Acta Informatica Pragensia
2026, Volume 15, Issue 3, Forthcoming article
Special Issue: Selected papers from the 14th International Conference on Information Technology in Asia
https://doi.org/10.18267/j.aip.310

PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

Article Open Access

Similarity Ranking-Based Instance Selection for Enhancing *k*-NN Classification Performances

Abdul Muqtasid bin Rushdi [1], Mohammad bin Hossin [1], Suhaila binti Saee [1], Norita binti Md Norwawi [2]

[1] Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia
[2] Department of Information Security and Assurance, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Nilai, Negeri Sembilan, Malaysia

Corresponding author: Mohammad bin Hossin (hmohamma@unimas.my)

Editorial Record

First submission received:
October 23, 2025

Revisions received:
January 13, 2026
January 18, 2026
March 6, 2026

Accepted for publication:
March 6, 2026

Special Issue Editors:
Stephanie Chua Hui Li
Universiti Malaysia Sarawak, Malaysia

Phei Chin Lim
Universiti Malaysia Sarawak, Malaysia

Siaw-Hong Liew
Universiti Malaysia Sarawak, Malaysia

Kim-Mey Chew
Universiti Malaysia Sarawak, Malaysia

Academic Editor:
Zdenek Smutny
Prague University of Economics and Business, Czech Republic

This article was accepted for publication by the Academic Editor upon evaluation of the reviewers' comments.

How to cite this article:
Rushdi, A. M. b., Hossin, M. b., Saee, S. b., & Norwawi, N. b. M. (2026). Similarity Ranking-Based Instance Selection for Enhancing k-NN Classification Performances. *Acta Informatica Pragensia*, 15(3), Forthcoming article. https://doi.org/10.18267/j.aip.310

Abstract

Background: The k-nearest neighbours (*k*-NN) is a well-established classifier in machine learning. Yet, its performance drops and computational costs rise with extensive or redundant datasets. Furthermore, current instance selection (IS) approaches often face scalability problems and are sensitive to parameter settings.

Objective: This study seeks to design a straightforward and efficient IS algorithm that reduces both dataset size and computational demands, yet preserves or enhances the accuracy of *k*-NN classification.

Methods: We propose Euclidean ranking-based instance selection (ERbIS), a novel IS approach that prioritises samples based on their Euclidean distance from a single anchor point. In this study, two anchor points are introduced: the first data anchor point (FD) and the mean of each column anchor point (MEC). Both ERbIS models (ERbIS-FD and ERbIS-MEC) are evaluated across 21 KEEL datasets. For performance comparison, the ERbIS models are benchmarked against current and state-of-the-art methods, including condensed nearest neighbour rule (CNN), edited nearest neighbour rule (ENN), adaptive threshold-based instance selection algorithm (ATISA1), decremental reduction optimization procedure (DROP3) and ranking-based instance selection (RIS1). The evaluation focuses on reduction speed, reduction rate and *k*-NN classification accuracy.

Results: The ERbIS models reduce dataset size by an average of 35 to 40% without compromising accuracy compared to the original *k*-NN and state-of-the-art IS models. Both ERbIS models also demonstrate superior computational efficiency in the reduction process relative to ENN and CNN. Notably, the ERbIS-MEC variant, which utilises the mean of each column as the anchor point, achieves the highest generalisation accuracy among all current and state-of-the-art models.

Conclusion: ERbIS offers an efficient and scalable approach for instance selection in *k*-NN classification, achieving significant data reduction and enhanced predictive accuracy with minimal parameter tuning. The model demonstrates strong potential for application to large datasets and may be further improved by investigating alternative distance metrics or integrating hybrid instance selection strategies.

Index Terms

Instance selection; k-nearest neighbours; Data reduction; Euclidean distance; Data classification.

1 INTRODUCTION

Recognised for its simplicity, the *k*-nearest neighbours (*k*-NN) algorithm remains a fundamental tool in machine learning classification. According to Boateng et al. (2020), *k*-NN works by assigning each new data point to the most common class among its nearest neighbours. Its non-parametric nature eliminates the need for a formal training stage, making it both flexible and easy to use.

Despite its straightforward design, *k*-NN has achieved strong results in fields such as visual recognition, textual analysis and medical diagnostics. The performance of *k*-NN depends heavily on both the quality and size of the data samples. Syriopoulus et al. (2023) and Zhang (2021) have highlighted that as datasets become larger, *k*-NN often encounters problems, including higher computational demands and increased vulnerability to noise and redundant information. Furthermore, as noted by Zhang (2021), the time required for making predictions increases in proportion to the number of data samples, which can pose significant challenges when working with extensive datasets. Xing and Bei (2020) reported that the presence of irrelevant or mislabelled data can substantially reduce classification accuracy, as *k*-NN does not differentiate between data points during the neighbour selection process.

To address these drawbacks, researchers have proposed instance selection (IS) and data reduction (DR) approaches. For example, Eleftheriadis et al. (2024) emphasized that the main objective is to identify and retain a representative subset of data, thereby preserving the necessary structure for accurate classification. By excluding data points that are redundant, noisy or irrelevant, these methods can accelerate the prediction process and potentially enhance the classification accuracy of *k*-NN. Previous studies, such as those by Cavalcanti and Soares (2020) and An et al. (2021), have indicated that these refined subsets can achieve similar or even improved accuracy compared to the full dataset. Moreover, reducing the data volume offers considerable gains in computational efficiency, making it feasible for *k*-NN to process larger datasets.

Although advances in DR methods have benefited *k*-NN classification, further development is necessary, particularly in terms of scalability and adaptability. As observed by Kordos et al. (2021), expanding datasets in both size and complexity presents challenges for many IS algorithms in managing increased data volume. The computational costs involved in the data reduction stage can at times undermine improvements achieved during classification, which is a well-recognised limitation among several IS algorithms.

Another notable concern is the potential for IS algorithms to eliminate infrequent but essential cases. Shi (2020) pointed out that this can erode the representation of minority classes and weaken the overall generalisation ability of the classifier. Furthermore, Blachnik and Kordos (2020) and Li et al. (2020) have noted that these algorithms frequently suffer from unstable performance and heightened sensitivity to parameter adjustments. The effectiveness of hybrid and ensemble methods often depends on parameter selection, which can differ greatly depending on data characteristics. Malhat et al. (2020) highlighted that such factors complicate implementation and can cause inconsistent outcomes across datasets. Consequently, there is a clear demand for more adaptive, scalable and context-sensitive IS algorithms.

This work introduces an innovative IS method that extends the use of Euclidean distance and ranking techniques. The aim is to decrease storage needs while improving *k*-NN classification. Our approach identifies a single anchor point and measures the Euclidean distance of all dataset instances from this anchor. Instances are then ranked based on their closeness. We call this method the Euclidean ranking-based instance selection (ERbIS).

This manuscript is an extended version of our previous conference paper, which introduced the ERbIS algorithm with the mean of each column (MEC) anchor point (Hossin & Rushdi, 2025). In the earlier work, we presented the basic approach and evaluated its performance on multiple datasets. In this extended study, we propose a new variant of the ERbIS algorithm by introducing an additional anchor point, referred to as the first data (FD) anchor-point. Both ERbIS models operate by calculating the distance between the selected anchor point and each data instance. The calculated distances are ranked to form an ordered list, L. For each class segment in this list, we select the first and last instances as the most informative prototypes and discard the remainder as redundant. This strategy is designed to address computational challenges related to instance selection and to reduce storage requirements, while aiming to preserve or improve the classification accuracy of *k*-NN.

In our previous paper (Hossin & Rushdi, 2025), we evaluated the ERbIS-MEC model using twenty-one datasets from the KEEL repository. We compared its performance with two leading models and three current instance selection algorithms, focusing on accuracy and reduction rate. In this paper, we present an extended analysis by evaluating both ERbIS models (ERbIS-MEC and ERbIS-FD) with additional comparison criteria, including reduction speed, reduction rate and classification accuracy. The results show that both ERbIS models achieve substantial reductions in training set size and demonstrate fast reduction speed. They consistently maintain or enhance the classification accuracy of *k*-NN. Overall, ERbIS demonstrates superior performance compared to state-of-the-art IS models and remains competitive with more recent IS models.

The rest of this paper is organised as follows. The subsequent section examines the fundamental aspects of *k*-NN and DR, with particular attention to IS and similarity distance metrics. Section 3 details the proposed methodology. In Section 4, the experimental framework is presented alongside a comprehensive analysis of the results and a discussion of their implications. The final section offers a summary of the key contributions and outlines future research directions.

2 RELATED WORKS

This section provides an overview of the *k*-NN algorithm and the principles of data reduction. The discussion pays attention to IS, which supports efficient data handling and robust classification performance. It further explores similarity distance measures used to assess the closeness of data observations in *k*-NN and related techniques.

2.1 Review on *k*-NN

The *k*-NN algorithm is regarded as one of the earliest and most fundamental classification methods in machine learning. Despite its long history, it continues to attract significant academic attention. Its enduring importance is attributed to its simplicity, resilience and adaptability. According to Amer et al. (2025) and Halder et al. (2024), numerous studies have sought to address its limitations and enhance its efficiency, particularly for applications involving large and complex datasets.

The *k*-NN classifier operates on the principle of proximity in feature space. When given an unlabelled input x, the algorithm computes its distance to all instances in the labelled training set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2, \cdots, (x_n, y_n))\}$, where each x_i represents a feature vector and y_i its corresponding class label. The class of x is predicted based on the majority among its k nearest neighbours. While Euclidean distance is commonly employed, Kalra et al. (2022) suggested that alternatives such as Manhattan and Mahalanobis distances are also effective, especially when datasets contain features with varying scales or internal correlations.

As noted by Feng and Zhang (2025) and Ortiz-Villaseñor et al. (2025), *k*-NN remains relevant largely due to its non-parametric nature, which enables it to capture complex and nonlinear patterns in data without presuming any underlying distribution. This flexibility supports its widespread application in fields such as pattern recognition, text classification, bioinformatics and recommendation systems. Moreover, the transparent decision-making process of *k*-NN, where each prediction directly relates to the nearest data points, enhances its value in domains that require interpretability.

Despite these strengths, *k*-NN is not without its drawbacks. As highlighted by Boateng et al. (2020) and Syriopoulus et al. (2023), the computational costs of *k*-NN increase linearly with the size of the training data, since each prediction necessitates computing distances to all training instances. This can make *k*-NN inefficient for large datasets, with memory usage and prediction time becoming significant concerns. Additionally, Zhang (2021) pointed out that *k*-NN is sensitive to irrelevant or redundant data, which can adversely affect classification accuracy.

To address these challenges, many researchers have focused on improving *k*-NN through efficient DR and IS strategies. According to Eleftheriadis et al. (2024), these methods work by selecting a representative subset of instances from the training set, thereby boosting computational efficiency and minimizing the influence of noise and redundancy. Recent advancements in instance selection algorithms have further enhanced the scalability and accuracy of *k*-NN in real-world applications. Ongoing research continues to reinforce the position of *k*-NN as a valuable and evolving tool within the machine learning landscape.

2.2 Review on IS algorithms

IS algorithms play a vital role in reducing data for supervised learning. Their goal is to find a smaller subset $\mathcal{S} \subset \mathcal{D}$ that maintains the discriminative power of the original dataset while lowering computational costs and reducing noise effects. These methods are especially useful in instance-based learning such as *k*-NN, where large datasets often cause excessive memory use and slow prediction performance.

According to Malhat et al. (2020), early IS algorithms, often referred to as classical IS, include the condensed nearest neighbour rule (CNN), edited nearest neighbour rule (ENN) and the decremental reduction optimisation procedure (DROP) family. These methods seek to reduce dataset size by eliminating redundant or misclassified samples to

improve storage efficiency and noise robustness. However, as discussed by Cavalcanti and Soares (2020), these algorithms may not always strike the optimal balance between reduction rate and accuracy, and their effectiveness can vary across datasets.

Recent advances have introduced more sophisticated IS algorithms. For example, Carbonera (2021) highlighted density-based approaches such as global density-based IS (GDIS) and enhanced global density-based IS (EGDIS), which utilise global or local density and relevance functions to identify representative instances. The goal, as described by Malhat et al. (2020), is to balance reduction rate, classification accuracy and computational efficiency. These advanced methods often outperform traditional density-based techniques, especially on large datasets.

The ranking-based instance selection (RIS) method, as presented by Cavalcanti and Soares (2020), evaluates instance similarity and removes those situated in both safe and indecision regions, aiming to maintain a balance between accuracy and reduction. In addition, evolutionary algorithms, including genetic and cooperative co-evolutionary frameworks, have been extensively applied for IS. Kordos et al. (2021) and Zhai and Song (2022) have noted that these approaches optimise instance subsets by maximising accuracy and minimising storage use, often employing multi-objective fitness criteria. Although these methods offer reliable and adaptable results, they typically require greater computational resources.

Ensemble strategies have also been explored for IS. Abdalla et al. (2025) described approaches such as bagging and boosting, which involve combining several algorithms or applying a single algorithm across different subsets to enhance stability and overall performance. However, it should be noted, as highlighted by Blachnik (2019), that such strategies may lead to a decrease in compression ability.

Recent hybrid methods integrate IS with preprocessing techniques such as feature selection or employ decomposition strategies such as the one-versus-one (OVO) approach for multiclass classification. As reported by Fang et al. (2023), OVO-based IS has demonstrated enhanced accuracy and higher reduction rates for classifiers including support vector machine (SVM) and *k*-NN. Moran et al. (2022) discussed the emergence of reinforcement learning (RL) and curiosity-driven frameworks, which offer flexible control over noise removal and data reduction. Cluster-oriented selection represents another promising direction. Studies by Abdelhay et al. (2025) and Saha et al. (2022) have demonstrated that clustering can be used to sample instances from both cluster centres and borders, resulting in a 2-3% improvement in *k*-NN accuracy over existing methods.

Despite these advancements, Eleftheriadis et al. (2024) emphasised that no single IS algorithm consistently outperforms others across all domains. Achieving a balance between reduction rate, accuracy and computational costs remains a significant challenge. Furthermore, the number and sensitivity of parameters can influence practical applications. For this reason, the choice of instance selection method should be guided by dataset characteristics, application requirements and available computational resources.

Overall, IS algorithms hold a central role in enabling k-NN to function effectively with large and complex datasets. Their use boosts classification efficiency and predictive reliability. Continued efforts should explore how to improve scalability, adaptability and robustness to achieve better IS and *k*-NN performance.

2.3 Review on similarity distance measurements

The choice of similarity or dissimilarity distance measure is pivotal in determining the performance and relevance of data analysis tasks. According to Levy et al. (2025), this selection directly affects the accuracy of results obtained from real datasets. Earlier investigations, as reported by Alfeilat et al. (2019), identified approximately 54 core measures spanning eight distinct families. More recently, Muniswamaiah et al. (2023) expanded this count to 78, illustrating the diversity of available similarity and dissimilarity measures. Nevertheless, Alfeilat et al. (2019) highlighted that no single distance measure consistently outperforms others across all datasets. The effectiveness of each measure is context-dependent, a finding that aligns with the no free lunch theorem. This underscores the importance of choosing a distance measure tailored to the specific characteristics of the data to ensure robust analytical outcomes.

In this study, we aim to enhance DR methods by extending the functionality of the classic Euclidean distance, recognised as part of the Minkowski family (Alfeilat et al., 2019). Euclidean distance is renowned for its ability to identify the shortest path between two points in a multidimensional space. As noted by Shirkhorshidi et al. (2015)

and Muniswamaiah et al. (2023), it remains a popular choice in machine learning (ML) due to its intuitive interpretation, suitability for continuous and uniformly scaled features and computational efficiency, which scales linearly with data size.

This efficiency makes Euclidean distance especially well-suited for high-dimensional or resource-constrained environments where rapid computation is necessary. Empirical evidence presented by Shirkhorshidi et al. (2015) demonstrates that Euclidean distance excels at identifying patterns in datasets characterised by compact groupings and distinct clusters, as it precisely captures variations in data magnitude. Consequently, Euclidean distance supports the objectives of our proposed approach, which is designed to uncover inherent patterns and structures in complex datasets.

3 PROPOSED METHOD

Our review shows that most IS algorithms still face difficulty in finding the best balance among reduction rate, accuracy and computational costs. Recent hybrid methods improve accuracy but often increase computational demands. Ensemble approaches can keep or exceed baseline accuracy yet usually reduce data compression and efficiency. Many IS algorithms also depend on extensive parameter tuning, where several parameters are both numerous and sensitive, requiring precise adjustment for best performance. These limitations have led us to design a simpler IS method with minimal reliance on complex parameter tuning. The proposed algorithm is named Euclidean ranking-based instance selection (ERbIS).

3.1 ERbIS

This section presents the stepwise procedure of the proposed ERbIS algorithm. For the sake of clarity, the notations used in the ERbIS procedure are introduced below.

- Let N be the total number of instances and d the number of features.
- Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be the full dataset, where:
  - $x_i \in \mathbb{R}^d$ is the feature vector for instance i;
  - y_i is the corresponding class label ($y_i \in Y$, the set of possible classes).
- Split the dataset into:
  - training set: $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{n_{train}}$
  - test set: $\mathcal{D}_{test} = \{(x_i, y_i)\}_{i=n_{train}+1}^{N}$
  - thus, $n_{train} + n_{test} = N$
- Let $X_{train} \in \mathbb{R}^{n_{train} \times d}$ and $y_{train} \in \mathbb{R}^{n_{train}}$ denote the feature matrix and label vector for training data; similarly, for X_{test}, y_{test}.
- Let the anchor point be $a \in \mathbb{R}^d$.
- Let P denote the set of selected prototypes.

Step 1: Finding an anchor data point

The first step in the ERbIS algorithm is to determine an anchor point $a \in \mathbb{R}^d$ in the feature space, which will serve as the reference for ranking instances in the subsequent steps. In this study, we consider two types of anchor points, as given in the following definitions.

Definition 1 (mean of each column anchor point): Given a training dataset $X_{train} \in \mathbb{R}^{n_{train} \times d}$, the MEC anchor point is defined as the mean vector of all training instances:

$$a_{MEC} = \left[\frac{1}{n_{train}} \sum_{i=1}^{n_{train}} x_i \right] \tag{1}$$

where x_i denotes the i-th row of X_{train}.

Definition 2 (first data anchor point): The FD anchor point is simply defined as the first instance in the training set:

$$a_{FD} = x_1 \tag{2}$$

where x_1 denotes the first row of X_{train}.

In practice, either a_{MEC} or a_{FD} is selected as the anchor point a for the subsequent distance computation and instance ranking steps.

Step 2: Compute Euclidean distances

For each training instance x_i, compute its Euclidean distance to the anchor:

$$d_i = \|x_i - a\|_2 = \sqrt{\sum_{j-1}^{d} \left(x_{ij} - a_j\right)^2} \tag{3}$$

for $i = 1, \cdots, n_{train}$.

Step 3: Rank instances by distance

Let π be the permutation of indices such that:

$$d_{\pi(1)} \leq d_{\pi(2)} \leq \cdots \leq d_{\pi(n_{train})} \tag{4}$$

Re-order X_{train} and y_{train} according to π:

$$\tilde{x}_i = x_{\pi(i)}, \tilde{y}_i = y_{\pi(i)} \text{ for } i = 1, \cdots, n_{train} \tag{5}$$

Step 4: Instance selection

Let $\left\{\left(\tilde{x}_i, \tilde{y}_i\right)\right\}_{i=1}^{n_{train}}$ denote the set of training instances, sorted by increasing distance to the anchor point as described in Step 3. The objective of this step is to select a minimal set of boundary instances (prototypes) that represent each contiguous segment of instances sharing the same class label in the ranked list. Define the prototype set as $P = \emptyset$. Proceed as follows:

For each unique class $c \in Y$, identify all contiguous segments S_c^j of instances with the label c in the ranked list. That is, for each segment $S_c^j = \left\{\left(\tilde{x}_k, \tilde{y}_k\right), \cdots, \left(\tilde{x}_l, \tilde{y}_l\right)\right\}$ such that $\tilde{y}_k = \cdots = \tilde{y}_l = c$ and either $k = 1$ or $\tilde{y}_{k-1} \neq c$ and either $l = n_{train}$ or $\tilde{y}_{l+1} \neq c$, add the **first** and **last** instance of the segment to the prototype set:

$$P \leftarrow P \cup \{(\tilde{x}_k, c), (\tilde{x}_l, c)\} \tag{6}$$

If $k = l$ (the segment contains only one instance), add only $(\tilde{x}_k, c)$. Repeat this process for all class segments in the ranked list. The resulting prototype set P thus contains the boundary instances for each contiguous class segment, which are expected to be most informative for preserving class distinctions near class boundaries.

Step 5: k-NN classification process

The final prototype set, denoted as $P = \left\{\left(p_j, q_j\right)\right\}_{j=1}^{m}$, where $p_j \in \mathbb{R}^d$ and $q_j \in Y$, serves as the reference set for classification. For a given test instance $x_{test} \in \mathbb{R}^d$, the class assignment is determined using the k-NN rule. Specifically, the Euclidean distance between x_{test} and each prototype p_j in P is computed as $d_j^{test} = \left\| x_{test} - p_j \right\|^2$. The k prototypes with the smallest distances to x_{test} are identified and the predicted class label $\hat{y}_{test}$ is assigned as the most frequent class among these k nearest prototypes, that is,

$$\hat{y}_{test} = \text{mode}(\{q_j : j \in N_k(x_{test})\}), \quad (7)$$

where $N_k(x_{test})$ denotes the set of indices corresponding to the k nearest prototypes in P. This approach enables efficient and accurate classification by utilising the compact and informative prototype set constructed in the earlier steps.

4 EXPERIMENTAL SETUP AND RESULTS

This section describes the experimental setup and presents the corresponding findings. Additionally, it discusses the implications of these results in the context of IS and classification performance.

4.1 Experimental design

To ensure a fair and comprehensive evaluation across multiple application areas, we utilised the 21 datasets introduced by Cavalcanti and Soares (2020), as listed in Table 1. These datasets are publicly available in the KEEL Repository (Alcalá-Fdez et al., 2011) at https://sci2s.ugr.es/keel/datasets.php. The experimental setup closely follows the procedure outlined by Cavalcanti and Soares (2020) to facilitate direct comparison of results.

For the experiments, all data types within each dataset were subjected to min–max normalisation, which scales feature values to the range [0, 1]. This approach was applied uniformly, without emphasis on the specific treatment of different data types, as the focus of the study was not on investigating their individual impacts. The normalisation process ensures that all features, regardless of their original scale or type, contribute equally to distance-based measures within the classification algorithms. Min–max normalisation for a feature x is defined as follows:

$$x = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (8)$$

where x_{min} and x_{max} are the minimum and maximum values of the feature x within the dataset, respectively. Additionally, all the datasets were complete and free from missing values, so no imputation procedures were necessary.

For evaluation, we employed stratified 10-fold cross-validation to partition the datasets into training and test sets. In stratified k-fold cross-validation, the original dataset D is partitioned into k mutually exclusive subsets (folds) $D_1, D_2, \cdots, D_k$ of approximately equal size, while preserving the percentage of samples for each class label across all folds. Specifically, for each iteration $i \in \{1, 2, \cdots, k\}$, the fold D_i is held out as the test set and the remaining $k - 1$ folds are combined to form the training set. The procedure is mathematically expressed as:

$$Accuracy = \frac{1}{k}\sum_{i=1}^{k} Accuracy_i \quad (9)$$

Where $Accuracy_i$ denotes the classification accuracy obtained in the i-th fold. Stratification ensures that each fold maintains the class distribution of the original dataset, thereby preventing bias towards majority classes and ensuring reliable evaluation across minority classes as well. By adhering to this experimental setup, we ensure that the evaluation of ERbIS is both rigorous and consistent with established benchmarks.

***Table 1**. Dataset characteristics.*

Datasets	#Instances	#Attributes	#Classes
appendicitis	106	7	2
balance	625	4	3
bupa	345	6	2
coil2000	9822	85	2
contraceptive	1473	9	3
haberman	306	3	2
hayes-roth	160	4	3
heart	270	13	2

Datasets	#Instances	#Attributes	#Classes
ionosphere	351	33	2
led7digit	500	7	10
marketing	6876	13	9
mon*k*-2	432	6	2
move-libras	360	90	15
pima	768	8	2
segment	2310	19	7
titanic	2201	3	2
vowel	990	13	11
wine	178	13	3
wineQ-red	1599	11	6
wineQ-white	4898	11	7
yeast	1484	8	10

4.1.1 Experimental evaluation

The experimental evaluation was conducted in two separate phases to assess the effectiveness of the proposed ERbIS algorithm. In the first phase, two models, ERbIS-FD and ERbIS-MEC, were tested against established baselines. Their performance was compared with the traditional *k*-NN algorithm and two recognised instance selection techniques, namely ENN and CNN. Three performance metrics were used to ensure a fair comparison, namely reduction time (T_{red}), reduction percentage ($R_{\%}$) and classification accuracy ($A_{\%}$).

Definition 3 (reduction time): Reduction time (T_{red}) is defined as the total computation time, measured in milliseconds (ms), required by instance selection to process the original dataset and produce a reduced dataset. Formally, let t_{start} and t_{end} denote the start and end timestamps of the reduction process, respectively, then $T_{red} = t_{end} - t_{start}$.

Definition 4 (reduction percentage): Reduction percentage ($R_{\%}$) quantifies the proportion of instances eliminated from the original dataset because of the instance selection process. Let N_{original} denote the number of instances in the original dataset and N_{selected} the number of instances remaining after selection. The reduction percentage is defined as: $R_{\%} = \frac{N_{\text{original}} - N_{\text{selected}}}{N_{\text{original}}} \times 100\%$.

Definition 5 (classification accuracy): Classification accuracy ($A_{\%}$) measures the proportion of correctly classified test instances in the evaluation phase. Let N_{correct} be the number of correctly classified instances and N_{total} the total number of test instances. The classification accuracy is given by: $A_{\%} = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\%$.

Previous research has mainly emphasised classification accuracy and reduction percentage while neglecting the importance of time efficiency. Consequently, comprehensive comparisons that include reduction times are limited. To bridge this gap, we conducted manual experiments and measured reduction time along with the other two common performance measures (Cavalcanti & Soares, 2020).

The second stage of our analysis compared ERbIS-FD and ERbIS-MEC with three established IS methods introduced by Cavalcanti and Soares (2020). These are RIS1, DROP3 and adaptive threshold-based instance selection (ATISA1). Each method offers a balance between accuracy and reduction rate. DROP3 maintains high generalisation accuracy even with a significant reduction of instances. ATISA1 selects data dynamically through adaptive thresholds, reducing noise while preserving relevance. RIS1 prioritises instances by rank to retain informative samples. Reduction rate and classification accuracy served as the main performance metrics. This allowed an assessment of how well the proposed models reduce data size while maintaining predictive accuracy.

Overall, the evaluation strategy promotes fair and detailed examination of the proposed IS algorithms in various contexts. The findings provide essential insight into their practical benefits for machine learning.

4.1.2 System and hardware requirements

All the experiments in phase one were conducted on a standard desktop computer with an Intel Core i5 7200U processor operating at 2.5 GHz, 12 GB of RAM and an NVIDIA GeForce GT 940MX GPU. This configuration offered sufficient computing power to process and analyse the selected datasets efficiently. The proposed ERbIS algorithm was implemented in Python 3.8 using reliable open-source libraries to ensure reproducibility and ease of use. Data handling and numerical operations were performed with NumPy and Pandas. Machine learning procedures, including the implementation of *k*-NN, ENN and CNN algorithms, along with stratified *k*-fold cross-validation and performance assessment, were carried out using the Scikit-learn library (Pedregosa et al., 2011). The use of Python and open-source tools not only ensure reproducibility but also promotes transparency, supporting future research and development in this field.

4.2 Results

In phase one, comprehensive parameter tuning was performed for *k* values of 1, 2, 3, 5 and 7 to determine the optimal setting for each classification model. The tuning process involved systematically evaluating the classification accuracy of each model across these *k* values to identify which setting yielded the highest performance. Selecting an appropriate *k* is crucial in the *k*-NN algorithm, as lower values may result in more sensitive or unstable predictions, while higher values can reduce sensitivity to noise but may overlook important local patterns. By rigorously testing a range of *k* values, this study ensures that each model is assessed under its most effective conditions. The *k* value that produced the best performance for each model is presented in Table 2, allowing a fair and focused comparison across different algorithms. This approach underscores the importance of parameter tuning in optimising classification models and ensures that the reported results accurately reflect the capabilities of each method under optimal circumstances.

For phase two, the reduction percentage ($R_{\%}$) and classification accuracy ($A_{\%}$) for RIS1, ATISA1 and DROP3 were taken from the study by Cavalcanti and Soares (2020). It is important to note that all the reported results are mean values, calculated using 10-fold cross-validation for each test model.

Table 2. *Phase 1 performance evaluation.*

Algorithms	IS models	Best *k* result	Classification models
k-nearest neighbour	-	1	1NN
Edited nearest neighbour	ENN	3	ENN-3
Condensed nearest neighbour	CNN	1	CNN-1
Euclidean ranking-based instance selection – first data	ERbIS-FD	3	ERbISFD-3
Euclidean ranking-based instance selection – mean of each column	ERbIS-MEC	3	ERbISMEC-3

4.2.1 Phase 1 evaluation results

All the analyses in phase one were carried out using Python. A stratified 10-fold cross-validation method was applied to ensure robust results. For each model, the mean values were calculated and are shown in Table 2. These mean values were then used for further analysis and comparison.

As shown in Table 3, the proposed IS models, namely ERbIS-FD and ERbIS-MEC, demonstrated a shorter computational time (T_{red}) for the reduction process compared with ENN and CNN. The CNN model exhibited the longest computational time (T_{red}) for instance reduction. In contrast, both ENN and the proposed models achieved comparable average times. A comparison between ERbIS-FD and ERbIS-MEC indicated that ERbIS-FD performed the reduction process marginally faster. This outcome may be attributed to the less complex procedure required for anchor determination in ERbIS-FD.

***Table 3**. Reduction time (T_{red}) results (in milliseconds (ms)).*

IS models / Datasets	ENN	CNN	ERbIS-FD	ERbIS-MEC
	T_{red}	T_{red}	T_{red}	T_{red}
appendicitis	11.81	154.02	17.19	31.94
balance	14.40	2141.41	24.44	40.34
bupa	8.63	1065.55	23.42	36.08
coil2000	525.12	118682.40	411.54	440.26
contraceptive	31.52	15488.99	43.19	60.40
haberman	12.04	1025.49	20.54	36.19
hayes-roth	7.64	171.45	17.19	32.45
heart	7.98	441.38	22.69	33.74
ionosphere	11.46	139.35	16.68	35.13
led7digit	18.75	568.46	18.02	36.77
marketing	415.53	95302.26	140.85	165.75
mon*k*-2	9.71	1206.42	24.65	38.25
move-libras	48.98	298.53	33.16	46.27
pima	20.21	4241.49	30.24	42.23
segment	59.76	1086.81	77.96	95.82
titanic	32.81	55968.83	47.87	68.77
vowel	29.04	471.07	33.36	50.32
wine	13.39	116.86	18.23	35.33
wineQ-red	60.02	2306.59	46.07	59.71
wineQ-white	404.59	6504.06	106.95	124.99
yeast	59.36	994.39	38.70	55.00
Average	99.13	14318.17	**63.70**	80.87
Min. value	7.64	116.86	16.68	31.94
Max. value	525.12	118682.40	411.54	440.26

Based on the data presented in Table 4, the CNN method achieved the highest average reduction percentage ($R_{\%}$) among all the approaches tested. Specifically, across the 21 datasets, CNN attained an average reduction of 57.29%. This superior reduction rate is attributed to the CNN design, which aggressively condenses the training set by retaining only a minimal subset of representative instances while eliminating redundant data points. ERbIS-FD and ERbIS-MEC followed with average reduction rates of 40.27% and 35.24%, respectively, while ENN exhibited the lowest average reduction performance at 30.13%. While these results underscore the effectiveness of CNN in reducing dataset size, it is important to note that a greater reduction rate does not necessarily guarantee improved classification accuracy; excessive data reduction may lead to loss of important information, potentially compromising the predictive performance of the model.

In terms of classification accuracy ($A_{\%}$), the proposed model ERbISMEC-3 achieved the highest average value at 66.46%, followed closely by ERbISFD-3 with an average accuracy of 65.57%. Both models outperformed the baseline 1-NN, which achieved 60.94%, and also surpassed the performance of state-of-the-art models, including ENN-3 at 57.28% and CNN-1 at 56.52%. When evaluated across the 21 datasets, both ERbISFD-3 and ERbISMEC-3 delivered better classification results than the baseline and benchmark models on 16 datasets.

Taken together, these findings demonstrate that while CNN excels in terms of reduction rate, the ERbISFD-3 and ERbISMEC-3 models are more effective at maintaining or improving classification accuracy, even after substantial data reduction. This highlights the importance of balancing reduction rate and predictive performance, as greater reduction does not automatically translate into better accuracy.

Table 4. *Phase 1 comparison – reduction percentage ($R_{\%}$) and classification accuracy ($A_{\%}$).*

Models / Datasets	1NN	ENN-3		CNN-1		ERbISFD-3		ERbISMEC-3	
	$A_{\%}$	$R_{\%}$	$A_{\%}$	$R_{\%}$	$A_{\%}$	$R_{\%}$	$A_{\%}$	$R_{\%}$	$A_{\%}$
appendicitis	72.08	21.70	77.93	57.65	79.18	75.68	86.82	75.69	**88.64**
balance	73.30	9.30	73.29	65.86	59.85	76.96	**80.34**	74.28	79.38
bupa	58.00	7.85	52.18	40.23	**60.28**	59.96	57.62	50.72	60.22
coil2000	90.05	23.95	**90.26**	36.87	87.47	34.86	84.37	30.74	87.47
contraceptive	**42.97**	32.72	37.41	45.93	40.93	39.11	40.46	37.11	41.48
haberman	61.81	41.61	60.77	31.98	62.73	35.17	67.38	33.33	**67.99**
hayes-roth	58.36	2.72	41.21	57.11	48.75	58.47	**68.13**	56.06	63.13
heart	74.81	41.42	76.67	78.06	75.93	57.53	**80.74**	45.49	**80.74**
ionosphere	84.89	47.76	86.88	71.20	85.74	37.33	85.18	31.69	**88.03**
led7digit	60.27	5.00	34.12	53.70	49.80	64.48	68.00	52.61	**70.20**
marketing	20.20	62.61	19.01	82.95	22.05	7.62	28.20	5.66	**28.26**
mon*k*-2	69.46	28.94	82.40	41.07	80.11	33.46	**84.24**	34.77	80.08
move-libras	66.20	21.74	56.67	31.33	66.67	51.02	67.50	36.04	**75.28**
pima	67.63	11.91	67.57	83.46	65.10	14.85	**72.40**	13.13	71.49
segment	96.30	77.88	91.89	98.60	70.95	12.60	96.75	11.54	**96.97**
titanic	60.10	46.21	31.10	27.07	**69.15**	92.07	53.93	91.72	60.92
vowel	65.35	72.39	50.20	96.60	49.49	31.24	**65.76**	30.25	65.35
wine	90.01	3.03	88.07	87.99	87.03	8.66	94.90	3.45	**94.97**
wineQ-red	48.03	14.85	51.41	78.40	31.08	43.59	52.66	31.21	**54.91**
wineQ-white	40.02	83.09	41.22	53.53	24.46	6.36	**49.88**	4.72	49.10
yeast	40.89	6.20	49.83	40.73	26.69	45.00	57.27	25.01	**57.47**
Average	60.94	30.13	57.28	57.29	56.52	40.27	65.57	35.24	**66.46**
Min. value	20.20	2.72	19.01	27.07	22.05	6.36	28.20	3.45	28.26
Max. value	96.30	83.09	91.89	98.60	87.47	92.07	96.75	91.72	96.97

4.2.2 Phase 2 evaluation results

In phase two evaluation, we compared our proposed models, ERbISFD-3 and ERbISMEC-3, with the current IS model, RIS1. We also included two established models, ATISA1 and DROP3, in the analysis. The comparison focused on reduction percentage ($R_{\%}$) and classification accuracy ($A_{\%}$). The results for these metrics are displayed in Table 5.

On average, our proposed model ERbISMEC-3 attained the highest classification accuracy ($A_{\%}$) across 21 datasets. This represents an improvement of 0.15% over the current model RIS1, which achieved 66.31% accuracy. In comparison, ERbISFD-3 showed a marginally lower average accuracy, falling short of RIS1 by 0.74%. When evaluated against ATISA1 and DROP3, both of our proposed models displayed consistently higher accuracy. The results show that both ERbISFD-3 and ERbISMEC-3 removed fewer instances, with reduction rates below 41%, compared to the three benchmark models, which all achieved rates above 50%. Although the DROP3 method produced the highest instance reduction at 81.34%, its generalisation performance was lower, reaching only 61.62%. In contrast, ERbISMEC-3 had the smallest reduction rate at 35.24% but achieved the highest generalisation ability of 66.46%. These outcomes indicate that the proposed methods are more effective at retaining informative instances. As a result, both proposed models help preserve and, in some cases, improve classification accuracy.

Table 5. *Phase 2 comparison – reduction percentage ($R_{\%}$) and classification accuracy ($A_{\%}$).*

Models / Datasets	RIS1		ATISA1		DROP3		ERbIS-FD		ERbIS-MEC	
	$R_{\%}$	$A_{\%}$	$R_{\%}$	$A_{\%}$	$R_{\%}$	$A_{\%}$	$R_{\%}$	$A_{\%}$	$R_{\%}$	$A_{\%}$
appendicitis	70.64	80.79	89.62	78.33	90.98	78.33	75.68	86.82	75.69	**88.64**
balance	76.53	**86.72**	73.07	84.92	83.25	85.24	76.96	80.34	74.28	79.38
bupa	43.83	57.97	44.93	56.86	74.30	56.00	59.96	57.62	50.72	**60.22**
coil2000	31.01	**94.03**	84.17	93.94	99.96	93.96	34.86	84.37	30.74	87.47
contraceptive	26.14	46.91	67.74	**50.34**	80.95	48.66	39.11	40.46	37.11	41.48
haberman	49.20	64.98	78.87	69.69	89.14	**70.00**	35.17	67.38	33.33	67.99
hayes-roth	44.79	66.42	81.25	36.11	87.44	44.44	58.47	**68.13**	56.06	63.13
heart	50.99	74.44	78.11	**81.11**	83.46	76.30	57.53	80.74	45.49	80.74
ionosphere	75.50	**90.62**	72.43	83.89	93.07	82.22	37.33	85.18	31.69	88.03
led7digit	16.53	**72.37**	85.84	41.27	94.82	39.45	64.48	68.00	52.61	70.20
marketing	82.96	18.28	99.67	21.14	99.79	21.32	7.62	28.20	5.66	**28.26**
mon*k*-2	72.79	92.11	37.73	**95.45**	96.73	79.32	33.46	84.24	34.77	80.08
move-libras	62.28	**78.11**	55.94	68.05	62.81	65.27	51.02	67.50	36.04	75.28
pima	53.78	63.40	61.47	71.69	82.29	71.95	14.85	**72.40**	13.13	71.49
segment	88.96	92.12	80.85	92.47	88.58	91.73	12.60	96.75	11.54	**96.97**
titanic	17.61	**69.06**	99.14	32.17	99.84	32.17	92.07	53.93	91.72	60.92
vowel	76.90	87.88	51.16	**88.89**	52.83	87.07	31.24	65.76	30.25	65.35
wine	86.64	93.39	74.66	90.53	84.39	89.47	8.66	94.90	3.45	**94.97**
wineQ-red	42.01	45.83	61.09	53.40	81.41	52.16	43.59	52.66	31.21	**54.91**
wineQ-white	41.83	41.55	62.00	42.08	81.82	43.34	6.36	**49.88**	4.72	49.10
yeast	36.27	41.77	65.00	48.63	81.67	47.32	45.00	57.27	25.01	**57.47**
Average	52.15	66.31	68.40	62.77	81.34	61.62	40.27	65.57	35.24	**66.46**
Min. value	16.53	18.28	37.73	21.14	52.83	21.32	6.36	28.20	3.45	28.26
Max. value	88.96	94.03	99.67	95.45	99.96	93.96	92.07	96.75	91.72	96.97

4.3 Discussion

The results of phase one experiments indicate that both of our proposed IS algorithms, as shown in Table 3, achieved slightly faster processing times than ENN and were noticeably faster than CNN. When the FD served as the anchor, computation times were shorter compared to using the MEC as the anchor. This is because calculating the mean involves additional computational steps ($O(n)$), while selecting the first data point does not require such processing. The process combines distance calculation, ranking from the anchor point ($O(n)$) and the instance selection procedure ($O(n)$). This combination offers clear computational benefits, especially for larger datasets with more than 4,000 instances. For example, in datasets such as coil2000 (9,822 instances), marketing (6,876 instances) and wineQ-white (4,898 instances), our models demonstrated greater efficiency than both ENN and CNN.

The approach of excluding inner instances while keeping the first and last instances for each class in the ranking list led to an improvement in the quality of the selected instances. This improvement is reflected in the generalisation ability of these prototypes, which managed to either maintain or enhance the classification accuracy achieved by the baseline *k*-NN classifier. Although the reduction in the number of instances, which ranged from 35 to 40%, was not as great as that observed with some alternative algorithms, including the existing IS model, the classification outcomes were better. This suggests that the instances selected using this method were of higher quality compared to those chosen by other IS algorithms.

Furthermore, the experimental results demonstrate that the MEC anchor, calculated as the mean of all data points in the dataset, consistently achieves higher classification accuracy than the FD anchor. By employing the overall

mean as the anchor point, this method offers a reference that represents the collective distribution of the dataset, rather than focusing on class-specific characteristics. Although the MEC anchor is susceptible to the influence of outliers, its central positioning within the data distribution generally aligns with regions of higher data density, which may contribute to improved classification performance. It should be noted, however, that in datasets characterised by significant outlier presence or class imbalance, the efficacy of the MEC anchor may be compromised. In such cases, the use of class-wise anchors or robust statistical estimators may offer enhanced performance (Shukla et al., 2025).

5 CONCLUSION AND FUTURE WORKS

This study addresses the significant challenges associated with applying the *k*-NN classifier to datasets of varying sizes and complexity. Specifically, it examines issues related to computational inefficiency and diminished classification accuracy, which frequently result from the presence of noisy or redundant data. To address these challenges, we proposed the ERbIS algorithm, which implements a straightforward and effective IS strategy based on ranking the distances of data points from a designated anchor point. Two anchor point selection methods were explored: one based on the mean value of each feature and the other on the first data point in the dataset. Both approaches offer flexibility and can be adapted to various data structures.

A comprehensive set of experiments was carried out using 21 benchmark datasets obtained from the KEEL repository. The results show that ERbIS is able to reduce the size of the training set by an average of 35-40%. This reduction does not compromise classification accuracy. In many cases, ERbIS even outperforms the baseline *k*-NN method and several well-known IS models, such as ENN, CNN, ATISA1, DROP3 and RIS1. The ERbIS-MEC model showed higher generalisation accuracy than the other models. This result highlights the value of choosing a central anchor. Such an approach helps keep important instances and reduces the risk of losing samples from the minority class. Additionally, the ERbIS-FD model demonstrated superior computational efficiency. Compared to more complex models, the ERbIS-FD requires fewer parameter settings and is able to process data more rapidly. These characteristics make ERbIS-FD particularly well-suited for large-scale or complex applications.

The results of this study demonstrate that utilising simple and effective strategies for selecting relevant instances can significantly enhance the performance of the *k*-NN classifier. Specifically, both ERbIS models achieve this improvement by carefully removing less important data, which reduces the dataset size while maintaining classification accuracy. Furthermore, this IS approach requires minimal parameter adjustment, making both ERbIS models easy to implement and robust across a wide variety of datasets.

Despite these advancements, some limitations remain. The current implementation of ERbIS relies exclusively on Euclidean distance, which may not be optimal for datasets exhibiting non-linear relationships or containing categorical features. Additionally, while the anchor-based selection strategy generally preserves samples from the minority class, there is a risk that rare but important instances may be inadvertently removed during the instance elimination process. Such omissions could negatively affect model performance, particularly in cases where the dataset is highly imbalanced.

Future research could explore the integration of alternative distance metrics or the combination of multiple similarity measures to further enhance ERbIS. Incorporating domain-specific knowledge in the process of anchor point selection may also yield additional benefits. Moreover, developing ensemble or hybrid frameworks that combine ERbIS with other DR techniques could further improve its overall effectiveness. Creating adaptive methods for anchor point selection, tailored to the unique characteristics of each dataset, may also contribute to better generalisation performance. In summary, ERbIS provides an effective solution for IS in *k*-NN classification and opens new opportunities for further research and use in demanding, data-rich settings.

ADDITIONAL INFORMATION AND DECLARATIONS

Acknowledgments: The authors would like to thank the *Faculty of Computer Science and Information Technology, UNIMAS*, for their continuous support and guidance throughout this research. Special appreciation goes to reviewers, colleagues and peers who provided valuable feedback during the preparation of this work.

Funding: This work was funded by the *Ministry of Higher Education Malaysia* and *Universiti Malaysia Sarawak (UNIMAS)*, Grant No. FRGS/1/2021/ICT02/UNIMAS/02/6.

Conflict of Interests: The authors declare no conflict of interest.

Author Contributions: A.M.B.R.: Writing – Original draft, Data Curation, Investigation, Methodology, Software. M.B.H.: Writing – Original draft, Supervision, Conceptualization, Investigation, Formal Analysis, Writing – Review & editing. S.B.S.: Supervision, Investigation, Writing – Review & editing. N.B.M.N.: Validation, Writing – Review & editing.

Statement on the Use of Artificial Intelligence Tools: The authors acknowledge the use of DeepAI to assist in improving the structure and wording of certain sections of this manuscript.

Data Availability: The data that support the findings of this study are openly available at https://sci2s.ugr.es/keel/datasets.php, see also (Alcalá-Fdez et al., 2011).

REFERENCES

Abdalla, H. I., Altaf, A., & Hamzah, A. A. (2025). A threefold-ensemble k-nearest neighbour algorithm. *International Journal of Computers and Applications*, 47(1), 70–83. https://doi.org/10.1080/1206212X.2024.2446896

Abdelhay, H. K., Benameur, Z., & Younes, G. (2025). NCBIS: A Novel Clustering-Based Approach for effective Instance Selection. In *2025 7th International Conference on Pattern Analysis and Intelligent System*s, (pp. 1–7). IEEE. https://doi.org/10.1109/PAIS66004.2025.11126049

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., & García, S. (2011). KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2–3), 255–287.

Alfeilat, H. a. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Salman, H. S. E., & Prasath, V. S. (2019). Effects of distance measure choice on K-Nearest Neighbor Classifier performance: a review. *Big Data*, 7(4), 221–248. https://doi.org/10.1089/big.2018.0175

Amer, A. A., Ravana, S. D., & Habeeb, R. a. A. (2025). Effective k-nearest neighbor models for data classification enhancement. *Journal of Big Data*, 12(1), Article 86. https://doi.org/10.1186/s40537-025-01137-2

An, S., Hu, Q., Wang, C., Guo, G., & Li, P. (2021). Data reduction based on NN-kNN measure for NN classification and regression. *International Journal of Machine Learning and Cybernetics*, 13, 765–781. https://doi.org/10.1007/s13042-021-01327-3

Blachnik, M. (2019). Ensembles of instance selection methods: A comparative study. *International Journal of Applied Mathematics and Computer Science,* 29, 151–168. https://doi.org/10.2478/amcs-2019-0012

Blachnik, M., & Kordos, M. (2020). Comparison of Instance Selection and Construction Methods with Various Classifiers. *Applied Sciences*, 10(11), Article 3933. https://doi.org/10.3390/app10113933

Boateng, E. Y., Otoo, J., & Abaye, D. A. (2020). Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, random Forest and Neural network: a review. *Journal of Data Analysis and Information Processing*, 8(4), 341–357. https://doi.org/10.4236/jdaip.2020.84020

Carbonera, J. L. (2021). A Global Density-based Approach for Instance Selection. In *Proceedings of the 23rd International Conference on Enterprise Information Systems*, (pp. 402–409). ScitePress. https://doi.org/10.5220/0010402104020409

Cavalcanti, G. D., & Soares, R. J. (2020). Ranking-based instance selection for pattern classification. *Expert Systems With Applications*, 150, 113269. https://doi.org/10.1016/j.eswa.2020.113269

Eleftheriadis, S., Evangelidis, G., Ougiaroglou, S. (2024). An Empirical Analysis of Data Reduction Techniques for k-NN Classification. In *Artificial Intelligence Applications and Innovations*, (pp. 83–97). Springer. https://doi.org/10.1007/978-3-031-63223-5_7

Fang, C., Wang, M., Tsai, C., Lin, W., & Liao, P. (2023). Instance selection using one-versus-all and one-versus-one decomposition approaches in multiclass classification datasets. *Expert Systems*, 40(6), e13217. https://doi.org/10.1111/exsy.13217

Feng, Z., & Zhang, J. (2025). Research on AI based Personalized Recommendation System for Foreign Language Learning using k-Nearest Neighbours. In *2025 International Conference on Intelligent Systems and Computational Networks*, (pp. 1–7). IEEE. https://doi.org/10.1109/ICISCN64258.2025.10934182

Gupta, S., Thakar, U., & Tokekar, S. (2025). A comprehensive survey on techniques for numerical similarity measurement. *Expert Systems With Applications*, 277, 127235. https://doi.org/10.1016/j.eswa.2025.127235

Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbour algorithm: a comprehensive review and performance analysis of modifications. *Journal of Big Data*, 11(1), Article 113. https://doi.org/10.1186/s40537-024-00973-y

Hossin, M., & Rushdi, A. M. (2025). Anchor-Point Based Euclidean Reduction for Enhanced Instance-based Classification. In *2025 14th International Conference on Information Technology in Asia*, (pp. 90–95). IEEE. https://doi.org/10.1109/CITA66455.2025.11198677

Kalra, V., Kashyap, I., & Kaur, H. (2022). Effect of Distance Measures on K-Nearest Neighbour Classifier. In *2022 Second International Conference on Computer Science, Engineering and Applications*, (pp. 1–7). IEEE. https://doi.org/10.1109/ICCSEA54677.2022.9936314

Kordos, M., Blachnik, M., & Scherer, R. (2021). Fuzzy clustering decomposition of genetic algorithm-based instance selection for regression problems. *Information Sciences*, 587, 23–40. https://doi.org/10.1016/j.ins.2021.12.016

Levy, A., Shalom, B. R., & Chalamish, M. (2025). A guide to similarity measures and their data science applications. *Journal of Big Data*, 12(1), Article 188. https://doi.org/10.1186/s40537-025-01227-1

Li, J., Zhu, Q., & Wu, Q. (2020). A parameter-free hybrid instance selection algorithm based on local sets with natural neighbours. *Applied Intelligence*, 50, 1527–1541. https://doi.org/10.1007/s10489-019-01598-y

Malhat, M., Menshawy, M. E., Mousa, H., & Sisi, A. E. (2020). A new approach for instance selection: Algorithms, evaluation, and comparisons. *Expert Systems With Applications*, 149, 113297. https://doi.org/10.1016/j.eswa.2020.113297

Moran, M., Cohen, T., Ben-Zion, Y., & Gordon, G. (2022). Curious instance selection. *Information Sciences*, 608, 794–808. https://doi.org/10.1016/j.ins.2022.07.025

Muniswamaiah, M., Agerwala, T., & Tappert, C. C. (2023). Applications of binary similarity and distance measures. arXiv preprint arXiv:2307.00411. https://doi.org/10.48550/arXiv.2307.00411

Ortiz-Villaseñor, D., Trujillo-Hernández, G., Real-Moreno, O., Castro-Toscano, M. J., Medina-Madrazo, L. D., & Barrera-Román, D. (2025). K-nearest neighbours regression and applications. In *Exploring Psychology, Social Innovation and Advanced Applications of Machine Learning*, (pp. 295–316). IGI Global. https://doi.org/10.4018/979-8-3693-6910-4.ch015

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). SciKit-Learn: Machine Learning in Python. *Journal of Machine Learning Research,* 12, 2825–2830. https://doi.org/10.5555/1953048.2078195

Saha, S., Sarker, P. S., Saud, A. A., Shatabda, S., & Newton, M. H. (2022). Cluster-oriented instance selection for classification problems. *Information Sciences*, 602, 143–158. https://doi.org/10.1016/j.ins.2022.04.036

Shi, Z. (2020). Improving K-Nearest Neighbors algorithm for imbalanced data classification. *IOP Conference Series Materials Science and Engineering*, 719(1), 012072. https://doi.org/10.1088/1757-899x/719/1/012072

Shirkhorshidi, A. S., Aghabozorgi, S., & Wah, T. Y. (2015). A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS One*, 10(12), e0144059. https://doi.org/10.1371/journal.pone.0144059

Shukla, S., Singh, A., & Vishwakarma, G. K. (2025). Predictive estimation for mean under median ranked set sampling: an application to COVID-19 data. *Indian Journal of Pure and Applied Mathematics*, 56(1), 218–229. https://doi.org/10.1007/s13226-023-00470-7

Syriopoulos, P. K., Kalampalikis, N. G., Kotsiantis, S. B., & Vrahatis, M. N. (2023). kNN Classification: A review. *Annals of Mathematics and Artificial Intelligence*, 93(1), 43–75. https://doi.org/10.1007/s10472-023-09882-x

Xing, W., & Bei, Y. (2020). Medical Health Big Data Classification Based on KNN Classification Algorithm. *IEEE Access*, 8, 28808–28819. https://doi.org/10.1109/ACCESS.2019.2955754

Zhai, J., & Song, D. (2022). Optimal instance subset selection from big data using genetic algorithm and open source framework. Journal of Big Data, 9, Article 87. https://doi.org/10.1186/s40537-022-00640-0

Zhang, S. (2021). Challenges in KNN Classification. *IEEE Transactions on Knowledge and Data Engineering*, 34(10), 4663–4675. https://doi.org/10.1109/TKDE.2021.3049250

Acta Informatica Pragensia is published by the Prague University of Economics and Business, Czech Republic | eISSN: 1805-4951