

THOMPSON RIVERS UNIVERSITY

**A new hybrid strategy for feature selection in classification
tasks based on the Four Vector Intelligent Metaheuristic and
the Black-Winged Kite Algorithm**

By

Thai Pham

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

Master of Science in Data Science

KAMLOOPS, BRITISH COLUMBIA

December, 2024

SUPERVISOR

Dr. Mohamed Tawhid

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Mohamed Tawhid, whose unwavering guidance, expert knowledge, and constant encouragement have been instrumental throughout this entire research process. Dr. Tawhid's dedication to my academic growth has been truly invaluable. His constructive feedback and insightful suggestions not only helped me shape the direction of this project but also provided me with the necessary tools to approach complex problems effectively. His patience in reviewing my work, as well as his ability to challenge my thinking in a constructive manner, has been a source of continuous motivation.

Dr. Tawhid's support went beyond mere academic mentorship; his thoughtful advice and timely interventions ensured that I remained focused and motivated throughout the course of this project. I am extremely grateful for his trust in my abilities and his willingness to invest his time and energy in guiding me to successfully complete this research.

I would also like to express my sincere thanks to Dr. Mridula Sharma, who graciously agreed to read and provide valuable feedback on my project. Her thoughtful comments and suggestions have greatly enriched the quality of my work, and I deeply appreciate the time and effort she dedicated to reviewing my research.

I am truly fortunate to have had such exceptional mentors, whose support has been crucial to the completion of this thesis. Their guidance, both academically and personally, has been beyond measure, and I will always be thankful for their contributions to my academic journey.

ABSTRACT

Feature selection is essential in classification tasks, particularly for large-scale datasets, as traditional methods often struggle to balance exploration and exploitation, limiting their capacity to identify optimal feature subsets in high-dimensional spaces. To address these limitations, this paper introduces a hybrid algorithm, named Four Vector - Black winged Kite (FVBK), which combines the migration behavior of the Black-Winged Kite Algorithm (BKA) with the position update mechanism of the Four Vector Intelligent Metaheuristic (FVIM). The proposed algorithm incorporates a Memory Buffer to effectively store promising solutions and employs a minimum redundancy maximum relevance criterion as its fitness function. Operating within an innovative filter/filter framework, FVBK is sequentially combined with filter algorithms, including Mutual Information, Fisher Score, ReliefF, and Chi-squared tests, to select the most informative features. FVBK's performance is evaluated across five classifiers: Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), Naive Bayes (NB), k-Nearest Neighbors (kNN), and Classification and Regression Trees (CART). Testing on nine datasets across biological, textual, and imaging domains demonstrates that FVBK consistently enhances classification accuracy, effectively balancing feature relevance and redundancy to improve model performance.

Key Words: Feature selection, Classification, Hybrid algorithm, Large-scale data

Contents

1	Introduction	1
2	Literature Review	7
3	Methodology	13
1	Standard algorithms	13
1.1	Four Vector Intelligence Metaheuristic (FVIM)	13
1.2	Black Winged Kite Algorithm (BKA)	18
2	Proposed hybrid algorithm	22
2.1	Feature subset evaluation criteria	22
2.2	Fitness function: Minimum redundancy maximum relevance (mRMR) .	23
2.3	Solution representation	25
2.4	Integration of Memory Buffer	25
2.5	Hybrid Algorithm Structure	26

2.6	Framework for Feature Selection	30
4	Experiments and discussion	32
1	Statistical analysis	32
2	Overall performance	57
5	Conclusion and future work	61

List of Figures

1.1	The basic architecture of Feature subset Selection	2
1.2	A taxonomy of feature selection	3
2.1	Evaluation metrics by category	8
2.2	Number of classifiers by year	8
2.3	Classifier metrics by year	8
2.4	Metaheuristic metrics by year	8
2.5	Classifier distribution	9
3.1	Flow chart of the FVIM algorithm	17
3.2	Attack strategies of Black-winged kites	18
3.3	The strategic changes of Black-winged kites during migration	21
3.4	Flowchart illustrating the process for feature selection using a hybrid approach	31
4.1	Accuracy of different algorithms by filters for the SVM classifier	37

4.2	Accuracy of different algorithms by filters for the LDA classifier	40
4.3	Accuracy of different algorithms by filters for the NB classifier	43
4.4	Accuracy of different algorithms by filters for the kNN classifier	46
4.5	Accuracy of different algorithms by filters for the CART classifier	49
4.6	Overall accuracy of different algorithms by filters	52
4.7	The Convergence Curve of FVBK across datasets	57

List of Tables

1.1	Advantages and Disadvantages of various Feature Selection methods	5
3.1	Hybrid algorithm FVBK pseudocode	28
3.2	Baseline algorithm pseudocode	29
4.1	Datasets Description	33
4.2	Parameter settings of classifiers for each dataset	33
4.3	Best accuracy achieved by algorithms with an SVM classifier using various filters	34
4.3	Best accuracy achieved by algorithms with an SVM classifier using various filters	35
4.3	Best accuracy achieved by algorithms with an SVM classifier using various filters	36
4.4	Best accuracy achieved by algorithms with an LDA classifier using various filters	37
4.4	Best accuracy achieved by algorithms with an LDA classifier using various filters	38
4.4	Best accuracy achieved by algorithms with an LDA classifier using various filters	39
4.5	Best accuracy achieved by algorithms with an NB classifier using various filters	40

4.5	Best accuracy achieved by algorithms with an NB classifier using various filters	41
4.5	Best accuracy achieved by algorithms with an NB classifier using various filters	42
4.6	Best accuracy achieved by algorithms with an kNN classifier using various filters	43
4.6	Best accuracy achieved by algorithms with an kNN classifier using various filters	44
4.6	Best accuracy achieved by algorithms with an kNN classifier using various filters	45
4.7	Best accuracy achieved by algorithms with an CART classifier using various filters	46
4.7	Best accuracy achieved by algorithms with an CART classifier using various filters	47
4.7	Best accuracy achieved by algorithms with an CART classifier using various filters	48
4.8	Average accuracy achieved by algorithms by filters	49
4.8	Average accuracy achieved by algorithms by filters	50
4.8	Average accuracy achieved by algorithms by filters	51
4.9	Classification accuracy results of FVBK	53
4.9	Classification accuracy results of FVBK	54
4.10	Classification accuracy results of BKVP	54
4.10	Classification accuracy results of BKVP	55
4.10	Classification accuracy results of BKVP	56
4.11	Wilcoxon signed-rank test between FVBK and FVIM	58
4.12	Wilcoxon signed-rank test between FVBK and BKA	58

4.13 Ranked by Friedman test based on the Best Fitness Value	59
4.14 Ranked algorithms by Friedman test	60

Chapter 1

Introduction

Classification is a fundamental task in machine learning, where algorithms are employed to derive insights and knowledge from a given dataset. As a form of supervised learning, the primary goal of classification is to develop a decision model that identifies patterns within the training data, enabling accurate predictions of class labels for previously unseen data. These class labels can be binary, such as distinguishing between spam and non-spam emails, or multiclass, such as identifying different species of flowers. Due to its wide-ranging applications and significance, classification remains a well-researched and pivotal topic in the fields of data mining and machine learning. The development of robust classification models is essential for tasks ranging from medical diagnosis to natural language processing.

However, as the volume of data continues to grow exponentially, the quality of data available for processing in classification tasks can deteriorate due to the presence of noisy, irrelevant, and redundant features. High-dimensional datasets not only introduce complexity but can also lead to overfitting and increased error rates in learning algorithms. This challenge highlights the importance of dimensionality reduction techniques, particularly Feature Selection (FS), during the preprocessing phase. By selecting a subset of relevant features, FS reduces dimensionality,

improves model performance, and enhances interpretability by eliminating unnecessary data. This process is crucial in classification tasks across various domains such as bioinformatics, text mining, and image recognition, where large-scale datasets often contain thousands of features. Implementing FS is vital for building efficient and accurate classification models that can handle the complexity of high-dimensional data.

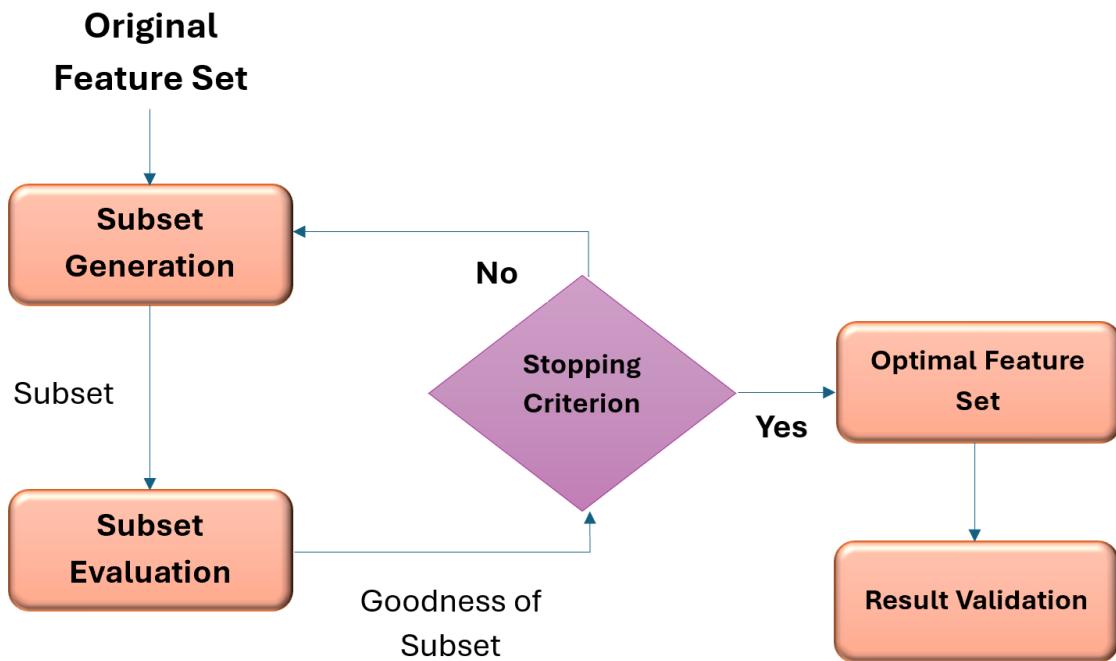


Figure 1.1: The basic architecture of Feature subset Selection

Various research efforts have proposed different techniques for subset evaluation and search, leading to a taxonomy of FS methods. Based on these criteria, feature selection can be classified into three categories: search strategy, evaluation criteria, and learning methods, as illustrated in figure 1.2. Specifically, based on evaluation criteria, FS methods are further divided into four main categories: filter methods, wrapper methods, embedded methods, and hybrid methods. Filter methods rely on statistical techniques to evaluate the relevance of each feature independently of any learning algorithm. They are computationally efficient and include tech-

niques such as correlation coefficients, mutual information, and chi-square tests. Wrapper methods, on the other hand, assess the performance of feature subsets by using a specific machine learning algorithm. These methods are more computationally intensive but often yield better performance since they consider feature interactions. Examples include recursive feature elimination, sequential forward selection, exhaustive selection, and metaheuristics. Embedded methods integrate feature selection within the training process of the algorithm itself. Regularization techniques like LASSO and decision tree-based methods like Random Forest are common examples of embedded methods. Hybrid methods combine elements of both filter and wrapper approaches to leverage the advantages of each. By initially using filter techniques to reduce the feature space and then applying wrapper methods to refine the selection, hybrid methods aim to balance computational efficiency with high-quality feature subsets.

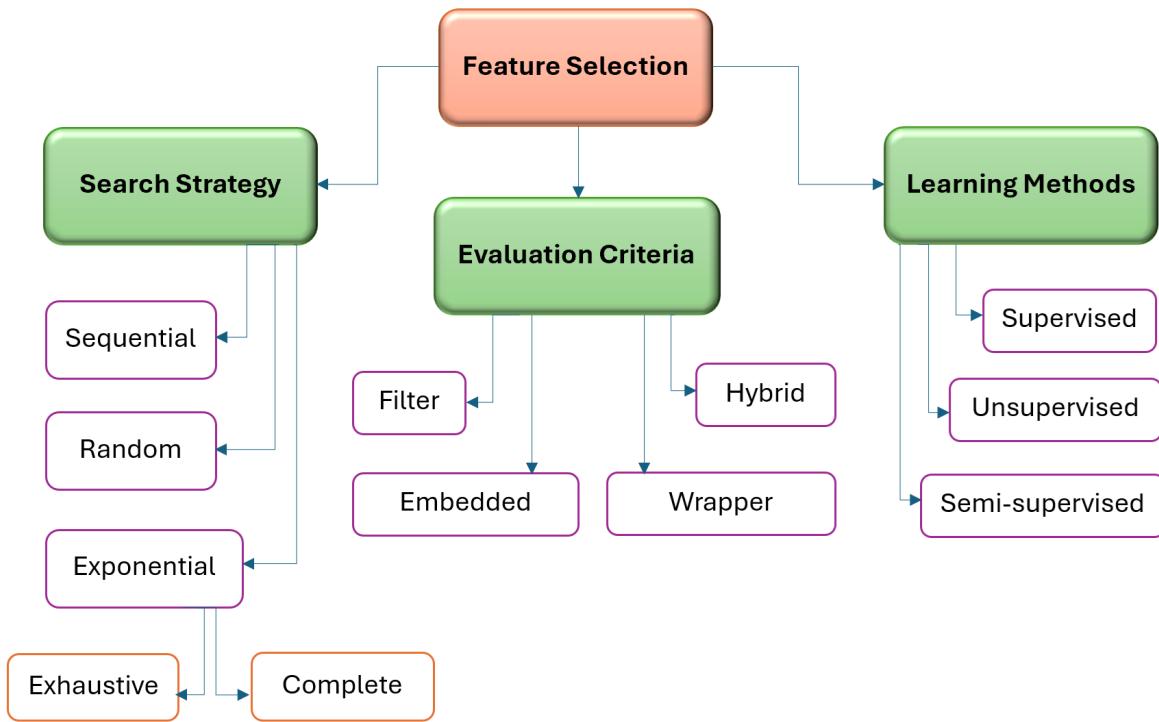


Figure 1.2: A taxonomy of feature selection

In the realm of metaheuristic optimization, the "No Free Lunch" theorem highlights that

no single algorithm universally excels across all types of problems. According to this theorem, the performance of any given algorithm is problem-specific, meaning that an algorithm that performs well on one class of problems might not necessarily perform well on another. This principle underscores the need for tailored approaches when solving complex optimization tasks such as feature selection. In light of this, numerous metaheuristic techniques have been employed to tackle feature selection challenges. Researchers face difficulties in effectively implementing and recommending modern metaheuristics because each algorithm has its own strengths and weaknesses. As a result, recent studies have increasingly focused on hybrid metaheuristics to enhance performance, adaptability, and efficiency by integrating the strengths of different metaheuristic algorithms. This hybrid approach increases the probability of identifying high-quality feature subsets. For instance, a hybrid approach might first employ statistical metrics, such as mutual information, correlation coefficients, or Fisher score, alongside statistical tests like the Chi-Squared test and ReliefF, to eliminate irrelevant features. Following this, a metaheuristic algorithm, such as Genetic Algorithm or Particle Swarm Optimization, is used to further refine the feature subset based on the performance of a classification model. This two-phase strategy combines the efficiency of filter methods with the precision of wrapper methods, leading to enhanced classification performance and reduced computational costs. Author Aziz et al. [2017] summarized the advantages and disadvantages of various feature selection methods, as presented in Table 1.1.

Building on established feature selection methodologies, this study emphasizes the integration of two advanced metaheuristic algorithms: the Four Vector Intelligent Metaheuristic (FVIM), developed by Fakhouri et al. [2024a], and the Black-winged Kite Algorithm (BKA), introduced by Wang et al. [2024]. FVIM is an advanced optimization algorithm designed to overcome the limitations of traditional Particle Swarm Optimization (PSO), particularly issues related to premature convergence. Unlike PSO, which updates particle positions based on a single global best value, FVIM employs a more sophisticated approach by integrating four distinct solution vectors. This method enhances both exploration and exploitation phases, significantly

Table 1.1: Advantages and Disadvantages of various Feature Selection methods

Model search	Advantages	Disadvantages	Examples
Filter methods			
a. Univariate	Fast, Scalable Independence of classifier	Overlooks feature relationships Ignores group feature significance Evaluates features independently	χ^2 test, t-test Euclidean distance Information gain
b. Multivariate	Accounts for feature relationships Classifier-agnostic Lower computational cost than wrapper methods	Slower than univariate methods Less scalable than univariate Ignores classifier interaction Includes redundant features	Correlation-based feature selection (CFS) Markov blanket filter (MBF) Fast correlation-based feature selection (FCBF)
Wrapper methods			
a. Sequential selection algorithms	Engages with the classifier Low overfitting risk Computational efficiency Susceptible to local optima Accounts for feature dependence	Higher susceptibility than randomized algorithms Local optimum trap (greedy search) Classifier-dependent approaches Suboptimal solutions	Sequential forward selection (SFS) Sequential backward elimination (SBE) Beam search
b. Evolutionary selection algorithms	Lower susceptibility to local optima Engages with the classifier Accounts for feature dependencies Better performance accuracy than filter methods	High computational cost Discriminative ability Shorter training times Classifier-dependent selection Greater overfitting risk compared to deterministic algorithms	Simulated annealing Randomized hill climbing Genetic algorithms Ant Colony Optimization Rough set methods Particle Swarm Optimization
Embedded methods			
	Engages with the classifier Better at modeling feature dependencies More efficient computationally compared to wrapper methods Superior performance and accuracy compared to filter methods Lower overfitting risk compared to wrapper methods Maintains data characteristics for interpretability	Selection based on classifier Accounts for feature dependencies	Decision trees Weighted naive Bayes Feature selection using the weight vector of SVM Random forests Least absolute shrinkage and selection operator (LASSO)
Hybrid methods			
	Superior performance and accuracy compared to filter methods More efficient computationally than wrapper methods More adaptable and robust with high-dimensional data	Methods specific to classifiers Dependent on the combination of various feature selection techniques	Recursive Feature Elimination with Cross-Validation (RFEcv) Combining Chi-squared Test and Sequential Forward Selection (SFS) Combining Lasso Regression and Recursive Feature Elimination (RFE) Combining Correlation-Based Filtering with Decision Trees

broadening the search space and improving the algorithm's ability to locate the global optimum. By balancing these two crucial aspects, FVIM effectively addresses convergence issues and enhances the optimization process, making it a powerful tool for complex problem-solving in various domains. The Black-winged Kite Algorithm (BKA) is a cutting-edge optimization method inspired by the hunting behavior of black-winged kites. This algorithm simulates the dynamic and adaptive strategies used by these birds in locating prey, employing a combination of exploration and exploitation techniques to find optimal solutions. BKA incorporates unique movement patterns and decision-making processes, enabling it to effectively navigate complex search spaces. Its ability to adaptively adjust its search behavior allows it to escape local optima and enhance convergence towards global optima. The BKA algorithm's innovative approach makes it a versatile and efficient tool for tackling diverse optimization challenges across various fields.

Leveraging the strengths of both FVIM and BKA, this study introduces a hybrid algorithm named Four Vector - Black winged Kite (FVBK). The FVBK combines FVIM's robust mechanisms for exploration and exploitation with BKA's effective search strategies to tackle feature selection problems in classification. This hybrid approach is designed to capitalize on FVIM's ability to explore various solution space regions and BKA's efficient convergence techniques. The performance of the FVIM and BKA algorithms, before being combined into the hybrid algorithm FVBK, has been thoroughly evaluated using various benchmark test suites: CEC-BC-2017, CEC-2013 LGSO, CEC2006 and 13 well-known Engineering problems. These benchmarks help assess the algorithms' efficacy in various optimization scenarios. Following this evaluation, the hybrid algorithm will be applied to feature selection for classification tasks. In the first stage of the feature selection process, statistical metrics such as Mutual Information, Fisher Score, Chi-Squared Test, and ReliefF will be employed to filter out and select elite features from the dataset. This step aims to identify and retain the most relevant features, enhancing the subsequent optimization phase. In the second stage, the FVBK algorithm will be used to further refine the selected features, optimizing them based on the Minimum Redundancy Maximum Relevance (mRMR) criterion. The goal is to maximize the relevance of the features while minimizing redundancy, ultimately improving the classification performance. The effectiveness of the selected features and the hybrid algorithm will be evaluated using five different classifiers: Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), Naive Bayes (NB), k-Nearest Neighbors (kNN), and Classification and Regression Trees (CART). Performance will be assessed based on accuracy metrics to determine how well the feature selection process enhances classification outcomes.

Chapter 2

Literature Review

Feature selection remains a critical area of research in machine learning, particularly for classification tasks. Recent advancements focus on developing hybrid methods that combine various strategies to enhance the selection process, especially in handling high-dimensional data. In a study of Barrera-García et al. [2024], a comprehensive statistical analysis was conducted to evaluate the use of various classifiers in the context of feature selection. They have compiled and analyzed a comprehensive set of statistics, including: evaluation metrics categorized by type (figure 2.1), which provide insights into the performance and effectiveness of different classifiers; the number of classifiers employed each year (figure 2.2), illustrating trends and shifts in usage over time; classifier performance metrics by year (figure 2.3), offering a temporal perspective on their effectiveness; metaheuristic metrics by year (figure 2.4), which shed light on the performance and evolution of optimization techniques; and the distribution of classifiers (figure 2.5), detailing their application across various contexts and datasets. From the statistics, we can discern a trend towards the growing adoption and evolution of classifiers, as evidenced by shifts in evaluation metrics by category, the annual number and performance metrics of classifiers, changes in metaheuristic metrics, and variations in classifier distribution over time.

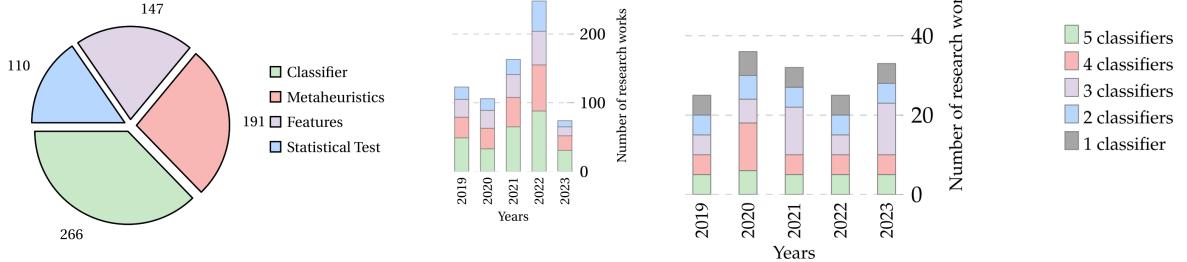


Figure 2.1: Evaluation metrics by category

Figure 2.2: Number of classifiers by year

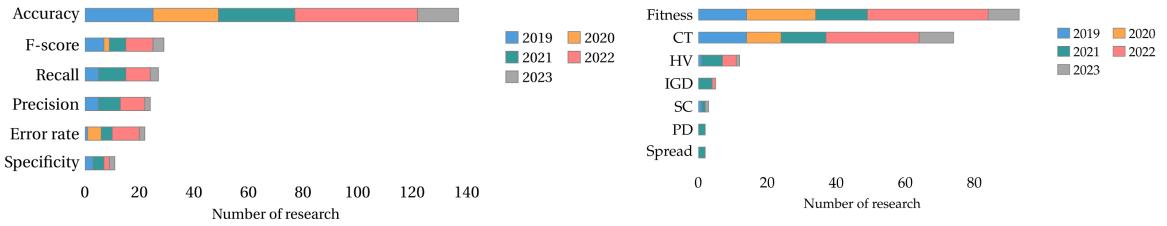


Figure 2.3: Classifier metrics by year

Figure 2.4: Metaheuristic metrics by year

Xue et al. [2013] explored the application of Particle Swarm Optimization (PSO) for feature selection in classification tasks. Their study demonstrated PSO's capability to balance exploration and exploitation, making it effective in navigating the feature space and identifying optimal subsets. This approach showed significant improvements in classification accuracy, particularly for high-dimensional datasets. Sabbah et al. [2018] proposed a machine learning-based feature ranking and selection method known as Support Vector Machine-based Feature Ranking Method (SVM-FRM). The method utilizes Support Vector Machine (SVM) to weight and select significant features, aiming to enhance classification performance. The paper also investigates hybridization techniques to further improve the SVM-FRM method's performance. Experimental results on three public text classification datasets showed that SVM-FRM achieved superior F-measure and accuracy compared to existing statistical feature selection methods, particularly on balanced datasets. Hybridization techniques further enhanced performance on unbalanced datasets. Alhassan and Zainon [2021] reviewed the ap-

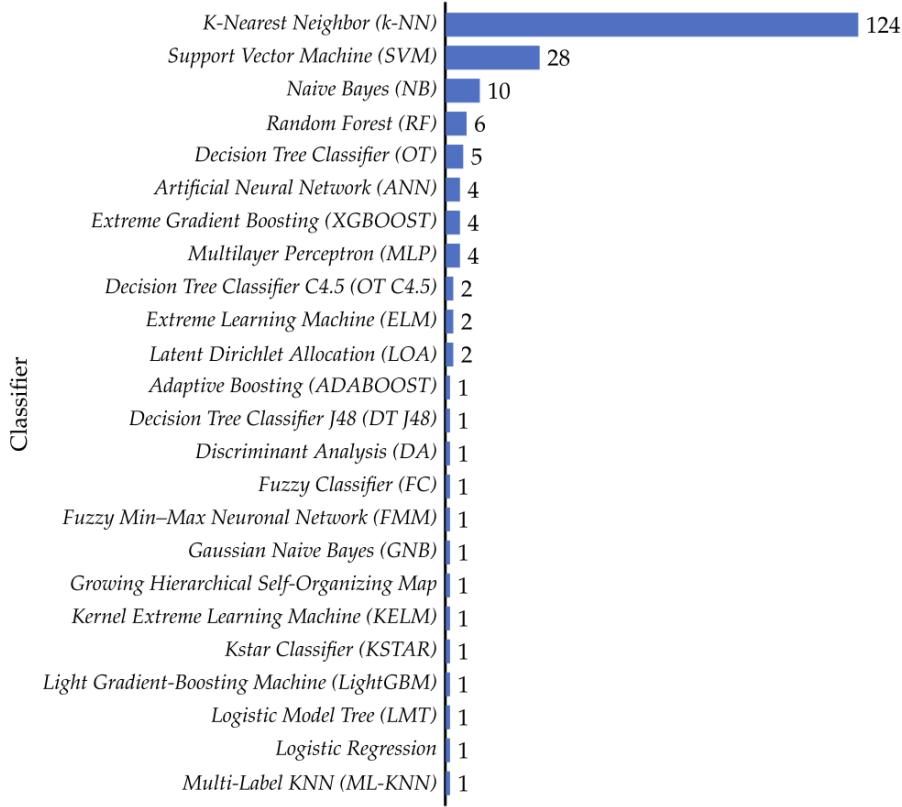


Figure 2.5: Classifier distribution

plication of feature selection, dimensionality reduction, and classification techniques in chronic disease prediction and diagnosis. Their work emphasized the importance of attribute selection and dimensionality reduction in enhancing classification accuracy and improving the overall performance of diagnostic systems. The study discussed various parallel and adaptive classification techniques to address computational costs and time, offering a comprehensive overview of these methods and their effectiveness in healthcare applications. Akinola et al. [2022] introduced a hybrid algorithm called Binary Dwarf Mongoose Optimization Algorithm with Simulated Annealing (BDMSAO), designed to enhance feature selection for high-dimensional multi-class datasets. By combining the global search capability of the binary variant of the Dwarf Mongoose Optimization (BDMO) with the local search efficiency of Simulated Anneal-

ing (SA), the BDMSAO overcomes the limitations of the original DMO’s exploitation phase. The algorithm was tested on 18 UCI datasets and three high-dimensional medical datasets, where it outperformed ten other methods, achieving high classification accuracy and effectively selecting relevant features. Yadav et al. [2023] explored the prediction of chronic diseases using machine learning models, focusing on datasets from Kaggle and two unseen real datasets. The authors addressed class imbalance with the Synthetic Minority Over-Sampling Technique (SMOTE) and used Boruta for feature selection. They introduced an innovative approach by combining Grid Search with Grey Wolf Optimization for hyper-parameter tuning. After evaluating various classifiers, the Stacking Classifier demonstrated superior performance. The results show that their method outperforms existing approaches, particularly on the PIMA dataset, and the study also discusses the ethical implications of applying explainable machine learning models in clinical practice. Al-Tashi et al. [2019] introduced a binary version of the hybrid Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO) algorithm, named BGWOPSO, specifically designed for feature selection tasks. This approach leverages the exploration capabilities of GWO and the exploitation strengths of PSO, adapting the original continuous search hybrid PSOGWO to suit the binary nature of feature selection problems. The authors employed a wrapper-based method, utilizing the K-nearest neighbors classifier with Euclidean distance, to evaluate potential feature subsets. Wajih et al. [2021] presented a hybrid approach, HBGWOHHO, which combines Binary Grey Wolf Optimizer (BGWO) with Harris Hawks Optimization (HHO) to improve feature selection in classification tasks. By using a sigmoid transfer function, the method converts the continuous search space into a binary format suitable for feature selection. A wrapper-based k-Nearest Neighbor technique evaluates the effectiveness of the selected features. The results demonstrate that HBGWOHHO outperforms the BGWO algorithm, showing enhancements in accuracy, a reduction in the number of selected features, and lower computational time. Additionally, it surpasses Binary Particle Swarm Optimization (BPSO), Binary Harris Hawks Optimizer (BHHO), Binary Genetic Algorithm (BGA), and Binary Hybrid BWOPSO in terms of accuracy and efficiency. Alyasiri et al. [2022] explored various metaheuristic algorithms used for

feature selection (FS) in text classification, addressing key questions about their effectiveness. It examines different FS approaches utilizing metaheuristic methods, compares their accuracy with traditional FS techniques, evaluates the performance of modified and hybrid algorithms, and identifies gaps in current research. The review of thirty-seven relevant studies reveals that hybrid FS approaches, which integrate multiple optimization strategies, significantly enhance performance in complex FS problems. The findings suggest that future research should focus on these hybrid methods to further improve text classification outcomes and explore new research opportunities in this area. Kumar and Phogat [2023] presented the Improved Binary Competitive Swarm Optimization Whale Optimization Algorithm (IBCSOWOA) for cancer classification using DNA microarray data. To address the challenges of high-dimensional gene datasets and overfitting, the technique integrates the Improved Binary Competitive Swarm Optimization (IBCSO) with Whale Optimization Algorithm (WOA). IBCSO is utilized to select an informative gene subset based on minimum redundancy maximum relevance (mRMR), and WOA is employed for tuning the parameters of an artificial neural network (ANN) model. Alyasiri et al. [2021] focused on feature selection for text classification by combining the Information Gain (IG) filter approach with the Gray Wolf Optimizer (GWO) search strategies. The study aims to first identify the top-N features with the highest relevance using IG, and then refine this feature set by applying GWO to find the most informative subset. The effectiveness of the proposed IG-GWO method was evaluated using a Naive Bayes (NB) classifier across nine benchmark document datasets. The results demonstrate that IG-GWO performs well in addressing the text feature selection problem, showing promise as an alternative to existing state-of-the-art feature selection algorithms.

The historical development of hybrid metaheuristic algorithms for feature selection in classification tasks has seen significant evolution. Initially, feature selection methods focused on traditional techniques like filter, wrapper, and embedded methods, which often struggled with high-dimensional data and overfitting. As the complexity of data and models increased, researchers began integrating metaheuristic algorithms, such as Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), to enhance feature

selection performance. The introduction of hybrid metaheuristic approaches marked a notable advancement. These methods combine the strengths of multiple metaheuristics to improve search efficiency and solution quality. For instance, hybridizing GA with PSO or incorporating local search strategies has led to more effective and computationally efficient feature selection processes. These hybrid approaches leverage the exploration capabilities of one algorithm with the exploitation strengths of another, addressing the limitations of individual methods. Recent advancements have focused on integrating newer metaheuristic algorithms, such as the Grey Wolf Optimizer (GWO) and Harris Hawks Optimization (HHO), with traditional techniques to further refine feature selection. The use of hybrid metaheuristics has shown promising results in reducing dimensionality, enhancing classification accuracy, and mitigating issues related to overfitting and high computational costs. This ongoing evolution underscores the growing sophistication and effectiveness of hybrid metaheuristic algorithms in tackling complex feature selection challenges in classification tasks.

Chapter 3

Methodology

The methodology of this research focuses on developing and evaluating a hybrid strategy that combines the Four Vector Intelligent Metaheuristic (FVIM) and Black-winged Kite Algorithm (BKA) for enhanced feature selection in classification tasks.

1 Standard algorithms

1.1 Four Vector Intelligence Metaheuristic (FVIM)

Proposed by Fakhouri et al. [2024a], the FVIM algorithm was developed to tackle issues of premature convergence in traditional swarm-based algorithm such as Particle Swarm Optimization (PSO). FVIM is designed to maintain a balance between exploration and exploitation across the search space, which increases its likelihood of reaching a global optimum. The algorithm is built on a mathematical framework that incorporates diverse agent behaviors and adaptively adjusts their movements to prevent early convergence—a frequent challenge in many optimization algorithms. The FVIM optimization process is structured into three main stages

that direct its operation. Initialization Phase: Set key parameters and randomly initialize agent positions, defining the upper and lower boundaries of the search space. Iteration Phase: Adjust agent positions using four distinct mathematical models, repeating the process until the stopping criteria are fulfilled. Optimal Solution Identification Phase: Determine the optimal solution by assessing the fitness of each agent's position and comparing it against the objective function. FVIM utilizes four leading agents to steer the search process, representing the top candidate solutions within the population and dynamically adjusting them at each iteration. These agents correspond to the four best solutions at any point: Leader 1: The highest-performing solution so far. Leader 2: The second most effective solution. Leader 3: The third top solution. Leader 4: The fourth leading solution. The "Update Solution Hierarchy" algorithm, which systematically updates the ranking of solution values and their associated positions, is presented in 1.

Algorithm 1 Update Solution Hierarchy

```

if fitness < fstVal then
    | [frthVal, thrdVal, sndVal, fstVal] ← [thrdVal, sndVal, fstVal, fitness]
    | [frthPos, thrdPos, sndPos, fstPos] ← [thrdPos, sndPos, fstPos, position]
end

else if fitness < sndVal then
    | [frthVal, thrdVal, sndVal] ← [thrdVal, sndVal, fitness] [frthPos, thrdPos, sndPos] ←
    | [thrdPos, sndPos, position]
end

else if fitness < thrdVal then
    | [frthVal, thrdVal] ← [thrdVal, fitness] [frthPos, thrdPos] ← [thrdPos, position]
end

else if fitness < frthVal then
    | frthVal ← fitness frthPos ← position
end

```

This algorithm takes a fitness value and its corresponding position as input and compares

this fitness value to the top four hierarchy values in the solution set. Depending on where the input value ranks among the top values, the algorithm shifts lower-ranking values down the hierarchy to maintain the order from best to fourth-best. In this way, it ensures the solution hierarchy remains up-to-date with the best available solutions at each level. These agents, actual members of the population, are updated according to their fitness values. They play a crucial role in influencing the position adjustments of other particles, guiding the search towards promising areas within the solution space. This multi-vector structure allows FVIM to conduct a more effective exploration compared to traditional single-point search methods. The functionality of FVIM relies heavily on the mechanism of its four guiding vectors. These vectors, representing the top-performing solutions, dynamically adjust according to their fitness values to navigate the search space effectively. Each vector leads a specific portion of the population, fostering a diverse exploration strategy. The lead vector directs the search, followed in sequence by the remaining vectors, each exploring distinct areas within the solution space. This hierarchical model is essential for achieving a strategic balance between exploring new regions and focusing on promising ones. The detailed process is expressed in the equation 3.1 and equation 3.2. For $t = 1, 2, 3, 4$

$$X_{t,i} = \begin{cases} P_{t,i} + (\alpha \times 2 \times r_1 - \alpha) \times \|r_2 \times P_{t,i} - P_i\|, & \text{if } r_3 < 0.5 \\ P_{t,i} - (\alpha \times 2 \times r_1 - \alpha) \times \|r_2 \times P_{t,i} - P_i\|, & \text{otherwise} \end{cases} \quad (3.1)$$

$$\bar{P}_i = \frac{X_{1,i} + X_{2,i} + X_{3,i} + X_{4,i}}{4} \quad (3.2)$$

Where $X_{t,i}$ represents the updated position for the t^{th} best agent in the i -th dimension. $P_{t,i}$ is the current position of the t^{th} best agent in the i -th dimension. \bar{P}_i represents the current average position of all agents in the i -th dimension. α is an adaptive coefficient. r_1 , r_2 , and r_3 represent random numbers uniformly distributed in $[0, 1]$. Initially, the parameter α is set to a substantial value of 1.5 during the FVIM initialization phase to enable extensive exploration of the solution space. However, as FVIM converges towards the optimal solution, α progressively decreases to zero, thereby shifting the focus towards exploitation. This adjustment is described

by equation below:

$$\alpha = 1.5 - \left(\frac{4}{\text{Max_iter}} \right)$$

A core feature of FVIM is its adaptive search strategy. FVIM uses dynamic parameters that adjust each vector's influence on the search agents, allowing it to effectively explore various regions, whether sparsely or densely populated with potential solutions. Another crucial aspect is FVIM's balance between exploration and exploitation. Exploration is enhanced by the stochastic updates of the vectors, reducing the risk of premature convergence to local optima. Meanwhile, exploitation is achieved by focusing the search around the most promising solutions identified by the vectors. This balance is key to FVIM's ability to find high-quality solutions, significantly improving the optimization process's overall effectiveness. The following flow chart (figure 3.1) provides a visual representation of the FVIM algorithm. It outlines the key phases and steps that FVIM follows, from initialization to the iterative updates and final solution identification. Each stage in the flow chart highlights how FVIM utilizes dynamic agent behaviors and statistical metrics to achieve an optimal balance between exploration and exploitation, ensuring a thorough search of the solution space and avoiding premature convergence.

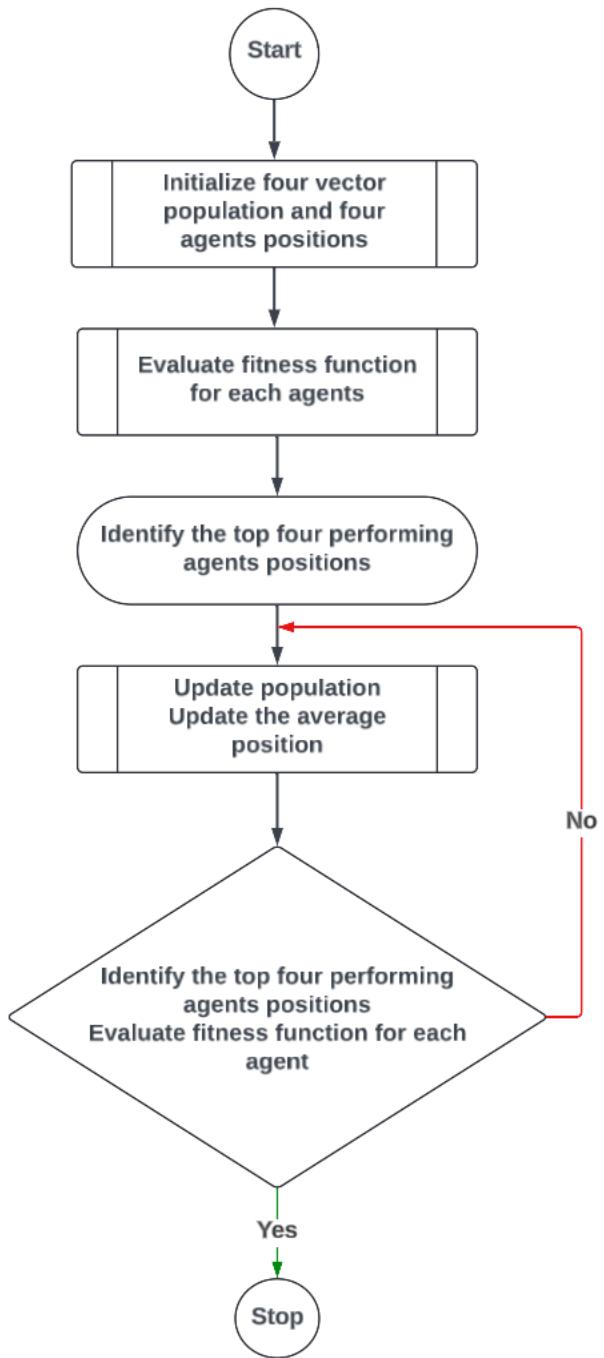


Figure 3.1: Flow chart of the FVIM algorithm

1.2 Black Winged Kite Algorithm (BKA)

Introduced by Wang et al. [2024], the Black Winged Kite Algorithm (BKA) is distinguished by its unique biological heuristic features, which emulate both the flight and predatory behaviors of black-winged kites, as well as their adaptability to environmental changes and target positions. This biological inspiration endows the algorithm with strong dynamic search capabilities, enabling it to handle evolving optimization environments effectively. Figure 3.2 illustrates two attack strategies of Black-winged kites.

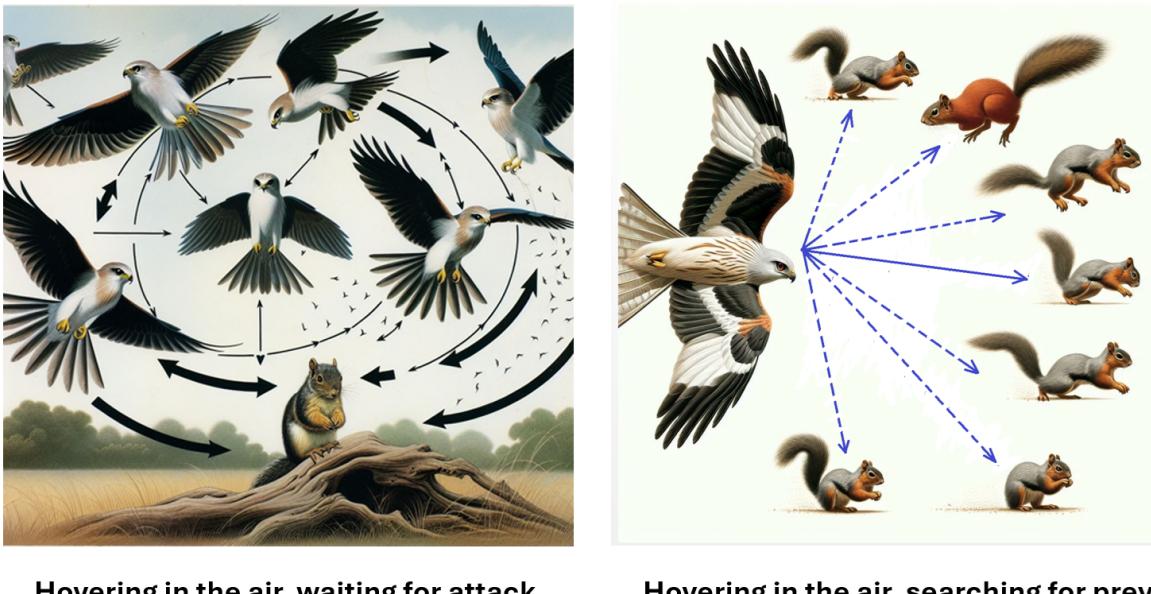


Figure 3.2: Attack strategies of Black-winged kites

The algorithm incorporates a Cauchy mutation strategy, which enhances its ability to escape local optima and increases the likelihood of finding better global solutions, particularly in high-dimensional optimization problems. Additionally, a leadership strategy is integrated, mirroring the role of leaders in a kite community, to optimize the use of the best solutions and guide the search direction. This approach improves the algorithm's efficiency in exploring the current search area while maintaining a balance between exploration and exploitation,

ensuring that promising new areas are not missed. This algorithm comprises three main phases: the Initialization phase, the Attacking Behavior, and the Migration Behavior. In the Initialization phase, the algorithm establishes the initial positions and parameters of the agents. The Attacking Behavior phase, as described by equation 3.3, models the aggressive interactions among agents to enhance their search capabilities by exploring diverse regions of the solution space. The Migration Behavior phase, represented by equation 3.4, enables agents to adjust to new positions based on their interactions and changes in the environment, thereby facilitating a dynamic adaptation of strategies to optimize overall performance.

$$v_{i,j}^{t+1} = \begin{cases} v_{i,j}^t + n(1 + \sin(\text{Rand})) \times v_{i,j}^t & \text{if } p < r \\ v_{i,j}^t + n \times (2\text{Rand} - 1) \times v_{i,j}^t & \text{else} \end{cases} \quad (3.3)$$

$$v_{i,j}^{t+1} = \begin{cases} v_{i,j}^t + C(0, 1) \times v_{i,j}^t, & \text{if } F_i < F_{ri} \\ v_{i,j}^t + C(0, 1) \times (L_j^t - L_j^t - m \times v_{i,j}^t), & \text{else} \end{cases} \quad (3.4)$$

The parameter n and m are defined as:

$$n = 0.05 \times e^{-2(\frac{t}{T})^2}$$

$$m = 2 \times \sin\left(\text{Rand} + \frac{\pi}{2}\right)$$

Where $v_{i,j}^t$ and $v_{i,j}^{t+1}$ represent the position of the i -th Black-winged kite in the j -th dimension at the t -th and $(t+1)$ -th iteration steps, respectively. The parameter Rand is a random number that ranges from 0 to 1, and p is a constant value of 0.9. T is the total number of iterations, and t is the number of iterations that have been completed so far. L_j^t is the leading scorer in the j -th dimension at the t -th iteration. $v_{i,j}^t$ and $y_{i,j}^{t+1}$ are the positions of the i -th kite in the j -th dimension at the t and $(t+1)$ -th iterations, respectively. F_i is the current position's fitness value. F_{ri} is the fitness value of a random position. $C(0, 1)$ is the Cauchy mutation.

The following pseudocode (2) outlines the Black-winged Kite Algorithm (BKA), emphasizing its key steps in navigating the search space to identify optimal solutions. This process begins by initializing the population within defined boundaries and evaluating their initial fitness.

In each iteration, the algorithm identifies the current best solution (leader) and applies two main behaviors—attacking and migration—to update each individual’s position. These behaviors leverage both the leader’s guidance and interactions with random individuals to enhance exploration and exploitation in the search space. Throughout the iterations, the algorithm continually updates the global best solution, ultimately yielding an optimized result and tracking the convergence trend.

Algorithm 2 Black-winged Kite Algorithm (BKA)

Input: Population size pop , maximum iterations T , lower bounds lb , upper bounds ub , dimensions dim , objective function $fobj$

Output: Best solution $Best_Fitness_BKA$, position $Best_Pos_BKA$, convergence history $Convergence_curve$

Initialize population positions $XPos$ within bounds $[lb, ub]$

Evaluate initial fitness $XFit$ for each individual in the population

for each iteration $t = 1$ **to** T **do**

- Identify** leader as the individual with best fitness $XLeader_Pos$
- for** each individual i in the population **do**

 - Attacking Behavior:**

 - Update position based on influence from leader $XLeader_Pos$

 - Migration Behavior:**

 - Adjust position using interactions with random individuals in population

 - Evaluate** new position and update if improvement is found

end

Update the global best solution $Best_Fitness_BKA$ if a better solution is found

Record current best fitness in $Convergence_curve$

end

Return $Best_Fitness_BKA$, $Best_Pos_BKA$, $Convergence_curve$

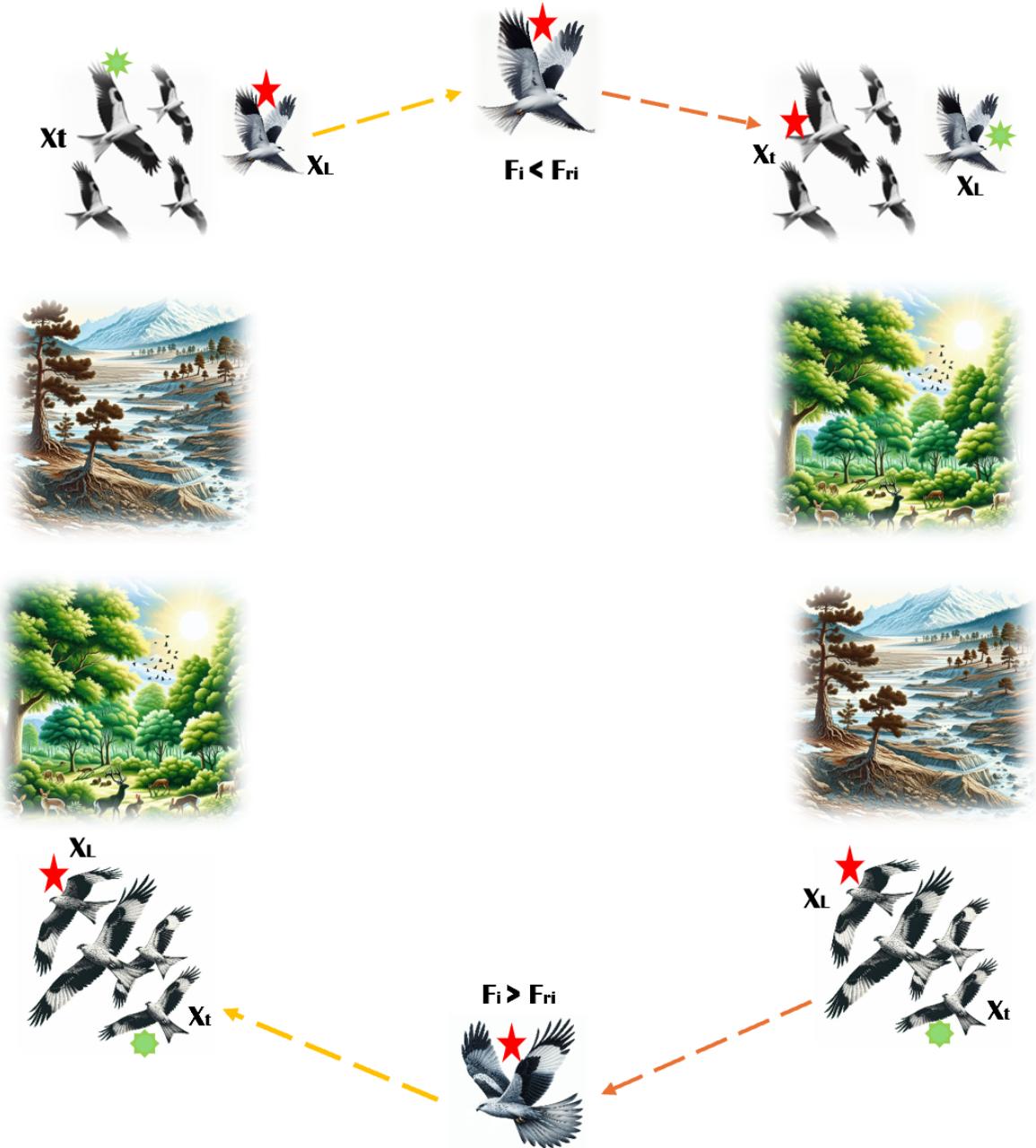


Figure 3.3: The strategic changes of Black-winged kites during migration

2 Proposed hybrid algorithm

2.1 Feature subset evaluation criteria

In this section, we utilize various statistical metrics to filter and identify elite features for classification tasks. Specifically, we employ Mutual Information, Fisher Score, ReliefF, and the Chi-squared test as our primary selection criteria.

Mutual Information (MI) is a measure of the interdependence between two variables, a and b . Equation 3.5 applies to discrete variables, while equation 3.5 is used for continuous variables, where $p(a, b)$ is the joint probability function of A and B . $p(a)$ and (p) are the marginal probability distribution functions of A and B . MI quantifies the amount of information one variable provides about the other, thereby capturing the extent of shared information between the two variables.

$$I(A, B) = \sum_{b \in B} \sum_{a \in A} p(a, b) \log \frac{p(a, b)}{p(a)p(b)} \quad (3.5)$$

$$I(A, B) = \int_A \int_B p(a, b) \log \frac{p(a, b)}{p(a)p(b)} da db \quad (3.6)$$

The Fisher score is a widely used supervised technique for feature selection that calculates individual scores for features across the data space. Although Fisher's criterion may not account for interactions between features or manage similar features effectively, it can offer optimal predictors under specific orthogonality conditions. The core concept behind the Fisher score is to maximize the separation between data points from different classes while minimizing the variation within the same class. The Fisher score can be computed using the following formula 3.7:

$$\text{Fisher Score}(x_i) = \frac{\sum_{c=1}^C n_c (\mu_c - \mu)^2}{\sum_{c=1}^C n_c \sigma_c^2} \quad (3.7)$$

The Chi-squared (χ^2) statistic assesses the independence between two variables by generating a score that reflects their level of independence. In feature selection, χ^2 is used to

measure the independence of features with respect to the class. The calculation assumes that the feature and class are independent. A higher χ^2 score suggests a stronger dependency between the feature and the class. The equation 3.8 represents the χ^2 statistic, where N signifies the complete dataset, r means the presence of the feature (r its absence), and c_i refers to the class, $P(r, c_i)$ is the probability that feature r occurs in class c_i , $P(r)$ is the likelihood that the feature resembles the dataset.

$$\chi^2(r, c_i) = \frac{N [P(r, c_i)P(\neg r, \neg c_i) - P(r, \neg c_i)P(\neg r, c_i)]^2}{P(r)P(\neg r)P(c_i)P(\neg c_i)} \quad (3.8)$$

The ReliefF algorithm is a feature selection method that ranks features based on how well they distinguish between instances that are near each other but belong to different classes. The mathematical formulation of the ReliefF algorithm involves updating the weight of each feature based on its ability to differentiate between nearest neighbors of the same class and different classes. For each feature A , ReliefF calculates a weight $W[A]$ that represents the importance of the feature according to the formula 3.9.

$$W[A] = W[A] - \frac{1}{m} \sum_{i=1}^m \left(\text{diff}(A, x_i, \text{nearestHit}(x_i)) - \sum_{c \neq y_i} \frac{P(c)}{1 - P(y_i)} \text{diff}(A, x_i, \text{nearestMiss}_c(x_i)) \right) \quad (3.9)$$

where the weight of feature A is denoted by $W[A]$. The number of iterations or sampled instances is represented by m . The i -th instance in the dataset is denoted by x_i , and its class label is y_i . The function $\text{nearestHit}(x_i)$ finds the nearest neighbor of x_i that belongs to the same class y_i . The function $\text{nearestMiss}_c(x_i)$ identifies the nearest neighbor of x_i from class c (where $c \neq y_i$). The function $\text{diff}(A, x_1, x_2)$ measures the difference in feature A between instances x_1 and x_2 . Finally, $P(c)$ denotes the prior probability of class c .

2.2 Fitness function: Minimum redundancy maximum relevance (mRMR)

Minimum Redundancy Maximum Relevance (mRMR) is a widely used feature selection method that aims to identify a subset of features that are both highly relevant to the target variable

and minimally redundant with respect to each other. The "maximum relevance" part ensures that the selected features are strongly correlated with the target variable, while the "minimum redundancy" part minimizes the redundancy between the selected features, ensuring that each feature adds unique information to the model. By balancing relevance and redundancy, mRMR helps in selecting a compact and informative set of features that can improve the performance of machine learning models. In our work, we have chosen mRMR as the fitness function for feature selection, as it effectively captures the trade-off between selecting highly relevant features and avoiding redundancy, leading to more robust and generalizable models.

Dataset features can be grouped into three categories: strongly relevant, weakly relevant, and irrelevant features. Strongly relevant features are critical and should always be part of the optimal subset. Weakly relevant features, however, are not always essential but may be needed under specific conditions to form the optimal feature subset (Salesi et al. [2021]). Given two random variables X and Y , their mutual information $I(X; Y)$ quantifies the amount of information shared between them. It is computed using their joint probability density function $P_{X,Y}(x, y)$ and their marginal probability functions $P_X(x)$ and $P_Y(y)$ as in equation 3.10

$$I(X; Y) = \int \int P_{X,Y}(x, y) \log \left(\frac{P_{X,Y}(x, y)}{P_X(x)P_Y(y)} \right) dx dy \quad (3.10)$$

To evaluate the information characteristics of a feature subset, relevance and redundancy criteria are utilized. Specifically, the relevance of a feature subset S is defined by:

$$\text{Rel} = \frac{1}{|S|} \sum_{x_i \in S} I(x_i; C)$$

where $|S|$ represents the number of features in the subset S , and $I(x_i; C)$ denotes the mutual information between the target class C and the i -th feature in subset S . The objective is to maximize the relevance Rel, which may lead to high inter-feature dependency (i.e., redundancy). Moreover, if two features are highly dependent, removing one from the subset S does not affect the class-discriminative capability. Therefore, the redundancy of a feature subset S

is defined as:

$$\text{Red} = \frac{1}{|S|^2} \sum_{i \neq j} I(x_i; x_j)$$

In this context, $I(x_i; x_j)$ represents the mutual information between the i -th and j -th features within the subset S . The goal of feature selection is to identify a subset S of N features that maximizes their collective relevance to the target class C while minimizing the redundancy among the features themselves. This approach results in a bi-criteria feature selection objective known as the Minimum Redundancy Maximum Relevance (mRMR) criterion, which is defined as:

$$\text{mRMR} = \text{Rel} - \text{Red}$$

2.3 Solution representation

In this study, integer-encoded solution representation is utilized as a method for encoding candidate solutions in discrete optimization problems. This technique involves representing each solution as a vector of integers, where each integer corresponds to a specific feature or decision variable. For feature selection tasks, integer encoding can be particularly effective. Given a dataset D with N features and a desired subset cardinality P (where $1 \leq P < N$), each feature is assigned a unique identifier ranging from 1 to N . To generate solutions, random vectors containing all N feature IDs are created. From each vector, the first P features are selected as the chosen subset. This approach ensures that each solution is defined by a distinct combination of features based on their IDs, facilitating the exploration of different feature subsets within the dataset.

2.4 Integration of Memory Buffer

We opted to integrate a memory buffer to enhance the efficiency of data management and processing. The memory buffer functions as a temporary storage area, facilitating the transfer

and processing of data. By leveraging this buffer, we enhance both performance and efficiency in data handling operations. Specifically, each updated solution is initially validated against the memory buffer. This buffer retains previously processed solutions and intermediate results. When a new solution is introduced, it is compared with the existing entries in the memory buffer to assess its novelty and potential advantages. This validation process enables the efficient detection and elimination of redundant or duplicate solutions, thereby minimizing unnecessary computations. Consequently, the focus remains on exploring innovative and potentially superior solutions.

2.5 Hybrid Algorithm Structure

The hybrid algorithm FVBK is a feature selection method that integrates sophisticated techniques to enhance optimization efficiency. To facilitate effective data processing and ensure convergence towards optimal solutions, a memory buffer is employed. This mechanism stores previously explored solutions, mitigating redundant evaluations and focusing computational efforts on unexplored, high-potential areas of the search space. The FVBK algorithm is constructed by embedding the migration behavior of the Black-winged Kite Algorithm (BKA) into the core framework of the Four Vector Intelligent Metaheuristic (FVIM). This strategic integration capitalizes on BKA's robust migration mechanisms, known for improving exploration capabilities in optimization scenarios. By synergizing these strengths with FVIM's refined vector-based position updates, the algorithm seeks to achieve a balanced trade-off between exploration and exploitation, thus enhancing its efficacy in identifying optimal feature subsets. Table 3.1 outlines the pseudocode of the FVBK algorithm, detailing the implementation of these integrated strategies throughout the optimization process. In addition to this hybrid approach, we implemented another combination to compare and contrast its effectiveness with FVBK. Specifically, we integrated only the position update strategy from FVIM into the entire BKA framework, aiming to capitalize on the robust update mechanisms of FVIM. This alternative algorithm is referred to as Black-winged Kite with Vector-based Position Update (BKVP)

algorithm. The purpose of this comparison is to determine which combination strategy offers superior performance in optimizing feature selection, providing insights into the effectiveness of each method. Table 3.2 presents the pseudocode of the baseline algorithm BKVP.

Pseudocode of FVBK Algorithm for Feature selection

Input: Dataset, MaxIter, PopSize, NFeat

Output: SelectedFeat (fstPos), BestFit (bstVal), ConvergenceCurve

Initialize

 Initialize memory buffer with given size.

 Initialize population Pop and positions (fstPos, sndPos, thrdPos, frthPos).

For each particle in the Pop :

 Randomly select feature indices.

 Calculate fitness of each particles using mRMR.

 Store selected features in the memory buffer.

End For

Iterative Optimization Loop

While iter \leq MaxIter:

For each particle in the Pop :

 Apply boundary conditions.

 Evaluate fitness using mRMR.

 Update solution hierarchy (fstPos, sndPos, thrdPos, frthPos) using Algorithm 1

 Store fitness and position history.

End For

Migration Behavior

For each particle in the Pop :

 Generate a new position using equation 3.4.

 Check new position's fitness using mRMR.

 Update memory buffer if the new position is unique.

 Store fitness and position history.

End For

Update Positions:

For each particle in the Pop :

 Adjust particle positions using equation 3.1

 Update the average position using equation 3.2

End For

 Identify the best-performing particles based on fstPos, sndPos, thrdPos, frthPos

 Update memory buffer

 Update the best value and convergence curve.

End While and return SelectedFeat, BestFit, ConvergenceCurve

Table 3.1: Hybrid algorithm FVBK pseudocode

Pseudocode of baseline algorithm BKVP

Input: Dataset, MaxIter, PopSize, NFeat

Output: SelectedFeat, BestFit, ConvergenceCurve

Initialize

 Initialize memory buffer with given size.

 Initialize population Pop

For each particle in Pop :

 Randomly select features

 Compute fitness using mRMR

 Store selected features in the memory buffer.

End For

Iterative Optimization Loop

For iter = 1 **to** MaxIter:

Attacking Behavior

For each particle in Pop :

 Adjust position using equation 3.3

End For

Migration Behavior

For each particle in Pop :

 Generate new position using equation 3.4

End For

Position Update

For each particle in Pop :

 Update position using equation 3.1 and equation 3.2

End For

Fitness Evaluation and Memory Update

For each particle in Pop :

 Recalculate fitness

 Update memory buffer if new position is unique

End For

 Update global best solution

End For and return SelectedFeat, BestFit, ConvergenceCurve

Table 3.2: Baseline algorithm pseudocode

2.6 Framework for Feature Selection

The feature selection process in this study is divided into two main stages. Initially, a filtering phase is conducted using various statistical metrics, including Chi-squared, ReliefF, Fisher score, and Mutual Information. These metrics are employed to identify and retain a set of elite features that exhibit significant relevance to the target variable. Following this, the elite features are passed through the FVBK algorithm, which refines the feature selection process by optimizing a fitness function based on the minimal-Redundancy-Maximal-Relevance (mRMR) criterion. This criterion ensures that the selected features are not only highly relevant to the outcome but also minimally redundant with each other, enhancing the overall quality of the feature subset.

Consequently, the selected features are evaluated using five diverse classifiers, each representing a distinct machine learning approach. The Support Vector Machine (SVM), a robust supervised learning model, is well-suited for both linear and non-linear classification tasks by finding the optimal hyperplane that separates classes. Linear Discriminant Analysis (LDA), on the other hand, focuses on dimensionality reduction while preserving class separability, making it effective in scenarios with normally distributed classes. The Naive Bayes (NB) classifier, grounded in Bayes' theorem, assumes feature independence and is particularly efficient in high-dimensional spaces, though it requires the assumption of conditional independence between features. The k-Nearest Neighbors (kNN) algorithm is a non-parametric method that classifies data points based on the majority class among the nearest neighbors, making it sensitive to the choice of k and the distance metric used. Lastly, the Classification and Regression Trees (CART) algorithm constructs decision trees by recursively splitting the dataset based on feature values, leading to interpretable decision rules but prone to overfitting without proper pruning. The framework depicted in Figure 3.4 outlines the comprehensive process of feature selection and classification utilized in this study.

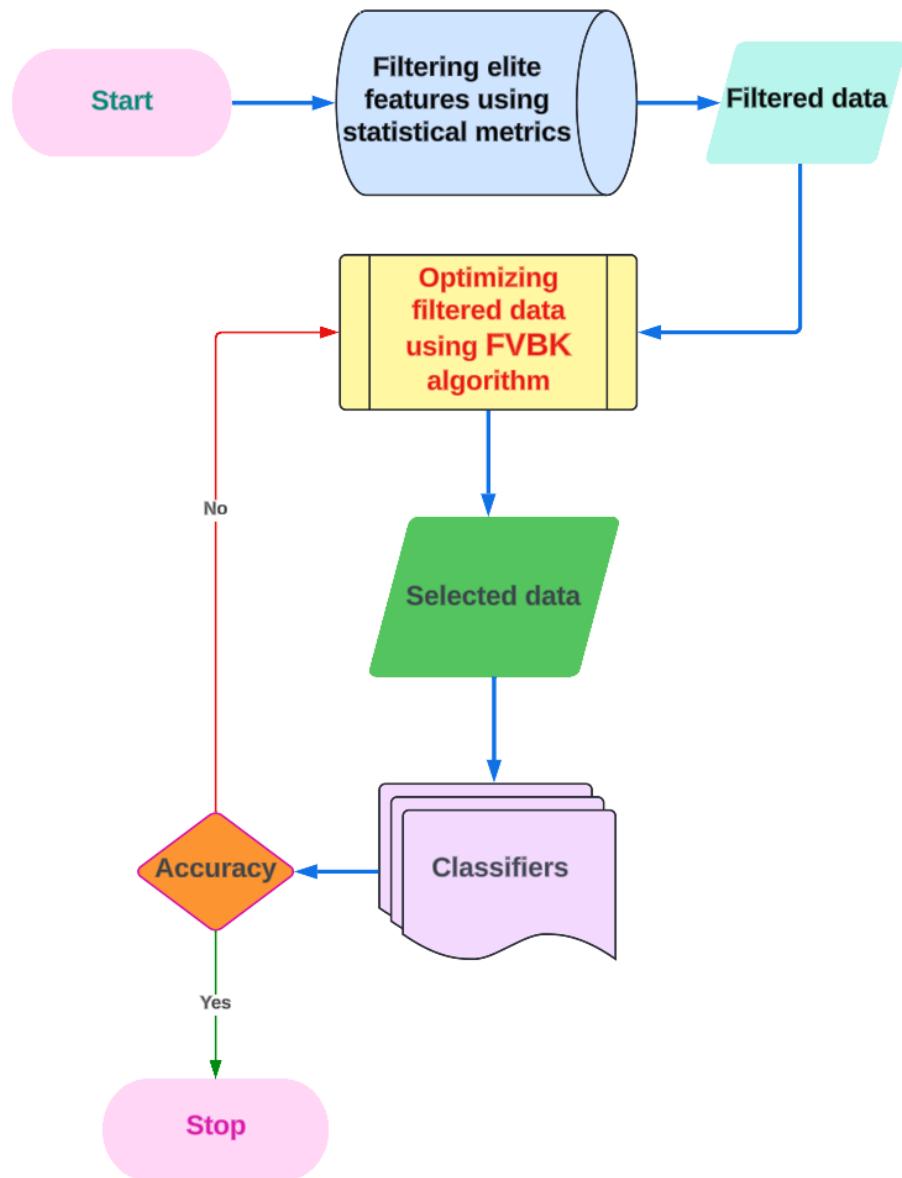


Figure 3.4: Flowchart illustrating the process for feature selection using a hybrid approach

Chapter 4

Experiments and discussion

1 Statistical analysis

The proposed algorithm was evaluated using nine real-world datasets, as summarized in Table 4.1. The parameter settings for the classifiers used with each dataset are detailed in Table 4.2. Across all experiments, the common parameters included a run count of 30, a population size (PopSize) of 100, and the number of features in the final subset set to 10. For most datasets, excluding DEX and DBE, the maximum number of iterations (MaxIter) was set to 100, and the number of elite features filtered out was 100. However, for the DEX and DBE datasets, more intensive processing was required, with MaxIter increased to 200 and the number of elite features filtered out expanded to 500. This adjustment reflects the complexity and larger feature space of these datasets, necessitating more iterations and a higher threshold for feature selection.

Table 4.1: Datasets Description

Dataset	Type	# Features	# Instances	# Classes
GLI_85 (GLI)	Biological	22283	85	2
Colon Cancer (CLN)	Biological	2000	62	2
DBWorld e-mails (DBE)	Text	4702	64	2
Dexter (DEX)	Text	20000	300	2
Lymphoma (LYM)	Biological	4026	96	9
NCI9 (NCI)	Biological	9712	60	9
Orlraws10P (ORP)	Image	10304	100	10
SMK_CAN_187 (SMK)	Biological	19993	187	2
TOX_171 (TOX)	Biological	5748	171	4

Table 4.2: Parameter settings of classifiers for each dataset

Dataset	LDA Type	SVM Kernel	NB Distribution
GLI	Linear	Linear	Normal
CLN	Diaglinear	Linear	Multivariate Multinomial
DBE	Diaglinear	Linear	Multivariate Multinomial
DEX	Diaglinear	Gaussian	Kernel Density Estimate
LYM	Linear	Linear	Multivariate Multinomial
NCI	Diaglinear	Linear	Multivariate Multinomial
ORP	Linear	Linear	Normal
SMK	Linear	Linear	Kernel Density Estimate
TOX	Linear	Linear	Normal

The findings highlight the best accuracy obtained from the FVBK algorithm, the baseline algorithm BKVP, and two standard algorithms, BKA and FVIM, in conjunction with eight additional swarm algorithms. The results, meticulously documented across nine datasets with varying filter selections, are systematically organized: Table 4.3 details the performance of the SVM classifier, Table 4.4 covers the LDA classifier, Table 4.5 presents the NB classifier, Table 4.6 illustrates the kNN classifier, and Table 4.7 highlights the CART classifier. The average accuracy across the five classifiers for each algorithm is presented in the Table 4.8, organized by each filter. We illustrate the maximum accuracy achieved by each algorithm across various filter configurations, organized by classifier, in the figures below. Specifically, Figure 4.1 illustrates the performance for the SVM classifier, Figure 4.2 for LDA, Figure 4.3 for NB, Figure 4.4 for kNN, and Figure 4.5 for CART. Additionally, Figure 4.6 provides an overview of the average accuracy across all classifiers for each filter, offering a comparative insight into overall classifier performance across filtering methods.

Table 4.3: Best accuracy achieved by algorithms with an SVM classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
Filter: Mutual Information									
FVBK	0.882	0.903	0.938	0.5	1	1	1	0.738	1
BKVP	0.871	0.919	0.938	0.5	1	1	1	0.733	1
AHA	0.894	0.952	0.953	0.5	1	1	1	0.743	1
BKA	0.871	0.919	0.953	0.5	1	1	1	0.743	1
CapSA	0.824	0.839	0.859	0.5	1	1	1	0.701	1
DAOA	0.882	0.919	0.922	0.5	1	1	1	0.727	1
ESOA	0.882	0.919	0.953	0.5	1	1	1	0.738	1
FVIM	0.894	0.887	0.938	0.5	1	1	1	0.727	1
HGSO	0.776	0.887	0.797	0.5	1	1	1	0.679	1
ILA	0.776	0.887	0.797	0.5	1	1	1	0.679	1
LOA	0.894	0.919	0.891	0.5	1	1	1	0.765	1

Table 4.3: Best accuracy achieved by algorithms with an SVM classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
LSO	0.871	0.887	0.906	0.5	1	1	1	0.738	1
Filter: Fisher score									
FVBK	0.941	0.935	0.922	0.5	1	1	1	0.786	1
BKVP	0.976	0.952	0.859	0.5	1	1	1	0.807	1
AHA	0.765	0.887	0.953	0.5	1	1	1	0.717	1
BKA	0.824	0.887	0.922	0.5	1	1	1	0.733	1
CapSA	0.494	0.758	0.859	0.5	1	1	1	0.658	1
DAOA	0.765	0.887	0.875	0.5	1	1	1	0.717	1
ESOA	0.765	0.887	0.938	0.5	1	1	1	0.743	1
FVIM	0.729	0.839	0.938	0.5	1	1	1	0.711	1
HGSO	0.682	0.645	0.563	0.5	1	1	1	0.519	1
ILA	0.682	0.645	0.563	0.5	1	1	1	0.519	1
LOA	0.812	0.887	0.922	0.5	1	1	1	0.701	1
LSO	0.741	0.79	0.906	0.5	1	1	1	0.711	1
Filter: ReliefF									
FVBK	0.976	0.903	0.922	0.5	1	1	1	0.797	1
BKVP	0.965	0.887	0.938	0.5	1	1	1	0.775	1
AHA	0.988	0.903	0.938	0.5	1	1	1	0.834	1
BKA	0.965	0.887	0.922	0.5	1	1	1	0.807	1
CapSA	0.835	0.839	0.859	0.5	1	1	1	0.658	1
DAOA	0.976	0.935	0.875	0.5	1	1	1	0.829	1
ESOA	0.965	0.887	0.922	0.5	1	1	1	0.786	1
FVIM	0.953	0.903	0.875	0.5	1	1	1	0.807	1
HGSO	0.8	0.758	0.859	0.5	1	1	1	0.658	1
ILA	0.8	0.758	0.859	0.5	1	1	1	0.658	1

Table 4.3: Best accuracy achieved by algorithms with an SVM classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
LOA	0.941	0.887	0.922	0.5	1	1	1	0.786	1
LSO	0.965	0.903	0.906	0.5	1	1	1	0.781	1
Filter: Chi-squared test									
FVBK	0.929	0.929	0.938	0.5	1	1	1	0.786	1
BKVP	0.894	0.871	0.875	0.5	1	1	1	0.824	1
AHA	0.941	0.919	0.953	0.5	1	1	1	0.791	1
BKA	0.894	0.935	0.953	0.5	1	1	1	0.781	1
CapSA	0.835	0.839	0.859	0.5	1	1	1	0.69	1
DAOA	0.918	0.935	0.922	0.5	1	1	1	0.765	1
ESOA	0.929	0.968	0.938	0.5	1	1	1	0.781	1
FVIM	0.941	0.952	0.906	0.5	1	1	1	0.797	1
HGSO	0.882	0.839	0.859	0.5	1	1	1	0.69	1
ILA	0.882	0.839	0.859	0.5	1	1	1	0.69	1
LOA	0.918	0.871	0.906	0.5	1	1	1	0.775	1
LSO	0.918	0.919	0.891	0.5	1	1	1	0.797	1

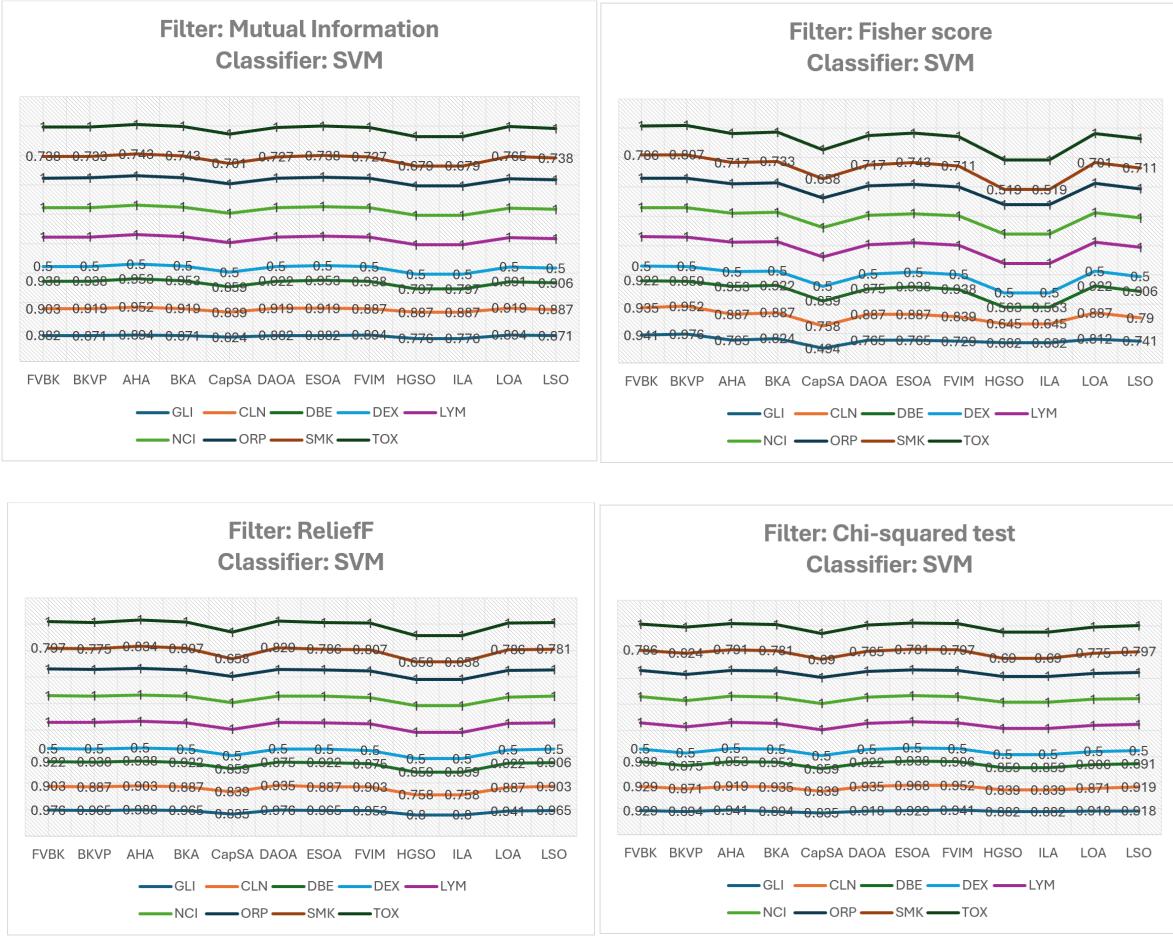


Figure 4.1: Accuracy of different algorithms by filters for the SVM classifier

Table 4.4: Best accuracy achieved by algorithms with an LDA classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
Filter: Mutual Information									
FVBK	0.941	0.919	0.922	0.502	0.896	0.7	0.95	0.738	0.743
BKVP	0.918	0.887	0.922	0.502	0.917	0.7	0.93	0.743	0.725
AHA	0.941	0.903	0.953	0.5	0.917	0.717	0.93	0.738	0.772
BKA	0.929	0.903	0.953	0.5	0.896	0.733	0.92	0.733	0.789

Table 4.4: Best accuracy achieved by algorithms with an LDA classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
CapSA	0.847	0.839	0.859	0.5	0.573	0.267	0.32	0.706	0.591
DAOA	0.929	0.887	0.938	0.5	0.896	0.717	0.93	0.754	0.754
ESOA	0.906	0.919	0.938	0.5	0.906	0.733	0.92	0.754	0.784
FVIM	0.941	0.903	0.922	0.5	0.917	0.733	0.94	0.743	0.731
HGSO	0.871	0.645	0.797	0.5	0.771	0.567	0.49	0.684	0.649
ILA	0.871	0.645	0.797	0.5	0.771	0.567	0.49	0.684	0.649
LOA	0.929	0.903	0.875	0.5	0.896	0.683	0.91	0.738	0.778
LSO	0.929	0.903	0.906	0.5	0.917	0.683	0.93	0.733	0.69
Filter: Fisher score									
FVBK	0.965	0.903	0.869	0.502	0.927	0.583	0.91	0.791	0.778
BKVP	0.965	0.903	0.852	0.502	0.896	0.567	0.91	0.802	0.749
AHA	0.906	0.806	0.953	0.5	0.594	0.383	0.67	0.722	0.696
BKA	0.894	0.823	0.906	0.5	0.552	0.367	0.67	0.717	0.678
CapSA	0.776	0.758	0.859	0.5	0.479	0.233	0.34	0.658	0.409
DAOA	0.882	0.823	0.859	0.5	0.594	0.283	0.68	0.69	0.649
ESOA	0.894	0.823	0.922	0.5	0.604	0.317	0.65	0.706	0.678
FVIM	0.894	0.823	0.938	0.5	0.583	0.35	0.7	0.706	0.684
HGSO	0.824	0.855	0.531	0.5	0.479	0.3	0.28	0.465	0.31
ILA	0.824	0.645	0.531	0.5	0.479	0.25	0.28	0.465	0.31
LOA	0.871	0.806	0.906	0.5	0.604	0.35	0.67	0.733	0.684
LSO	0.882	0.823	0.891	0.5	0.573	0.35	0.71	0.711	0.655
Filter: ReliefF									
FVBK	0.965	0.903	0.891	0.502	0.885	0.583	0.78	0.77	0.766
BKVP	0.965	0.887	0.891	0.502	0.896	0.6	0.77	0.818	0.754
AHA	0.976	0.903	0.938	0.5	0.875	0.567	0.77	0.802	0.76

Table 4.4: Best accuracy achieved by algorithms with an LDA classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
BKA	0.965	0.903	0.906	0.5	0.833	0.567	0.76	0.797	0.743
CapSA	0.871	0.839	0.859	0.5	0.51	0.233	0.41	0.663	0.532
DAOA	0.953	0.903	0.875	0.502	0.854	0.567	0.72	0.813	0.725
ESOA	0.965	0.903	0.922	0.5	0.865	0.583	0.77	0.834	0.778
FVIM	0.976	0.903	0.906	0.5	0.875	0.483	0.76	0.786	0.76
HGSO	0.906	0.758	0.859	0.5	0.625	0.45	0.21	0.658	0.474
ILA	0.906	0.758	0.859	0.5	0.625	0.45	0.21	0.658	0.474
LOA	0.965	0.903	0.891	0.5	0.865	0.583	0.77	0.781	0.749
LSO	0.965	0.919	0.875	0.5	0.865	0.6	0.76	0.791	0.754
Filter: Chi-squared test									
FVBK	0.929	0.887	0.938	0.502	0.896	0.567	0.9	0.797	0.731
BKVP	0.929	0.903	0.938	0.502	0.927	0.75	0.93	0.791	0.713
AHA	0.941	0.903	0.953	0.5	0.906	0.683	0.89	0.807	0.754
BKA	0.929	0.919	0.938	0.5	0.917	0.683	0.9	0.797	0.725
CapSA	0.871	0.839	0.859	0.5	0.573	0.25	0.63	0.695	0.48
DAOA	0.941	0.919	0.922	0.5	0.927	0.7	0.85	0.77	0.713
ESOA	0.941	0.903	0.938	0.5	0.906	0.733	0.89	0.775	0.778
FVIM	0.941	0.903	0.922	0.5	0.917	0.717	0.85	0.797	0.725
HGSO	0.8	0.645	0.859	0.5	0.917	0.15	0.48	0.695	0.462
ILA	0.8	0.645	0.859	0.5	0.771	0.15	0.48	0.695	0.462
LOA	0.953	0.919	0.891	0.5	0.958	0.7	0.87	0.775	0.749
LSO	0.941	0.903	0.906	0.5	0.927	0.7	0.87	0.786	0.737

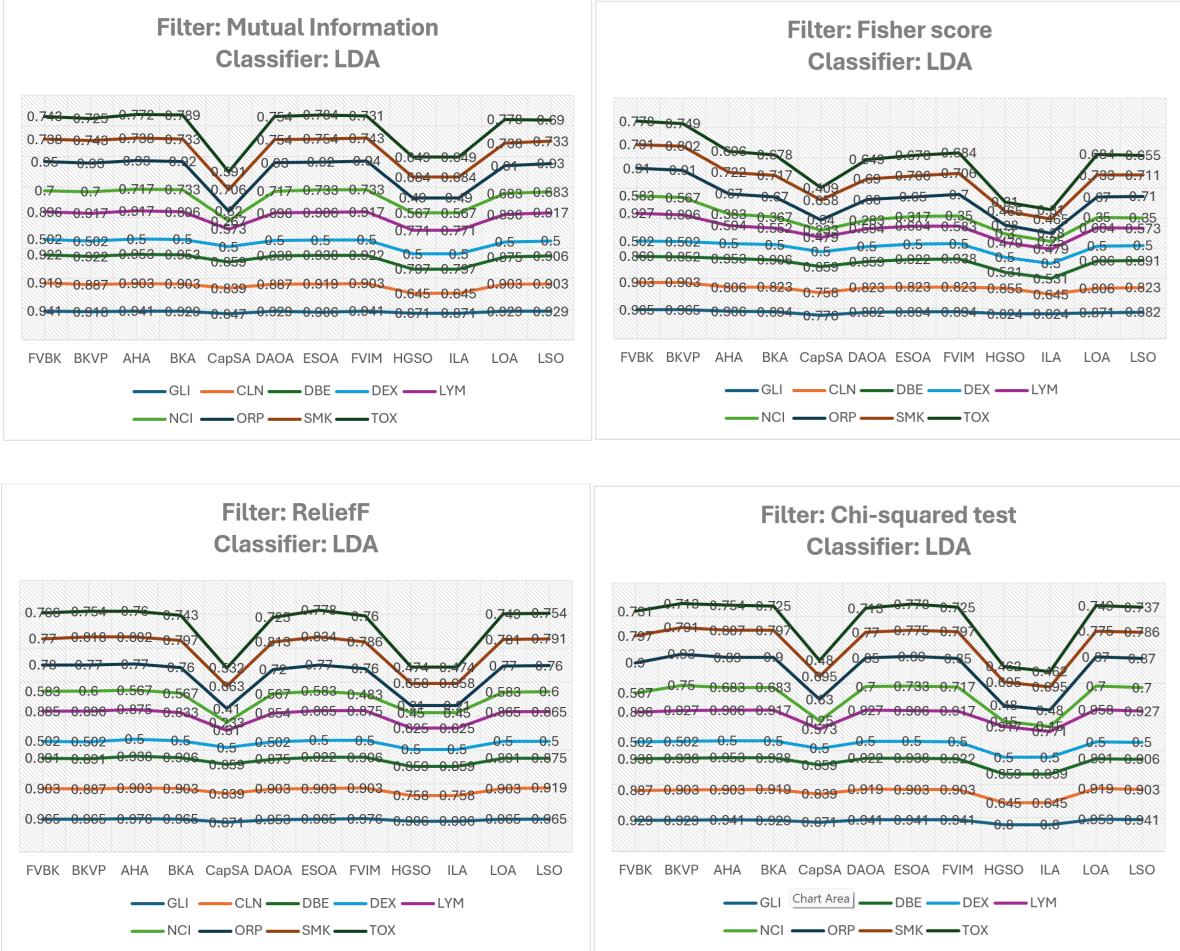


Figure 4.2: Accuracy of different algorithms by filters for the LDA classifier

Table 4.5: Best accuracy achieved by algorithms with an NB classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
Filter: Mutual Information									
FVBK	0.953	0.903	0.953	0.5	0.917	0.717	0.92	0.781	0.725
BKVP	0.941	0.903	0.953	0.5	0.927	0.767	0.94	0.786	0.731
AHA	0.953	0.919	0.953	0.5	0.927	0.7	0.91	0.759	0.731
BKA	0.929	0.919	0.953	0.5	0.885	0.75	0.93	0.77	0.713

Table 4.5: Best accuracy achieved by algorithms with an NB classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
CapSA	0.859	0.839	0.547	0.5	0.604	0.333	0.35	0.69	0.556
DAOA	0.941	0.887	0.906	0.5	0.917	0.683	0.92	0.754	0.719
ESOA	0.965	0.919	0.953	0.5	0.917	0.717	0.93	0.759	0.784
FVIM	0.941	0.919	0.938	0.5	0.885	0.717	0.93	0.765	0.76
HGSO	0.765	0.839	0.891	0.5	0.698	0.583	0.51	0.695	0.585
ILA	0.765	0.839	0.547	0.5	0.698	0.583	0.51	0.695	0.585
LOA	0.941	0.887	0.906	0.5	0.917	0.7	0.92	0.754	0.772
LSO	0.929	0.903	0.906	0.5	0.906	0.683	0.92	0.775	0.725
Filter: Fisher score									
FVBK	0.976	0.903	0.5	0.5	0.906	0.667	0.97	0.797	0.731
BKVP	0.965	0.903	0.516	0.5	0.896	0.7	0.97	0.807	0.713
AHA	0.835	0.823	0.969	0.5	0.531	0.333	0.78	0.738	0.667
BKA	0.859	0.823	0.938	0.5	0.552	0.317	0.79	0.701	0.655
CapSA	0.741	0.565	0.547	0.5	0.479	0.333	0.38	0.652	0.415
DAOA	0.835	0.758	0.906	0.5	0.542	0.333	0.83	0.701	0.655
ESOA	0.824	0.823	0.922	0.5	0.542	0.317	0.81	0.722	0.643
FVIM	0.859	0.823	0.953	0.5	0.531	0.333	0.86	0.722	0.719
HGSO	0.729	0.645	0.641	0.5	0.198	0.083	0.28	0.54	0.374
ILA	0.729	0.645	0.641	0.5	0.198	0.083	0.28	0.54	0.374
LOA	0.882	0.79	0.922	0.5	0.573	0.35	0.82	0.711	0.649
LSO	0.824	0.823	0.875	0.5	0.521	0.317	0.79	0.711	0.673
Filter: ReliefF									
FVBK	0.976	0.919	0.906	0.5	0.844	0.55	0.84	0.797	0.789
BKVP	0.976	0.903	0.938	0.5	0.844	0.617	0.82	0.786	0.737
AHA	0.976	0.903	0.922	0.5	0.833	0.567	0.81	0.791	0.737

Table 4.5: Best accuracy achieved by algorithms with an NB classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
BKA	0.976	0.903	0.938	0.5	0.854	0.567	0.82	0.818	0.725
CapSA	0.882	0.839	0.547	0.5	0.479	0.233	0.46	0.658	0.509
DAOA	0.965	0.887	0.875	0.5	0.844	0.6	0.84	0.77	0.789
ESOA	0.965	0.887	0.875	0.5	0.844	0.6	0.84	0.77	0.789
FVIM	0.965	0.903	0.922	0.5	0.844	0.533	0.85	0.77	0.743
HGSO	0.871	0.839	0.547	0.5	0.573	0.367	0.24	0.647	0.415
ILA	0.871	0.839	0.547	0.5	0.573	0.367	0.24	0.647	0.415
LOA	0.965	0.887	0.891	0.5	0.854	0.517	0.83	0.781	0.76
LSO	0.976	0.887	0.938	0.5	0.865	0.533	0.87	0.77	0.766
Filter: Chi-squared test									
FVBK	0.953	0.953	0.953	0.5	0.948	0.717	0.92	0.802	0.713
BKVP	0.965	0.903	0.953	0.5	0.896	0.7	0.9	0.791	0.737
AHA	0.965	0.903	0.953	0.5	0.906	0.7	0.91	0.813	0.749
BKA	0.976	0.919	0.953	0.5	0.896	0.733	0.88	0.781	0.708
CapSA	0.882	0.839	0.859	0.5	0.573	0.383	0.62	0.679	0.48
DAOA	0.953	0.903	0.922	0.5	0.896	0.683	0.92	0.791	0.69
ESOA	0.965	0.903	0.953	0.5	0.896	0.733	0.93	0.802	0.743
FVIM	0.953	0.919	0.922	0.5	0.917	0.717	0.9	0.813	0.719
HGSO	0.788	0.839	0.547	0.5	0.896	0.4	0.71	0.69	0.532
ILA	0.788	0.839	0.547	0.5	0.677	0.4	0.71	0.69	0.532
LOA	0.965	0.919	0.906	0.5	0.896	0.683	0.91	0.791	0.731
LSO	0.953	0.903	0.906	0.5	0.917	0.683	0.91	0.791	0.702

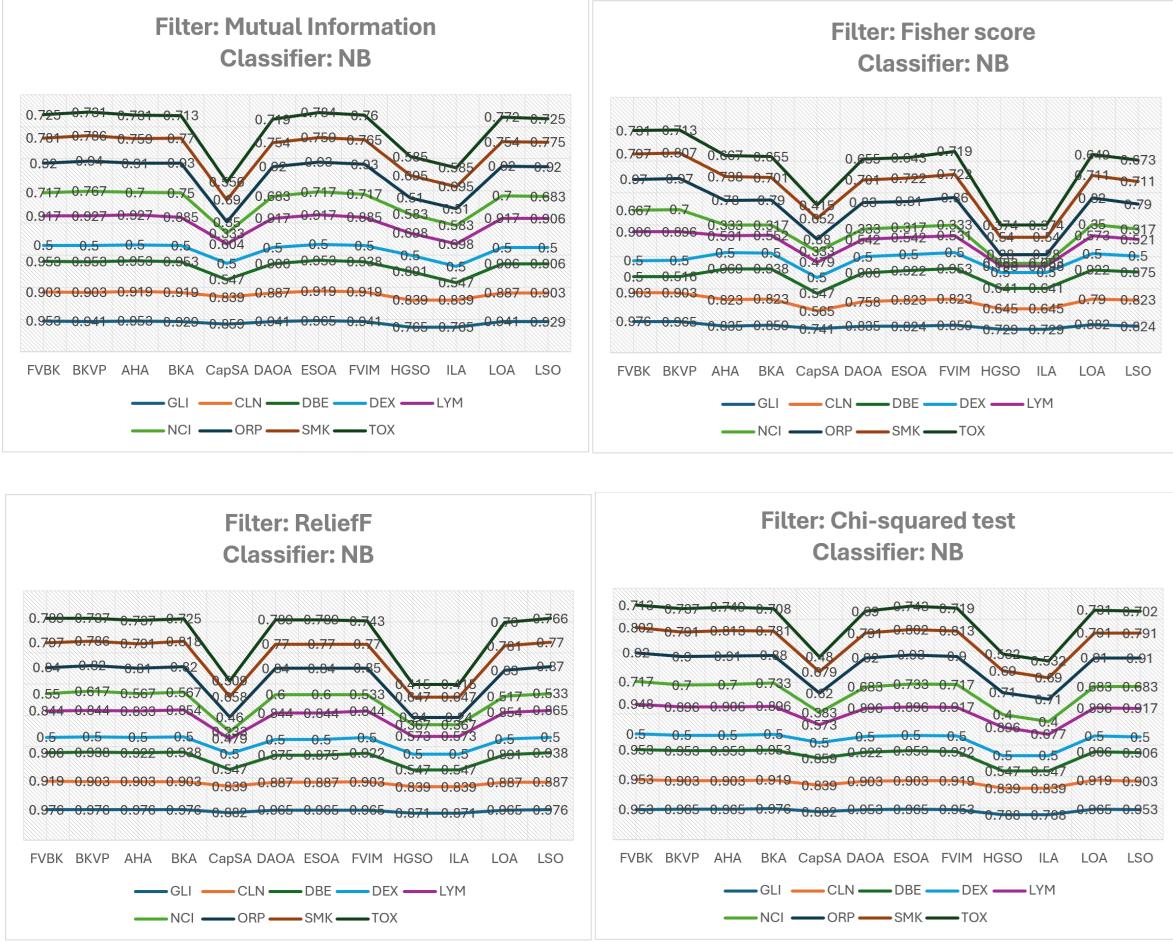


Figure 4.3: Accuracy of different algorithms by filters for the NB classifier

Table 4.6: Best accuracy achieved by algorithms with an kNN classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
Filter: Mutual Information									
FVBK	0.929	0.919	0.938	0.49	0.885	0.717	0.93	0.733	0.754
BKVP	0.953	0.919	0.938	0.49	0.906	0.7	0.95	0.727	0.766
AHA	0.929	0.903	0.969	0.49	0.833	0.617	0.87	0.754	0.737
BKA	0.918	0.903	0.938	0.49	0.906	0.7	0.94	0.722	0.749

Table 4.6: Best accuracy achieved by algorithms with an kNN classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
CapSA	0.824	0.839	0.859	0.49	0.573	0.333	0.5	0.54	0.556
DAOA	0.941	0.935	0.875	0.49	0.917	0.7	0.95	0.749	0.719
ESOA	0.918	0.919	0.953	0.49	0.896	0.7	0.91	0.706	0.754
FVIM	0.906	0.919	0.953	0.5	0.865	0.717	0.93	0.727	0.76
HGSO	0.824	0.839	0.547	0.49	0.698	0.517	0.59	0.578	0.673
ILA	0.824	0.839	0.547	0.49	0.698	0.517	0.59	0.578	0.673
LOA	0.929	0.887	0.891	0.49	0.885	0.717	0.93	0.717	0.749
LSO	0.941	0.887	0.875	0.49	0.854	0.717	0.95	0.722	0.76
Filter: Fisher score									
FVBK	0.976	0.903	0.891	0.49	0.906	0.7	0.97	0.797	0.708
BKVP	0.965	0.903	0.813	0.49	0.927	0.683	0.97	0.749	0.69
AHA	0.918	0.839	0.969	0.49	0.625	0.433	0.71	0.706	0.766
BKA	0.894	0.887	0.891	0.49	0.635	0.433	0.72	0.711	0.76
CapSA	0.8	0.694	0.859	0.49	0.479	0.183	0.48	0.604	0.462
DAOA	0.871	0.806	0.813	0.5	0.635	0.45	0.74	0.717	0.76
ESOA	0.906	0.887	0.891	0.49	0.615	0.45	0.73	0.69	0.766
FVIM	0.871	0.871	0.938	0.5	0.646	0.45	0.75	0.701	0.749
HGSO	0.694	0.419	0.547	0.49	0.479	0.267	0.44	0.61	0.52
ILA	0.694	0.419	0.547	0.49	0.479	0.267	0.44	0.61	0.52
LOA	0.918	0.806	0.938	0.49	0.615	0.55	0.72	0.695	0.789
LSO	0.882	0.839	0.844	0.49	0.646	0.4	0.74	0.695	0.807
Filter: ReliefF									
FVBK	0.988	0.903	0.922	0.49	0.854	0.533	0.84	0.77	0.754
BKVP	0.953	0.903	0.922	0.49	0.854	0.567	0.81	0.791	0.749
AHA	0.965	0.919	0.938	0.49	0.833	0.473	0.81	0.765	0.731

Table 4.6: Best accuracy achieved by algorithms with an kNN classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
BKA	0.965	0.919	0.922	0.49	0.844	0.633	0.84	0.754	0.766
CapSA	0.871	0.839	0.859	0.49	0.49	0.233	0.57	0.652	0.55
DAOA	0.918	0.871	0.859	0.49	0.802	0.567	0.8	0.727	0.649
ESOA	0.965	0.903	0.922	0.49	0.833	0.567	0.82	0.754	0.69
FVIM	0.941	0.919	0.859	0.5	0.802	0.6	0.77	0.77	0.754
HGSO	0.859	0.839	0.859	0.5	0.656	0.433	0.11	0.529	0.468
ILA	0.859	0.839	0.859	0.5	0.656	0.433	0.11	0.529	0.468
LOA	0.941	0.919	0.813	0.49	0.813	0.567	0.79	0.738	0.784
LSO	0.976	0.887	0.875	0.49	0.875	0.533	0.83	0.759	0.76
Filter: Chi-squared test									
FVBK	0.953	0.903	0.938	0.49	0.885	0.885	0.89	0.754	0.719
BKVP	0.941	0.919	0.938	0.49	0.885	0.75	0.93	0.781	0.719
AHA	0.965	0.935	0.953	0.49	0.917	0.733	0.92	0.786	0.725
BKA	0.941	0.903	0.938	0.49	0.875	0.7	0.88	0.791	0.754
CapSA	0.871	0.839	0.859	0.49	0.531	0.25	0.62	0.636	0.526
DAOA	0.953	0.903	0.875	0.49	0.896	0.7	0.87	0.775	0.719
ESOA	0.953	0.903	0.938	0.49	0.917	0.75	0.86	0.775	0.737
FVIM	0.953	0.903	0.922	0.5	0.917	0.683	0.9	0.765	0.743
HGSO	0.918	0.839	0.859	0.49	0.448	0.25	0.9	0.636	0.585
ILA	0.918	0.839	0.859	0.49	0.448	0.25	0.71	0.636	0.585
LOA	0.941	0.903	0.859	0.49	0.896	0.733	0.87	0.77	0.713
LSO	0.953	0.903	0.891	0.49	0.906	0.717	0.86	0.77	0.725

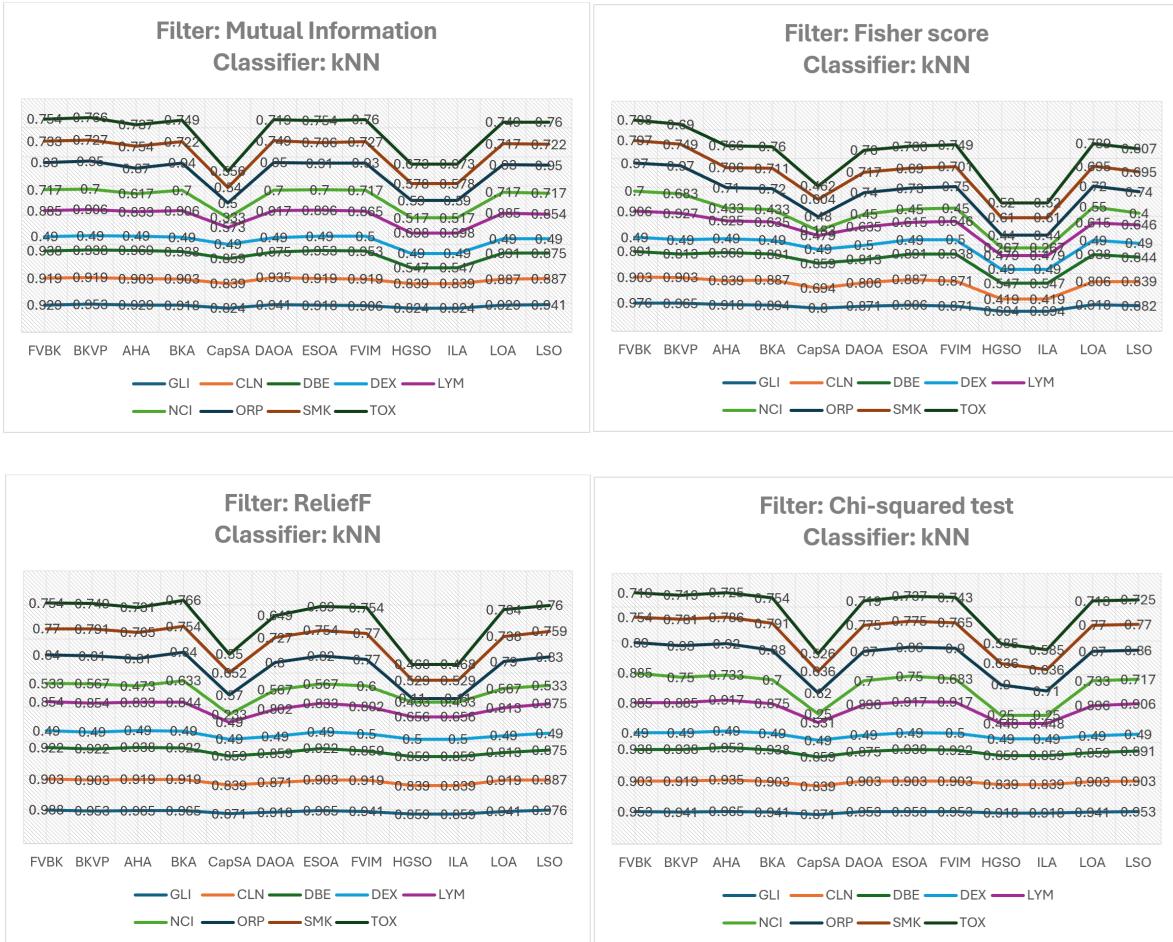


Figure 4.4: Accuracy of different algorithms by filters for the kNN classifier

Table 4.7: Best accuracy achieved by algorithms with an CART classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
Filter: Mutual Information									
FVBK	0.929	0.935	0.922	0.5	0.833	0.667	0.85	0.695	0.673
BKVP	0.941	0.903	0.922	0.5	0.833	0.633	0.85	0.722	0.702
AHA	0.929	0.919	0.875	0.5	0.813	0.633	0.83	0.754	0.678
BKA	0.906	0.887	0.922	0.5	0.865	0.667	0.86	0.743	0.69

Table 4.7: Best accuracy achieved by algorithms with an CART classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
CapSA	0.824	0.839	0.859	0.5	0.573	0.333	0.5	0.54	0.556
DAOA	0.941	0.968	0.906	0.5	0.833	0.633	0.88	0.717	0.69
ESOA	0.953	0.919	0.922	0.5	0.844	0.683	0.84	0.711	0.719
FVIM	0.929	0.887	0.922	0.5	0.844	0.667	0.86	0.717	0.719
HGSO	0.859	0.839	0.797	0.5	0.74	0.567	0.58	0.535	0.591
ILA	0.859	0.839	0.797	0.5	0.74	0.567	0.58	0.535	0.591
LOA	0.953	0.919	0.859	0.5	0.813	0.617	0.84	0.711	0.713
LSO	0.918	0.903	0.906	0.5	0.865	0.683	0.82	0.738	0.673
Filter: Fisher score									
FVBK	0.929	0.935	0.844	0.5	0.854	0.667	0.84	0.754	0.667
BKVP	0.918	0.919	0.828	0.5	0.833	0.683	0.84	0.765	0.684
AHA	0.859	0.758	0.938	0.5	0.583	0.433	0.69	0.701	0.69
BKA	0.847	0.887	0.938	0.5	0.542	0.317	0.69	0.711	0.673
CapSA	0.812	0.758	0.859	0.5	0.49	0.25	0.51	0.61	0.468
DAOA	0.835	0.887	0.875	0.5	0.563	0.317	0.7	0.722	0.673
ESOA	0.882	0.871	0.938	0.5	0.604	0.45	0.69	0.733	0.696
FVIM	0.894	0.887	0.891	0.5	0.573	0.45	0.66	0.679	0.62
HGSO	0.729	0.516	0.547	0.5	0.49	0.283	0.31	0.594	0.526
ILA	0.729	0.516	0.547	0.5	0.49	0.283	0.31	0.594	0.526
LOA	0.871	0.903	0.906	0.5	0.573	0.433	0.67	0.743	0.649
LSO	0.847	0.823	0.891	0.5	0.625	0.417	0.73	0.749	0.69
Filter: ReliefF									
FVBK	0.941	0.935	0.938	0.5	0.813	0.5	0.77	0.759	0.684
BKVP	0.929	0.919	0.922	0.5	0.802	0.583	0.8	0.797	0.713
AHA	0.906	0.903	0.938	0.5	0.706	0.533	0.75	0.786	0.731

Table 4.7: Best accuracy achieved by algorithms with an CART classifier using various filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
BKA	0.941	0.903	0.906	0.5	0.802	0.6	0.73	0.754	0.69
CapSA	0.859	0.839	0.859	0.5	0.542	0.233	0.56	0.62	0.556
DAOA	0.929	0.919	0.859	0.5	0.74	0.583	0.79	0.749	0.731
ESOA	0.953	0.903	0.953	0.5	0.792	0.617	0.8	0.765	0.696
FVIM	0.941	0.919	0.922	0.5	0.75	0.483	0.8	0.754	0.731
HGSO	0.859	0.839	0.891	0.5	0.656	0.433	0.11	0.529	0.468
ILA	0.859	0.839	0.859	0.5	0.656	0.433	0.11	0.529	0.468
LOA	0.918	0.919	0.891	0.5	0.854	0.55	0.78	0.781	0.713
LSO	0.906	0.903	0.922	0.5	0.76	0.55	0.81	0.765	0.667
Filter: Chi-squared test									
FVBK	0.929	0.935	0.938	0.5	0.802	0.717	0.86	0.775	0.775
BKVP	0.953	0.935	0.938	0.5	0.885	0.683	0.87	0.749	0.713
AHA	0.918	0.919	0.922	0.5	0.792	0.7	0.89	0.775	0.708
BKA	0.929	0.903	0.906	0.5	0.813	0.65	0.84	0.781	0.702
CapSA	0.859	0.839	0.859	0.5	0.573	0.25	0.69	0.583	0.48
DAOA	0.941	0.903	0.859	0.5	0.844	0.667	0.87	0.775	0.719
ESOA	0.941	0.919	0.922	0.5	0.833	0.667	0.9	0.775	0.743
FVIM	0.929	0.935	0.906	0.5	0.813	0.617	0.85	0.77	0.725
HGSO	0.871	0.839	0.859	0.5	0.573	0.25	0.66	0.583	0.561
ILA	0.871	0.839	0.859	0.5	0.573	0.25	0.66	0.583	0.561
LOA	0.953	0.903	0.891	0.5	0.802	0.667	0.86	0.759	0.696
LSO	0.941	0.903	0.875	0.5	0.833	0.683	0.9	0.781	0.69

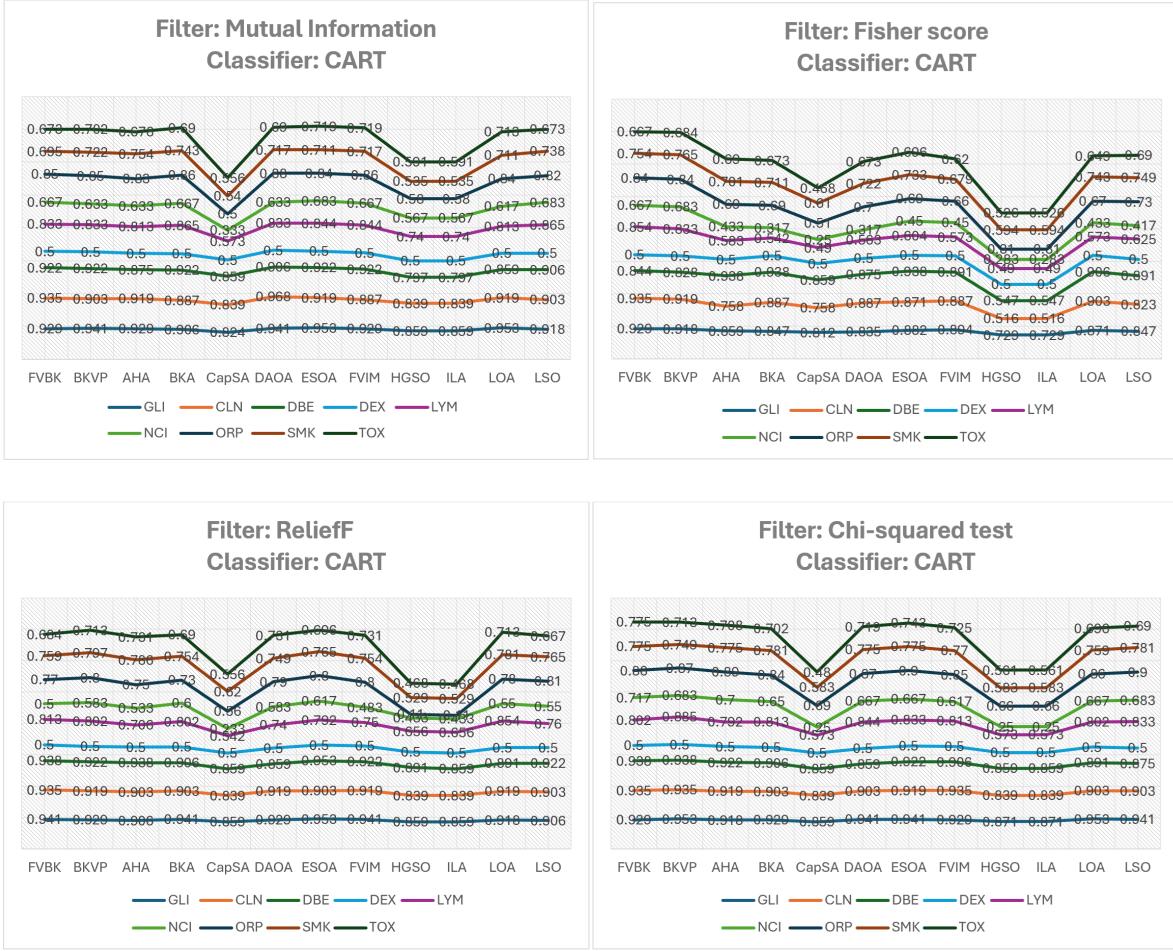


Figure 4.5: Accuracy of different algorithms by filters for the CART classifier

Table 4.8: Average accuracy achieved by algorithms by filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
	Filter: Mutual Information								
FVBK	0.927	0.916	0.934	0.498	0.906	0.768	0.934	0.737	0.779
BKVP	0.925	0.906	0.934	0.498	0.917	0.768	0.934	0.742	0.785
AHA	0.922	0.914	0.921	0.498	0.898	0.733	0.908	0.75	0.784
BKA	0.911	0.906	0.924	0.498	0.91	0.77	0.93	0.742	0.788

Table 4.8: Average accuracy achieved by algorithms by filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
CapSA	0.835	0.839	0.797	0.498	0.665	0.453	0.534	0.635	0.651
DAOA	0.927	0.914	0.909	0.498	0.913	0.747	0.93	0.74	0.777
ESOA	0.925	0.914	0.934	0.498	0.913	0.767	0.92	0.734	0.808
FVIM	0.922	0.903	0.934	0.5	0.902	0.767	0.932	0.736	0.794
HGSO	0.819	0.81	0.766	0.498	0.781	0.647	0.634	0.634	0.699
ILA	0.819	0.81	0.697	0.498	0.781	0.647	0.634	0.634	0.699
LOA	0.921	0.903	0.884	0.498	0.902	0.743	0.92	0.737	0.802
LSO	0.918	0.897	0.9	0.498	0.908	0.753	0.924	0.741	0.77
Filter: Fisher score									
FVBK	0.958	0.916	0.805	0.498	0.919	0.723	0.938	0.786	0.777
BKVP	0.958	0.916	0.774	0.498	0.91	0.727	0.938	0.785	0.767
AHA	0.856	0.823	0.956	0.498	0.667	0.517	0.77	0.717	0.764
BKA	0.864	0.861	0.919	0.498	0.656	0.487	0.774	0.714	0.753
CapSA	0.725	0.706	0.797	0.498	0.585	0.4	0.542	0.636	0.551
DAOA	0.838	0.832	0.866	0.5	0.667	0.477	0.79	0.709	0.747
ESOA	0.854	0.858	0.922	0.498	0.673	0.507	0.776	0.719	0.757
FVIM	0.849	0.848	0.931	0.5	0.667	0.517	0.794	0.704	0.754
HGSO	0.732	0.616	0.566	0.498	0.529	0.387	0.462	0.545	0.546
ILA	0.732	0.574	0.566	0.498	0.529	0.377	0.462	0.545	0.546
LOA	0.871	0.839	0.919	0.498	0.673	0.537	0.776	0.717	0.754
LSO	0.835	0.819	0.881	0.498	0.673	0.497	0.794	0.716	0.765
Filter: ReliefF									
FVBK	0.969	0.913	0.916	0.498	0.879	0.633	0.846	0.779	0.799
BKVP	0.958	0.9	0.922	0.498	0.879	0.673	0.84	0.794	0.791
AHA	0.962	0.906	0.934	0.498	0.85	0.628	0.828	0.796	0.792

Table 4.8: Average accuracy achieved by algorithms by filters

	GLI	CLN	DBE	DEX	LYM	NCI	ORP	SMK	TOX
BKA	0.962	0.903	0.919	0.498	0.867	0.673	0.83	0.786	0.785
CapSA	0.864	0.839	0.797	0.498	0.604	0.387	0.6	0.65	0.629
DAOA	0.948	0.903	0.869	0.498	0.848	0.663	0.83	0.778	0.779
ESOA	0.962	0.9	0.931	0.498	0.869	0.657	0.845	0.79	0.786
FVIM	0.955	0.91	0.897	0.5	0.854	0.62	0.836	0.778	0.798
HGSO	0.859	0.806	0.803	0.5	0.702	0.537	0.334	0.604	0.565
ILA	0.859	0.806	0.797	0.5	0.702	0.537	0.334	0.604	0.565
LOA	0.946	0.903	0.881	0.498	0.877	0.643	0.834	0.773	0.801
LSO	0.958	0.9	0.903	0.498	0.873	0.643	0.854	0.773	0.789
Filter: Chi-squared test									
FVBK	0.939	0.922	0.941	0.498	0.906	0.777	0.914	0.783	0.788
BKVP	0.936	0.906	0.928	0.498	0.919	0.777	0.926	0.787	0.777
AHA	0.93	0.916	0.947	0.498	0.904	0.763	0.922	0.795	0.787
BKA	0.934	0.916	0.938	0.498	0.9	0.753	0.9	0.786	0.778
CapSA	0.864	0.839	0.859	0.498	0.65	0.427	0.712	0.657	0.593
DAOA	0.941	0.913	0.9	0.498	0.913	0.75	0.902	0.775	0.768
ESOA	0.93	0.919	0.938	0.498	0.91	0.777	0.916	0.782	0.7
FVIM	0.944	0.923	0.916	0.5	0.913	0.747	0.9	0.788	0.782
HGSO	0.852	0.8	0.797	0.498	0.767	0.41	0.75	0.659	0.628
ILA	0.852	0.8	0.797	0.498	0.694	0.41	0.712	0.659	0.628
LOA	0.931	0.903	0.891	0.498	0.91	0.757	0.902	0.774	0.778
LSO	0.941	0.906	0.894	0.498	0.917	0.757	0.908	0.785	0.771

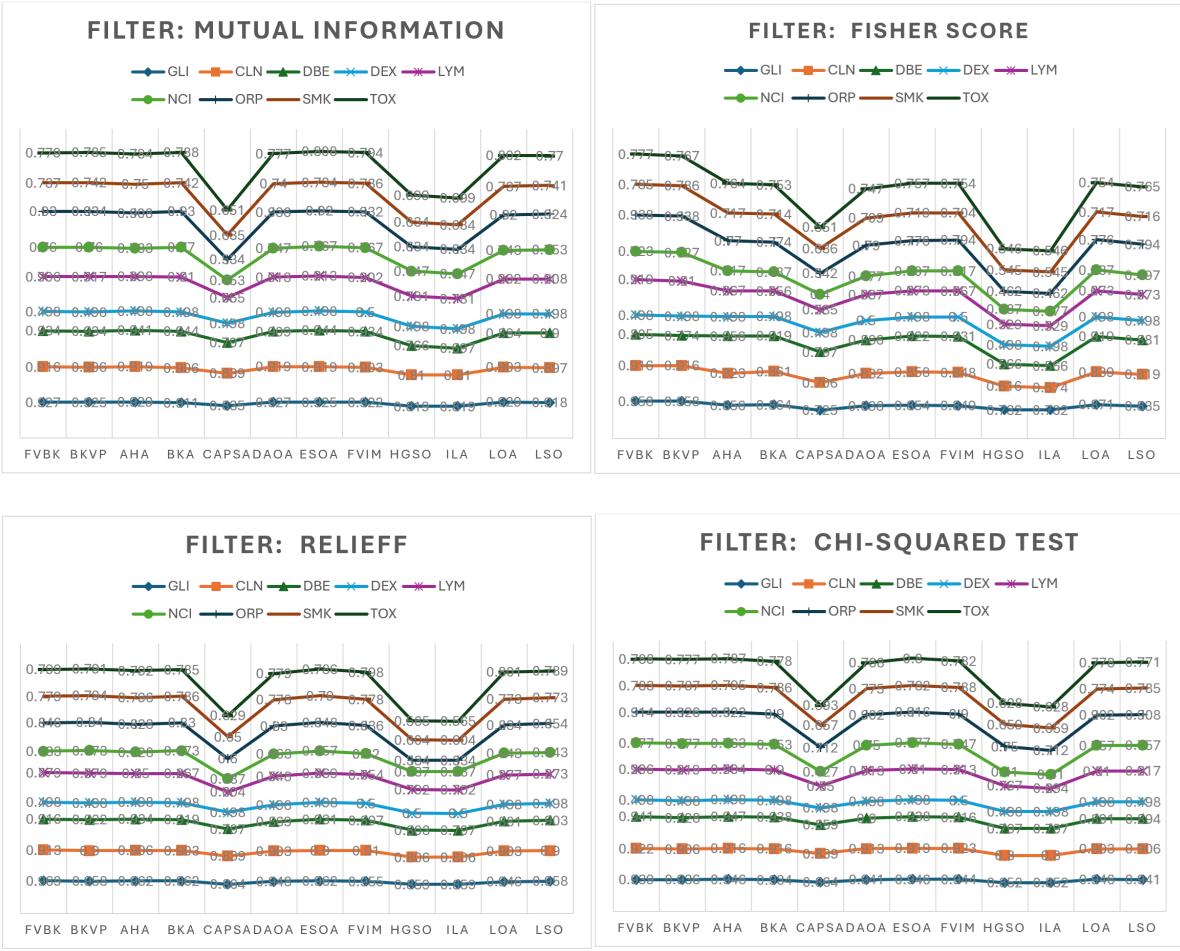


Figure 4.6: Overall accuracy of different algorithms by filters

In addition to our main findings, we emphasize the effectiveness and suitability of the filters utilized in Phase 1 of the feature selection framework. Table 4.9 illustrates the accuracy of the FVBK algorithm across various datasets, with significant results emphasized in bold for clarity. Following this, Table 4.10 presents the outcomes of the baseline algorithm, BKVP, allowing for a comparative analysis of its performance alongside the FVBK results. This comprehensive approach enables us to assess not only the algorithms' accuracies but also the impact of the chosen filters on overall effectiveness.

Table 4.9: Classification accuracy results of FVBK

Statistic metric	SVM	LDA	NB	kNN	CART	Average
GLI dataset						
Mutual Information	0.882	0.941	0.953	0.929	0.929	0.927
ReliefF	0.976	0.965	0.976	0.988	0.941	0.969
Chi-squared test	0.929	0.929	0.953	0.953	0.929	0.939
Fisher score	0.941	0.965	0.976	0.976	0.929	0.958
CLN dataset						
Mutual Information	0.903	0.919	0.903	0.919	0.935	0.916
ReliefF	0.903	0.903	0.919	0.903	0.935	0.913
Chi-squared test	0.929	0.887	0.953	0.903	0.935	0.922
Fisher score	0.935	0.903	0.903	0.903	0.935	0.916
DBE dataset						
Mutual Information	0.938	0.922	0.953	0.938	0.922	0.934
ReliefF	0.922	0.891	0.906	0.922	0.938	0.916
Chi-squared test	0.938	0.938	0.953	0.938	0.938	0.941
Fisher score	0.922	0.869	0.500	0.891	0.844	0.805
DEX dataset						
Mutual Information	0.500	0.502	0.500	0.490	0.500	0.498
ReliefF	0.500	0.502	0.500	0.490	0.500	0.499
Chi-squared test	0.500	0.502	0.500	0.490	0.500	0.498
Fisher score	0.500	0.502	0.500	0.490	0.500	0.498
LYM dataset						
Mutual Information	1.000	0.896	0.917	0.885	0.833	0.906
ReliefF	1.000	0.885	0.844	0.854	0.813	0.879
Chi-squared test	1.000	0.896	0.948	0.885	0.802	0.906
Fisher score	1.000	0.927	0.906	0.906	0.854	0.919

Table 4.9: Classification accuracy results of FVBK

Statistic metric	SVM	LDA	NB	kNN	CART	Average
NCI dataset						
Mutual Information	1.000	0.700	0.717	0.717	0.667	0.760
ReliefF	1.000	0.583	0.550	0.533	0.500	0.633
Chi-squared test	1.000	0.567	0.717	0.885	0.717	0.777
Fisher score	1.000	0.583	0.667	0.700	0.667	0.723
ORP dataset						
Mutual Information	1.000	0.950	0.920	0.930	0.850	0.930
ReliefF	1.000	0.780	0.840	0.840	0.770	0.846
Chi-squared test	1.000	0.900	0.920	0.890	0.860	0.914
Fisher score	1.000	0.910	0.970	0.970	0.840	0.938
SMK dataset						
Mutual Information	0.738	0.738	0.781	0.733	0.695	0.737
ReliefF	0.797	0.770	0.797	0.770	0.759	0.779
Chi-squared test	0.786	0.797	0.802	0.754	0.775	0.783
Fisher score	0.786	0.791	0.797	0.797	0.754	0.785
TOX dataset						
Mutual Information	1.000	0.743	0.725	0.754	0.673	0.779
ReliefF	1.000	0.766	0.789	0.754	0.684	0.799
Chi-squared test	1.000	0.731	0.713	0.719	0.775	0.788
Fisher score	1.000	0.778	0.731	0.708	0.667	0.777

Table 4.10: Classification accuracy results of BKVP

Statistic metric	SVM	LDA	NB	kNN	CART	Average
GLI dataset						
Mutual Information	0.871	0.918	0.941	0.953	0.941	0.925

Table 4.10: Classification accuracy results of BKVP

Statistic metric	SVM	LDA	NB	kNN	CART	Average
ReliefF	0.965	0.965	0.976	0.953	0.929	0.958
Chi-squared test	0.894	0.929	0.965	0.941	0.953	0.936
Fisher score	0.976	0.965	0.965	0.965	0.918	0.958
CLN dataset						
Mutual Information	0.919	0.887	0.903	0.919	0.903	0.906
ReliefF	0.887	0.887	0.903	0.903	0.919	0.900
Chi-squared test	0.871	0.903	0.903	0.919	0.935	0.906
Fisher score	0.952	0.903	0.903	0.903	0.919	0.916
DBE dataset						
Mutual Information	0.938	0.922	0.953	0.938	0.922	0.934
ReliefF	0.938	0.891	0.938	0.922	0.922	0.922
Chi-squared test	0.875	0.938	0.953	0.938	0.938	0.928
Fisher score	0.859	0.852	0.516	0.813	0.828	0.774
DEX dataset						
Mutual Information	0.500	0.502	0.500	0.490	0.500	0.498
ReliefF	0.500	0.502	0.500	0.490	0.500	0.499
Chi-squared test	0.500	0.502	0.500	0.490	0.500	0.498
Fisher score	0.500	0.502	0.500	0.490	0.500	0.498
LYM dataset						
Mutual Information	1.000	0.917	0.927	0.906	0.833	0.917
ReliefF	1.000	0.896	0.844	0.854	0.802	0.879
Chi-squared test	1.000	0.927	0.896	0.885	0.885	0.919
Fisher score	1.000	0.896	0.896	0.927	0.833	0.910
NCI dataset						
Mutual Information	1.000	0.700	0.767	0.700	0.633	0.760

Table 4.10: Classification accuracy results of BKVP

Statistic metric	SVM	LDA	NB	kNN	CART	Average
ReliefF	1.000	0.600	0.617	0.567	0.583	0.673
Chi-squared test	1.000	0.750	0.700	0.750	0.683	0.777
Fisher score	1.000	0.567	0.700	0.683	0.683	0.727
ORP dataset						
Mutual Information	1.000	0.930	0.940	0.950	0.850	0.934
ReliefF	1.000	0.770	0.820	0.810	0.800	0.840
Chi-squared test	1.000	0.930	0.900	0.930	0.870	0.926
Fisher score	1.000	0.910	0.970	0.970	0.840	0.938
SMK dataset						
Mutual Information	0.733	0.743	0.786	0.727	0.722	0.742
ReliefF	0.775	0.818	0.786	0.791	0.797	0.793
Chi-squared test	0.824	0.791	0.791	0.781	0.749	0.787
Fisher score	0.807	0.802	0.807	0.749	0.765	0.786
TOX dataset						
Mutual Information	1	0.725	0.731	0.766	0.702	0.785
ReliefF	1	0.754	0.737	0.749	0.713	0.791
Chi-squared test	1	0.713	0.737	0.719	0.713	0.777
Fisher score	1	0.749	0.713	0.69	0.684	0.767

Furthermore, Figure 4.7 depicts the Convergence Curve of the FVBK algorithm across the datasets, showcasing its performance trajectory and demonstrating the efficiency of the algorithm in reaching optimal solutions over time.

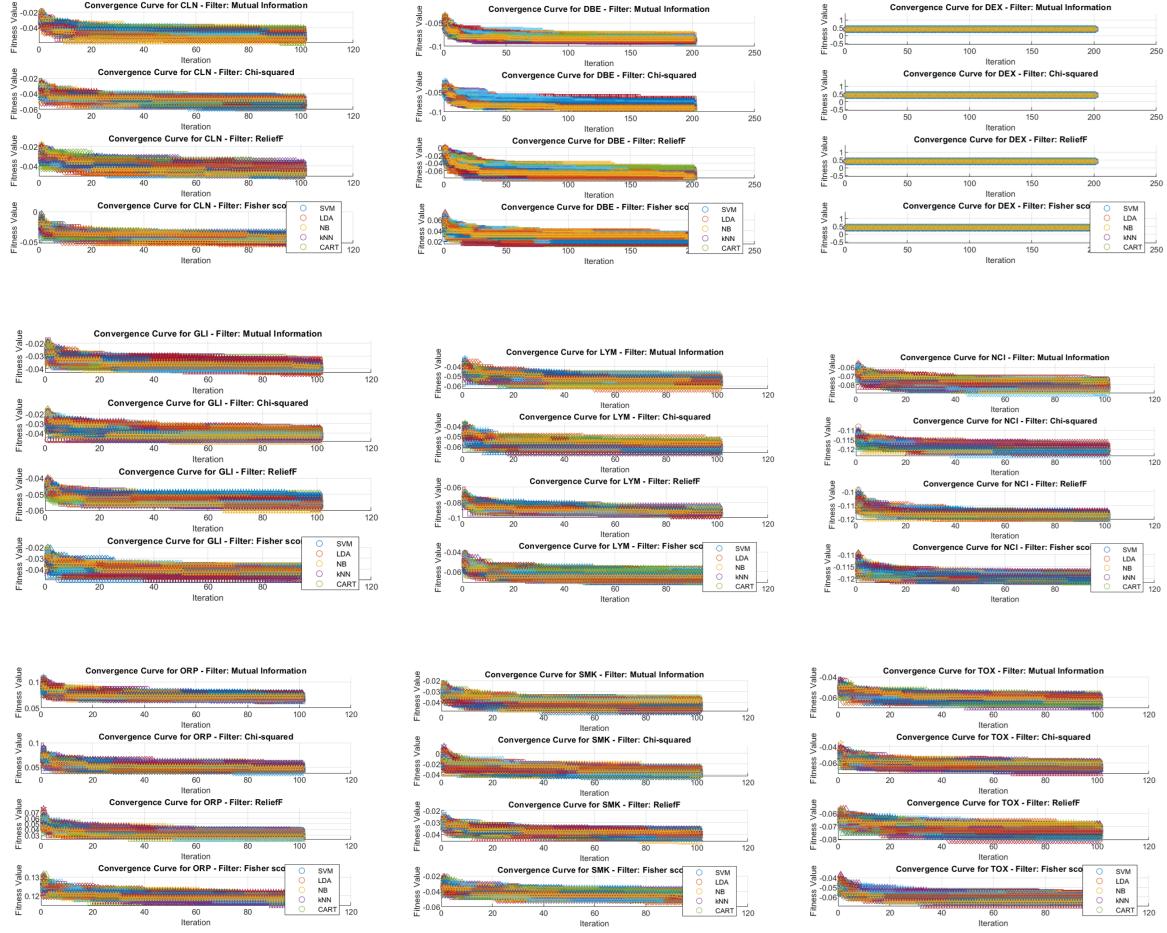


Figure 4.7: The Convergence Curve of FVBK across datasets

2 Overall performance

The Wilcoxon signed-rank test was conducted to compare the performance of the FVBK algorithm against FVIM and BKA across various datasets and filters. This non-parametric test assesses whether there is a statistically significant difference in the paired classification accuracy results between the algorithms. The test specifically examines whether the median difference between paired observations is significantly different from zero. The results reveal

that FVBK consistently outperforms FVIM when using the ReliefF filter and demonstrates superior performance compared to BKA with both the ReliefF and Chi-squared test filters. These findings highlight the ReliefF filter as the most effective option for optimizing the performance of the FVBK algorithm. The results of these comparisons, with significant differences noted, are detailed in Tables 4.11 and 4.12.

Table 4.11: Wilcoxon signed-rank test between FVBK and FVIM

Filter	p_value	Signed Rank Statistics	Winner	Losser
Mutual Information	0.9453125	19	4	4
Fisher score	0.07421875	7	7	2
ReliefF	0.01953125	3	8	1
Chi-squared test	0.49609375	16	4	5

Table 4.12: Wilcoxon signed-rank test between FVBK and BKA

Filter	p_value	Signed Rank Statistics	Winner	Losser
Mutual Information	0.7421875	21	3	5
Fisher score	0.0546875	6	8	1
ReliefF	0.02578125	15	7	2
Chi-squared test	0.01953125	3	8	1

We conducted the Friedman test to assess the statistical significance of the performance differences among the algorithms. This non-parametric test was chosen because it is well-suited for comparing multiple algorithms across several datasets, accounting for the repeated measures within the data. By ranking the algorithms on each dataset and analyzing these ranks, the Friedman test helps determine whether the observed performance differences are statistically significant or simply due to random variation. The purpose of this test was to identify if any algorithm consistently outperforms the others across the datasets, providing robust validation for the effectiveness of the proposed algorithms compared to standard meth-

ods. To ensure fairness in the comparison, both the standard algorithms and the algorithms used for comparison were integrated with a Memory Buffer and subjected to the same feature selection process, using identical parameters and classifier setups. The ranking results from the Friedman test are presented in Table 4.14, showcasing the relative performance of each algorithm across the datasets. The results indicate that FVBK consistently ranked first across most classifiers, except for the SVM, where it ranked second, just behind AHA. However, FVBK maintained a higher rank than both standard algorithms across all classifiers. Furthermore, the BKVP algorithm also demonstrated strong performance, ranking second only to FVBK in most cases, except for the SVM classifier. These findings suggest that while the standard algorithms did not achieve high ranks when tested on benchmark functions (Table 4.13), the hybrid algorithms effectively maximized their strengths, leading to outstanding performance. This highlights the superiority of hybrid approaches in enhancing algorithmic efficiency and accuracy.

Table 4.13: Ranked by Friedman test based on the Best Fitness Value

Rank	Engineering problem	CEC2013	CEC2017	CEC2006
1	AHA	CapSA	AHA	ESOA
2	CapSA	AHA	CapSA	CapSA
3	ILA	LSO	FVIM	AHA
4	BKA	LOA	ESOA	FVIM
5	FVIM	ESOA	BKA	BKA
6	ESOA	FVIM	LOA	LOA
7	HGSO	HGSO	HGSO	HGSO
8	LSO	BKA	LSO	LSO
9	DAOA	ILA	ILA	DAOA
10	LOA	DAOA	DAOA	ILA
p-value	1.69E-15	2.91E-07	3.02E-45	1.68E-30

Table 4.14: Ranked algorithms by Friedman test

Rank	SVM	LDA	NB	kNN	CART	Overall accuracy
1	AHA	FVBK	FVBK	FVBK	FVBK	FVBK
2	FVBK	BKVP	BKVP	BKVP	BKVP	BKVP
3	FVIM	AHA	FVIM	FVIM	ESOA	AHA
4	ESOA	FVIM	AHA	AHA	FVIM	ESOA
5	BKA	ESOA	ESOA	BKA	AHA	FVIM
6	DAOA	LOA	BKA	ESOA	LSO	BKA
7	BKVP	BKA	LOA	LSO	BKA	LOA
8	LOA	LSO	LSO	DAOA	DAOA	DAOA
9	LSO	DAOA	DAOA	LOA	LOA	LSO
10	CSA	HGSO	CSA	HGSO	CSA	CSA
11	HGSO	CSA	HGSO	ILA	HGSO	HGSO
12	ILA	ILA	ILA	CSA	ILA	ILA
<hr/>						
p-value	8.45E-02	3.81E-08	5.63E-08	1.29E-07	1.71E-07	4.21E-09

Chapter 5

Conclusion and future work

This project presented a novel hybrid algorithm, FVBK, integrated into a two-stage hybrid framework termed the filter/filter approach. In the first stage, various statistical metrics were utilized to identify the most relevant features, which were then used in the second stage. Here, FVBK, which incorporates the migration behavior of BKA within the FVIM framework, was employed to refine the feature selection further. Additionally, a Memory Buffer was integrated to enhance the algorithm's efficiency and effectiveness. Comprehensive experiments were carried out across five different classifiers and nine diverse datasets to assess the performance of FVBK. The results were benchmarked against two standard algorithms and several leading feature selection methods. The findings confirmed that FVBK consistently outperformed the other algorithms, showcasing its superior capability in feature selection.

Although the hybrid algorithm has achieved promising results, there remain several avenues for future research moving forward:

- Improving Filtering Algorithms: The choice of filtering algorithm and the number of features filtered out during the first stage can significantly influence the overall performance of both FVBK and other feature selection methods. Future work should focus on

developing more efficient filtering algorithms to enhance dimensionality reduction.

- Optimizing Subset Cardinality: Determining the optimal size of the feature subset is crucial. While the current method involves exploring a user-defined range of subset sizes, future research should aim to create methods that can automatically determine the ideal subset size. Additionally, advancing towards dynamic parameter settings that adapt to varying conditions—rather than relying on fixed iterations and runs—could further improve the algorithm’s performance.
- Enhanced Memory Structures: Instead of a traditional Memory Buffer, employing multi-layered memory structures or recurrent memory components may offer better historical data management, especially for highly dynamic solution spaces. Incorporating techniques like reinforcement-based memory retention could further enhance learning and adaptation.
- Testing Alternative Objective Functions: Instead of using the mRMR function as the fitness function, future research could explore alternative objective functions, such as correlation-based evaluations, to potentially improve feature relevance and selection quality.

References

- Rabia Aziz, C.K. Verma, and Namita Srivastava. Dimension reduction methods for microarray data: a review. *AIMS Bioengineering*, 4:179–197, 03 2017. doi: 10.3934/bioeng.2017.2.179.
- Hussam Fakhouri, Feras Awaysheh, Sadi Alawadi, Mohannad Alkhalaileh, and Faten Hamad. Four vector intelligent metaheuristic for data optimization. *Computing*, 106:1–39, 04 2024a. doi: 10.1007/s00607-024-01287-w.
- Jun Wang, Wen chuan Wang, Xiao xue Hu, Lin Qiu, and Hong fei Zang. Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems. *Artificial Intelligence Review*, 57(4), March 2024. doi: 10.1007/s10462-024-10723-4. URL <https://www.mathworks.com/matlabcentral/fileexchange/161401-black-winged-kite-algorithm-bka>.
- José Barrera-García, Felipe Cisternas-Caneo, Broderick Crawford, Mariam Gómez Sánchez, and Ricardo Soto. Feature selection problem and metaheuristics: A systematic literature review about its formulation, evaluation and applications. *Biomimetics*, 9(1), 2024. ISSN 2313-7673. doi: 10.3390/biomimetics9010009. URL <https://www.mdpi.com/2313-7673/9/1/9>.
- Bing Xue, Mengjie Zhang, and Will Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, 43:1656–1671, 12 2013. doi: 10.1109/TSMCB.2012.2227469.

Thabit Sabbah, Mosab Ayyash, and Mahmood Ashraf. Hybrid support vector machine based feature selection method for text classification. *The International Arab Journal of Information Technology*, 15(3A):365–374, 2018. URL http://iajit.org/2018/15/3A/15_3A_2018_365_374.pdf.

Afnan Alhassan and Wan Mohd Nazmee Zainon. Review of feature selection, dimensionality reduction and classification for chronic disease diagnosis. *IEEE Access*, PP:1–1, 06 2021. doi: 10.1109/ACCESS.2021.3088613.

Olatunji Akinola, Absalom Ezugwu, Oyelade Olaide, and Ovre Agushaka. A hybrid binary dwarf mongoose optimization algorithm with simulated annealing for feature selection on high dimensional multi-class datasets. *Scientific Reports*, 12:1–22, 09 2022. doi: 10.1038/s41598-022-18993-0.

Pooja Yadav, S. C. Sharma, Rajesh Mahadeva, and Shashikant P. Patole. Exploring hyperparameters and feature selection for predicting non-communicable chronic disease using stacking classifier. *IEEE Access*, July 2023.

Qasem Al-Tashi, Said Jadid Abdul Kadir, Helmi Md Rais, Seyedali Mirjalili, and Hitham Alhussian. Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access*, 7:39496–39508, 2019. doi: 10.1109/ACCESS.2019.2906757.

Ranya Wajih, Said Jadid Abdulkadir, NorShakirah Aziz, Qasem Al-Tashi, and Noureen Talpur. Hybrid binary grey wolf with harris hawks optimizer for feature selection. *IEEE Access*, 9: 31662–31677, 02 2021. doi: 10.1109/ACCESS.2021.3060096.

Osamah Mohammed Alyasiri, Yu-N Cheah, Ammar Kamal Abasi, and Omar Mustafa Al-Janabi. Wrapper and hybrid feature selection methods using metaheuristic algorithms for english text classification: A systematic review. *IEEE Access*, 10:39833–39852, 2022. doi: 10.1109/ACCESS.2022.3165814.

Dharmender Kumar and Manu Phogat. A hybrid metaheuristics based technique for mutation

based disease classification. *International journal of electrical and computer engineering systems*, 14:635–646, 07 2023. doi: 10.32985/ijeces.14.6.3.

Osamah Alyasiri, Yu-N Cheah, and Ammar Abasi. Hybrid filter-wrapper text feature selection technique for text classification. pages 80–86, 06 2021. doi: 10.1109/ICICT52195.2021.9567898.

Sadegh Salesi, Georgina Cosma, and Michalis Mavrovouniotis. Taga: Tabu asexual genetic algorithm embedded in a filter/filter feature selection approach for high-dimensional data. *Information Sciences*, 565, 01 2021. doi: 10.1016/j.ins.2021.01.020.

Aklima Lima, Sujoy Das, M. Ph. D., Akibur Rahman Prodeep, and Yutaka Watanobe. A comprehensive survey on the process, methods, evaluation, and challenges of feature selection. *IEEE Access*, 09 2022. doi: 10.1109/ACCESS.2022.3205618.

Prachi Agrawal, Hattan Abutarboush, Ganesh Talari, and Ali Wagdy. Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *IEEE Access*, PP:1–1, 02 2021. doi: 10.1109/ACCESS.2021.3056407.

Manal Othman and Ku Ku-Mahamud. Hybrid metaheuristic algorithms for feature selection in classification: A systematic literature review. 05 2024. doi: 10.21203/rs.3.rs-4286826/v1.

B Venkatesh and J. Anuradha. A review of feature selection and its methods. *Cybernetics and Information Technologies*, 19:3, 03 2019. doi: 10.2478/cait-2019-0001.

Tansel Dökeroğlu, Firdevsi Ayça Deniz, and Hakan Kızılıöz. A comprehensive survey on recent metaheuristics for feature selection. 494, 04 2022.

El-Sayed El-kenawy, Seyedali Mirjalili, Fawaz Alassery, Yudong Zhang, Marwa Eid, Shady El-Mashad, Bandar Aloyaydi, Abdelhameed Ibrahim, and Abdelaziz Abdelhamid. Novel metaheuristic algorithm for feature selection, unconstrained functions and engineering problems. *IEEE Access*, pages 1–1, 04 2022. doi: 10.1109/ACCESS.2022.3166901.

Shilan Hameed, Olutomilayo Petinrin, Abdirahman Hashi, and F. Saeed. Filter-wrapper combination and embedded feature selection for gene expression data. *International Journal of Advances in Soft Computing and its Applications*, 10:90–105, 03 2018.

Hussam Fakhouri, Abdelraouf Ishtaiwi, Sharif Naser Makhadmeh, and Faten Hamad. Hybrid four vector intelligent metaheuristic andde for solving complex and engineering designoptimization problems. 05 2024b. doi: 10.21203/rs.3.rs-4409293/v1.

Weiguo Zhao, Liying Wang, and Seyedali Mirjalili. Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, 388:114194, 2022. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2021.114194>. URL <https://www.sciencedirect.com/science/article/pii/S0045782521005259>.

Kang Chen, Liuxin Chen, and Gang Hu. Pso-incorporated hybrid artificial hummingbird algorithm with elite opposition-based learning and cauchy mutation: A case study of shape optimization for csgc-ball curves. *Biomimetics*, 8(4), 2023. ISSN 2313-7673. doi: 10.3390/biomimetics8040377. URL <https://www.mdpi.com/2313-7673/8/4/377>.

P. N. Suganthan J. J. Liang N. H. Awad, M. Z. Ali and B. Y. Qu. Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization. *Artificial Intelligence Review*, pages 8–28, October 2016. doi: 10.1109/CEC38257.2017.

Malik Braik, Alaa Sheta, and Heba Al-Hiary. A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. 09 2021. doi: 10.1007/s00521-020-05145-6.

Nima Khodadadi, Vaclav Snasel, and Seyedali Mirjalili. Dynamic arithmetic optimization algorithm for truss optimization under natural frequency constraints. *IEEE Access*, pages 1–1, 01 2022. doi: 10.1109/ACCESS.2022.3146374.

Zuyan Chen, Adam Francis, Shuai Li, Bolin Liao, Dunhui Xiao, Tran Thu Ha, Jianfeng Li, Lei Ding, and Xinwei Cao. Egret swarm optimization algorithm: An evolutionary computation approach for model free optimization. *Biomimetics*, 7(4), 2022. ISSN 2313-7673. doi: 10.3390/biomimetics7040144. URL <https://www.mdpi.com/2313-7673/7/4/144>.

Fatma A. Hashim, Essam H. Houssein, Mai S. Mabrouk, Walid Al-Atabany, and Seyedali Mirjalili. Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems*, 101:646–667, 2019. ISSN 0167-739X. doi: <https://doi.org/10.1016/j.future.2019.07.015>. URL <https://www.sciencedirect.com/science/article/pii/S0167739X19306557>.

Masoomeh Mirrashid and Hosein Naderpour. Incomprehensible but intelligible-in-time logics: Theory and optimization algorithm. *Knowledge-Based Systems*, 264:110305, 2023. ISSN 0950-7051. doi: <https://doi.org/10.1016/j.knosys.2023.110305>. URL <https://www.sciencedirect.com/science/article/pii/S0950705123000552>.

Mohammad Dehghani, Gulnara Bektemyssova, Zeinab Montazeri, Galymzhan Shaikemelev, O.P. Malik, and Gaurav Dhiman. Lyrebird optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics*, 8:507, 10 2023. doi: 10.3390/biomimetics8060507.

Mohamed Abdel-Basset, Reda Mohamed, Karam M. Sallam, and Ripon K. Chakrabortty. Light spectrum optimizer: A novel physics-inspired metaheuristic optimization algorithm. *Mathematics*, 10(19), 2022. ISSN 2227-7390. doi: 10.3390/math10193466. URL <https://www.mdpi.com/2227-7390/10/19/3466>.

Hadi Bayzidi, Siamak Talatahari, Meysam Saraei, and Charles-Philippe Lamarche. Social network search for solving engineering optimization problems. *Computational Intelligence and Neuroscience*, 2021:32, 09 2021. doi: 10.1155/2021/8548639.

Jing Liang, Thomas Runarsson, Efrén Mezura-Montes, Maurice Clerc, Ponnuthurai Suganthan, Carlos Coello, and Kalyan Deb. Problem definitions and evaluation criteria for the cec

2006 special session on constrained real-parameter optimization. *Nanyang Technological University, Singapore, Tech. Rep*, 41:16–17, 01 2006.

Xiaodong Li, Ke Tang, Mohammmd Nabi Omidvar, Zhenyu Yang, and Kai Qin. Benchmark functions for the cec'2013 special session and competition on large-scale global optimization. 01 2013.

Zeng Meng, Gang Li, Xuan Wang, Sadiq Sait, and Ali Yildiz. A comparative study of metaheuristic algorithms for reliability-based design optimization problems. *Archives of Computational Methods in Engineering*, 28, 05 2020. doi: 10.1007/s11831-020-09443-z.

Zbigniew Telec Tadeusz Lasota Bogdan Trawiński, Magdalena Smetek. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *International Journal of Applied Mathematics and Computer Science*, 22(4):867–881, 2012. URL <http://eudml.org/doc/244548>.

Mohammad Akbari, Mohsen Zare, Rasoul Azizipanah-abarghooyee, Seyedali Mirjalili, and Mohamed Deriche. The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems. *Scientific Reports*, 12:10953, 06 2022. doi: 10.1038/s41598-022-14338-z.

Vinod Chandra and Anand S. Nature inspired meta heuristic algorithms for optimization problems. *Computing*, 104:1–19, 02 2022. doi: 10.1007/s00607-021-00955-5.

V. Tomar, M. Bansal, and P. Singh. Metaheuristic algorithms for optimization: A brief review. *Engineering Proceedings*, 59:238, 2023. doi: 10.3390/engproc2023059238. URL <https://doi.org/10.3390/engproc2023059238>.

Zeng Meng, Gang Li, Xuan Wang, Sadiq M. Sait, and Ali Rıza Yıldız. A comparative study of metaheuristic algorithms for reliability-based design optimization problems. *Archives of Computational Methods in Engineering*, 28(3):1853–1869, May 2021. ISSN 1886-1784. doi: 10.1007/s11831-020-09443-z. URL <https://doi.org/10.1007/s11831-020-09443-z>.

Byeng Dong Youn, Kyung Choi, and Y.H. Park. Hybrid analysis method for reliability-based design optimization. *Journal of Mechanical Design - J MECH DESIGN*, 125, 06 2003. doi: 10.1115/1.1561042.

Héctor Migallón, Akram Belazi, José-Luis Sánchez-Romero, Héctor Rico, and Antonio Jimeno-Morenilla. Settings-free hybrid metaheuristic general optimization methods. *Mathematics*, 8(7), 2020. ISSN 2227-7390. doi: 10.3390/math8071092. URL <https://www.mdpi.com/2227-7390/8/7/1092>.

Trinav Bhattacharyya, Bitanu Chatterjee, Pawan Kumar Singh, Jin Hee Yoon, Zong Woo Geem, and Ram Sarkar. Mayfly in harmony: A new hybrid meta-heuristic feature selection algorithm. *IEEE Access*, 8:195929–195945, 2020. doi: 10.1109/ACCESS.2020.3031718.

Anezka Kazikova, Michal Pluhacek, and Roman Senkerik. How does the number of objective function evaluations impact our understanding of metaheuristics behavior? *IEEE Access*, 9: 44032–44048, 2021. doi: 10.1109/ACCESS.2021.3066135.

Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of reliefF and rreliefF. *Machine Learning*, 53(1):23–69, Oct 2003. ISSN 1573-0565. doi: 10.1023/A:1025667309714. URL <https://doi.org/10.1023/A:1025667309714>.

Hussam N. Fakhouri, Ahmad Sami Al-Shamayleh, Abdelraouf Ishtaiwi, Sharif Naser Makhadmeh, Sandi N. Fakhouri, and Faten Hamad. Hybrid four vector intelligent metaheuristic with differential evolution for structural single-objective engineering optimization. *Algorithms*, 17(9), 2024c. ISSN 1999-4893. doi: 10.3390/a17090417. URL <https://www.mdpi.com/1999-4893/17/9/417>.

Zheng Zhang, Xiangkun Wang, and Yinggao Yue. Heuristic optimization algorithm of black-winged kite fused with osprey and its engineering application. *Biomimetics*, 9(10), 2024. ISSN 2313-7673. doi: 10.3390/biomimetics9100595. URL <https://www.mdpi.com/2313-7673/9/10/595>.

M. Haohao, A. As'arry, F. Yanwei, et al. Improved black-winged kite algorithm and finite element analysis for robot parallel gripper design. *Advances in Mechanical Engineering*, 16 (10), 2024. doi: 10.1177/16878132241288402.