Multi-objective Feature Selection Optimization Using NSGA-2

Hamza Rangwala
Department of Science
Wilfrid Laurier University
Waterloo, ON, Canada
hamzarangwala51@gmail.com

Azam Bidgoli
Faculty of Science
Wilfrid Laurier University
Waterloo, ON, Canada
aasilianbidgoli@wlu.ca

Shahryar Rahnamayan
Department of Engineering
Brock University
St. Catherines, ON, Canada
srahnamayan@brocku.ca

ABSTRACT

Feature selection is critical in machine learning, particularly with high-dimensional datasets and decision making while handling big data is getting difficult. This paper introduces an innovative approach using the NSGA-II (Non-dominated Sorting Genetic Algorithm II) for feature selection, aiming to minimize the number of features and classification error simultaneously. A custom function was implemented to facilitate this process. The K-Nearest Neighbors (KNN) classifier evaluated the selected features across multiple runs and folds to ensure robustness. The methodology involved optimizing feature selection on training sets and evaluating on validation sets, identifying Pareto front solutions, and testing on the test set for generalization capability. Results showed effective feature reduction while maintaining or improving classification accuracy. Key metrics, such as average training error, minimum validation error, and test error, were tracked to highlight consistency. Non-dominated binary vectors of optimal feature subsets were analyzed for further feature selection via voting. This framework aims to narrow the disparity between the classification error observed on the training and test datasets, thereby enhancing model generalization. This study provides a robust framework for feature selection for a set of features having high dimension, leveraging evolutionary algorithms and cross-validation techniques to enhance machine learning model performance and reduce the error difference between train and test sets while solving the issue of data driven evolutionary computation.

KEYWORDS

Insert keyword text, Insert keyword text, Insert keyword text, Insert keyword text

ACM Reference format:

FirstName Surname, FirstName Surname and FirstName Surname. 2018. Insert Your Title Here: Insert Subtitle Here. In *Proceedings of ACM Woodstock conference (WOODSTOCK'18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1. INTRODUCTION

In the era of big data, the exponential growth of available information has led to increasingly high-dimensional datasets, posing significant challenges for machine learning applications. These datasets, characterized by many features, often include redundant, irrelevant, or noisy data, which can negatively impact the performance of machine learning models. Consequently, feature selection has emerged as a critical preprocessing step to enhance model accuracy, reduce computational complexity, and improve interpretability. Given the recent explosion in the dimensionality of real-world datasets, which now often contain thousands or even millions of features, the importance of feature selection methods has surged dramatically [1]. Big data is playing an important role for many years due to the vast and vide growth of data in various fields and it has become difficult to find the sense from all those large collective data. Since there is an increase in the large amount of data it is difficult to make better decisions [2].

This study aims to identify the most relevant subset of features that contribute significantly to the predictive accuracy of a model which in turn helps us in better decision making. By eliminating unnecessary features, the dimensionality of the data is reduced, leading to simpler models that are easier to interpret and faster to train. Moreover, effective feature selection can mitigate the risk of overfitting, thereby enhancing the generalization capability of the model on unseen data. We define the feature selection as a problem including two competing objectives and we try to find a set of optimal solutions so called Pareto-optimal solutions instead of a single optimal solution.[3]

Feature selection is reported to be an NP-hard problem that tries to find the best subset from 2^{N-1} possible subsets of features given a dataset with N features [4]. Traditional feature selection methods, such as filter, wrapper, and embedded techniques, have been widely used. However, these methods often struggle with the curse of dimensionality and may not efficiently explore the vast search space of potential feature subsets. To address these limitations, evolutionary algorithms have gained popularity due to their ability to perform global searches and handle complex optimization problems. Among these, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) has shown promise in multi-objective optimization tasks, making it a suitable candidate for feature selection.

Multi-objective optimization algorithms such as genetic algorithms (GAs) have been successfully applied to various optimization problems, ranging from evolutionary design [5]. This study introduces an innovative approach utilizing NSGA-II for feature selection, with a dual objective of minimizing the number of features and the classification error simultaneously. It aims to solve the data driven evolutionary computation problem starting with this procedure of finding a robust set of features.

Simulation results on difficult test problems show that the proposed NSGA-II, in most problems, is able to find much better spread of solutions and better convergence near the true Pareto-optimal front compared to Pareto-archived evolution strategy and strength-Pareto EA—two other elitist MOEAs that pay special attention to creating a diverse Pareto-optimal front [6] .The NSGA-II algorithm is well-suited for this task as it can effectively balance the trade-off between these conflicting objectives by identifying a set of Pareto optimal solutions. Each solution on the Pareto front represents a potential feature subset that optimizes the trade-off between the two objectives.

To facilitate the feature selection process, a custom function was developed which is used to diversify the initial population based on the number of features and generating random values to set the cells of the individuals and integrated with the NSGA-II algorithm. The K-Nearest Neighbor Algorithm is the simplest of all machine learning algorithms. It is based on the principle that the samples that are similar generally lie in close vicinity [7]. The selected feature subsets were evaluated using the K-Nearest Neighbors (KNN) classifier, a widely used non-parametric method known for its simplicity and effectiveness. The performance of the KNN classifier, in conjunction with the selected features, was assessed across multiple runs and folds of the dataset, ensuring the robustness and generalizability of the results.

The dominance-based MOEAs evaluate solutions based on dominance relationship and select solutions according to Pareto-based and diversity-based selection criteria [8]. The methodology employed in this study involved dividing the dataset into training and validation sets. The feature selection process was optimized on the training set, and the selected features were then evaluated on the validation set to identify Pareto front solutions. These solutions were further analyzed for their performance on the validation set, and the subsets with the minimum classification error were tested on a separate test set to evaluate their generalization capability.

The results of this study demonstrated that the proposed approach effectively reduced the number of features while improving classification accuracy. Key performance metrics, such as average training error, average minimum validation error, and average test error which were tracked across multiple runs, highlighting the consistency and reliability of the feature selection process. Additionally, the non-dominated binary vectors representing the optimal feature subsets were saved and analyzed for further selection using a voting mechanism, providing valuable insights into the feature selection process.

In summary, this research offers a robust framework for feature selection and finding a more robust set of features on different set of datasets which are high in dimension, leveraging the strengths of evolutionary algorithms and rigorous cross-validation techniques. The findings contribute to the field by offering a scalable and effective solution for feature selection, potentially improving the performance of various machine learning models and we expect that after using our voting mechanism our set of features will help in reducing the difference present between the training and test sets.

2. RELATED WORK

2.1 NSGA-II Algorithm vs. other MOEAs

Compared with NSGA-II, NSGA-III has no obvious advantage for multi-objective optimization problems with two or three objectives because of their general similar structures [9]. The main difference between NSGA-II and NSGA-III lies in the selection mechanism. NSGA-II adopts the crowding distance approach to select non-dominated solutions, ensuring diversity by maintaining a spread of solutions across the Pareto front. In contrast, NSGA-III utilizes a selection operator based on reference points to choose non-dominated solutions, which is particularly advantageous for problems with a large number of objectives. Despite these differences, for problems with fewer objectives, both algorithms perform similarly due to their comparable structures and mechanisms for maintaining diversity and convergence in the solution space

Multi objective EAs using fitness based on dominance counting and identification of nondominated solutions can be improved significantly in terms of running time by using efficient algorithms known from computer science instead of inefficient O (MN/sup 2/) algorithms [10]. These improved algorithms reduce computational complexity and enhance the efficiency of identifying and maintaining non-dominated solutions in large and complex problem spaces. This advancement is crucial for handling the increasing computational demands of modern multi-objective optimization problems, making these EAs more practical and scalable for real-world applications.

Computational experiments that NSGA-III does not always outperform NSGA-II even for ten-objective problems. That is, their comparison results depend not only on the number of objectives but also on the type of test problems. The choice of test problems has a larger effect on their comparison results than the number of objectives in our computational experiments [11]. The nature of the test problems can have a more substantial impact on the results than the sheer number of objectives, highlighting the importance of choosing appropriate benchmark problems for algorithm evaluation. These findings emphasize that the effectiveness of NSGA-III over NSGA-II is not universal and must be assessed in the context of specific problem characteristics and domains.

2.2 NSGA-II Algorithm

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is a widely used evolutionary algorithm designed for solving multi-objective optimization problems. NSGA-II improves upon its predecessor, NSGA, by addressing key issues such as computational complexity, lack of elitism, and the need for specifying a sharing parameter. NSGA-II outperforms two other contemporary MOEAs: Pareto-archived evolution strategy (PAES) and strength Pareto EA (SPEA) in terms of finding a diverse set of solutions and in converging near the true Pareto-optimal set [6]. NSGA-II operates by simulating the process of natural evolution. The algorithm maintains a population of potential solutions that evolve over successive generations. Each solution in the population is evaluated based on multiple objective functions, which, in this study, are the minimization of feature count and classification error.

In optimisation problems, the location of the global optimum solution is unknown a priori, and initialisation is a stochastic process. The initialisation control parameters of population-based metaheuristic algorithms play a significant role in improving the performance of the algorithms [12]. During the initialization of the algorithm the population needs to be diverse, it will ensure that the initial population covers the overall solution space and converges to an optimal set of solutions. We generate individuals with a value in between 1 and n where n represents the total number of features in the dataset. After doing that we set n random cells of the individual to 1, where n represents the number of selected features in the individual. This is being reiterated multiple times to suffice the number of individuals required to accommodate the initial population size. Generating the initial population is one of the important steps in evolutionary algorithms. A poor initial population may unnecessarily increase the number of searche's or it may cause the algorithm to converge at local optima [13].

Similar to other population-based evolutionary algorithms, a set of uniform random individuals are produced as an initial population. Feature selection is originally a high dimensional binary optimization problem. Each individual in the population is a vector in the length of the number of all features. The variable in the vector indicate the selection status of the features by using a binary value (0 or 1). The value of one indicates the selection of corresponding feature and value of zero means that the feature is not selected. In order to compute the error of classification on each subset (each solution in the population), features associated with value of one are selected to form the dataset.[14]

3. PROPOSED METHODOLOGY

Our methodology integrates evolutionary algorithms namely NSGA-II with the K-Nearest Neighbors (KNN) classifier to identify optimal feature subsets. By minimizing both the number of features and classification error, we aimed to enhance model performance while ensuring robustness through cross-validation.

3.1 Tailored Feature Selection Methodology

We developed a function for the feature selection process which is designed to optimize two primary objectives:

1) Minimizing the Number of Features.
2) Minimizing the Classification Error.

Minimizing Feature Count: This function calculates the total number of features selected in each solution. The fewer the features, the better the solution is considered in this context.

Minimizing Classification Error: This function evaluates the classification error using the K-Nearest Neighbors (KNN) classifier. It identifies the selected features from the binary vector and trains the KNN classifier on these features. The classification error is computed as one minus the accuracy of the classifier on the training data.

The initialization of the function depends upon the number of decision variables corresponding to the total number of features in the dataset, two objective functions, and one inequality constraint. The constraint ensures that at least one feature is selected in each solution. This prevents the algorithm from considering solutions with no selected features, which would be meaningless for the classification task. The evaluation function processes the population of solutions, calculating the values for both objective functions and the constraint for each solution. It evaluates the number of features selected, the classification error, and ensures the constraint is satisfied. This function's evaluation method is invoked to assess the fitness of each solution in the population. This involves computing the number of features, the classification error, and applying the constraint to ensure validity. The algorithm iterates through multiple generations, continuously evaluating and evolving the population towards optimal solutions.

This function is integrated with the NSGA-II algorithm to achieve an effective multi-objective optimization for feature selection. The following details describe the implementation and integration process.

3.2 Evaluation Procedure for Feature Selection

In this study, we employ a thorough experimental procedure to assess the effectiveness of our proposed feature selection method using NSGA-II. The process begins by dividing the dataset into separate training and test sets. The test set is set aside for final evaluation, ensuring that its class distribution is consistent with the overall dataset. The training set is further divided into multiple subsets using K-Fold cross-validation, which allows different portions of the data to be used for training and validation in each iteration, enhancing the robustness and generalizability of the model.

During the optimization phase, the NSGA-II algorithm is applied to the training data to identify optimal feature subsets. Each solution generated by the algorithm is evaluated on the validation set, where the number of selected features and the validation error are calculated. Non-Dominated Sorting (NDS) is then performed on these solutions to create a Pareto front plot, which helps in identifying solutions with minimal error rates. This process is repeated across multiple iterations, and the average error rate on the validation set is computed for further analysis.

The non-dominated binary vectors representing the optimal feature subsets are saved for additional analysis on the test data. From these solutions, we identify the one with the minimum error rate and evaluate it on the initial test set. This involves retrieving the test data and using the selected features to make predictions. The accuracy obtained from these predictions is used to calculate the average test error across multiple runs, providing insights into the performance of the feature subsets and their associated features.

3.3 Voting Mechanism

In this study, a voting mechanism is implemented to further refine the selection of non-dominated binary vectors representing optimal feature subsets which are obtained from the solutions on the validation set. Each run of the optimization process results in a set of non-dominated binary vectors, representing feature subsets that have the minimum error rate on the validation set. These binary vectors are saved to files for each run and later loaded for analysis. An array is initialized to count the number of times each feature is selected across all runs. For each solution in the Pareto front, the corresponding feature positions in the array are incremented by one. Features are sorted based on their selection frequency in descending order, and the top features are identified. Figure 1 gives a visual representation of the binary vector showcasing that if a feature is present it is represented with 1 and 0 if it is absent, across multiple runs that would be required for tally the frequency of the features for the voting mechanism.

The top features are based on their selection frequency and are used to create a new feature subset. This subset is then used to train a K-Nearest Neighbors (KNN) classifier on the training data having the most n frequent features selected, and the classifier's performance is evaluated on the test set which has been restored from the machine to determine the minimum error rate on test after voting and choosing the most frequent features. This process is repeated iteratively for the top n features, by selecting the most frequently occurring feature in the first iteration and subsequently adding the next most frequent feature in each subsequent run. The average of the number of features across n iterations on the validation front would give an estimate of the number of features that need to be selected from the most frequent feature list. This would let us validate the frequent features on test set which has been restored.

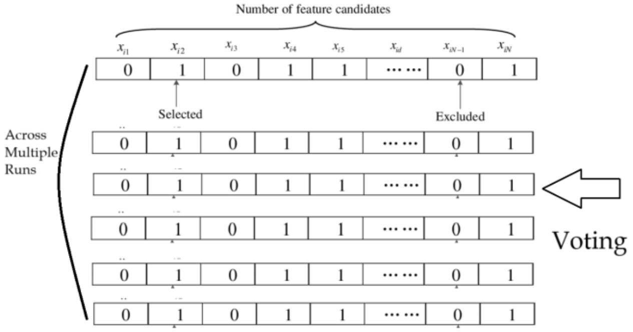

Number of feature candidates
x_{i1} x_{i2} x_{i3} x_{i4} x_{i5} x_{id} x_{iN-1} x_{iN}
0 1 0 1 1 ⋯⋯ 0 1
Selected
Excluded
Across Multiple Runs
Voting

Figure 1. Binary vector solutions considered during voting

3.4 Evaluation Metrics

Evaluating the performance of the feature selection process is critical to ensure that the selected features lead to an effective and robust machine learning model. In this study, several key metrics are used to assess the quality of the feature selection process, including average training error, minimum validation error, and test error after voting. Each of these metrics provides unique insights into the model's performance at different stages of the selection and evaluation process.

Average Training Error = $1/N \sum_{i=1}^{N} E_{train}^{(i)}$

where$E_{train}^{(i)}$ is the training error for the ith run or fold, and N is the total number of folds.

Average training error is the mean classification error observed on the training dataset over multiple runs and folds. It provides an indication of how well the model fits the training data. A lower average training error suggests that the model can capture the underlying patterns in the training data effectively.

Average Minimum Validation Error = $1/N \sum_{i=1}^{N} \min \{Eval_{val}^{(i)}\}$

Average minimum validation error is the lowest classification error observed on the validation dataset during the feature selection and model training process. Hyper parameter tuning guides the tuning process to find the optimal settings that minimize the error on the validation set, reducing the risk of overfitting. It serves as a critical metric for model selection and hyperparameter tuning. Helps in selecting the model configuration (e.g., feature subset, hyperparameters) that generalizes best to unseen data.

Test Error After Voting = $1-\frac{1}{N}\sum_{i=1}^{N} I\ (\hat{y}_i = y_i)$

After the optimization process using NSGA-II, the feature selection framework includes a voting mechanism to further refine the selection of features. This step involves identifying the most frequently selected features across the binary vector solution having the lowest classification error on the validation pareto front and using these to build the final model. The test error after voting is a crucial metric to evaluate the effectiveness of this approach in generalizing to unseen data. This approach aims to produce robust and efficient machine learning models suitable for high-dimensional datasets.

Systematically evaluating these metrics, the research demonstrates the effectiveness of the proposed feature selection method in improving the overall performance of machine learning models, making it a valuable contribution to the field.

4. EXPERIMENTAL SETUP

This section includes the details of the datasets that are being used, the steps taken to carry out this unique framework for splitting, evaluation and optimization of the datasets for feature selection as well as the models and techniques used to bridge the gap between the training and test error which has been significant earlier.

4.1 Dataset Description

This study utilized a diverse set of biological datasets to evaluate the feature selection process. The datasets vary in size, number of features, and classification tasks, providing a comprehensive assessment of the proposed method's performance. Primarily the datasets have been taken from a category representing the biological data with ranges of instance between 50 to 174 and the feature ranges in between 2000 and 24481, it spans a broad range in terms of both the number of instances and features, providing a comprehensive basis for evaluating the proposed feature selection method across various biological contexts. The classification of the datasets includes both binary and multi-class.

By covering a wide range of instances, features, and classification types, these datasets provide a robust framework for assessing the efficacy and generalizability of the proposed feature selection method in various biological contexts.

4.2 Experimental Procedure

This study follows a comprehensive experimental process to evaluate the effectiveness of the proposed feature selection method using NSGA-II. The process includes dataset splitting, optimization using the NSGA-II, validation on train, and testing after validation. The dataset is initially split into training and test sets using `train_test_split` with a 70-30 ratio. This ensures that 30% of the data is reserved for final testing. The split is stratified based on class labels to maintain the class distribution in both sets. The training set is further divided into training and validation sets in the ration of 70-30 using K-Fold cross-validation where K=10. This ensures that different portions of the data are used for training and validation in each fold, promoting robustness and generalizability. We will be taking the training set (70 of 70 %) after cross-validation for multiple iterations and feeding that data to the algorithm for optimization and getting solutions from the pareto optimal front made by the algorithm.

A function has been designed which evaluates each solution from the optimization function of the algorithm on the validation set (30 of 70%), computing the number of selected features and validation error. After doing that, we perform NDS (Non-Dominant Sorting) on the solutions to plot a pareto front using the validation set. This plot helps us in identifying the solutions with the minimum error rate and calculating the average error rate on validation across multiple runs, which will help in further analysis. The non-dominated binary vectors representing optimal feature subsets are also saved for further analysis on independent test data for frequent feature selection optimization. The identification of the solution having the minimum error rate is done from these solutions and using that solution we evaluate it on the test data set which had been initially split and stored on the local machine. The test data is firstly retrieved using the pickle library and after that it is used to evaluate on the solution containing the minimum error rate on validation pareto front. The accuracy calculated using these solutions helps us in calculating the average test error for every run and the features associated with them.

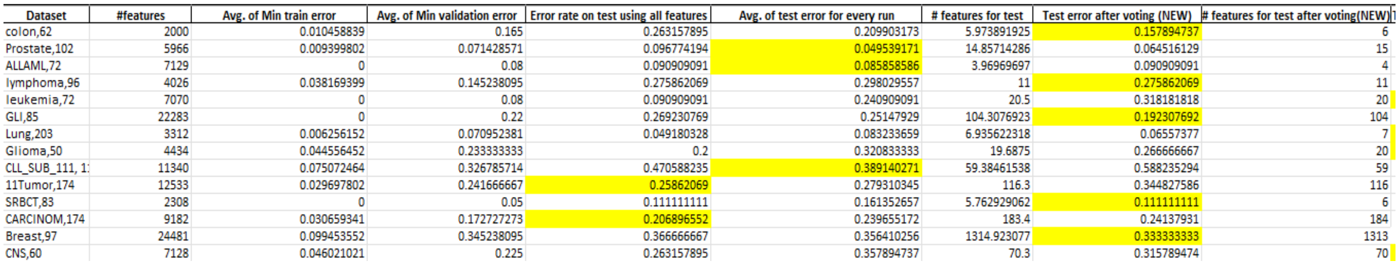
The voting mechanism comes into effect after this step, once we get a solution having the minimum error rate on the validation pareto front, we calculate the number of frequent features using the voting mechanism which calculates the frequency or the occurrence of a particular feature across 10 runs on the validation front and assigns an index value to every feature. The binary vector which we stored earlier is retrieved and is used to help us in the process. The frequency of each feature being selected is printed and analyzed. This helps to identify the features that are consistently selected across multiple runs. Features are sorted based on their selection frequency in descending order, and the top 100 features are identified. Once we get the frequent features list across the 10 runs, they can then used to select those features from the training set which are present in the frequent feature list. Now, to decide the number of features from the top frequent features list which needs to be selected, we use the average of the number of features (Min. Error from validation front) across multiple runs. For a dataset named Lymphoma having the average number of features selected after the optimization and validation on the test set is close to 11, then to calculate the error on test we would use the top 11 most frequent features from the list and select those top 11 features to evaluate it on test. This subset of features is then used to train a K-Nearest Neighbors (KNN) classifier on the training data using selected features, and the classifier's performance is evaluated on the test set. The test error is computed for the selected feature subsets, and the results are analyzed to understand the impact of the voting mechanism on the test set performance.

The voting mechanism for feature selection is an effective approach to refine the feature subsets obtained from the NSGA-II optimization process.

5. Results and Analysis

The experiments conducted in this study focused on evaluating the effectiveness of the proposed feature selection method using NSGA-II. The performance of the selected feature subsets through optimization was assessed based on two primary metrics: the number of features and the classification accuracy. The goal was to minimize the number of features and the classification error. The proposed voting mechanism and framework of optimization, validation and using those solutions on test set have shown a more robust set of features and created a model that has shown these features are essential while considering feature selection. The problem while implementing a machine learning model on a dataset having a lot of features is reducing the redundant features and reduce the computational time on a system. The selected feature subsets consistently performed well on the test sets after voting, achieving test accuracies in the range of 41% to 93% according to each datasets considered.

The datasets considered in the experiment are very diverse in nature and provide a clear indication of the methods performance. The reduction in the number of features while maintaining classification accuracy is crucial for decision-making in various applications. By identifying the most relevant features, the proposed method allows for the development of more efficient and interpretable models. The use of NSGA-II enables the identification of Pareto-optimal solutions that represent the best trade-offs, thus providing a robust framework for feature selection. The results indicate that the feature subsets selected using the proposed voting method demonstrate strong generalization capability, as evidenced by high test accuracies and low-test error rates across a variety of datasets. This demonstrates that the proposed method is effective in identifying the most relevant features that contribute to accurate classification, even when applied to unseen data.



Dataset	#features	Avg. of Min train error	Avg. of Min validation error	Error rate on test using all features	Avg. of test error for every run	# features for test	Test error after voting (NEW)	# features for test after voting(NEW)
colon,62	2000	0.010458839	0.165	0.263157895	0.209903173	5.973891925	0.157894737	6
Prostate,102	5966	0.009399802	0.071428571	0.096774194	0.049539171	14.85714286	0.064516129	15
ALLAML,72	7129	0	0.08	0.090909091	0.085858586	3.96969697	0.090909091	4
lymphoma,96	4026	0.038169399	0.145238095	0.275862069	0.298029557	11	0.275862069	11
leukemia,72	7070	0	0.08	0.090909091	0.240909091	20.5	0.318181818	20
GLI,85	22283	0	0.22	0.269230769	0.25147929	104.3076923	0.192307692	104
Lung,203	3312	0.006256152	0.070952381	0.049180328	0.083233659	6.935622318	0.06557377	7
Glioma,50	4434	0.044556452	0.233333333	0.2	0.320833333	19.6875	0.266666667	20
CLL_SUB_111, 1:	11340	0.075072464	0.326785714	0.470588235	0.389140271	59.38461538	0.588235294	59
11Tumor,174	12533	0.029697802	0.241666667	0.25862069	0.279310345	116.3	0.344827586	116
SRBCT,83	2308	0	0.05	0.111111111	0.161352657	5.762929062	0.111111111	6
CARCINOM,174	9182	0.030659341	0.172727273	0.206896552	0.239655172	183.4	0.24137931	184
Breast,97	24481	0.099453552	0.345238095	0.366666667	0.356410256	1314.923077	0.333333333	1313
CNS,60	7128	0.046021021	0.225	0.263157895	0.357894737	70.3	0.315789474	70

Figure 2. Overview of the results containing dataset description, average minimum error on train, validation, test error on ###

In Figure 2. we see that to determine the number of frequent features that are being selected are close to the average number of features that were considered during the validating it on the test-set after getting the solution from the validation set having the least error. We are solving the data-driven evolutionary computation problem by optimizing feature selection in a manner that balances the trade-off between feature reduction and classification performance starting with this method. We expect that that after voting our set of features will validate the effectiveness in achieving our set objectives. In Figure 2. it demonstrates that the set of features selected after voting consistently performed better than the classification error using all the features of the dataset.

The scalability of the proposed method is demonstrated by its successful application to datasets with varying sizes and feature counts. The use of NSGA-II allows for efficient handling of high-dimensional data, making it suitable for large-scale problems. The method can be parallelized to further enhance its scalability, making it practical for real-world applications. Potential applications of the proposed method extend beyond biological data to other domains, include healthcare, finance, image and text classification, engineering. It can be used in in the healthcare sector in identifying key biomarkers for disease diagnosis and prognosis.

It could also be used in enhancing the interpretability and efficiency of predictive models in medical research helping in creating cancer research studies etc. The models can help in selecting relevant features for credit scoring models. It could help in the finance sector with improving the accuracy and efficiency of algorithmic trading strategies to help stockbrokers and traders. In the classification sector it would help in reducing the dimensionality of image and text data for more efficient and accurate classification, enhancing the performance of deep learning models by focusing on the most informative features. In the engineering department using of IOT products and collecting information from sensors can be used to create an anomaly detection in sensor networks optimizing the feature selection in it. It can also increase the efficiency of predictive maintenance models in industrial applications. Industry 4.0 could take help of these models in creating an environment of sustainable and reliable tools. The proposed NSGA-II based feature selection method demonstrates strong generalization capability and competitive performance compared to traditional methods. Its scalability and versatility make it a valuable tool for various machine learning applications, enabling more efficient and accurate data-driven decision-making.

6. Conclusion remarks

This study presents an innovative approach to feature selection (FS) utilizing the NSGA-II optimization algorithm and a comprehensive experimental methodology. Our experiments involved dataset splitting, optimization, validation, and rigorous testing. The primary aim was to evaluate the effectiveness of our feature selection method in reducing the number of features while maintaining or improving classification accuracy. By employing a K-Nearest Neighbors (KNN) classifier, we ensured robust validation and testing of selected feature subsets.

The results indicated a significant reduction in the number of features required for accurate classification, thereby enhancing model interpretability and computational efficiency. Our method's ability to maintain high accuracy with fewer features selection subset highlights its potential for practical applications in various domains. The analysis on the test-set performance demonstrated the generalization capability of the selected feature subsets. Our method consistently produced accurate and efficient models across different datasets, validating its robustness and effectiveness. Compared to traditional feature selection methods, our approach showed superior performance in terms of both feature reduction and accuracy. Scalability and versatility are key strengths of our proposed method. It can be applied to other machine learning problems beyond feature selection, offering a promising solution for data-driven evolutionary computation challenges. Future research could explore the scalability of our method on datasets with higher instances and its applicability to many multi-objective problems.

In conclusion, our study underscores the importance of diverse and effective feature selection techniques in optimizing machine learning models. The proposed method not only reduces the number of features but also ensures that the frequent features selected are robust in nature thereby contributing to more efficient and interpretable models.

REFERENCES

[1] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014, doi: https://doi.org/10.1016/j.compeleceng.2013.11.024.

[2] J. B. Jane and D. E. N. Ganesh, "A Review On Big Data With Machine Learning And Fuzzy Logic For Better Decision Making," vol. 8, no. 10, 2019.

[3] G. Goos *et al.*, "Lecture Notes in Computer Science".

[4] A.-D. Li, B. Xue, and M. Zhang, "Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies," *Applied Soft Computing*, vol. 106, p. 107302, 2021, doi: https://doi.org/10.1016/j.asoc.2021.107302.

[5] Bentley, P.J., Wakefield, J.P.: Generic evolutionary design. In: Chawdhry, P.K., Roy, R., Pant, R.K. (eds.) Soft Computing in Engineering Design and Manufacturing, pp. 289–298. Springer, London (1998). https://doi.org/10.1007/978-1-4471-0427-8_31

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Computat.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[7] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.

[8] J.-S. Pan, N. Liu, and S.-C. Chu, "A competitive mechanism based multi-objective differential evolution algorithm and its application in feature selection," *Knowledge-Based Systems*, vol. 245, p. 108582, Jun. 2022, doi: 10.1016/j.knosys.2022.108582.

[9] H. Ma, Y. Zhang, S. Sun, T. Liu, and Y. Shan, "A comprehensive survey on NSGA-II for multi-objective optimization and applications," *Artif Intell Rev*, vol. 56, no. 12, pp. 15217–15270, Dec. 2023, doi: 10.1007/s10462-023-10526-z.

[10] M. T. Jensen, "Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms," *IEEE Trans. Evol. Computat.*, vol. 7, no. 5, pp. 503–515, Oct. 2003, doi: 10.1109/TEVC.2003.817234.

[11] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "Performance comparison of NSGA-II and NSGA-III on various many-objective test problems," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada: IEEE, Jul. 2016, pp. 3045–3052. doi: 10.1109/CEC.2016.7744174.

[12] O. Agushaka and A. Ezugwu, "Initialisation Approaches for Population-Based Metaheuristic Algorithms: A Comprehensive Review," *Applied Sciences*, pp. 1–34, Jan. 2022, doi: 10.3390/app12020896.

[13] A. Deniz and H. E. Kiziloz, "On initial population generation in feature subset selection," *Expert Systems with Applications*, vol. 137, pp. 11–21, Dec. 2019, doi: 10.1016/j.eswa.2019.06.063.

[14] A. Asilian Bidgoli, H. Ebrahimpour-Komleh, and S. Rahnamayan, "Reference-point-based multi-objective optimization algorithm with opposition-based voting scheme for multi-label feature selection," *Information Sciences*, vol. 547, pp. 1–17, Feb. 2021, doi: 10.1016/j.ins.2020.08.004.