

## Enhancing software fault prediction using wrapper-based metaheuristic feature selection methods

Ha Thi Minh Phuong, Dang Thi Kim Ngan, Dao Khanh Duy, Nguyen Thanh Binh

The University of Danang, Vietnam - Korea University of Information and Communication Technology, Danang, Vietnam

### Article Info

#### Article history:

Received Jun 6, 2025

Revised Jul 18, 2025

Accepted Aug 1, 2025

#### Keywords:

Datasets

Feature selection methods

Machine learning

Software fault prediction

Wrapper-based feature

selection methods

### ABSTRACT

The application of software fault prediction (SFP) to predict faulty components at the early stage has been investigated in various studies. Reducing feature redundancy is key to enhancing the predictive accuracy of SFP models. Feature selection methods are utilized to select and retain the features that contribute the most information while eliminating irrelevant or redundant features from software fault datasets. However, feature selection (FS) in the field of SFP remains a broad and continuously evolving field, encompassing a diverse range of techniques and methodologies. In this work, we study and perform empirical evaluation of ten wrapper FS methods, namely artificial butterfly optimization (ABO), atom search optimization (ASO), equilibrium optimizer (EO), Henry gas solubility optimization (HGSO), poor and rich optimization (PRO), generalized normal distribution optimization (GNDO), slime mold algorithm, Harris hawk's optimization, pathfinder algorithm (PFA) and manta ray foraging optimization for resolving the data redundancy issue in SFP datasets. Experimental results on nine fault datasets from the PROMISE and AEEEM repositories show that the EO achieves the best performance, with PRO and HGSO ranking next. The comparative analysis revealed that ten wrapper-based FS methods demonstrated a substantial improvement in handling data redundancy issues for SFP.

This is an open access article under the [CC BY-SA](#) license.



### Corresponding Author:

Nguyen Thanh Binh

The University of Danang, Vietnam - Korea University of Information and Communication Technology

470 Tran Dai Nghia, Ngu Hanh Son District, Danang 55000, Vietnam

Email: ntbinh@vku.udn.vn

## 1. INTRODUCTION

The primary objective of feature selection is to extract a subset of relevant features from the original set, aiming to reduce dimensionality without compromising classification performance. Feature selection methods are typically categorized into three types: filter-based, wrapper-based, and hybrid approaches. Filter-based techniques rank features according to specific criteria and discard those that do not meet a predefined threshold [1]. Wrapper-based feature selection (FS) techniques utilize classification models to evaluate the effectiveness of feature subsets, often resulting in superior performance compared to filter-based methods [2]. Hybrid approaches combine the advantages of both filter and wrapper methods to achieve a balance between computational efficiency and predictive accuracy [3]. Prior research has shown that wrapper-based approaches generally outperform filter-based techniques [4]. Nevertheless, a large number of metaheuristic variants remain underexplored in the context of feature selection. Therefore, this study presents an empirical evaluation of metaheuristic algorithms within wrapper-based FS methods to reduce data redundancy in common software fault prediction (SFP) datasets, with the goal of improving model efficiency while preserving predictive performance. Specifically, we investigate a range of wrapper-based FS techniques

applied to software fault datasets, including artificial butterfly optimization (ABO), atom search optimization (ASO), equilibrium optimizer (EO), Henry gas solubility optimization (HGSO), poor and rich optimization (PRO), generalized normal distribution optimization (GNDO), slime mould algorithm, Harris hawk's optimization, pathfinder algorithm (PFA), and Manta Ray Foraging Optimization. Specifically, the proposed wrapper-based FS methods are evaluated against a baseline that applies learning algorithms directly to the original software fault datasets. Experiments were conducted on nine datasets derived from PROMISE and AEEEM repositories. To assess classification performance, we employed three learning models: random forest, extra trees, and AdaBoost. Evaluation metrics included precision, recall, F1-score, and area under the curve (AUC). To determine the statistical significance of performance differences between the ten wrapper-based FS techniques and the baseline, the Wilcoxon signed-rank test was performed at a 0.05 significance level. Each experiment was repeated ten times to ensure reliability, producing ten unique test sets. The results indicate that the wrapper-based methods consistently outperform the baseline. Among them, the EO achieved the best overall performance, followed by PRO and HGSO.

This study specifically addresses the following research questions:

- a. How effective are the applied FS techniques in enhancing SFP by filtering out irrelevant or redundant software metrics?
- b. Which wrapper-based FS method performs best for selecting the optimal features for SFP?

The structure of the paper is organized as. Section 2 reviews the related literature, while section 3 outlines the research methodology. Section 4 presents and discusses the experimental findings, and section 5 highlights the main conclusions and recommendations.

## 2. RELATED WORK

A hybrid FS method was introduced by Anju *et al.* [5]. This study proposed a method that combines quantum particle swarm optimization (QPSO) and principal component analysis (PCA). The results demonstrate that the proposed model, employing an artificial neural network (ANN) classifier, achieved higher accuracy and precision compared to existing approaches. The authors further suggest that these findings hold significant implications for both academia and the software industry. According to Ali *et al.* [6], metaheuristic approaches within wrapper-based FS methods outperformed traditional techniques such as Best First Search and Greedy Stepwise Search. Some examples of these algorithms include: whale optimization algorithm (WOA) [7], genetic algorithm (GA) [8] and particle swarm optimization (PSO) [8] are popular metaheuristic algorithms used for FS in SFP. Ali *et al.* [6] emphasized that removing irrelevant or redundant features may bring better predictive performance. However, incorrect FS or omitting important features can lead to a decline in model performance. Therefore, analyzing and evaluating different FS methods is essential to identify the most effective approach for software defect prediction. Balogun *et al.* [9] proposed a hybrid FS method that combines multiple filter techniques with a wrapper approach using rank aggregation. By combining filter and wrapper techniques, this approach aims to boost the efficacy of SFP models. It was shown through their experiments that the proposed methodology considerably enhanced the predictive performance of SFP models. Shah and Das [10] investigated the effectiveness of PSO for FS in conjunction with k-nearest neighbors (k-NN), naive Bayes and decision tree classifiers. Their experimental results demonstrate that integrating PSO with these classifiers enhances predictive performance across multiple datasets.

## 3. RESEARCH METHOD

### 3.1. Experimental design

Figure 1 indicates the main steps of our methodology. After splitting the data into training and testing parts, we filled in missing values and normalized the data. Subsequently, ten FS techniques based on wrapper strategies were employed to identify the most relevant set of features. Specifically, we utilized an open-source toolkit, wrapper-feature-selection-toolbox, which implements over 40 different wrapper methods. Finally, we utilized three ML classifiers, namely random forest, extra trees, and AdaBoost. precision, recall, F1-score, and AUC are the metrics used to determine the performance of the selected wrapper-driven approaches.

### 3.2. Wrapper-based feature selection techniques

There are various wrapper-based FS methods that have been applied in software fault prediction, including binary genetic algorithm (BGA), binary particle swarm optimization (BPSO), and binary ant colony optimization (BACO) [11]. We consider the following wrapper-based FS approaches in this study. ABO, inspired by butterfly behavior, was proposed by Qi *et al.* [12]. This algorithm is based on the preference of speckled woods, which seek out warm sunspots in woods and areas. ASO was introduced by Zhao *et al.* [13] for solving optimization challenges. Zhao *et al.* [13] demonstrated that ASO outperforms

many classic and emerging algorithms in benchmark tests and they have shown that ASO is a promising solution for real-world engineering problems. The EO, proposed by Faramarzi *et al.* [14], is a physics-inspired algorithm that regulates the volume mass balance model to determine dynamic and equilibrium states. This mechanism enables EO to effectively balance exploration and exploitation during the search process. Similarly, the HGSO algorithm [15], grounded in Henry's law [16], solves complex optimization tasks by simulating gas molecule clustering, which helps maintain exploration-exploitation balance and avoid premature convergence. The PRO algorithm, proposed by Moosavi and Bardsiri [17], is inspired by the financial behavior of individuals within society, modeled through the interaction between wealthier and poorer groups striving to improve their status. The generalized normal distribution optimization (GNDO) algorithm, introduced by Zhang *et al.* [18], operates in three stages: initial solution dispersion, convergence toward the optimal region, and refinement, using multiple normal distributions with gradually reduced variance. the slime mold algorithm (SMA), developed by Li *et al.* [19], is based on the foraging behavior of *Physarum polycephalum*, which identifies optimal paths to food sources without a central nervous system. Harris hawks optimization (HHO), introduced by Heidari *et al.* [20], simulates the cooperative hunting strategy of Harris hawks. The algorithm begins by randomly initializing hawk agents across the search space and guides them through exploration and exploitation phases. The PFA, proposed by Yapici and Cetinkaya [21], is a metaheuristic method that simulates the collective foraging behavior of animal groups, where individuals follow a leader while relying on their own perception to explore the search space. The manta ray foraging optimization (MRFO) algorithm, introduced by Zhao *et al.* [22], is inspired by the unique foraging strategies of manta rays, which search large areas and dynamically adjust their positions toward regions with greater resource availability.

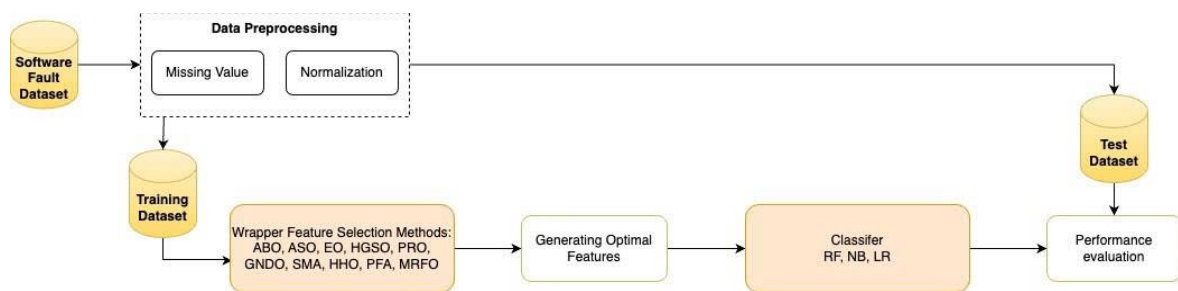


Figure 1. The main steps of the evaluation

### 3.3. Dataset

In this study, we obtained nine datasets from the PROMISE and AEEEM repositories, both of which are widely recognized and frequently utilized in SFP research. These datasets comprise independent variables, such as lines of code (LOCcode) and lines of comments (LOComment), along with a dependent variable indicating the status of a software component, classifying it as either faulty or non-faulty. A notable characteristic of the dataset is the predominance of non-faulty samples over faulty ones, which introduces an imbalanced data problem. The details are presented in Table 1.

Table 1. The datasets used in the study

| Dataset | Projects | Instances | Faulty instances | Non-faulty instances | Faulty ratio (%) |
|---------|----------|-----------|------------------|----------------------|------------------|
| CM1     | PROMISE  | 505       | 48               | 457                  | 9.50             |
| KC1     | PROMISE  | 2107      | 325              | 1782                 | 15.42            |
| KC2     | PROMISE  | 522       | 107              | 415                  | 20.50            |
| PC1     | PROMISE  | 1107      | 76               | 1031                 | 6.87             |
| EQ      | AEEEM    | 324       | 129              | 195                  | 39.81            |
| JDT     | AEEEM    | 997       | 105              | 892                  | 10.53            |
| Lucene  | AEEEM    | 691       | 64               | 627                  | 9.26             |
| Mylyn   | AEEEM    | 1862      | 245              | 1617                 | 13.16            |
| PDE     | AEEEM    | 1497      | 209              | 1288                 | 13.96            |

### 3.4. Evaluation measures

During the fault prediction model development process, performance evaluation metrics are applied to systematically assess the effectiveness of the models and identify the most appropriate one for a given dataset. In this study, we adopt precision (P), recall (R), F1-score (F1) and AUC as evaluation measures.

### 3.5. Machine learning techniques

Random forest [23] represents a supervised learning approach primarily used for regression and classification tasks. It operates by combining multiple decision trees, with the final prediction based on the aggregated results of these trees. Random forest is known for its high classification accuracy and its resistance to overfitting. AdaBoost [24] is a well-known Boosting algorithm designed to enhance weak classifiers by iteratively creating new models and adjusting their weights. It operates by sequentially training multiple weak classifiers, with each new model prioritizing the correction of errors made by its predecessors. ET [25] is a variant of random forest that selects attributes randomly for classification rather than choosing the optimal one, as random forest does. This approach enhances training speed and reduces overfitting; however, in some cases, it may result in lower accuracy.

## 4. RESULTS AND DISCUSSION

This section presents experimental findings to systematically address the two aforementioned research questions.

### 4.1. Research question 1: How effective are the presented feature selection methods for reducing the irrelevant/redundant metrics for SFP?

To answer research question 1, we perform various experiments to compare ten FS approaches using three ML classifiers described in section 3.5. In Tables 2 and 3, the best experimental results are highlighted in bold.

- a. Random forest results: Table 2 shows that, among all evaluated FS methods, the combination of EO and random forest achieved the highest average performance across all metrics, with PRO ranking second. Notably, EO achieved the highest average values across all evaluation metrics, with precision of 0.69, recall of 0.83, AUC of 0.76 and F1-score of 0.87. PRO achieved the second-highest performance metrics, with precision, recall, F1-score and AUC values of 0.68, 0.84, 0.76 and 0.84, respectively. The analysis results demonstrate that EO and PRO achieved comparatively superior performance when used as FS methods for the Random Forest classifier, outperforming the other FS evaluated in this study.

Table 2. The performance of the wrapper-based FS methods when applied with the RF model and the AdaBoost model

| Dataset | Metric         | RF       |      |      |      |      |      |      |      |      |      |      |          | AdaBoost |      |      |      |      |      |      |      |      |      |  |  |
|---------|----------------|----------|------|------|------|------|------|------|------|------|------|------|----------|----------|------|------|------|------|------|------|------|------|------|--|--|
|         |                | Baseline | ABO  | ASO  | EO   | HGSO | PRO  | GNDO | SMA  | HHO  | PFA  | MRFO | Baseline | ABO      | ASO  | EO   | HGSO | PRO  | GNDO | SMA  | HHO  | PFA  | MRFO |  |  |
| CM1     | P              | 0.5      | 0.55 | 0.49 | 0.5  | 0.62 | 0.48 | 0.54 | 0.55 | 0.49 | 0.61 | 0.48 | 0.5      | 0.55     | 0.49 | 0.5  | 0.62 | 0.48 | 0.49 | 0.5  | 0.49 | 0.49 | 0.49 |  |  |
|         | R              | 0.56     | 0.46 | 0.71 | 0.96 | 0.7  | 0.81 | 0.63 | 0.71 | 0.46 | 0.96 | 0.45 | 0.7      | 0.71     | 0.46 | 0.5  | 0.86 | 0.45 | 0.46 | 0.45 | 0.46 | 0.46 | 0.46 |  |  |
|         | F1             | 0.55     | 0.47 | 0.62 | 0.7  | 0.61 | 0.72 | 0.58 | 0.62 | 0.47 | 0.75 | 0.47 | 0.6      | 0.62     | 0.47 | 0.6  | 0.72 | 0.46 | 0.47 | 0.5  | 0.47 | 0.47 | 0.47 |  |  |
|         | AUC            | 0.61     | 0.83 | 0.82 | 0.87 | 0.78 | 0.79 | 0.77 | 0.54 | 0.51 | 0.72 | 0.69 | 0.74     | 0.76     | 0.71 | 0.83 | 0.72 | 0.59 | 0.67 | 0.63 | 0.61 | 0.65 | 0.68 |  |  |
|         | P              | 0.58     | 0.65 | 0.62 | 0.65 | 0.67 | 0.65 | 0.62 | 0.58 | 0.64 | 0.63 | 0.6  | 0.5      | 0.54     | 0.62 | 0.56 | 0.53 | 0.55 | 0.59 | 0.58 | 0.53 | 0.59 | 0.57 |  |  |
| KC1     | R              | 0.65     | 0.73 | 0.7  | 0.81 | 0.72 | 0.81 | 0.7  | 0.71 | 0.75 | 0.69 | 0.68 | 0.6      | 0.62     | 0.75 | 0.94 | 0.68 | 0.7  | 0.72 | 0.68 | 0.65 | 0.72 | 0.79 |  |  |
|         | F1             | 0.61     | 0.68 | 0.66 | 0.72 | 0.69 | 0.72 | 0.66 | 0.64 | 0.69 | 0.66 | 0.64 | 0.6      | 0.58     | 0.68 | 0.7  | 0.6  | 0.61 | 0.65 | 0.63 | 0.59 | 0.65 | 0.66 |  |  |
|         | AUC            | 0.74     | 0.81 | 0.8  | 0.82 | 0.8  | 0.79 | 0.81 | 0.8  | 0.81 | 0.82 | 0.81 | 0.76     | 0.77     | 0.79 | 0.66 | 0.8  | 0.77 | 0.8  | 0.8  | 0.79 | 0.79 | 0.82 |  |  |
|         | P              | 0.65     | 0.65 | 0.64 | 0.67 | 0.64 | 0.67 | 0.56 | 0.65 | 0.65 | 0.73 | 0.74 | 0.69     | 0.75     | 0.68 | 0.56 | 0.67 | 0.67 | 0.71 | 0.68 | 0.75 | 0.79 | 0.83 |  |  |
|         | R              | 0.76     | 0.69 | 0.72 | 0.71 | 0.74 | 0.81 | 0.59 | 0.67 | 0.65 | 0.77 | 0.76 | 0.71     | 0.78     | 0.73 | 0.94 | 0.71 | 0.79 | 0.68 | 0.75 | 0.7  | 0.79 | 0.79 |  |  |
| KC2     | F1             | 0.7      | 0.67 | 0.67 | 0.69 | 0.69 | 0.72 | 0.58 | 0.66 | 0.65 | 0.75 | 0.75 | 0.7      | 0.76     | 0.7  | 0.69 | 0.73 | 0.7  | 0.72 | 0.73 | 0.79 | 0.81 |      |  |  |
|         | AUC            | 0.81     | 0.86 | 0.78 | 0.83 | 0.85 | 0.79 | 0.56 | 0.62 | 0.81 | 0.81 | 0.89 | 0.79     | 0.88     | 0.81 | 0.66 | 0.82 | 0.81 | 0.75 | 0.71 | 0.78 | 0.8  | 0.9  |  |  |
|         | P              | 0.51     | 0.6  | 0.63 | 0.69 | 0.59 | 0.65 | 0.62 | 0.63 | 0.55 | 0.59 | 0.65 | 0.5      | 0.5      | 0.5  | 0.56 | 0.59 | 0.5  | 0.63 | 0.65 | 0.5  | 0.56 | 0.53 |  |  |
|         | R              | 0.52     | 0.85 | 0.81 | 0.78 | 0.72 | 0.81 | 0.72 | 0.81 | 0.57 | 0.72 | 0.73 | 0.5      | 0.47     | 0.47 | 0.84 | 0.72 | 0.6  | 0.76 | 0.73 | 0.65 | 0.64 | 0.59 |  |  |
|         | F1             | 0.52     | 0.7  | 0.71 | 0.73 | 0.65 | 0.72 | 0.67 | 0.71 | 0.56 | 0.65 | 0.69 | 0.5      | 0.48     | 0.48 | 0.7  | 0.65 | 0.6  | 0.69 | 0.69 | 0.6  | 0.59 | 0.56 |  |  |
| EQ      | AUC            | 0.6      | 0.77 | 0.83 | 0.94 | 0.89 | 0.79 | 0.84 | 0.74 | 0.65 | 0.83 | 0.85 | 0.74     | 0.82     | 0.81 | 0.66 | 0.89 | 0.85 | 0.82 | 0.66 | 0.7  | 0.84 | 0.82 |  |  |
|         | P              | 0.73     | 0.84 | 0.83 | 0.9  | 0.81 | 0.89 | 0.74 | 0.67 | 0.8  | 0.77 | 0.7  | 0.67     | 0.52     | 0.77 | 0.81 | 0.88 | 0.68 | 0.7  | 0.7  | 0.71 | 0.67 | 0.66 |  |  |
|         | R              | 0.72     | 0.83 | 0.84 | 0.9  | 0.8  | 0.91 | 0.74 | 0.69 | 0.8  | 0.76 | 0.7  | 0.7      | 0.88     | 0.77 | 0.83 | 0.87 | 0.69 | 0.7  | 0.7  | 0.72 | 0.67 | 0.67 |  |  |
|         | F1             | 0.73     | 0.84 | 0.83 | 0.9  | 0.8  | 0.9  | 0.74 | 0.68 | 0.8  | 0.77 | 0.7  | 0.68     | 0.65     | 0.77 | 0.82 | 0.88 | 0.69 | 0.7  | 0.7  | 0.72 | 0.67 | 0.67 |  |  |
|         | AUC            | 0.76     | 0.89 | 0.88 | 0.93 | 0.9  | 0.92 | 0.75 | 0.7  | 0.81 | 0.77 | 0.72 | 0.67     | 0.65     | 0.87 | 0.83 | 0.92 | 0.82 | 0.83 | 0.71 | 0.79 | 0.79 | 0.79 |  |  |
| JDT     | P              | 0.7      | 0.73 | 0.8  | 0.8  | 0.8  | 0.73 | 0.73 | 0.77 | 0.68 | 0.69 | 0.66 | 0.58     | 0.71     | 0.82 | 0.72 | 0.68 | 0.77 | 0.69 | 0.69 | 0.71 | 0.69 | 0.63 |  |  |
|         | R              | 0.75     | 0.86 | 0.87 | 0.89 | 0.86 | 0.86 | 0.78 | 0.82 | 0.79 | 0.79 | 0.75 | 0.79     | 0.77     | 0.89 | 0.81 | 0.71 | 0.87 | 0.71 | 0.8  | 0.74 | 0.74 | 0.7  |  |  |
|         | F1             | 0.72     | 0.79 | 0.84 | 0.84 | 0.83 | 0.79 | 0.75 | 0.79 | 0.73 | 0.73 | 0.7  | 0.67     | 0.74     | 0.85 | 0.76 | 0.7  | 0.82 | 0.7  | 0.74 | 0.73 | 0.72 | 0.66 |  |  |
|         | AUC            | 0.83     | 0.87 | 0.9  | 0.9  | 0.89 | 0.87 | 0.84 | 0.85 | 0.86 | 0.87 | 0.83 | 0.71     | 0.82     | 0.83 | 0.76 | 0.81 | 0.88 | 0.8  | 0.84 | 0.85 | 0.84 | 0.81 |  |  |
|         | P              | 0.53     | 0.7  | 0.63 | 0.77 | 0.73 | 0.73 | 0.62 | 0.57 | 0.68 | 0.58 | 0.73 | 0.57     | 0.57     | 0.7  | 0.75 | 0.62 | 0.54 | 0.55 | 0.61 | 0.66 | 0.66 | 0.58 |  |  |
| Lucene  | R              | 0.62     | 0.89 | 0.97 | 0.91 | 0.9  | 0.97 | 0.84 | 0.66 | 0.7  | 0.8  | 0.73 | 0.88     | 0.63     | 0.79 | 0.88 | 0.77 | 0.86 | 0.56 | 0.68 | 0.8  | 0.8  | 0.8  |  |  |
|         | F1             | 0.57     | 0.79 | 0.76 | 0.83 | 0.8  | 0.83 | 0.71 | 0.61 | 0.69 | 0.67 | 0.73 | 0.69     | 0.6      | 0.74 | 0.85 | 0.68 | 0.69 | 0.56 | 0.64 | 0.72 | 0.72 | 0.67 |  |  |
|         | AUC            | 0.79     | 0.91 | 0.85 | 0.85 | 0.77 | 0.89 | 0.81 | 0.73 | 0.64 | 0.83 | 0.83 | 0.71     | 0.75     | 0.81 | 0.84 | 0.88 | 0.64 | 0.74 | 0.72 | 0.78 | 0.8  | 0.78 |  |  |
|         | P              | 0.59     | 0.65 | 0.6  | 0.68 | 0.66 | 0.65 | 0.6  | 0.58 | 0.57 | 0.59 | 0.57 | 0.5      | 0.54     | 0.58 | 0.6  | 0.58 | 0.58 | 0.56 | 0.53 | 0.57 | 0.55 | 0.59 |  |  |
|         | R              | 0.74     | 0.8  | 0.74 | 0.82 | 0.83 | 0.77 | 0.75 | 0.81 | 0.71 | 0.72 | 0.73 | 0.7      | 0.72     | 0.67 | 0.8  | 0.71 | 0.71 | 0.64 | 0.74 | 0.68 | 0.59 | 0.68 |  |  |
| Mylyn   | F1             | 0.66     | 0.72 | 0.66 | 0.75 | 0.73 | 0.71 | 0.67 | 0.67 | 0.63 | 0.65 | 0.64 | 0.6      | 0.62     | 0.62 | 0.72 | 0.64 | 0.64 | 0.6  | 0.62 | 0.62 | 0.57 | 0.63 |  |  |
|         | AUC            | 0.88     | 0.89 | 0.88 | 0.9  | 0.9  | 0.89 | 0.85 | 0.68 | 0.84 | 0.84 | 0.84 | 0.63     | 0.8      | 0.81 | 0.75 | 0.71 | 0.75 | 0.82 | 0.69 | 0.81 | 0.83 | 0.8  |  |  |
|         | P              | 0.56     | 0.58 | 0.61 | 0.59 | 0.58 | 0.65 | 0.57 | 0.57 | 0.58 | 0.55 | 0.61 | 0.52     | 0.6      | 0.57 | 0.56 | 0.59 | 0.59 | 0.57 | 0.52 | 0.6  | 0.56 | 0.68 |  |  |
|         | R              | 0.61     | 0.83 | 0.76 | 0.69 | 0.76 | 0.81 | 0.74 | 0.69 | 0.76 | 0.72 | 0.76 | 0.88     | 0.74     | 0.69 | 0.84 | 0.67 | 0.69 | 0.77 | 0.93 | 0.79 | 0.75 | 0.76 |  |  |
|         | F1             | 0.59     | 0.68 | 0.67 | 0.64 | 0.66 | 0.72 | 0.64 | 0.63 | 0.66 | 0.62 | 0.68 | 0.65     | 0.66     | 0.62 | 0.7  | 0.63 | 0.64 | 0.65 | 0.67 | 0.68 | 0.64 | 0.72 |  |  |
| PDE     | AUC            | 0.58     | 0.75 | 0.79 | 0.76 | 0.79 | 0.79 | 0.8  | 0.68 | 0.78 | 0.75 | 0.79 | 0.65     | 0.68     | 0.74 | 0.66 | 0.75 | 0.74 | 0.77 | 0.65 | 0.72 | 0.76 | 0.72 |  |  |
|         | Average values |          |      |      |      |      |      |      |      |      |      |      |          |          |      |      |      |      |      |      |      |      |      |  |  |
|         | P              | 0.59     | 0.66 | 0.65 | 0.69 | 0.68 | 0.68 | 0.62 | 0.62 | 0.63 | 0.64 | 0.64 | 0.56     | 0.59     | 0.64 | 0.62 | 0.64 | 0.6  | 0.61 | 0.61 | 0.61 | 0.62 | 0.62 |  |  |
|         | R              | 0.66     | 0.77 | 0.79 | 0.83 | 0.78 | 0.84 | 0.72 | 0.73 | 0.69 | 0.77 | 0.7  | 0.72     | 0.7      | 0.69 | 0.82 | 0.74 | 0.71 | 0.67 | 0.72 | 0.69 | 0.68 | 0.69 |  |  |
|         | F1             | 0.63     | 0.7  | 0.71 | 0.76 | 0.72 | 0.76 | 0.67 | 0.67 | 0.65 | 0.69 | 0.67 | 0.63     | 0.63     | 0.66 | 0.73 | 0.69 | 0.65 | 0.64 | 0.66 | 0.65 | 0.65 | 0.65 |  |  |
| AUC     | 0.73           | 0.84     | 0.84 | 0.87 | 0.84 | 0.84 | 0.78 | 0.7  | 0.75 | 0.8  | 0.81 | 0.71 | 0.77     | 0.8      | 0.74 | 0.81 | 0.76 | 0.78 | 0.71 | 0.76 | 0.79 | 0.79 |      |  |  |

- b. AdaBoost results: The results summarized in Table 2 reveal that the HGSO consistently yielded the highest precision and AUC values in the majority of the experimental datasets. Specifically, on the KC1, KC2, PC1, EQ, and Lucene datasets, the highest AUC values were recorded in HGSO with 0.80, 0.82, 0.89, 0.92 and 0.88, respectively. In addition, EO obtained the greatest values on recall values on most of the datasets. Finally, EO achieved the greatest average recall and F1-score values of 0.82 and 0.73, whereas HGSO demonstrated superior average precision and AUC with values of 0.64 and 0.81, respectively.
- c. ET results: It can be observed in Table 3, EO achieved superior recall, F1-score, and AUC compared to the other wrapper-based FS methods for the CM1, KC1, KC2 and PC1 datasets. ASO obtained the highest precision on the EQ, JDT and Lucene datasets, with values of 0.82, 0.73 and 0.74, respectively. For the EQ dataset, EO highlighted better prediction performance than all other FS models with the greatest recall and AUC values of 0.96 and 0.86. For the EQ dataset, EO demonstrated better predictive performance compared to all other FS models, achieving the highest recall and AUC values of 0.96 and 0.86, respectively. Again, for the average values of recall, F1-score and AUC, EO outperformed all other FS models in terms of recall, F1-score and AUC values, with 0.77, 0.70 and 0.84, while the highest average precision value was recorded in ASO method, with a value of 0.67.

Table 3. The performance of the wrapper-based FS methods when applied with the ET model

| Dataset        | Metrics | Baseline    | ABO         | ASO         | EO          | HGSO        | PRO         | GNDO        | SMA  | HHO  | PFA  | MRFO        |
|----------------|---------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------|------|------|-------------|
| CM1            | P       | 0.55        | 0.55        | 0.59        | <b>0.61</b> | 0.54        | 0.47        | 0.6         | 0.55 | 0.49 | 0.61 | 0.48        |
|                | R       | 0.71        | 0.71        | 0.66        | <b>0.96</b> | 0.70        | 0.45        | 0.71        | 0.71 | 0.46 | 0.96 | 0.45        |
|                | F1      | 0.62        | 0.62        | 0.63        | <b>0.75</b> | 0.61        | 0.46        | 0.65        | 0.62 | 0.47 | 0.75 | 0.47        |
|                | AUC     | 0.85        | 0.85        | 0.80        | <b>0.86</b> | 0.81        | 0.70        | 0.75        | 0.47 | 0.54 | 0.77 | 0.66        |
| KC1            | P       | 0.65        | 0.65        | 0.62        | <b>0.66</b> | 0.64        | 0.62        | 0.64        | 0.56 | 0.64 | 0.63 | 0.62        |
|                | R       | 0.70        | 0.70        | 0.68        | <b>0.75</b> | 0.69        | 0.70        | 0.69        | 0.69 | 0.70 | 0.67 | 0.70        |
|                | F1      | 0.67        | 0.67        | 0.65        | <b>0.70</b> | 0.66        | 0.66        | 0.67        | 0.62 | 0.67 | 0.65 | 0.66        |
|                | AUC     | 0.81        | 0.81        | 0.79        | <b>0.81</b> | 0.78        | 0.80        | 0.82        | 0.81 | 0.81 | 0.81 | 0.79        |
| KC2            | P       | 0.70        | 0.70        | 0.70        | 0.72        | 0.64        | 0.71        | 0.56        | 0.65 | 0.65 | 0.72 | <b>0.78</b> |
|                | R       | 0.75        | 0.75        | 0.73        | 0.75        | 0.69        | 0.80        | 0.59        | 0.67 | 0.65 | 0.75 | <b>0.81</b> |
|                | F1      | 0.73        | 0.73        | 0.71        | 0.73        | 0.66        | 0.75        | 0.58        | 0.66 | 0.65 | 0.73 | <b>0.80</b> |
|                | AUC     | 0.86        | 0.86        | 0.81        | 0.83        | 0.78        | 0.83        | 0.58        | 0.62 | 0.78 | 0.82 | <b>0.89</b> |
| PC1            | P       | 0.52        | 0.56        | 0.63        | <b>0.69</b> | 0.66        | 0.62        | 0.59        | 0.63 | 0.51 | 0.62 | 0.66        |
|                | R       | 0.55        | <b>0.80</b> | 0.76        | 0.78        | 0.75        | 0.72        | 0.66        | 0.76 | 0.52 | 0.72 | 0.79        |
|                | F1      | 0.53        | 0.66        | 0.69        | <b>0.73</b> | 0.70        | 0.67        | 0.62        | 0.69 | 0.52 | 0.67 | 0.72        |
|                | AUC     | 0.60        | 0.78        | 0.82        | <b>0.94</b> | 0.87        | 0.85        | 0.83        | 0.74 | 0.64 | 0.82 | 0.88        |
| EQ             | P       | 0.77        | 0.77        | <b>0.82</b> | 0.61        | 0.68        | 0.72        | 0.77        | 0.67 | 0.81 | 0.81 | 0.68        |
|                | R       | 0.76        | 0.76        | 0.81        | <b>0.96</b> | 0.67        | 0.72        | 0.77        | 0.69 | 0.8  | 0.8  | 0.69        |
|                | F1      | 0.77        | 0.77        | <b>0.82</b> | 0.75        | 0.68        | 0.72        | 0.77        | 0.68 | 0.81 | 0.8  | 0.68        |
|                | AUC     | 0.83        | 0.83        | 0.84        | <b>0.86</b> | 0.83        | 0.80        | 0.84        | 0.73 | 0.89 | 0.84 | 0.83        |
| JDT            | P       | <b>0.78</b> | 0.73        | 0.73        | 0.72        | 0.73        | 0.75        | 0.72        | 0.75 | 0.69 | 0.73 | 0.69        |
|                | R       | 0.82        | <b>0.86</b> | 0.80        | 0.81        | 0.82        | 0.78        | 0.77        | 0.78 | 0.77 | 0.84 | 0.77        |
|                | F1      | <b>0.80</b> | 0.79        | 0.77        | 0.76        | 0.77        | 0.76        | 0.74        | 0.76 | 0.73 | 0.78 | 0.73        |
|                | AUC     | 0.87        | 0.86        | 0.88        | 0.87        | 0.89        | <b>0.90</b> | 0.83        | 0.83 | 0.87 | 0.88 | 0.79        |
| Lucene         | P       | 0.56        | 0.58        | <b>0.74</b> | 0.65        | 0.57        | 0.61        | 0.65        | 0.61 | 0.68 | 0.64 | 0.72        |
|                | R       | 0.63        | <b>0.96</b> | 0.81        | 0.69        | 0.71        | 0.76        | 0.72        | 0.68 | 0.7  | 0.67 | 0.69        |
|                | F1      | 0.59        | 0.73        | <b>0.77</b> | 0.67        | 0.63        | 0.68        | 0.68        | 0.64 | 0.69 | 0.66 | 0.70        |
|                | AUC     | 0.55        | <b>0.84</b> | 0.79        | <b>0.84</b> | 0.71        | 0.63        | 0.83        | 0.71 | 0.64 | 0.81 | 0.83        |
| Myllyn         | P       | 0.55        | 0.60        | <b>0.61</b> | 0.53        | 0.52        | 0.56        | 0.61        | 0.56 | 0.61 | 0.61 | 0.60        |
|                | R       | 0.63        | <b>0.71</b> | <b>0.71</b> | 0.59        | 0.55        | 0.63        | 0.7         | 0.77 | 0.72 | 0.71 | 0.71        |
|                | F1      | 0.59        | 0.65        | <b>0.66</b> | 0.56        | 0.53        | 0.59        | 0.65        | 0.65 | 0.66 | 0.66 | 0.65        |
|                | AUC     | 0.54        | 0.79        | <b>0.85</b> | 0.80        | 0.76        | 0.74        | 0.82        | 0.68 | 0.83 | 0.85 | 0.81        |
| PDE            | P       | 0.57        | 0.60        | 0.58        | 0.57        | 0.56        | 0.60        | <b>0.61</b> | 0.58 | 0.59 | 0.59 | 0.62        |
|                | R       | 0.61        | 0.77        | 0.71        | 0.62        | <b>0.80</b> | 0.68        | 0.78        | 0.69 | 0.78 | 0.75 | 0.74        |
|                | F1      | 0.59        | 0.67        | 0.64        | 0.60        | 0.66        | 0.64        | <b>0.69</b> | 0.63 | 0.67 | 0.66 | 0.67        |
|                | AUC     | 0.60        | 0.76        | 0.77        | 0.73        | <b>0.79</b> | 0.75        | 0.77        | 0.67 | 0.76 | 0.74 | 0.76        |
| Average values | P       | 0.63        | 0.64        | <b>0.67</b> | 0.64        | 0.62        | 0.63        | 0.64        | 0.62 | 0.63 | 0.66 | 0.65        |
|                | R       | 0.68        | 0.78        | 0.74        | <b>0.77</b> | 0.71        | 0.69        | 0.71        | 0.72 | 0.68 | 0.76 | 0.71        |
|                | F1      | 0.65        | 0.70        | 0.70        | <b>0.70</b> | 0.66        | 0.66        | 0.67        | 0.66 | 0.65 | 0.71 | 0.68        |
|                | AUC     | 0.72        | 0.82        | 0.82        | <b>0.84</b> | 0.80        | 0.78        | 0.79        | 0.70 | 0.75 | 0.82 | 0.80        |

#### 4.2. Research question 2: which wrapper-based feature selection method performs best for selecting the optimal features for SFP?

Figure 2(a)-(i) in Appendix, presents the AUC performance of the proposed wrapper-based FS methods. The x-axis indicates the names of the FS algorithms, while the y-axis shows their corresponding AUC scores. According to the chart, EO and PRO consistently delivered superior results across several datasets. Specifically, EO attained top AUC scores of 0.78, 0.90, 0.84, and 0.75 on KC1, EQ, Lucene, and

PDE, respectively. Similarly, PRO achieved notable AUC values of 0.88, 0.93, and 0.86 on PC1, EQ, and JDT. To further analyze overall performance, the average AUC across different machine learning classifiers was computed. All evaluated FS methods achieved mean AUC values above 0.65. Among them, EO in combination with random forest yielded the highest average score of 0.87, followed closely by PRO with random forest at 0.84. On the other hand, SMA coupled with ET obtained the lowest average AUC, with a score of 0.65. These outcomes suggest that EO offers the most reliable FS strategy for SFP, effectively enhancing classifier performance. PRO and HGSO also demonstrated strong and consistent performance across multiple scenarios.

### 4.3. Statistical analysis

To compare the performance of ten wrapper-based FS methods (ABO, ASO, EO, HGSO, PRO, GNDO, SMA, HHO, PFA, and MRFO) with that of the baseline method, the Wilcoxon signed-rank test was applied with  $\alpha = 0.05$ . Table 4 presents the detailed statistical test results for all evaluated ML algorithms and wrapper-based FS methods with respect to precision, recall, F1-score, and AUC. Table 4 presents the statistical analysis results, including  $P$ -values, effect size ( $r$ ), and mean differences. An asterisk (\*) denotes a statistically significant difference, corresponding to a  $P$ -value less than 0.05. The effect size  $r$  quantifies the magnitude of the performance difference between the compared methods, while the mean difference indicates which method achieves a higher average score. According to Table 4, several FS methods demonstrated statistically significant improvements in F1-score compared to the baseline. For example, ASO achieved a statistically significant improvement with a  $P$ -value of 0.026 and a large effect size ( $r = 0.913$ ), along with a mean difference of 0.071. Similarly, HGSO and PRO demonstrated meaningful gains, with  $P$ -values of 0.022 and 0.033, and moderate effect sizes of 0.520 and 0.561, respectively. In contrast, EO showed a moderate effect size ( $r = 0.489$ ) but a non-significant  $P$ -value of 0.293, indicating that its performance difference from the baseline may not be statistically reliable. In contrast, SMA and HHO showed  $P$ -values of 1.000, indicating no statistical difference from the baseline, even though their effect sizes were moderate (0.534 and 0.633). These findings suggest that not all numerical differences translate into meaningful statistical significance. Overall, ASO, HGSO, PRO, PFA, and MRFO stood out as statistically and practically effective FS methods for random forest and should be prioritized for real-world deployment in SFP models.

Table 4. Statistical test results for the comparison between ten wrapper-based FS methods and the baseline method for random forest, ET and AdaBoost

| Comparison group | Measure         | Random forest |        |        |        | ET     |        |        |        | AdaBoost |        |        |        |
|------------------|-----------------|---------------|--------|--------|--------|--------|--------|--------|--------|----------|--------|--------|--------|
|                  |                 | AUC           | F1     | P      | R      | AUC    | F1     | P      | R      | AUC      | F1     | P      | R      |
| ABO & Baseline   | P-value         | 0.08          | 0.08   | 0.416  | 0.043* | 0.08   | 0.08   | 0.416  | 0.043* | 0.018*   | 0.052* | 0.236  | 0.858  |
|                  | Effect r        | 0.695         | 0.782  | 0.974  | -0.062 | 0.695  | 0.782  | 0.974  | -0.062 | 0.589    | 0.814  | 0.549  | 0.481  |
|                  | Mean-difference | 0.097         | 0.044  | 0.01   | 0.096  | 0.097  | 0.044  | 0.01   | 0.096  | 0.059    | 0.002  | 0.028  | 0.016  |
| ASO & Baseline   | P-value         | 0.013*        | 0.109  | 0.107  | 0.053* | 0.013* | 0.109  | 0.107  | 0.053* | 0.018*   | 0.362  | 0.035* | 0.59   |
|                  | Effect r        | 0.256         | 0.585  | 0.714  | 0.184  | 0.256  | 0.585  | 0.714  | 0.184  | -0.104   | 0.768  | 0.703  | 0.53   |
|                  | Mean-difference | 0.093         | 0.05   | 0.041  | 0.057  | 0.093  | 0.05   | 0.041  | 0.057  | 0.087    | 0.027  | 0.078  | 0.027  |
| EO & Baseline    | P-value         | 0.051*        | 0.293  | 0.574  | 0.050* | 0.051* | 0.293  | 0.574  | 0.050* | 0.407    | 0.018* | 0.068  | 0.123  |
|                  | Effect r        | 0.205         | 0.54   | 0.435  | 0.525  | 0.205  | 0.54   | 0.435  | 0.525  | -0.284   | 0.509  | 0.532  | -0.01  |
|                  | Mean-difference | 0.114         | 0.04   | 0.012  | 0.083  | 0.114  | 0.04   | 0.012  | 0.083  | 0.028    | 0.096  | 0.066  | 0.102  |
| HGSO & Baseline  | P-value         | 0.208         | 0.022* | 0.189  | 0.033* | 0.208  | 0.022* | 0.189  | 0.033* | 0.011*   | 0.092  | 0.011* | 0.528  |
|                  | Effect r        | 0.446         | 0.52   | 0.765  | 0.123  | 0.446  | 0.52   | 0.765  | 0.123  | 0.226    | 0.401  | 0.776  | -0.013 |
|                  | Mean-difference | 0.079         | 0.001  | -0.012 | 0.024  | 0.079  | 0.001  | -0.012 | 0.024  | 0.1      | 0.056  | 0.081  | 0.027  |
| PRO & Baseline   | P-value         | 0.284         | 0.033* | 1      | 0.498  | 0.284  | 0.033* | 1      | 0.498  | 0.155    | 0.205  | 0.207  | 0.905  |
|                  | Effect r        | 0.487         | 0.561  | 0.837  | 0.165  | 0.487  | 0.561  | 0.837  | 0.165  | 0.042    | 0.597  | 0.677  | 0.488  |
|                  | Mean-difference | 0.054         | 0.004  | 0.001  | 0.009  | 0.054  | 0.004  | 0.001  | 0.009  | 0.05     | 0.021  | 0.037  | 0.011  |
| GNDO & Baseline  | P-value         | 0.372         | 0.045* | 0.028* | 0.062  | 0.372  | 0.045* | 0.028* | 0.062  | 0.050*   | 0.752  | 0.024* | 0.401  |
|                  | Effect r        | -0.361        | 0.491  | 0.685  | 0.144  | -0.361 | 0.491  | 0.685  | 0.144  | -0.372   | 0.113  | 0.769  | -0.21  |
|                  | Mean-difference | 0.062         | 0.018  | 0.011  | 0.026  | 0.062  | 0.018  | 0.011  | 0.026  | 0.067    | 0.003  | 0.051  | 0.051  |
| SMA & Baseline   | P-value         | 0.089         | 1      | 0.026* | 0.041* | 0.089  | 1      | 0.026* | 0.041* | 1        | 0.31   | 0.028* | 0.575  |
|                  | Effect r        | -0.061        | 0.534  | 0.757  | -0.063 | -0.061 | 0.534  | 0.757  | -0.063 | 0.199    | 0.314  | 0.745  | 0.326  |
|                  | Mean-difference | -0.028        | 0.007  | -0.01  | 0.031  | -0.028 | 0.007  | -0.01  | 0.031  | 0.001    | 0.024  | 0.048  | 0      |
| HHO & Baseline   | P-value         | 0.499         | 1      | 1      | 1      | 0.499  | 1      | 1      | 1      | 0.173    | 0.037* | 0.017* | 0.373  |
|                  | Effect r        | 0.231         | 0.633  | 0.78   | 0.309  | 0.231  | 0.633  | 0.78   | 0.309  | -0.198   | 0.7    | 0.9    | 0.552  |
|                  | Mean-difference | 0.028         | -0.002 | 0.002  | -0.007 | 0.028  | -0.002 | 0.002  | -0.007 | 0.048    | 0.019  | 0.054  | 0.03   |
| PFA & Baseline   | P-value         | 0.023*        | 0.035* | 0.065  | 0.025* | 0.023* | 0.035* | 0.065  | 0.025* | 0.024*   | 0.405  | 0.017* | 0.553  |
|                  | Effect r        | 0.203         | 0.786  | 0.894  | 0.483  | 0.203  | 0.786  | 0.894  | 0.483  | -0.133   | 0.686  | 0.886  | 0.413  |
|                  | Mean-difference | 0.092         | 0.052  | 0.034  | 0.079  | 0.092  | 0.052  | 0.034  | 0.079  | 0.078    | 0.014  | 0.059  | 0.033  |
| MRFO & Baseline  | P-value         | 0.207         | 0.035* | 0.593  | 0.483  | 0.207  | 0.035* | 0.593  | 0.483  | 0.013*   | 0.373  | 0.028* | 0.514  |
|                  | Effect r        | -0.207        | 0.293  | 0.471  | -0.076 | -0.207 | 0.293  | 0.471  | -0.076 | 0.426    | 0.671  | 0.784  | 0.398  |
|                  | Mean-difference | 0.081         | 0.021  | 0.022  | 0.021  | 0.081  | 0.021  | 0.022  | 0.021  | 0.08     | 0.018  | 0.059  | 0.024  |

## 5. CONCLUSION

The objective of SFP models is to build high-quality software with minimal testing efforts by identifying faulty modules early in the software development process. The performance of SFP models, however, is influenced by the quality of the software defect datasets, which commonly suffer from high dimensionality. Therefore, the elimination of irrelevant and redundant features from software fault datasets is an important step in building a robust SFP model. In this study, we present an evaluation of these wrapper strategies to select optimal features. Subsequently, we applied various ML techniques to the datasets with the previously extracted optimal features. Moreover, this study conducted a comparative performance analysis between the proposed wrapper-based FS methods and the baseline approach. The experimental results showed that ten presented wrapper-based FS methods achieved better performance compared to the baseline method which applied ML techniques to the original dataset. Moreover, the results indicated that EO was the most effective, with PRO and HGSO following. However, this study still has certain limitations. First, the datasets used are static and relatively well-balanced, which may not fully capture the challenges posed by real-world scenarios involving imbalanced distributions or data that change over time. Second, evaluating the methods using only three classifiers may limit the generalizability of the findings across other machine learning models. In the future, we aim to explore the integration of the proposed wrapper-based FS methods with sampling techniques such as SMOTE or ADASYN to address the challenges of high-dimensional and imbalanced datasets, thereby enhancing the performance of SFP models.

## FUNDING INFORMATION

Funding for this research was provided by the Science and Technology Development Fund of the University of Danang, project number B2022-DN07-02.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author     | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|--------------------|---|---|----|----|----|---|---|---|---|---|----|----|---|----|
| Ha Thi Minh Phuong | ✓ | ✓ | ✓  | ✓  | ✓  | ✓ |   | ✓ | ✓ | ✓ |    |    | ✓ |    |
| Dang Thi Kim Ngan  | ✓ | ✓ |    |    |    | ✓ |   | ✓ | ✓ | ✓ | ✓  | ✓  |   |    |
| Dao Khanh Duy      | ✓ |   | ✓  | ✓  |    | ✓ | ✓ |   |   | ✓ | ✓  |    | ✓ | ✓  |
| Nguyen Thanh Binh  | ✓ | ✓ | ✓  | ✓  | ✓  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓  | ✓  | ✓ | ✓  |

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

## CONFLICT OF INTEREST STATEMENT

There are no conflicts of interest regarding the publication of this paper.

## DATA AVAILABILITY

The datasets and source code used can be accessed at <https://github.com/Duycan17/Wrapper-Based>.

## REFERENCES

- [1] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.
- [2] E. Llobet *et al.*, "Efficient feature selection for mass spectrometry based electronic nose applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 85, no. 2, pp. 253–261, Feb. 2007, doi: 10.1016/j.chemolab.2006.07.002.
- [3] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8144–8150, Jul. 2011, doi: 10.1016/j.eswa.2010.12.156.
- [4] M. Mabayoje, A. Balogun, S. Bello, and J. Atoyebi, "Wrapper feature selection based Heterogeneous classifiers for software defect prediction," *Adeleke University Journal of Engineering and Technology (AUJET)*, vol. 2, no. 1, pp. 1–11, 2019.
- [5] A. J. Anju and J. E. Judith, "Hybrid feature selection method for predicting software defect," *Journal of Engineering and Applied Science*, vol. 71, no. 1, p. 124, 2024.
- [6] M. Ali *et al.*, "Analysis of feature selection methods in software defect prediction models," *IEEE Access*, vol. 11, pp. 145954–145974, 2023, doi: 10.1109/ACCESS.2023.3343249.



- [7] M. Mafarja, I. Jaber, S. Ahmed, and T. Thaher, "Whale optimisation algorithm for high-dimensional small-instance feature selection," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 36, no. 2, pp. 80–96, Mar. 2021, doi: 10.1080/17445760.2019.1617866.
- [8] A. Amirullah, U. L. Yuhana, and M. Alfian, "Improve software defect prediction using particle swarm optimization and synthetic minority over-sampling technique," *Scientific Journal of Informatics*, vol. 11, no. 4, pp. 1127–1136, 2024.
- [9] A. O. Balogun *et al.*, "A novel rank aggregation-based hybrid multifilter wrapper feature selection method in software defect prediction," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/5069016.
- [10] H. Shah and H. Das, "A wrapper-based feature selection approach using particle swarm optimization for software fault prediction," in *International conference on computational intelligence in pattern recognition*, 2022, pp. 375–386.
- [11] H. Turabieh, M. Mafarja, and X. Li, "Iterated feature selection algorithms with layered recurrent neural network for software fault prediction," *Expert Systems with Applications*, vol. 122, pp. 27–42, May 2019, doi: 10.1016/j.eswa.2018.12.033.
- [12] X. Qi, Y. Zhu, and H. Zhang, "A new meta-heuristic butterfly-inspired algorithm," *Journal of Computational Science*, vol. 23, pp. 226–239, Nov. 2017, doi: 10.1016/j.jocs.2017.06.003.
- [13] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowledge-Based Systems*, vol. 163, pp. 283–304, Jan. 2019, doi: 10.1016/j.knsys.2018.08.030.
- [14] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: a novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, p. 105190, Mar. 2020, doi: 10.1016/j.knsys.2019.105190.
- [15] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 646–667, Dec. 2019, doi: 10.1016/j.future.2019.07.015.
- [16] Y. Li, W. Zhang, L. Wang, F. Zhao, W. Han, and G. Chen, "Henry's law and accumulation of weak source for crust-derived helium: A case study of Weihe Basin, China," *Journal of Natural Gas Geoscience*, vol. 2, no. 5–6, pp. 333–339, Dec. 2017, doi: 10.1016/j.jnggs.2018.02.001.
- [17] S. H. Samareh Moosavi and V. K. Bardsiri, "Poor and rich optimization algorithm: A new human-based and multi populations algorithm," *Engineering Applications of Artificial Intelligence*, vol. 86, pp. 165–181, Nov. 2019, doi: 10.1016/j.engappai.2019.08.025.
- [18] Y. Zhang, Z. Jin, and S. Mirjalili, "Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models," *Energy Conversion and Management*, vol. 224, p. 113301, Nov. 2020, doi: 10.1016/j.enconman.2020.113301.
- [19] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, Oct. 2020, doi: 10.1016/j.future.2020.03.055.
- [20] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.
- [21] H. Yapici and N. Cetinkaya, "A new meta-heuristic optimizer: pathfinder algorithm," *Applied Soft Computing*, vol. 78, pp. 545–568, May 2019, doi: 10.1016/j.asoc.2019.03.012.
- [22] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103300, Jan. 2020, doi: 10.1016/j.engappai.2019.103300.
- [23] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [24] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/jcss.1997.1504.

## APPENDIX

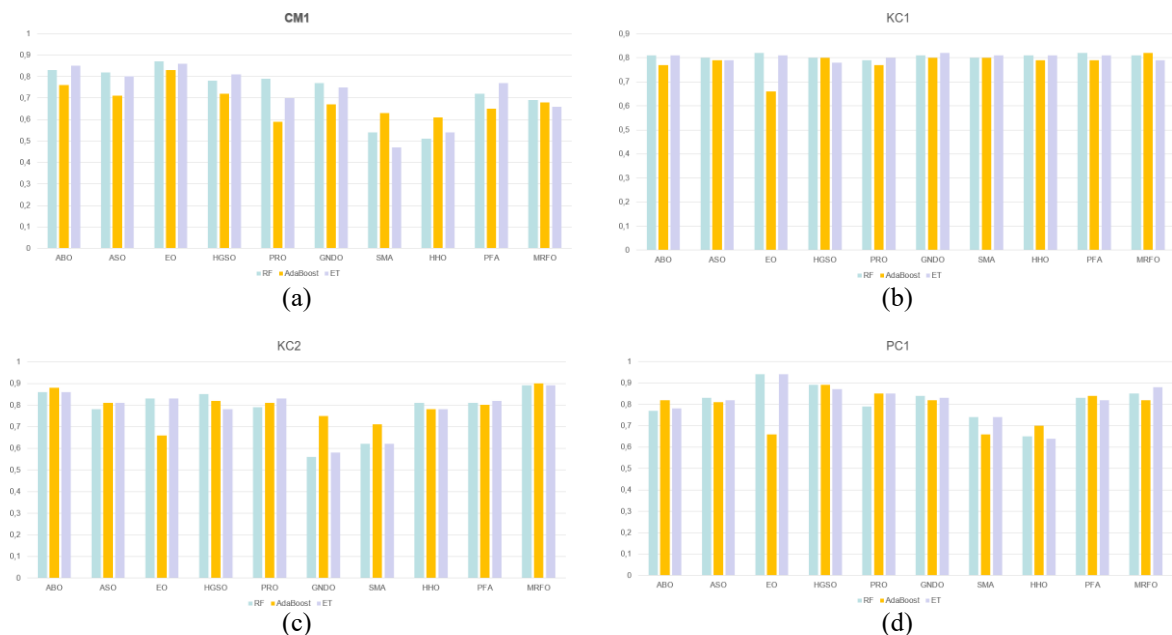


Figure 2. Comparison of AUC values for the wrapper-based FS methods across the different datasets, (a) the comparative AUC values on CM1, (b) the comparative AUC values on KC1, (c) the comparative AUC values on KC2, (d) the comparative AUC values on PC1 (*Continue*)



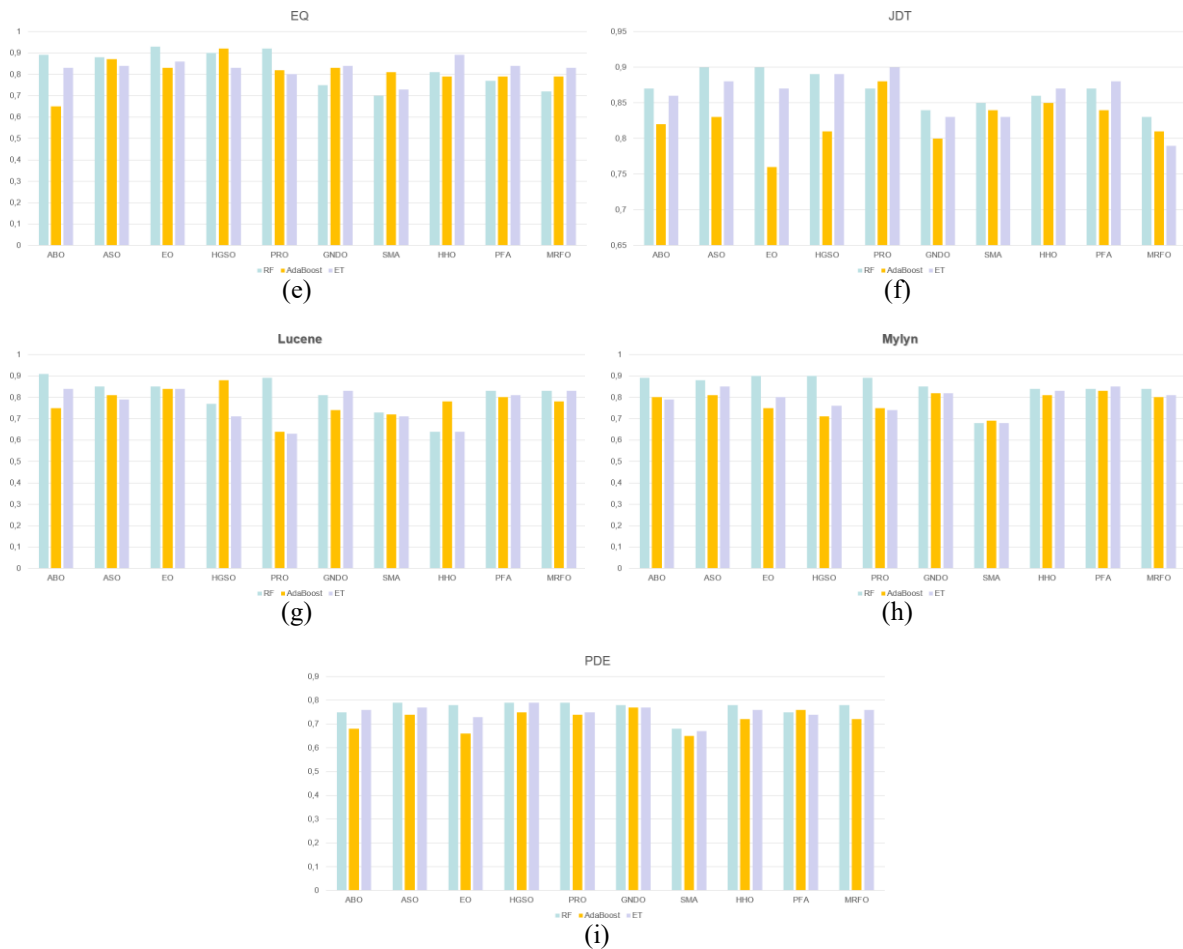








Figure 2. Comparison of AUC values for the wrapper-based FS methods across the different datasets, (e) the comparative AUC values on EQ, (f) the comparative AUC values on JDT, (g) the comparative AUC values on Lucene, (h) the comparative AUC values on Mylyn, (i) the comparative AUC values on PDE

## BIOGRAPHIES OF AUTHORS






**Ha Thi Minh Phuong**    received bachelor's degree in information technology from The University of Danang-University of Science and Technology in 2010. She earned M.Sc. degree in computer science at Yuan Ze University, Taiwan in 2013. She is a lecturer at The University of Danang-Vietnam-Korea University of Information and Communication Technology. She is currently pursuing a Ph.D. degree in computer science at the University of Danang. Her current research focuses on software testing, deep learning. He can be contacted at email: htmphuong@vku.udn.vn.






**Dang Thi Kim Ngan**    graduated bachelor's degree in information management at the University of Danang-University of Economics in 2011. She got MSc degree of at Chinese Culture University, Taiwan in 2014. She is currently serving as a lecturer at the Department of Computer Science, The University of Danang-Vietnam-Korea University of Information and Communication Technology. Her research focuses on software engineering. He can be contacted at email: dtkngan@vku.udn.vn.



**Dao Khanh Duy**    is currently pursuing a Bachelor of Engineering at The University of Danang Vietnam-Korea University of Information and Communication Technology. He has been gaining valuable hands-on experience in research and development through internships at King Mongkut's University of Technology North Bangkok and MGM technology partners. His current research focuses on software testing and software effort estimation. He can be contacted at email: [duydk.22git@vku.udn.vn](mailto:duydk.22git@vku.udn.vn).



**Nguyen Thanh Binh**    graduated in information technology from the University of Danang-University of Science and Technology in 1997. He received PhD. degree in Information Technology at Grenoble Institute of Technology, France in 2004. He has been qualified as Associate Professor since 2013. He is currently working at the University of Danang-Vietnam-Korea University of Information and Communication Technology. His research interests include software engineering and software quality. He can be contacted at email: [ntbinh@vku.udn.vn](mailto:ntbinh@vku.udn.vn).