integrated clustering and BERT framework for improved topic modeling. *International Journal of Information Technology, 15*(4), 2187-2195.

Gunay, H. B., & Shi, Z. (2020). Cluster analysis-based anomaly detection in building automation systems. *Energy and Buildings, 228*, 110445.

Hamidzadeh, J., Monsefi, R., & Yazdi, H. S. (2015). IRAHC: instance reduction algorithm using hyperrectangle clustering. *Pattern Recognition, 48*(5), 1878-1889.

He, Z., Shen, X., Zhou, Y., & Wang, Y. (2024). *Application of K-means clustering based on artificial intelligence in gene statistics of biological information engineering.* Paper presented at the Proceedings of the 2024 4th International Conference on Bioinformatics and Intelligent Computing.

He, Z., Xu, X., & Deng, S. (2002). Squeezer: an efficient algorithm for clustering categorical data. *Journal of Computer Science and Technology, 17*(5), 611-624.

He, Z., Xu, X., & Deng, S. (2005). Clustering mixed numeric and categorical data: A cluster ensemble approach. *arXiv preprint cs/0509011*.

Hernández-Castaño, J. A., Villuendas-Rey, Y., Camacho-Nieto, O., & Yáñez-Márquez, C. (2018). Experimental platform for intelligent computing (EPIC). *Computación y Sistemas, 22*(1), 245-253.

Hernández-Castaño, J. A., Villuendas-Rey, Y., Nieto, O. C., & Rey-Benguría, C. F. (2018). A New Experimentation Module for the EPIC Software. *Res. Comput. Sci., 147*(12), 243-252.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery, 2*(3), 283-304.

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences, 622*, 178-210.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*: Prentice-Hall, Inc.

Jain, D., Borah, M. D., & Biswas, A. (2024). A sentence is known by the company it keeps: Improving Legal Document Summarization Using Deep Clustering. *Artificial Intelligence and Law, 32*(1), 165-200.

Jain, M., Kaur, G., & Saxena, V. (2022). A K-Means clustering and SVM based hybrid concept drift detection technique for network anomaly detection. *Expert Systems with Applications, 193*, 116510.

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

Lin, C., Han, G., Qi, X., Du, J., Xu, T., & Martínez-García, M. (2021). Energy-optimal data collection for unmanned aerial vehicle-aided industrial wireless sensor network-based agricultural monitoring system: A clustering compressed sampling approach. *IEEE Transactions on Industrial Informatics, 17*(6), 4411-4420.

Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2*(1), 86-97.

Olvera-López, J. A., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2008). *Prototype selection via prototype relevance.* Paper presented at the Progress in Pattern Recognition, Image Analysis and Applications: 13th Iberoamerican Congress on Pattern Recognition, CIARP 2008, Havana, Cuba, September 9-12, 2008. Proceedings 13.

Paek, J., & Ko, J. (2015). $ K $-Means clustering-based data compression scheme for wireless imaging sensor networks. *IEEE Systems Journal, 11*(4), 2652-2662.

Pradana, M. G., & Ha, H. T. (2021). Maximizing strategy improvement in mall customer segmentation using k-means clustering. *Journal of Applied Data Sciences, 2*(1), 19-25.

Reyes-González, R., & Ruiz-Shulcloper, J. (1999). *Un algoritmo de estructuración restringida de espacios.* Paper presented at the SIARP 1999, La Habana, Cuba.

Roy, D. K., & Sharma, L. K. (2010). Genetic k-means clustering algorithm for mixed numeric and categorical data sets. *International Journal of Artificial Intelligence & Applications, 1*(2), 23-28.

Saha, S., Sarker, P. S., Al Saud, A., Shatabda, S., & Newton, M. H. (2022). Cluster-oriented instance selection for classification problems. *Information Sciences, 602*, 143-158.

Sarkar, M., Puja, A. R., & Chowdhury, F. R. (2024). Optimizing Marketing Strategies with RFM Method and K-Means Clustering-Based AI Customer Segmentation Analysis. *Journal of Business and Management Studies, 6*(2), 54-60.

Trinidad, J. F. M. n., Shulcloper, J. R., & Cortés, M. S. L. (2000). Structuralization of universes. *Fuzzy sets and systems, 112*(3), 485-500.

Tsai, C.-F., Lin, W.-C., Hu, Y.-H., & Yao, G.-T. (2019). Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Information Sciences, 477*, 47-54.

Tusell-Rey, C. C., Camacho-Nieto, O., Yáñez-Márquez, C., & Villuendas-Rey, Y. (2022). Customized instance random undersampling to increase knowledge management for multiclass imbalanced data classification. *Sustainability, 14*(21), 14398.

Villuendas-Rey, Y. (2022). Hybrid data selection with preservation rough sets. *Soft Computing, 26*(21), 11197-11223.

Villuendas-Rey, Y., Barroso-Cubas, E., Camacho-Nieto, O., & Yáñez-Márquez, C. (2021). A general framework for mixed and incomplete data clustering based on swarm intelligence algorithms. *Mathematics, 9*(7), 786.

Wilson, D. R., & Martinez, T. R. (1997). Improved heterogeneous distance functions. *Journal of artificial intelligence research, 6*, 1-34.

Xu, C., & Zhang, S. (2024). A Genetic Algorithm-based sequential instance selection framework for ensemble learning. *Expert Systems with Applications, 236*, 121269.

Zhu, X., Nie, S., Wang, C., Xi, X., Wang, J., Li, D., & Zhou, H. (2020). A noise removal algorithm based on OPTICS for photon-counting LiDAR data. *IEEE Geoscience and Remote Sensing Letters, 18*(8), 1471-1475.