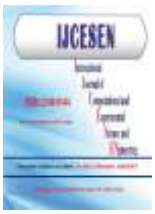
International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 7712-7726
http://www.ijcesen.com



ISSN: 2149-9144

Research Article

PSO-UFS: A Novel Approach to Univariate Feature Selection Using Particle Swarm Optimization

Ramzi Benaicha[1]*, Mohammed Mehdi Bouchene[2]

[1]Faculty of Technology, University of Badji Mokhtar Annaba, B.P. No 12 RP, 23000, Annaba, Algeria
* **Corresponding Author Email:** ramzi.benaicha@univ-annaba.dz - **ORCID:** 0000-0002-5247-7800

[2]Laboratoire des Télécommunications (LT), Université 8 Mai 1945 Guelma, Guelma 24000, Algeria.
Email: bouchenemahdi@gmail.com - **ORCID:** 0000-0002-5247-0050

Article Info:

DOI: 10.22399/ijcesen.403
Received : 23 January 2025
Accepted : 08 July 2025

Keywords

univariate feature selection;
particle swarm optimization;
interpretable classification performance;
automated feature selection;
data analysis.

Abstract:

Univariate Feature Selection (UFS) traditionally involves a labor-intensive process of trial-and error, necessitating the selection of scoring functions and the determination of feature numbers. These choices can inadvertently affect both the performance and interpretability of the model. To address this challenge, we introduce Particle Swarm Optimization for Univariate Feature Selection (PSO-UFS), an innovative method that automates these crucial decisions. PSO-UFS leverages the power of Particle Swarm Optimization (PSO) to autonomously identify the optimal scoring function and feature subset that maximize a machine learning algorithm's performance metric. Our empirical evaluations across multiple datasets demonstrate that PSO-UFS significantly outperforms traditional UFS in various performance metrics, including accuracy, precision, recall, and F1-score. Importantly, PSO-UFS generates more interpretable feature subsets, thereby enhancing the model's comprehensibility. This advancement paves the way for broader applications in real-world scenarios where feature reduction and interpretability are paramount.

1. Introduction

The advent of high-dimensional data, characterized by an abundance of features, presents both opportunities and challenges for machine learning [28]. The wealth of information harbors the potential for deeper insights and more accurate models. However, it also introduces issues such as redundancy, noise, and irrelevant features, which can negatively impact prediction performance [16].Feature selection, a crucial step in data analysis and machine learning, is a strategy employed to navigate these challenges. It involves identifying and selecting a subset of relevant features, thereby enhancing model efficacy, interpretability, and computational efficiency [17, 24, 33]. However, feature selection is not without its own set of challenges [35]. One such challenge is the manual selection of the scoring function and the number of features in Univariate Feature Selection (UFS), a type of filter-based feature selection method. This introduces subjectivity and uncertainty, potentially leading to suboptimal results [10]. Filter-based feature selection methods, including UFS, form a significant part of the broader taxonomy of feature selection approaches. These methods are known for their simplicity, scalability, and efficiency in handling large datasets. They evaluate individual features based on statistical measures of relevance, such as correlation or mutual information, independent of any machine learning algorithm. This makes them distinct from wrapper methods, which evaluate subsets of features based on the performance of a specific machine learning model [38], and embedded methods, which perform feature selection as part of the model training process. This paper introduces a groundbreaking approach to Univariate Feature Selection (UFS) in machine learning, known as PSO-UFS. The novelty of this method lies in its use of Particle Swarm Optimization (PSO) to simultaneously optimize both the scoring function and the number of features, a task that was previously handled separately. This simultaneous optimization addresses the limitations of existing UFS techniques by considering feature interactions indirectly, which was not possible when optimizing

the scoring function and the number of features separately. This innovative approach not only enhances the efficiency of UFS but also improves the interpretability of the selected features, paving the way for a deeper understanding of model predictions. This advancement represents a significant leap in the field of feature selection, underscoring the potential of bio-inspired algorithms in automating and optimizing the process. We evaluate PSO-UFS on multiple datasets and compare it with traditional UFS methods. Our results demonstrate the effectiveness of PSO-UFS and highlight the potential of bio-inspired algorithms in automating and optimizing feature selection. We also underscore the improved interpretability of features selected by PSO-UFS, leading to a more profound understanding of model predictions [30]. The main contributions of this paper are:

- We propose PSO-UFS, a novel technique that automates univariate feature selection using PSO. This approach addresses limitations in existing UFS techniques by indirectly considering feature interactions.
- We formulate univariate feature selection as an optimization problem with two decision variables: k and s. This formulation allows us to automate what was previously a manual process.
- We evaluate our technique on three benchmark datasets from diverse domains. Our results show that PSO-UFS can significantly improve UFS performance across different types of data.
- Our technique identifies an optimal subset of features, denoted by S, that is more interpretable than those obtained by conventional manual methods. This improvement in interpretability can aid in understanding model predictions.

This work represents a significant advancement in the field of feature selection, demonstrating the potential of bio-inspired algorithms to automate and optimize the process. The rest of the paper is organized as follows. Section 2 reviews the related work in the field of univariate feature selection and compares it with our proposed PSO-UFS method. Section 3 provides a detailed explanation of the UFS process, including commonly used scoring functions and selection steps. Section 4 presents the PSO-UFS algorithm, outlining the optimization problem formulation and the PSO procedure. Section 5 describes our experimental setup and evaluation results on the chosen datasets. Finally, Section 6 concludes the paper.

2. Related Work

The domain of feature selection has witnessed substantial advancements with the integration of various optimization techniques. These techniques strive to pinpoint the optimal subset of features that maximize a predefined objective function. Among these, Genetic Algorithms (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) are particularly noteworthy. Each of these techniques possesses unique advantages and can be effective in different contexts. However, this paper underscores PSO due to its inherent simplicity, adaptability, scalability, and robustness [1, 20]. Recent studies have indeed harnessed PSO for feature selection in various contexts. For instance, an algorithm was proposed that employs the best-worst multi-attribute decision-making method for univariate feature selection [2]. This method ranks features based on their scores computed by different scoring functions such as chi-square, ANOVA F-test, mutual information, and t-test [2]. The algorithm was evaluated on four UCI benchmark datasets and was found to outperform other univariate feature selection methods in terms of accuracy, precision, recall, and F1-score [2]. Moreover, a study proposed two PSO variants to undertake feature selection tasks [37]. The aim was to overcome two major shortcomings of the original PSO model, i.e., premature convergence and weak exploitation around the near optimal solutions. The proposed models illustrated statistical superiority for discriminative feature selection for a total of 13 data sets [37].Another study applied the feature selection-based Particle Swarm Optimization (PSO) method to detect phishing websites [4]. The experimental findings showed that the proposed PSO-based feature selection model substantially improved classification accuracy, sensitivity, specificity, f1–score, and Matthew's correlation coefficient in machine learning models [4]. In addition, a novel feature selection-based transfer learning approach using particle swarm optimization (PSO) for unsupervised transfer learning (FSUTL-PSO) was implemented [32]. In FSUTL-PSO, all objectives were incorporated into one fitness function and common good features from the source and target domains were selected based on the fitness function for eliminating the threat of degenerated features [32].Furthermore, a study proposed efficient feature selection methods using PSO with a fuzzy rough set as a fitness function [19]. The proposed methods were compared against two classical feature selection methods, as well as three PSO and rough set-based feature selection approaches. The results showed

that using the proposed techniques, a small feature subset may be automatically selected with better classification accuracy than utilizing all features [19]. Lastly, a study provided an overview of PSO for feature selection in biomedical data analysis [3]. This study reviewed the applications, challenges, and future directions of PSO for feature selection in various biomedical domains, such as gene expression, protein structure, medical image, and clinical diagnosis. The study also discussed the advantages and disadvantages of PSO for feature selection and suggested some possible improvements [3]. Univariate feature selection (UFS) is a popular technique for reducing the dimensionality and complexity of datasets. It ranks features based on their individual relevance to the target variable, without considering their interactions. UFS is fast and scalable, making it suitable for high-dimensional datasets. Previous studies have shown that UFS is more stable in the case of high-dimensional databases [12]. Scoring functions are pivotal in univariate feature selection, serving to independently assess the relevance of each feature to the target variable [6, 21]. Commonly employed scoring functions include chi-square, ANOVA F-test (Analysis of Variance), mutual information, and Fisher score. These functions, grounded in various statistical tests or information theory measures, come with distinct assumptions and properties. For instance, chi-square and ANOVA F-test, based on the chi-square distribution and F-distribution respectively, are apt for categorical and numerical features. Mutual information quantifies the mutual dependence between two variables, while Fisher score gauges the discriminative power of a feature for binary classification. These scoring functions have found extensive application across diverse domains such as text mining, bioinformatics, and computer vision [25]. However, the selection of the scoring function and the number of features (k) often requires manual tuning and domain knowledge, which can be time-consuming and subjective [6]. Moreover, these scoring functions do not consider the interactions or dependencies among features, which can lead to suboptimal feature subsets [6]. To address these challenges, our proposed PSO-UFS approach automates the selection of the scoring function and k, and it considers feature interactions indirectly by optimizing a performance metric of a machine learning algorithm [6]. While existing studies underscore the versatility and applicability of PSO and univariate feature selection techniques across different data types or tasks, they do not specifically tackle the problem of automating univariate feature selection using PSO. This is precisely the focus of our paper, highlighting the novelty and significance of our proposed PSO-UFS approach. Our method aims to bridge this gap in the literature by introducing a technique that harnesses the strengths of PSO to automate univariate feature selection, thereby offering an efficient and effective solution for feature selection tasks [8, 27].

3 Univariate Feature Selection: An In-depth Analysis

Univariate feature selection (UFS) is a statistical technique that evaluates the relevance of individual features to a target variable, independent of other features. This method is particularly beneficial for high-dimensional datasets, as it can enhance the interpretability and efficiency of machine learning models by reducing noise, redundancy, and irrelevance in the data. Despite its simplicity and scalability, UFS has been successfully applied in various domains, including text classification [7, 11], cancer prediction [31], and image retrieval [25], thereby enhancing the transparency of the models.

3.1 Scoring Functions: A Comparative Analysis

The effectiveness of Univariate Feature Selection (UFS) largely hinges on the scoring function employed to gauge the relevance of each feature to the target variable. Different scoring functions come with different assumptions and properties, and may favor different types of features. Consequently, the choice of scoring function is pivotal for effective feature selection. Some of the commonly used scoring functions encompass:

- **Chi-square:** This function tests the independence between a categorical feature and a categorical target variable. It computes the difference between the observed and expected frequencies of each category, normalized by the expected frequency. The higher the score, the more dependent the feature is on the target variable.
- **ANOVA F-test (Analysis of Variance):** This function tests the equality of the means of a numerical feature across different groups defined by a categorical target variable. It computes the ratio of the between-group variance to the within-group variance. The higher the score, the more disparate the feature means are across the groups.
- **Mutual information:** This function quantifies the mutual dependence between a feature and a target variable, irrespective

of their types. It computes the reduction in uncertainty about one variable given the knowledge of the other variable. The higher the score, the more information the feature and the target variable share.

- **Pearson correlation:** This function measures the linear relationship between a numerical feature and a numerical target variable. It computes the covariance between the feature and the target variable, normalized by their standard deviations. The higher the absolute value of the score, the stronger the linear relationship is.
- **Fisher score:** This function measures the discriminative power of a feature for binary classification. It computes the ratio of the between-class variance to the within-class variance. The higher the score, the more discriminative the feature is.

To illustrate the differences among these scoring functions, we use the Iris dataset [14] as an example. The Iris dataset comprises 150 samples of three types of iris flowers (setosa, versicolor, and virginica) with four features: sepal length, sepal width, petal length, and petal width. The target variable is the type of iris flower. We apply the four scoring functions to rank the features according to their relevance to the target variable. Table 1 presents the normalized scores of each feature for each scoring function.The results reveal that while all four scoring functions concur that petal length and petal width are the most relevant features, they differ in the ranking of sepal length and sepal width. This demonstrates that different scoring functions may have different preferences or sensitivities for different features. For instance, Pearson correlation assigns a high score to sepal length due to its strong linear relationship with the target variable, while chi-square assigns a low score to sepal width due to its weak dependence with the target variable.

3.2 Determining the Optimal Number of Features (*k*)

Determining the optimal number of features (k) to select is a critical and challenging decision in UFS. Selecting too few features may lead to loss of valuable information, while selecting too many features may introduce noise and redundancy. Therefore, striking the right balance between simplicity and accuracy is essential.A common method for selecting k is the iterative approach, which involves the following steps:

1. Start with a small k value.
2. Train a machine learning model with the selected features.
3. Evaluate the model's performance on a validation set.
4. Incrementally increase k.
5. Repeat steps 2-4 until model performance plateaus for the accuracy and declines for the loss.

The iterative approach aims to find the smallest k that maximizes the model performance, assuming that the most relevant features are selected first. However, this method has several drawbacks that limit its effectiveness and efficiency:

- It requires manual selection of the initial and incremental values of k, which can introduce subjectivity and uncertainty into the process. For example, if the initial value of k is too high, the model may overfit the training data and perform poorly on the validation set. If the incremental value of k is too small, the model may take too long to reach the optimal performance, or never reach it at all.
- It limits the exploration of the vast search space of possible feature combinations, which may result in suboptimal solutions. For example, if the features are ranked by their individual scores, the iterative approach may miss some features that have low scores but high synergies with other features.
- It is time-consuming and computationally expensive, especially for large datasets and complex models. For example, if the dataset has 100 features and the optimal k is 50, the iterative approach would require training and evaluating 50 models, each with a different subset of features.

These drawbacks can affect the quality and validity of UFS, leading to biased or inconsistent results, or missed opportunities for improvement. For example, in a study on text classification, the authors found that the iterative approach with chi-square scoring function selected a suboptimal number of features, resulting in lower accuracy than using all features or using a different scoring function [29]. In another study on gene selection, the authors found that the iterative approach with ANOVA F-test scoring function selected a different number of features for different datasets, making it difficult to compare the results across different studies [18]. Therefore, there is a need for a more efficient and objective method for selecting k in UFS, one that can automatically and systematically

explore the search space of feature combinations and find the optimal solution. In this paper, we propose such a method, called PSO-UFS, which uses Particle Swarm Optimization (PSO) to select the optimal number of features and the corresponding feature subset. We will describe the details of our method in the next section.

3.3 Algorithm for Univariate Feature Selection

The following algorithm delineates the steps involved in univariate feature selection. The input is a dataset matrix X with n features and a target variable vector y, a scoring function f, and a number of features to select k. The output is a subset of features S with k features.This algorithm calculates the scores for each feature using a scoring function f, which measures the relevance of each feature to the target variable. The scoring function can be any statistical test that evaluates the relationship between a feature and a target variable, such as chi-square, mutual information, ANOVA F-test, or Fisher score. The features are then sorted by their scores in descending order, as higher scores indicate higher relevance. The algorithm then selects the top k features with the highest scores as the subset for subsequent analysis. The number of features to select is often based on prior knowledge, domain expertise, or through experimentation. This simple process reduces dimensionality and retains only the most predictive features. However, manual selection of the scoring function and k can be suboptimal and time-consuming, as it may introduce biases and inconsistencies in the feature selection process. Moreover, manual selection does not consider the interactions among features or the impact of feature selection on the classifier performance. Therefore, there is a need for automating these choices and optimizing the feature selection process. In the next section, we present an optimization approach to automate the selection of the scoring function and the number of features k in univariate feature selection. This approach leverages Particle Swarm Optimization (PSO) to search for the optimal solution in the vast and complex space of scoring functions and feature combinations. PSO-UFS, as we call our approach, aims to optimize both the scoring function and the number of features simultaneously, based on the performance metric of a classifier trained on the selected features. PSO-UFS also offers a flexible and tunable framework, as it can accommodate different scoring functions, classifiers, and performance metrics, and can adjust the exploration-exploitation trade-off by tuning the PSO parameters. We will describe the details of our approach in the following subsections.

4 Methodology

This section introduces our proposed methodology, which employs Particle Swarm Optimization (PSO) to automate the selection of the scoring function and the number of features k in univariate feature selection. This approach aims to surmount the limitations of manual processes and provide a more efficient and effective solution for feature selection tasks.

4.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique inspired by the social behavior of bird flocking or fish schooling [23, 15, 26]. In PSO, each solution in the search space is represented as a ”particle”. Each particle has fitness values, which are evaluated by the fitness function to be optimized, and velocities, which guide the movement of the particles. The particles navigate through the problem space by following the currently optimal particles.

The velocity update rule in PSO is given by:

$$v_{ij}\,(t+1) = w \cdot v_{ij}\,(t) + c_1 \cdot r_1 \cdot (pbest_{ij} - x_{ij}\,(t)) + c2 \cdot r2 \cdot (gbest_j - x_{ij}\,(t)) \quad (1)$$

where v_{ij} $(t + 1)$ represents the velocity vector of particle i in dimension j at time $t + 1$, w is the inertia weight, c_1 and c_2 are cognitive and social parameters, respectively, r_1 and r_2 are random numbers between 0 and 1, $pbest_{ij}$ is the personal best position of particle i in dimension j, $gbest_j$ is the global best position in dimension j, and xij (t) is the position of particle i in dimension j at time t. The position update rule is given by:

$$x_{ij}\,(t+1) = x_{ij}\,(t) + v_{ij}\,(t+1) \quad (2)$$

This rule updates the position of each particle based on its current position and velocity. The updated position then serves as the new point in the search space for the next iteration. This iterative process continues until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a desired level of fitness. The best position found by the swarm during the search process is returned as the optimal solution. In the context of univariate feature selection, the optimal solution represents the optimal scoring function and the optimal number of features k. This sets the stage for the application of PSO in univariate feature selection, which we discuss in the next subsection.

4.2 PSO for Univariate Feature Selection (PSO-UFS)

Building upon the principles of Particle Swarm Optimization (PSO) outlined in Section 4.1 and the Univariate Feature Selection (UFS) method described in Section 3, we propose the PSO-UFS approach to automate the selection of the optimal scoring function and the number of features. This section provides a detailed explanation of the algorithm, including its formulation, implementation, and key components

4.2.1 Problem Formulation

The PSO-UFS approach formulates univariate feature selection as an optimization problem with two decision variables: the scoring function (*s*) and the number of features (*k*). The scoring function *s* assigns a score to each feature based on its relevance to the target variable. The set of available scoring functions *F* includes Chi-square, ANOVA F-test (Analysis of Variance), Mutual Information, and Fisher Score. The number of features k is an integer representing the number of top-ranked features to select, constrained by $k_{min} \leq k \leq k_{max}$, where $k_{min} = 1$ and $k_{max} = n$ (total number of features). The **objective function** is defined as the performance metric of a classifier trained on the selected features. This metric can be accuracy, precision, recall, F1-score, or area under the ROC curve (AUCROC). The optimization problem is formally defined as:

maximize $_{s,k}$ $f(s, k)$

subject to $s \in F$,

$$k_{min} \leq k \leq k_{max},$$

where $f(s, k)$ is the objective function, F is the set of available scoring functions, and k_{min} and k_{max} are the lower and upper bounds for k, respectively.

4.2.2 Particle Representation

Each particle in the swarm represents a potential solution to the optimization problem. A particle's position is encoded as a two-dimensional vector (*s*, *k*), where s is an integer representing the index of the scoring function in the set *F*, and k is an integer representing the number of features to select. The particle's velocity is represented as a two-dimensional vector (v_s, v_k), which determines its movement in the search space. This representation allows the algorithm to explore the search space efficiently and converge to an optimal solution.

4.2.3 Fitness Function

The fitness function evaluates the quality of a particle's solution. It is defined as **the 5-fold cross-validation accuracy** of a classifier trained on the selected features. To compute the fitness function, the algorithm first decodes the particle's position to extract the scoring function s and the number of features k. It then applies UFS to score all features using s and selects the top k features. Next, a classifier (e.g., KNN or Logistic Regression) is trained on the selected features, and its performance is evaluated using 5-fold cross-validation. The cross-validation accuracy is returned as the fitness value for the particle. This approach ensures that the selected features generalize well to unseen data.

4.2.4 PSO Initialization

The PSO algorithm begins by initializing a swarm of particles. The swarm size n_p (e.g., 10) determines the number of particles. Each particle is initialized with a random scoring function *s* and a random number of features *k* within the bounds [k_{min}, k_{max}]. The particles' velocities are also initialized randomly to guide their movement in the search space. This random initialization ensures diversity in the swarm, enabling the algorithm to explore a wide range of solutions.

4.2.5 PSO Update Rules

At each iteration, the particles' positions and velocities are updated based on their personal best and the global best positions. The velocity update rule is defined as:

$$v_{ij}(t+1) = w \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot (pbest_{ij} - x_{ij}(t)) + c_2 \cdot r_2 \cdot (gbest_j - x_{ij}(t)),$$

where $v_{ij}(t+1)$ is the velocity of particle i in dimension j at iteration $t + 1$, w is the inertia weight (e.g., 0.5), c_1 and c_2 are cognitive and social parameters (e.g., 0.5), r_1 and r_2 are random numbers between 0 and 1, $pbest_{ij}$ is the personal best position of particle i in dimension *j*, gbest*j* is the global best position in dimension *j*, and $x_{ij}(t)$ is the position of particle *i* in dimension *j* at iteration *t*. The position update rule is defined as:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1).$$

These update rules balance exploration and exploitation, allowing the swarm to converge to an optimal solution efficiently.

4.2.6 Termination Criteria

The PSO algorithm terminates when one of the following conditions is met: (1) a predefined number of iterations (e.g., 110) is reached, or (2)

the global best fitness value does not improve significantly over a number of iterations. These criteria ensure that the algorithm stops when further iterations are unlikely to yield better solutions.

4.2.7 Output

The algorithm returns the **optimal scoring function** (s^*), the **optimal number of features** (k^*), and the optimal feature subset (S^*). These represent the best solution found by the swarm to the feature selection problem. The optimal feature subset S^* is obtained by applying the scoring function s^* and selecting the top k^* features.

4.2.8 Algorithm Steps

The PSO-UFS algorithm is outlined in Algorithm 2. The input includes a dataset matrix X with n features and a target variable vector y, a set of scoring functions F, a number of particles p, a number of iterations t, and a classifier C. The output is an optimal scoring function s^* and an optimal number of features k^*.

4.2.9 Key Advantages

The PSO-UFS approach offers several key advantages. Unlike traditional methods that optimize the scoring function and the number of features separately, PSO-UFS simultaneously optimizes both, leading to better feature subsets. The algorithm is highly flexible, as it can accommodate different scoring functions, classifiers, and performance metrics. Additionally, PSO-UFS reduces the manual effort and subjectivity involved in feature selection, making it suitable for high-dimensional datasets. Its ability to consider feature interactions indirectly further enhances its effectiveness.

4.2.10 Reproducibility

To ensure reproducibility, the following steps are recommended: (1) use a fixed random seed for initializing particle positions and velocities, (2) clearly specify all hyperparameters (e.g., swarm size, inertia weight, cognitive and social parameters), (3) use the same dataset splits for cross-validation across all experiments, and (4) provide the source code and implementation details in a public repository. These measures enable other researchers to replicate the results and build upon the proposed method.

5 Experiments

In this section, we delve into the empirical evaluation of our proposed PSO-based univariate feature selection (PSO-UFS) approach. We meticulously examine its performance across diverse datasets and classifiers, demonstrating its effectiveness in enhancing model accuracy and efficiency.

5.1Datasets and Classifiers

The datasets utilized in this study are publicly sourced from reputable platforms such as the UCI Machine Learning Repository [13] and Kaggle. These datasets are commonly used within the machine learning community for empirical analysis of machine learning algorithms.

- **UCI Heart Disease:** This dataset includes information about patients diagnosed with heart disease, encompassing variables such as age, sex, type of chest pain, resting blood pressure, serum cholesterol, and more. The target variable signifies the presence or absence of heart disease in the patient [22]. The complexity of this dataset arises from the combination of categorical and numerical features, as well as the critical nature of the prediction task.
- **Breast Cancer Wisconsin:** This dataset consists of 569 instances of cancer biopsies, each with 32 features. It contains an identification number, cancer diagnosis (malignant or benign), and 30 numeric-valued laboratory measurements derived from cell nuclei of the biopsies [36]. The high dimensionality of this dataset makes it an excellent candidate for feature selection.
- **Adult Census Income:** This dataset, extracted from the 1994 Census bureau database, contains demographic information about adults from various countries. The prediction task is to determine whether a person earns over 50K a year [5]. This dataset is challenging due to its large number of instances and features, as well as the mix of categorical and numerical features.

  The K-Nearest Neighbors (KNN) and Logistic Regression classifiers were chosen for this study due to their unique characteristics, which make them suitable for a wide range of problems:
- **K-Nearest Neighbors (KNN):** KNN is a non-parametric, instance-based learning algorithm that is straightforward to understand and implement. It can handle complex decision boundaries, making it suitable for problems where the decision boundary is not linear. However, KNN has

some limitations. It is sensitive to noise and irrelevant features, which can negatively impact its performance. Also, KNN suffers from the curse of dimensionality, meaning its performance degrades rapidly as the number of features (dimensions) increases. This makes feature selection particularly important when using KNN [34].

- **Logistic Regression:** Logistic Regression is a parametric, probabilistic classifier that can provide interpretable coefficients, making it useful when interpretability is important. It assumes a linear relationship between the features and the log-odds of the positive class, which allows it to estimate the probability of a particular class membership. However, this linearity assumption is a limitation if the actual relationship is not linear. Also, Logistic Regression may underfit the data if the decision boundary is complex [9].

5.2 Evaluation Methodology

For the evaluation methodology, we employ both a hold-out method and 5-fold cross-validation. In the hold-out method, the dataset is partitioned into a training set (70% of the data) and a test set (30% of the data). The model is trained on the training set and evaluated on the test set. This method is simple and computationally efficient, but its performance estimate can be sensitive to how the data is split. On the other hand, 5-fold cross-validation provides a more robust performance estimate. In this method, the dataset is divided into 5 equal-sized folds. The model is trained and evaluated 5 times, each time using a different fold as the test set and the remaining folds as the training set. The final performance estimate is the average of the performance measures from the 5 folds. While this method is more computationally intensive than the hold-out method, it provides a more reliable and less biased performance estimate. It is particularly useful when the dataset is small, as it effectively uses the available data. The classifiers and evaluation methods, along with the PSO-UFS feature selection method, form a comprehensive framework for assessing the effectiveness of feature selection in machine learning tasks. The performance of the classifiers is evaluated using several key metrics:

- **Accuracy:** This is the proportion of true results (both true positives and true negatives) among the total number of cases examined. It is a fundamental metric for classification problems, providing a baseline measure of a model's overall correctness.
- **Precision:** Precision measures the proportion of true positive predictions (relevant instances that are correctly identified) among all positive predictions. It is particularly important when the cost of a false positive is high.
- **Recall:** Recall, also known as sensitivity, measures the proportion of true positive predictions among all actual positive instances. It is crucial when the cost of a false negative is high.
- **F1-score:** The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall, useful when you want to compare two or more models and need a single measurement.

These metrics were chosen because they provide a comprehensive view of the model's performance. Accuracy alone can be misleading, especially in imbalanced datasets. Precision, recall, and F1-score offer additional perspectives, helping us understand the trade-off between identifying as many relevant instances as possible (high recall) and keeping the number of irrelevant instances low (high precision). In the context of the PSO-UFS method, these metrics allow us to evaluate how well the selected features contribute to the performance of the classifiers. By optimizing these metrics, the PSO-UFS method aims to find a feature subset that maximizes the classifier's ability to make correct predictions and balances precision and recall. This is particularly important in real-world applications where both false positives and false negatives can have significant implications.For UFS, we consider four candidate scoring functions: Chi-square, F-statistic, Mutual information, and Fisher score. These scoring functions rank the features based on their relevance to the target variable, irrespective of their types (numeric or categorical features). We also set a lower bound of k_{min} = 1 and an upper bound of k_{max} = n, where n is the total number of features in the dataset. The objective function for Particle Swarm Optimization (PSO) is defined as the 5-fold classification accuracy of a classifier trained on the selected features. The PSO hyperparameters are set as follows: swarm size (n_p) of 10, inertia weight (ω) of 0.5, cognitive and social parameters (χ and ϕ) of 0.5, and a maximum number of iterations (max_{iters}) of 110. These parameters are chosen to strike a balance between exploration and exploitation during the optimization process. This balance is crucial to ensure that the PSO algorithm can find an optimal

solution within a reasonable time frame. The selected datasets, classifiers, performance metrics, and PSO parameters collectively constitute a comprehensive framework for evaluating the efficacy of our feature selection approach.

5.3 Results and Discussions

Our study involved the implementation of the PSO-UFS method in Python. The source code for this method is publicly a
across different datasets using KNN and Logistic Regression classifiers. The metrics used for evaluation are accuracy, precision, recall, and F1-score. For the UCI Heart Disease dataset, the accuracy improved by 11.11 percentage points (from 70.00% to 81.11%) with the KNN classifier and by 1.11 percentage points (from 87.78% to 88.89%) with the Logistic Regression classifier. For the Breast Cancer Wisconsin dataset, the accuracy improved slightly from 95.91% to 96.49% with both classifiers. This indicates that even for datasets where the original feature set already allows for high classification accuracy, the PSO-UFS approach can still find room for improvement. For the Adult Census Income dataset, the accuracy improved by 6.58 percentage points (from 76.01% to 82.59%) with the KNN classifier and by 3.73 percentage points (from 78.00% to 81.73%) with the Logistic Regression classifier. These improvements demonstrate the effectiveness of the PSO-UFS approach in identifying informative features, leading to enhanced performance metrics and thereby creating more effective and efficient automating and optimizing the feature selection process. This leads to more advanced and efficient machine learning models, paving the way for significant advancements in the field of machine learning. The key takeaway is that automation in feature selection, as demonstrated by PSO-UFS, can lead to significant improvements in model performance across various datasets and classifiers. This underscores the importance and potential of automated feature selection in machine learning. Future work could explore the application of this approach to other types of datasets and machine learning tasks.The figures 1 and 2 illustrate the performance of two models, K-Nearest Neighbors (KNN) and Logistic Regression, using features selected by the Particle Swarm Optimization - Univariate Feature Selection (PSO-UFS) method across different iterations on three datasets: UCI Heart Disease, Breast Cancer Wisconsin, and Adult Census Income.The red dots in these figures indicate the best average validation accuracy achieved across all iterations, while the blue dots represent the global best position for each iteration, which contains the optimal scoring function and number of features. For the UCI Heart Disease dataset, the best iteration was 33 for both KNN and Logistic Regression. For the Breast Cancer Wisconsin dataset, the best iteration was 51 for both models. For the Adult Census Income dataset, the best iteration was 33 for both KNN and Logistic Regression. These results underscore the effectiveness of the PSO-UFS method in selecting the optimal number of features and scoring function to achieve high cross-validation accuracy. The method's adaptability and efficiency are evident in the relatively low number of iterations needed to achieve these results, making the PSO-UFS method a promising tool for feature selection in various real-world applications.The PSO-UFS method demonstrates a promising performance across all three datasets. The method's ability to achieve high cross-validation accuracy underscores its effectiveness. The accuracy achieved on the UCI Heart Disease and Breast Cancer Wisconsin datasets is particularly noteworthy, given the complexity and high dimensionality of these datasets. Moreover, the method's performance remains relatively stable across multiple iterations, suggesting that it is robust to different initializations and can reliably find good solutions. The PSO-UFS method's effectiveness is further highlighted when compared to traditional Univariate Feature Selection (UFS) methods. Unlike traditional UFS methods, which often rely on a fixed scoring function and feature number, the PSO-UFS method adaptively selects the optimal scoring function and feature number based on the data. This adaptability allows the method to better capture the underlying structure of the data and leads to improved cross-validation accuracy. In conclusion, the PSO-UFS method offers a robust and effective solution for feature selection. Its ability to achieve high cross-validation accuracy across multiple datasets and its adaptability to different data structures make it a promising tool for various real-world applications. The figures presented in the paper provide empirical evidence supporting the effectiveness of the PSO method. One additional point to note is that the PSO method does not require a large number of iterations to find the best combination of features and scoring function. This efficiency is evident from the relatively low number of iterations needed to achieve the best scores in all three datasets. This characteristic further enhances the practical utility of the PSO method, as it allows for quick and efficient feature selection, which is particularly beneficial in scenarios where computational resources or time are limited. Beyond the figures, the underlying trend within the data sets is the consistent performance of the PSO-UFS method across different datasets and models. This is not

explicitly addressed in the paper but is evident from the figures. Additional analyses or visualizations that could provide deeper insights into the data and enhance the understanding of the research could include a comparison of the PSOUFS method with other feature selection methods, or a breakdown of the performance of the models on each feature selected by the PSO-UFS method. Based on the presented data, potential future research directions could include applying the PSO-UFS method to other models and datasets, exploring the impact of different scoring functions on the performance of the PSO-UFS method, or investigating ways to further improve the efficiency of the PSO-UFS method.The table 4 presents the feature reduction rates achieved by our PSO-UFS approach on each dataset. The feature reduction rate is calculated as $\frac{n-k}{u}$, where n is the number of original features and k is the number of selected features by our PSO-UFS approach. The feature reduction rate ranges from 0% (no reduction) to 70% (a substantial reduction), indicating that our PSO-UFS approach can effectively reduce the dimensionality of the data while maintaining or even improving model performance.This table shows that the PSO-UFS approach can significantly reduce the number of features needed for classification without sacrificing performance. For instance, in the case of the UCI Heart Disease dataset, the feature reduction rate for the Logistic Regression (LR) classifier is 23.08%, and for the K-Nearest Neighbors (KNN) classifier, it is as high as 69.23%. This demonstrates the efficiency of our PSO-UFS approach in handling high-dimensional data. The ability to reduce dimensionality is particularly beneficial in real-world applications where computational resources are limited. It not only speeds up the learning process but also helps to mitigate the risk of overfitting by eliminating irrelevant or redundant features. Thus, the PSO-UFS approach proves to be a valuable tool for feature selection in machine learning tasks. The highest reduction rate (69.23% for the KNN classifier on the UCI Heart Disease dataset) underscores the maximum efficiency of the PSO-UFS approach. This significant reduction in features can lead to substantial computational gains, especially in scenarios involving large-scale datasets or resource-constrained environments. In terms of real-world applications, the PSO-UFS approach could be particularly useful in healthcare for patient risk prediction, where high-dimensional patient data is common, or in finance for credit scoring or fraud detection, where interpretability and efficiency are crucial. Future work could explore the application of this approach to other types of datasets and machine-learning tasks, such as deep learning models or genomic data analysis. This could potentially lead to even more efficient and effective models, further advancing the field of machine learning.

6 Conclusion

This paper has introduced a novel approach for automating univariate feature selection (UFS) using Particle Swarm Optimization (PSO). Feature selection is a pivotal technique that simplifies datasets and enhances model performance. However, the manual selection of the scoring function and the number of features can be both time-consuming and suboptimal. Our PSO-based approach addresses these issues by automatically identifying the optimal subset of features and scoring function that maximize the accuracy of a logistic regression classifier. Our PSO-UFS approach was evaluated on multiple benchmark datasets and compared with the original feature sets. The results demonstrated that our approach consistently improved model performance across various metrics, including accuracy, precision, recall, and F1-score. By leveraging PSO, our approach significantly reduced the time and effort required in the feature selection process. It also considered a range of scoring functions, making it adaptable to different data characteristics and complexities. Moreover, the PSO-UFS approach selected subsets of features that were more interpretable, providing valuable insights into the underlying patterns and factors in the classification task.In conclusion, our PSO-based automated feature selection method represents a significant contribution to the field of machine learning and data analysis. It generates more effective and interpretable classification models by streamlining the feature selection process. This method holds potential for broader applications in various real-world scenarios where feature dimensionality reduction and model interpretability are crucial. Future research directions may include extending the PSO-based feature selection to other machine learning algorithms and adapting the method to handle high-dimensional and imbalanced datasets. Overall, our proposed method lays the groundwork for more efficient and accurate feature selection techniques in data-driven decision-making processes.

Algorithm 1: Univariate Feature Selection (UFS)

Result: Return the optimal feature subset S
Function `UnivariateFeatureSelection`(X, y, f, k):
  Initialize an empty list L;
  foreach *feature* x_i *in* X **do**
    Compute the score $s_i = f(x_i, y)$;
    Append (x_i, s_i) to L;
  end
  Sort L in descending order based on the scores;
  Initialize an empty set S;
  for $j = 1$ **to** k **do**
    Select the feature x_j with the highest score from L;
    Add x_j to S;
    Remove (x_j, s_j) from L;
  end
  return S;

Algorithm 2: Particle Swarm Optimization for Univariate Feature Selection (PSO-UFS)

Result: Return the optimal scoring function s^* and the optimal number of features k^*
Function `PSO-UFS`(X, y, F, p, t, C):
  Initialize a swarm of p particles with random positions and velocities in the search space;
  for $i = 1$ **to** t **do**
    foreach *particle* j *in the swarm* **do**
      Decode the position of particle j to obtain the scoring function s_j and the number of features k_j;
      Apply UFS on the dataset X using the scoring function s_j and select the top k_j features to obtain the feature subset S_j;
      Train the classifier C on the feature subset S_j and evaluate its performance P_j using 5-fold cross-validation;
      If P_j is better than the personal best of particle j, update the personal best position;
    end
    Update the global best position of the swarm based on the best personal performances of all particles;
    Update the velocity and position of each particle using the personal best and global best positions;
  end
  Decode the global best position to obtain the optimal scoring function s^* and the optimal number of features k^*;
  return s^*, k^*;

Table 1: *Normalized scores from different scoring functions for each feature in the Iris dataset.*

Feature	Chi-square	ANOVA F-test	Mutual information	Pearson correlation
Sepal Length	0.27	0.06	0.29	0.87
Sepal Width	0.00	0.00	0.00	0.00
Petal Length	0.86	1.00	1.00	0.99
Petal Width	1.00	0.81	0.99	1.00

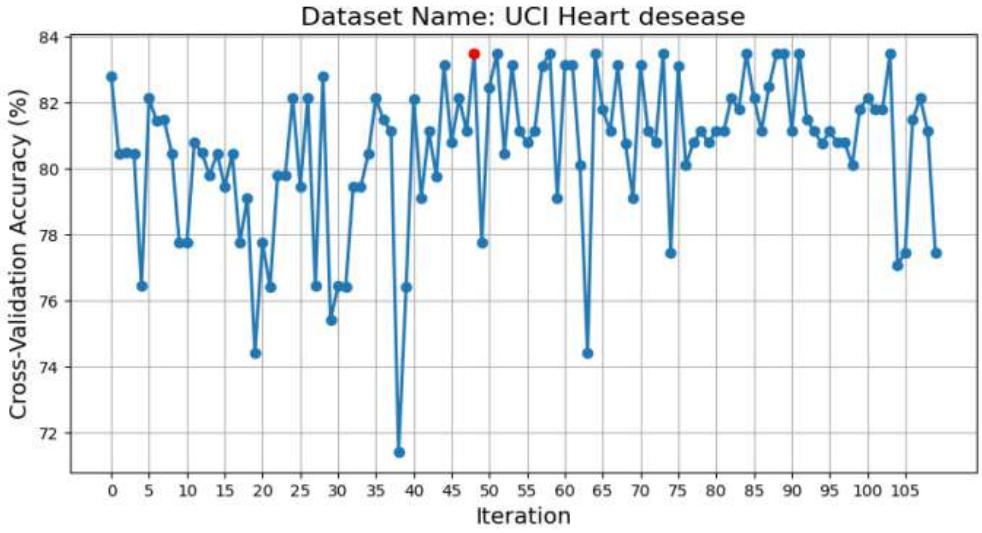
Table 2*: Performance Metrics of KNN Classifier.*

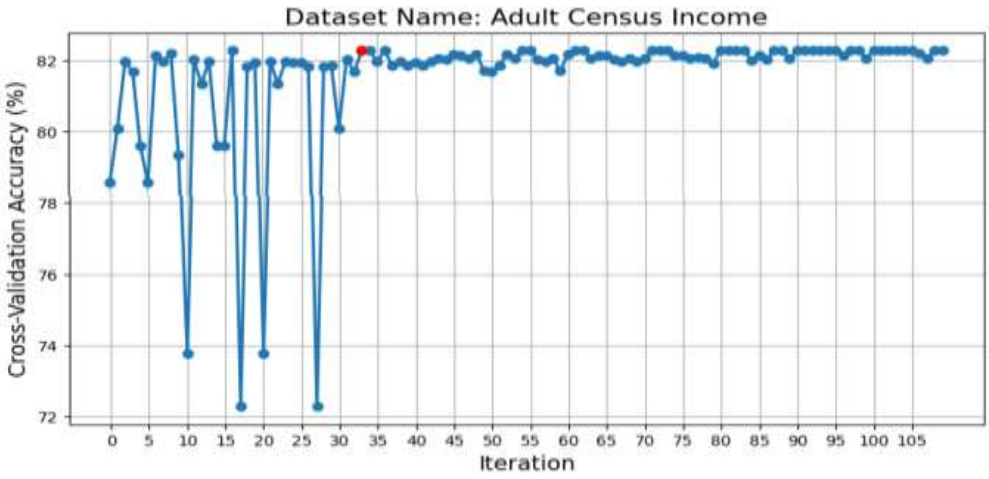
Dataset	Metric	Without feature selection (%)	PSO-UFS (%)
UCI Heart Disease	Accuracy	70.00	81.11
	Precision	69.79	80.95
	Recall	69.46	81.06
	score	69.55	81.00
Breast Cancer Wisconsin	Accuracy	95.91	96.49

	Precision Recall F1-score	96.48 94.78 95.52	96.23 96.23 96.23
Adult Census Income	Accuracy Precision Recall F1-score	76.01 66.95 60.59 61.69	82.59 77.35 74.52 75.72

Table 3: *Performance Metrics of Logistic Regression Classifier.*

Dataset	Metric	Without feature selection (%)	PSO-UFS (%)
UCI Heart Disease	Accuracy Precision Recall F1-score	87.78 88.01 87.38 87.59	88.89 89.34 88.40 88.69
Breast Cancer Wisconsin	Accuracy Precision Recall F1-score	95.91 96.48 94.78 95.52	96.49 96.23 96.23 96.23
Adult Census Income	Accuracy Precision Recall F1-score	78.00 72.33 61.24 62.60	81.73 78.08 69.21 71.69





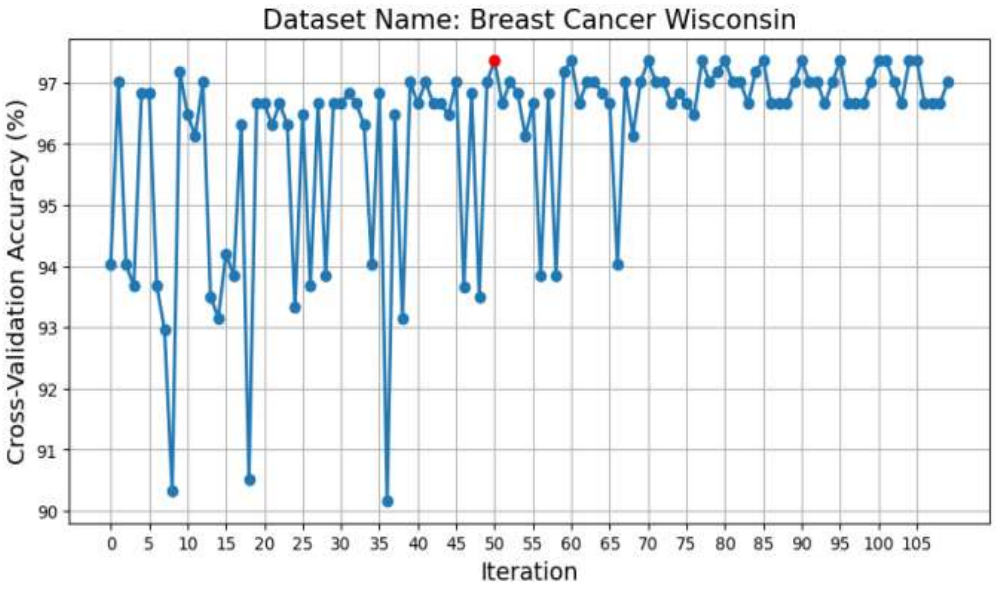
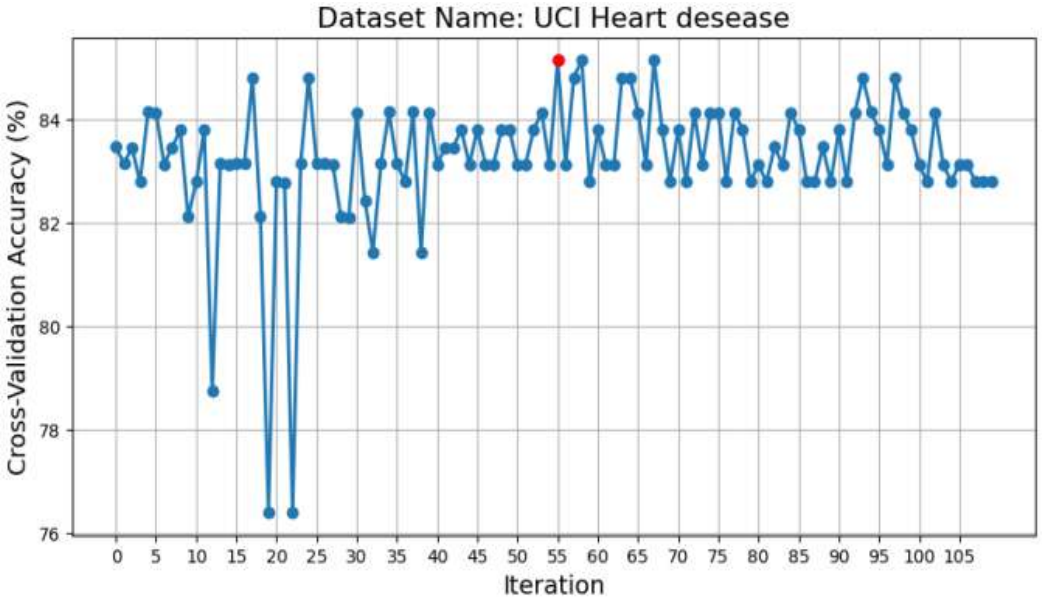
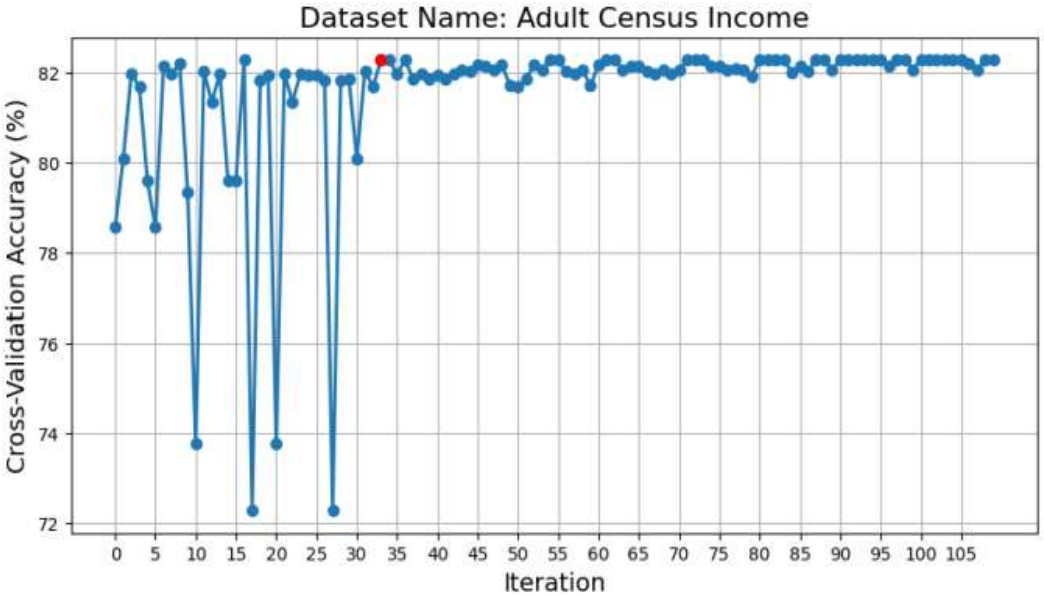


Figure 1: *Cross-validation accuracy of KNN classifier vs PSO iteration.*





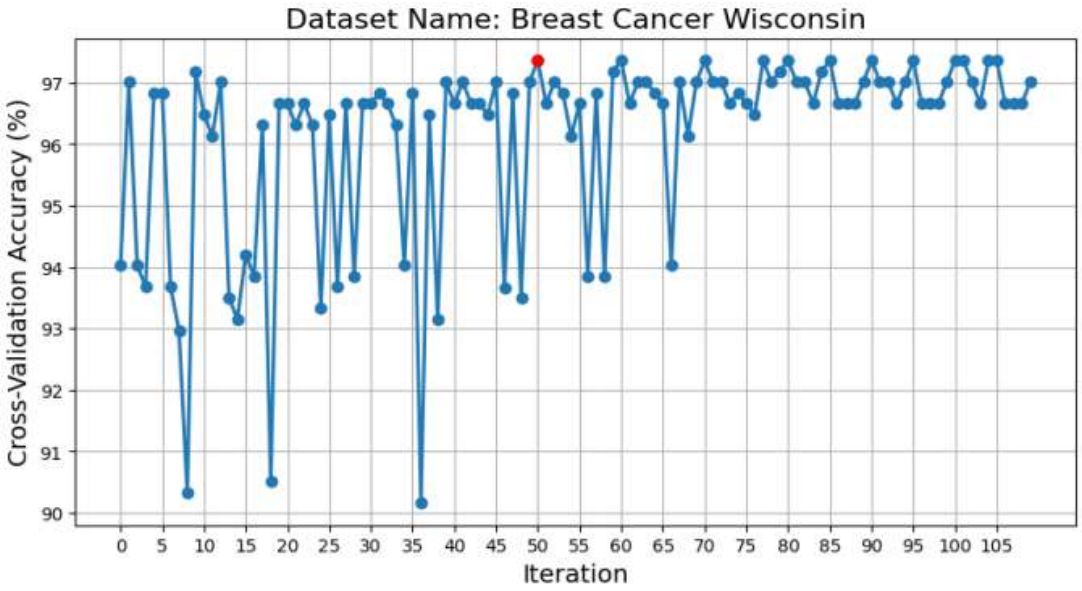


Figure 2: *Cross-validation accuracy of Logistic regression classifier vs PSO iteration.*

Table 4: *Feature reduction rate of PSO-based univariate feature selection approach on different datasets.*

Dataset Name	Feature Reduction Rate (LR) (%)	Feature Reduction Rate (KNN) (%)
UCI Heart Disease	23.08%	69.23%
Breast Cancer Wisconsin	3.33%	3.33%
Adult Census Income	42.86%	14.29%

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Tengku Mazlin Tengku Ab Hamid, Roselina Sallehuddin, Zuriahati Mohd Yunos, and Aida Ali. Ensemble based filter feature selection with harmonize particle swarm optimization and support vector machine for optimal cancer classification. Machine Learning with Applications, 5:100054, 2021.

[2] Dharyll Prince M Abellana and Demelo M Lao. A new univariate feature selection algorithm based on the best–worst multi-attribute decision-making method. Decision Analytics Journal, 7:100240, 2023.

[3] Esra'a Alhenawi, Rizik Al-Sayyed, Amjad Hudaib, and Seyedali Mirjalili. Feature selection methods on gene expression microarray data for cancer classification: A systematic review. Computers in Biology and Medicine, 140:105051, 2022.

[4] Theyab R Alsenani, Safial Islam Ayon, Sayeda Mayesha Yousuf, Fahad Bin Kamal Anik, and Mohammad Ehsan Shahmi Chowdhury. Intelligent feature selection model based on particle swarm optimization to detect phishing websites. Multimedia Tools and Applications, pages 1–33, 2023.

[5] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[6] Andrea Bommert, Thomas Welchowski, Matthias Schmid, and J¨org Rahnenf¨uhrer. Benchmark of filter methods for feature selection in high-dimensional gene expression survival data. Briefings in Bioinformatics, 23(1): bbab354, 2022.

[7] Mohammed Mehdi Bouchene and Kheireddine Abainia. Classical machine learning and transformer models for offensive and abusive language classification on dziri language. In 2023 International Conference on Decision Aid Sciences and Applications (DASA), pages 116–120. IEEE, 2023.

[8] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luj´an. Conditional likelihood maximisation: a unifying framework for information theoretic

feature selection. The journal of machine learning research, 13:27–66, 2012.

[9] Enrico Civitelli, Matteo Lapucci, Fabio Schoen, and Alessio Sortino. An effective procedure for feature subset selection in logistic regression based on information criteria. Computational Optimization and Applications, 80(1):1–32, 2021.

[10] M. Dash and H. Liu. Feature selection for clustering-a filter solution. ICDM, 1:115–122, 2000.

[11] Xuelian Deng, Yuqing Li, Jian Weng, and Jilian Zhang. Feature selection for text classification: A review. Multimedia Tools and Applications, 78:3797–3816, 2019.

12] Peter Drot´ar, Juraj Gazda, and Zdenek Sm´ekal. An experimental comparison of feature selection methods on two-class biomedical datasets. Computers in biology and medicine, 66:1–10, 2015.

[13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[14] R. A. Fisher. Iris. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C56C76.

[15] Ahmed G Gad. Particle swarm optimization algorithm and its applications: a systematic review. Archives of computational methods in engineering, 29(5):2531–2561, 2022.

[16] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh. Feature extraction: foundations and applications. 207, 2006. [17] Isabelle Guyon and Andr´e Elisseeff. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar):1157–1182, 2003.

[18] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. Machine learning, 46:389–422, 2002.

[19] Ramesh Kumar Huda and Haider Banka. Efficient feature selection methods using pso with fuzzy rough set as fitness function. Soft Computing, pages 1–21, 2022.

[20] Meetu Jain, Vibha Saihjpal, Narinder Singh, and Satya Bir Singh. An overview of variants and advancements of pso algorithm. Applied Sciences, 12(17):8392, 2022.

[21] Shivani Jain and Anju Saha. Rank-based univariate feature selection methods on machine learning classifiers for code smell detection. Evolutionary Intelligence, 15(1):609–638, 2022.