Science of Computer Programming 205 (2021) 102621

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Improving software effort estimation using bio-inspired algorithms to select relevant features: An empirical study

Asad Ali *, Carmine Gravino

University of Salerno, Department of Computer Science, Via Giovanni Paolo II, 132, 84084 Fisciano (SA), Italy

ARTICLE INFO

Article history:
Received 27 May 2020
Received in revised form 21 October 2020
Accepted 25 January 2021
Available online 28 January 2021

Keywords:
Bio-inspired algorithms
Feature selection
Effort estimation
Estimation techniques

ABSTRACT

Context: Bio-inspired feature selection algorithms got the attention of the researchers in the domain of Software Development Effort Estimations (SDEE) because they can improve the prediction accuracy of existing estimation techniques, such as machine learning methods.
Objective: This paper aims to analyze different feature selection algorithms and assess the role they can play to increase the accuracy of software development effort predictions.
Method: We have performed an empirical study considering commonly used bio-inspired feature selection algorithms in the domain of SDEE, i.e., Genetic Algorithm (GA), Particle Swarm Optimization, Ant Colony Optimization, Tabu Search, Harmony Search (HS), and Firefly algorithm, and four traditional non-bio-inspired algorithms, i.e., Best-First Search (BFS), Greedy Stepwise, Subset Forward Selection, and Random Search, used in combination with five widely used estimation techniques and applied to eight widely used SDEE datasets.
Results: The performed analysis suggests that almost all (bio-inspired) feature selection algorithms have outperformed the baseline estimation techniques (i.e., techniques employed without any feature selection algorithms) in the majority of the experiments and hence we can conclude that feature selection algorithms can help in the domain of SDEE to increase the prediction accuracy. Similarly, HS and GA are considered as best performed bio-inspired algorithms because they provided significantly better results than the non-bio-inspired algorithms in a greater number of experiments. Moreover, we also compared the results of various employed bio-inspired algorithms, and, again, GA and HS came out as the best performed bio-inspired feature selection algorithms.
Conclusion: From our results, if we have to pick feature selection algorithms (from both bio- and non-bio-inspired) and recommend them for future investigations, we would suggest HS because it provided better effort predictions in more combinations of datasets and estimation techniques than the other considered bio- and non-bio-inspired algorithms. Among the non-bio-inspired algorithms, BFS is the one that provided better predictions.

* Corresponding author.
E-mail addresses: aali@unisa.it (A. Ali), gravino@unisa.it (C. Gravino).

https://doi.org/10.1016/j.scico.2021.102621
0167-6423/

1. Introduction

According to a study, about 66% of the total software projects investigated are characterized either by a delay in the schedule or have passed the foreseen budget, causing the loss of huge resources for the involved organizations [1]. The US economy, from 2005-2010, faced a loss of about 25 to 75 billion dollars due to the impact of the software project's failure [2]. Both overestimations and underestimations of the software project efforts are a disaster for the organizations. Indeed, one of the reasons for software project failures is the inaccurate and inefficient estimation of the software development effort [2]. Thus, there is a need for novel and sophisticated methods to increase the accuracy of SDEE in order to decrease the loss of budgets software companies encounter every year.

In the past three decades, a lot of research has been done in the area of software effort estimation which includes traditional and widely used approaches for building estimation models, like COnstructive COst MOdel (COCOMO) [3], and more recent machine learning (ML) methods, such as radial basis function (RBF) neural networks [4], regression tree [5], Support Vector Regression (SVR) [6] and bagging predictors [7]. Some investigations focused on the analysis of specific (functional) size measures aiming at improving the accuracy of effort predictions (e.g., [8,9]) or to obtain early the predictions (e.g., [10–12]). The used estimation techniques get the data from the past completed projects to build a regression model that is used for software effort estimation. However, there is no specific technique that can be considered more accurate than the others, and mostly it depends on factors such as the dataset type, size, and other characteristics of the dataset. However, several studies have shown that not all the attributes of a dataset are relevant and affect the accuracy of the estimation (e.g., [13–15]). For this reason, many researchers have investigated some feature selection algorithms to reduce the dataset features to select those that allow obtaining a better prediction accuracy. Thus, feature selection allows the identification and filtering of the irrelevant and redundant features which do not contribute to the accuracy of an estimation model (i.e., they reduce the overall estimation accuracy). Furthermore, the selection of features can also reduce model complexity and lead to low computational resources. And this can have an impact on the activities of the Software Quality Assurance (SQA) team to accurately and timely predict the most realistic amount of effort, expressed either in person-hours or person-months. Thus, the use of feature selection techniques is considered an important preprocessing step that has been investigated in the domain of Software Development Effort Estimation (SDEE) for two decades. According to a Systematic Mapping Study (SMS) [16], around 18 different features selection algorithms have been employed from 2000 till 2017 and the use of FS algorithms in SDEE has provided better predictions with the respect to the use of the full datasets. To further motivate the use of FS techniques in the context of SDEE, we want to highlight that researchers have highlighted not only their contributions in improving prediction accuracy but also other aspects. As an example, Menzies et al. [17] proposed that feature subset selection should be routinely carried out in the domain of SDEE because it does not only improve the prediction accuracy but also improve the stability of estimation techniques. Stability of an estimation technique can be measured by calculating the standard deviation of the prediction accuracy achieved by employing a feature selection algorithm divided by the prediction accuracy achieved by using the estimation technique alone (i.e., without any feature selection algorithm).

The performance of the machine learning technique largely depends on the quality and characteristics of the data that are used to train the system. However, the data can be redundant, noisy, and incomplete which could lead to the degradation of the trained model performance [18,19]. Feature selection is the process of choosing the subset of (most suitable) features which aims to design a better-performed learning system [18,19]. In particular, the relevancy of individual features is the first found and then the most appropriate and relevant subset features are identified and selected [20]. It is an optimization problem that searches all the possible feature subsets and comes out with an optimized one. An optimized subset of features may affect the accuracy and the learning time of the classifiers used for estimating problems [20,21].

Feature selection was the topic of debate from the early 90s and the earliest feature selection techniques were used by [21] considering dynamic programming and branch and bound. Feature selection methods are divided into: Filter [22], Wrapper [23], and Hybrid [24] methods. After that, different researchers worked for the selection of the most suitable features using different search techniques and evaluation functions. For instance, in [25] the authors compared the performance of two feature selection methods, i.e., correlation-based feature selection and wrapper feature selection, for the prediction of faults in a software system. The wrapper feature selection method resulted to be 33% more accurate than the correlation-based feature selection.

Various types of feature selections algorithms are used in the area of software effort estimation to select only the relevant subset of features. For this purpose, apart from the traditional feature selection algorithms, the researchers have more focused in the recent past on algorithms that are inspired by nature. These nature-inspired feature selection algorithms are usually called bio-inspired or simply metaheuristic algorithms. As the real-world problems getting more complex and multi-dimensional, the traditional feature selection algorithms take a significant amount of time and almost fail to solve the problem in a rational amount of time [26]. On the other hand, the bio-inspired algorithms manage to effectively provide a 'near to the best solution' for complex and multi-dimensional optimization problems in an acceptable amount of time [27].

After successfully employed in various optimization problems, bio-inspired algorithms have been recently used in the domain of software effort estimation to select the most relevant features of the datasets and have provided a better prediction accuracy. To obtain information about the application of bio-inspired feature selection algorithms in the context of effort estimation, we have recently performed a Systematic Literature Review (SLR) [28]. This SLR highlighted that in more than 90% of experiments the (bio-inspired) feature selection algorithms allowed to obtain better predictions with respect to the use of estimation techniques alone (i.e., when no features selection algorithms are used). Moreover, the analysis of

the selected papers has allowed us to highlight the list of widely used bio-inspired algorithms, the (bio-inspired) algorithms that had upper-hand over the other algorithms in terms of accurate estimations, and the datasets on which these algorithms were employed. Analyzing all the studies selected by our SLR we can also find that:

(a) Bio-inspired feature selection algorithms are compared with the baseline estimation techniques and not with other bio-inspired algorithms;
(b) Very few studies comparing the use of different bio-inspired algorithms consider in the analysis a maximum of two algorithms;
(c) No study compares the performance of bio-inspired algorithms with non-bio-inspired algorithms.

So, the literature seems to not provide a clear picture of which bio-inspired algorithms are able to have an impressive prediction accuracy, when compared with (1) baseline estimation techniques, i.e., applied alone, (2) other bio-inspired algorithms, and 3) other non-bio-inspired algorithms. We also believe that since the internal heuristics of the various bio-inspired algorithms are completely different to each other (i.e., pheromone trail of ACO, inertia weight for PSO, mutation operator for GA), their performance can be completely different from each other when employed with different datasets and hence need to be elaborated in detail.

All the above considerations motivate the empirical study presented here which not only evaluates and compares the performance of various bio-inspired algorithms to each other but also against the traditional non-bio-inspired algorithms, considering a variety of commonly used SDEE datasets. The bio-inspired algorithms are specially designed for problems like non-deterministic polynomial-time hard (NP-hard) and, since searching for the best features in search space is an NP-hard problem, the bio-inspired algorithms could be considered suitable for feature selection [29]. And, taking into account that bio-inspired algorithms as feature selection algorithms have not been investigated so far in a single comprehensive study and bio-inspired algorithms perform better with NP-hard problems, we conjecture that they can perform better than non-bio-inspired algorithms in the context of SDEE. Hence, we decided to perform an empirical study to evaluate the impact of various bio-inspired feature selection algorithms, comparing them to the traditional feature selection algorithms. The selection of the various employed bio-inspired feature selection algorithms takes into account the findings of our previous SLR [28]. Performing a comprehensive study can help future researches, suggesting in which cases bio-inspired algorithms need to be used and in which situations the traditional feature selection algorithms are enough as well as the case where best results are achieved without employing any feature selection algorithms.

In our study we employ six bio-inspired algorithms, i.e., Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Tabu Search (TS), Harmony Search (HS), and Firefly algorithm (FA). The selection of these algorithms is mainly based on the findings of our SLR [28], which states that GA and PSO are widely employed bio-inspired feature selection algorithms in the domain of SDEE. The other algorithms that are used in more than one study (among those selected by our SLR) are ACO, TS, HS, Differential Evolution (DE), and Artificial Bee Colony (ABC). However, we do not consider DE because we believe it is a special type of GA (which has already been employed in our study) and do not employ ABC because, in our initial analysis before finalizing our algorithms, we have found that ABC selects almost the same subsets of features as selected by PSO. Furthermore, we employ four traditional non-bio-inspired algorithms, namely Random Search (RS), Best-First Search (BFS), Subset Size Forward Selection (SSFS), Greedy Stepwise Search (GSS).

In order to assess the usefulness of the considered bio-inspired algorithms, we first provide a sanity check to evaluate if the (subset) selection of features has any significance, which ultimately leads to the improvement of the accuracy of the classifiers. In other words, we empirically investigate if the use of the (bio-inspired) feature selection algorithms provides better results than the estimation techniques employed without any feature selection algorithms. To this end, the first research question of our study is:

RQ1: Do the use of (bio-inspired) feature selection algorithms provide better results than the estimation techniques applied alone?

The second research question is to identify which bio-inspired algorithms if there are any, select the best subset of features and hence provide a prediction accuracy better than the non-bio-inspired algorithms. This research question also allows us to identify non-bio-inspired algorithms that outperform some bio-inspired algorithms.

RQ2: Which bio-inspired algorithms perform better than non-bio-inspired algorithms?

Our recent performed SLR [26] has highlighted that GA and PSO are the two bio-inspired algorithms (as feature selections) that have been employed in the earlier studies investigating bio-inspired algorithms for feature selection in the context of the effort estimation problem. So, our third research question aims to compare their accuracy with the one achieved with other algorithms recently proposed for various optimization problems, such as Ant Colony Optimization (ACO), Tabu Search (TS), Harmony Search (HS), and Firefly algorithm (FA).

RQ3: Do the recently employed bio-inspired algorithms (i.e., ACO, TS, HS, and FA) allow to obtain effort predictions better than those achieved with bio-inspired algorithms employed earlier (i.e., GA and PSO)?

Thus, to address RQ1, RQ2, and RQ3, we perform a variety of experiments considering six bio-inspired feature selection algorithms (GA, PSO, ACO, TS, HS, and FA) and four traditional non-bio-inspired algorithms (Best-First Search, Greedy

Stepwise, Subset Forward Selection, and Random Search), used in combination with five widely used estimation techniques (MultiLayer Perceptron, Support Vector Regression, Random Forest, Linear Regression, and M5P algorithm) and applied to eight publicly available datasets widely used in the SDEE community (Albrecht, China, COCOMO, Finnish, Kemerer, Maxwell, Miyazaki, and NASA).

The paper is organized as follows. In Section 2 we present the background and related work of our investigation. Section 3 describes the design of the performed empirical study, while Section 4 presents and discusses the achieved results. The summary and main findings are presented in Section 5. Conclusions will conclude the paper.

2. Background and related work

While the bio-inspired algorithms are employed in SDEE for parameter optimization to configure the hyperparameters of the estimation techniques [30–34], very few studies have investigated them for feature selection. For instance, the authors in [35] used Genetic Algorithm (GA) for feature selection (from ISBSG dataset) and compared its performance with the baseline estimation techniques (i.e., estimation techniques that are employed without any feature selection algorithms). GA managed to improve the prediction accuracy of the estimation techniques such as SVM and regression models. In [36], the authors employed GA with a variety of datasets, such as Desharnais, COCOMO, Albrecht, to extract the best features and produced a best prediction accuracy when compared to the one of the baselines, i.e., Neural Networks (NN), SVM and Bayesian models. Similarly, [37] also used GA as feature selection algorithm to improve the performance of ensemble methods. The authors in [38] have employed and compared the performance of three bio-inspired algorithms (ACO, GA, and PSO) which provided different results when used with different datasets. In [39], the authors used PSO to improve the performance of the baseline ANN. Similarly, the authors of [40] used PSO and compared the achieved performance with Albrecht and Gaffney and Kemerer models. Benala et al. [41] have compared the performance of Differential Evolution (DE) with PSO and found that DE provided better accuracy than PSO when used in combination with the Analogy based estimation approach. In [42], GA is used both for parameter tuning and feature selection to improve the prediction accuracy (when using Nasa, Albrecht, Desharnais datasets). Apart from the bio-inspired algorithms, some traditional feature selection algorithms are also investigated in the context of SDEE. In [43] two traditional feature selection algorithms, namely, Correlation-based Feature Selection (CFS) and RRelief, have been used in combination with the estimation techniques Support Vector Regression (SVR), Multilayer Perceptron, and Decision Tree, employing a variety of datasets. Similarly, the study in [43] has used Principal Component Analysis and Correlation-based Feature Selection (with the COCOMO dataset) and the results show that the effort predictions were better than those obtained with Artificial Neural Network. Note that bio-inspired algorithms were also investigated in the software engineering context for similar problems. As examples, Sarro et al. analyzed the use of the Genetic Algorithm (GA) to configure Support Vector Machine for inter-release fault prediction [44] and Andrews et al. used GA for finding parameters for randomized unit testing to optimize the test coverage [45].

As mentioned in the introduction we have recently performed a Systematic Literature Review (SLR) [28] to obtain information about the application of bio-inspired feature selection algorithms in the context of effort estimation. The analysis of the 30 different studies selected by our SLR revealed that the employed bio-inspired feature selection algorithms are Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Tabu Search (TS), Harmony Search (HS), Artificial Bee Colony (ABC), Differential Evolution (DE), Satin Bowerbird Optimization (SBO), Chaos Optimization Algorithm (COA), Cuckoo Search (CS), and, among them, those mostly used are GA (43%) and PSO (36%). ACO was used in only 3 (10%) studies while ABC, TS, HS, and DE were all employed in 2 (7%) studies. SBO and COA are the least used algorithms since both were investigated in a single study. The widely used datasets for the assessment of these techniques are NASA (12 studies) and Albrecht (6 studies), while COCOMO and ISBSG were used in 5 and 2 studies, respectively. Desharnais, CF, and China were used in only 1 study, while 2 studies did not provide information. The evaluation criteria frequently employed are Mean Magnitude of Relative Error (19 studies) and Prediction at level l=25 (11 studies), followed by Median of Magnitude of Relative Error (5 studies). Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are used in a few studies (3 and 2, respectively).

When comparing different bio-inspired feature selection algorithms to identify the one that allows to obtain better effort estimations, we can focus only on GA and PSO because they were employed in a higher number of studies. Among them, PSO can be considered as a better option because GA always performed worse when compared with other bio-inspired algorithms. Overall, we can observe that there is no bio-inspired algorithm that can be considered as the best algorithm. Indeed, an algorithm that performed well in a study considering some datasets and estimation techniques was outperformed by other algorithms in another study that employed other datasets and estimation techniques.

We want to highlight that 23 (77%) studies assessed whether the use of bio-inspired feature selection algorithms improved the prediction accuracy of estimation techniques. Furthermore, the best performed bio-inspired algorithms when compared with the baseline estimation techniques are PSO and GA**,** which improved the prediction accuracy of the employed estimation techniques in 9 different studies (for PSO the studies are [46–54] and for GA, the studies include [55–63] reported in our SLR [28]).

We also verified whether estimation techniques used alone (i.e., without any feature selection algorithms) provided better effort predictions than the same estimation techniques used in combination with bio-inspired feature selection algorithms. We found that ANN (Artificial neural networks) alone performed better than ANN together with PSO and GA in only three studies, followed by SVR (Support Vector Regression) which alone performed better in two studies [50,64] with respect to

its use in combination with PSO or TS to select the features. So, taking into account that we analyzed several estimation techniques, e.g., ANN, SVR, Classification and regression tree (CART), Analogy based estimation (ABE), Linear Regression, and Fuzzy logic, applied in 30 studies, we can conclude that the use of bio-inspired feature selection algorithms in SDEE can help since in the majority of the analyzed studies they allowed to improve the accuracy of the effort predictions.

A detailed discussion of the achieved results is presented in [28]. Above, we have reported a summary that can help to understand the motivations behind the empirical study we are presenting in this paper and the decisions we made to design the study.

3. Design of the study

In this section, we present the design of the performed empirical study by describing the employed datasets (Section 3.1), estimation techniques (Section 3.2), feature selection algorithms (Section 3.3), validation method (Section 3.4), evaluation criteria (Section 3.5), while threats to validity are discussed in Section 5.

3.1. Datasets

We have used eight different freely and publicly available datasets widely employed in the software effort estimation research community and we do believe that this number of different datasets can be considered good for our research goal**.** Prior research shows that by employing a relatively large number of publicly available SDEE datasets helps to make a stable conclusion [65]. The datasets used are Albrecht [66], China [67], COCOMO [68], Finnish [69] Kemerer [70], Miyazaki [71], Maxwell [72], and NASA [73]. The datasets Albrecht, Desharnais, and Maxwell are each one from a single company while the remaining are set of project data collected from different companies. The employed datasets have a high degree of diversity such as different number of features, ranging from 8 to 27, number of observations ranging from 15 to 499, differ in terms of technical characteristics (i.e., data from software projects developed in different programming languages, different application domains like telecommunications to commercial information systems) and geography locations (i.e., software projects came from countries like China, Canada, and Finland, etc). The dependent variable in all the datasets is "Effort", expressed in person-hours or person-month (for Nasa and COCOMO datasets). Features of these datasets are about:

- (Size attributes), information about the size of the project measured in terms of different measures, e.g., Lines of Code (LOC), Function Points etc.
- (Environment attributes): background information about the company, development team, the number of developers, experience of the developers, etc.
- (Development attributes): technical information about the project such as the type of database and programming language used in the project.
- (Project related attributes): about purpose, type, and requirements of the specific project.

The Albrecht dataset developed by IBM DP service organization consists of eight features and 24 observations from industrial software projects. It is characterized in terms of commonly used Kilo Source Lines of Code (KSLOC) and Function Points, which is a weighted sum of the numbers of inputs, outputs, files, and inquiries by a software system.

China dataset includes data of projects developed by Chinese companies and considers 19 features and 499 observations. Functional components to determine the number of Function Points (i.e., Input, Output, Inquiry, File, and Interface) are used as independent variables.

The COCOMO dataset is developed by Nasa and considers 17 features and 63 observations. The features include: rely (reliability of the software), data (size of the datasets), cplx (process complexity), time (cpu time constraint), stor (main memory constraint), virt (volatility of the machine), turn (turnaround time), acap (capability of analysts), aexp (application experience), pcap (capability of programmers), vexp (virtual machine experience), lexp (language experience), modp (modern programming practices), tool (use of software tools), sced (schedule constraint, loc (line of code).

Finnish dataset contains data of industrial software projects developed by 9 finish software companies and considers 9 features and 38 observations. The independent variables include PROD (productivity in terms of effort), HW (type of hardware), FP (Function Points).

Kemerer is one of the small size datasets which includes 7 independent variables and 15 observations. The independent variables have two categorical features (Language and Hardware), Adjusted Function Points (i.e., AdjFP), and Raw Function Points (i.e., RAWFP) which are based on KSLOC. The project duration and the total effort are the two dependent variables.

The Maxwell dataset includes information of industrial software projects developed by one of the biggest commercial banks in Finland and contains information about 62 projects. The important independent features include Function Points (SizeFP), T01 (customer participation), T02 (development environment adequacy), T03 (staff availability), T04 (standards used), T05 (methods used), T06 (tools used), T07 (softwares logical complexity), T08 (requirements volatility), T09 (quality requirements), T10 (efficiency requirements), T11 (installation requirements), T12 (staff analysis skills), T13 (staff application knowledge), T14 (staff tool skills), and T15 (staff team skill).

Miyazaki is a mid-sized dataset containing information of projects developed by 20 different software companies of Fujitsu Large Systems Users Group and considers 48 observations. The dataset considers 8 independent variables, while

Table 1
Datasets employed.

Dataset	Number of features	Number of observations
Albrecht	7	24
China	19	499
COCOMO	17	63
Finnish	9	38
Kemerer	8	15
Miyazaki	9	48
Maxwell	27	62
Nasa	24	93

the dependent variable is represented by the number of person-hours needed to carried out the development from system design to system test. The important dependent variables are different input or output screens (SCRN), different report forms (FORM), and number of different record formats (FILE).

Nasa93 includes information about 93 different projects developed by different Nasa centers between 1971-87. It consists of 15 discrete independent variables: rely (required software reliability), data (database size), cplx (process complexity), time (time constraint for cpu), stor (main memory constraint), virt (machine volatility), turn (turnaround time), acap (analyst capability), aexp (application experience), pcap (programmers capability), vexp (virtual machine experience), lexp (language experience), modp (modern programing practices), tool (use of software tools), sced (schedule constraint) in the range of Very Low to Extra High. It also has the variable software size measures in KSLOC.

Table 1 shows the details about the number of features and the number of observations for each dataset. Albrecht, Finnish, Kemerer, and Miyazaki are considered small datasets, in terms of both numbers of attributes and instances while China, COCOMO, Maxwell, and NASA can be considered as medium/large datasets.

These datasets represent an interesting set of software projects, containing data from a single software company or different companies that can represent a diversity of application domains and projects' characteristics. As mentioned above these datasets have been used in many empirical studies in the past for evaluating effort estimation methods [74–77]. In our empirical study, we did not consider the datasets CF and ISBSG, because CF is only employed in one study [52] selected by our SLR [28] and ISBSG dataset is used only in two studies [50] and [64 selected by our SLR [28]). Moreover, CF is not a widely used dataset in the SDEE literature and ISBSG dataset is not freely available (see threats to validity section for further discussion about this point).

3.2. Estimation techniques

We have used five estimation techniques from different families: Support Vector Regression (for SVR) [78], Random Forest (for Decision Tree) [79], MLP (for ANN) [80], Linear Regression [81], and M5P (for regression) [82].

Support Vector Machines (SVM): Introduced by Vapnik in 1995, SVM is used to solve classification and regression problems. When it is used in regression analysis, it is called SVR and can be used for linear and non-linear data analysis. In high-dimensional space, SVM creates single or multiple hyperplanes and maximizes the margin of separation around the separating hyperplanes. Various kernel functions (linear, polynomial, radial basis, and sigmoid) can be used to maximize the distance between the hyperplanes.

Random Forest (RF): RF is an ensemble classifier and a robust combination of tree predictors that have a lower percentage of overfitting and high accuracy of classifications [78**].** In the training phase, it creates different decision trees and they are combined by majority vote, so, the classification is given by the class that is given by the majority of the decision trees.

MultiLayer Perceptron (MLP): MLP is the type of feed-forward ANN which is inspired by the human brain and has a parallel and distributed processing structure [79]. ANN contains the uniform processing elements in layers (input, hidden, and output), called neurons which are connected with a specific architecture. Data in MLP is fed into the input layer, predicted in the output layer, and has some hidden layers which provide the level of abstractions.

Linear Regression: Linear regression is a linear model which attempts to draw a linear relationship between (explanatory) independent variables and a single dependent variable, by fitting a linear equation [80]: $Y = a + bX$, where X is the independent variable, Y is the dependent variable, a is the intercept, and b is the slope of the line.

M5P: M5P algorithm is the implementation of Quinlan's algorithm M5, which includes both the Model Trees and Regression Trees [81]. M5P can also be considered as a flavor of the decision tree with the linear regression function at the leaves of the tree and used to predict the continuous dependent variables.

The idea behind selecting these five techniques are to choose one estimation technique from the different flavors of the machine learning/regression techniques like ANN, SVM, Decision Tree, and Regression models. Moreover, the results of our previous SLR [28] summarized in Section 2 have revealed that ANN, SVR, Random Forest/DT, and Linear Regression are widely used estimation techniques. Also, ANN and SVR were the best-performed estimation techniques in most of the

studies selected by the SLR and MLP is the mostly employed flavor of the ANN. Finally, we selected M5P because it is available in the Weka tool. Indeed, we applied the implementation of the above estimation techniques provided by the tool Weka [83].

3.3. Feature selection algorithms

In general, preprocessing affects the performance of classifiers and feature selection is considered as one of the important steps in performing preprocessing, which exclude irrelevant and redundant features and hence produce more accurate and less complex models [84]. Moreover, ML algorithms are executed faster with the reduced data. The traditional feature selection algorithms perform well when the problem to be solved is not very big and complex. Indeed, when the problem size increases exponentially, their performance degrades gradually [85]. For huge, complex, and dynamic problems, bio-inspired algorithms performed well in recent years, working on the principle of the behavior of organisms [85]. The search and optimization problems like non-deterministic polynomial-time hard (NP-hard) ones require an increased amount of time to be solved, so the traditional algorithms can fail. Since the search of the optimal subset in feature space is an NP-hard problem, the use of bio-inspired algorithms are ideal as they can guarantee outstanding performances in many NP-hard problems [36]. In particular, bio-inspired algorithms create consecutive populations of individuals, considered as feasible solutions, aiming to search for a solution which gives the best approximation of the optimum for the effort estimation problem. To reach this goal, a fitness function is employed to evaluate the goodness (i.e., fitness) of the solutions represented by the individuals and different bio-inspired operators/factors (e.g., crossover and mutation with GA) are used to create new populations (e.g., as chromosomes with GA). So, we have to define a suitable fitness function to determine whether an estimation model leads to better predictions than another. In our case, the fitness function is given in terms of a widely used evaluation criteria, i.e., Mean of Absolute Error (MAE), to determine if a feature needs to be selected in the final list of features or should be dropped out. For that purpose, each feature has been allotted a 'fitness score' to determine if it is suitable for the final list selection [86]. Apart from the fitness function, we have also specified our stopping criteria, which are: 1) a predefined maximum number of iterations are reached, i.e., 20; 2) the best fitness value (i.e., in terms of MAE) of the population does not change appreciably for several iterations.

The behavior of the employed bio-inspired algorithms is different from each other when selecting the final subsets of features. For instance, GA works via different operators (i.e., crossover and mutation operators), while the other algorithms do not have such operators. PSO reaches the final subset of features through the best positions and velocities of the particles, while ACO achieves it via evaporation and pheromone intensity between two features. Similarly, TS works with a so-called 'move' operator, FA operates via the brightness of the fireflies and HS considers harmony memory and pitch adjustment, which acts like a mutation operator in GA [88].

In the following, we provide a brief description of each considered bio-inspired feature selection algorithm. Before that we want to clarify that in all our experiments, we have used the default parameters of bio-inspired algorithms (i.e., population size, number of iterations, etc.) set by the Weka tool as done in similar studies (e.g., [89–91]). Our main motivation of sticking with the default parameter settings is that we just wanted to let our readers know which bio-inspired algorithms (and non-bio-inspired algorithms) provided better prediction accuracy with which datasets and estimation techniques, rather than to focus on improving the prediction accuracy. Thus, we decided to use the following values for the three parameters shared by all the employed bio-inspired algorithms: Population Size=20, Iterations=20, Mutation Probability=0.01. Similarly, we have also selected the specific parameter values of bio-inspired algorithms (i.e., HMCR of HS, pheromone rate of ACO) already set by the Weka team (i.e., default values). Indeed, according to Eibe Frank (author of the Weka tool), most of the default parameter values set by Weka have been empirically found to work better in general [92]. In the threats to validity section we provide a further discussion about the choice of default parameter settings. Note that to apply bio-inspired algorithms, we have employed the package called Metaphor Search [87], created by researchers of Data Analytics and Collaborative Computing Laboratory (University of Macau) and embedded in Weka from version 3.7.4.

Genetic Algorithm (GA): The use of GA, proposed in the 1970s by John Holland, is motivated by Darwin's theory of evolution [93]. It is inspired by the natural selection and is based on four factors: the size of the population, mutation, crossover, and number of generations. To identify the best individual in the whole population, genes of the corresponding individuals are assessed against the objective function [94]. Crossover and mutation are the two operators through which a new offspring (individual) is produced. The entire (evolutionary) mechanism is continued until some convergence criteria are satisfied. GA can be used to produce the optimal solutions which make it ideal for the applications of feature selections. Apart from the population size, the number of generations, and mutation probability as defined above, for GA the value of crossover is set to 0.6. The procedure for selecting features is as follows. GA starts considering the original full dataset (with all features) as input, while the output is the reduced subset of features. In the first step, GA generates randomly the population of candidate solutions and evaluates the population using an estimation technique (e.g., RF). The best individuals (features) are then selected and reproduction is performed as long as the stopping criteria do not meet. The selected individuals are produced using crossover and mutation operations. The fitness of new individuals (features) is assessed and the worst individual (i.e., the one with the highest prediction error) is replaced with the best individual having less prediction error.

Particle Swarm Optimization (PSO): Particle Swarm Optimization, developed by J. Kennedy and R. Ederhart in 1995, mimics the behavior of flocking birds moving in a solution space [95]. This flocking behavior of birds determines the optimal

solution, which is finding the food. To reach the food, the birds follow some routes, and the shortest path followed by a bird is considered to be local or particle best solution. Particles have their local best solution and also have the knowledge of the global best solution, which is the shortest path found by any particle at a particular instance. Each particle also has a velocity through which it gets acceleration towards the local and global best path. The particles communicate to each other and thus reach their optimal solution (shortest path). This process continues until the maximum of the iterations or the stopping criterion are not met [95]. The inertia weight parameter of PSO is the weight factor which controls the exploration of the search space, while the individual weight and social weight are the acceleration constants that control the velocity of a particle towards the local or global best. We applied this setting: Individual Weight = 0.34, Inertia Weight = 0.33, and Social Weight = 0.33. PSO starts considering the original full dataset (with all features) as input, while the output is the reduced subset of features, and randomly initializes the particle position and velocity. At every iteration, each particle is updated with respect to two best values, i.e., (1) the best fitness value that the particle has obtained so far, denoted by pBest, and (2) another value considered as the best value obtained so far by any particle in the swam (gBest). The suitability of each particle (feature) is evaluated with an estimation technique (e.g., RF). While the maximum number of iterations or the stopping condition are not met, the gBest and pBest values are computed with the considered fitness function and the applied estimation technique. The velocity is measured for each particle and the positions are updated accordingly. In particular, when pBest and gBest values are identified, the particle updates its position and velocity with the following equation [96]:

$$v[] = v[] + c1 * \text{rand}() * \big(\text{pbest}[] - \text{present}[]\big) + c2 * \text{rand}() * \big(\text{gbest} - \text{present}[]\big)$$
$$present[] = present[] + v[]$$

where v[] is the velocity of a particle, present[] is the current particle, pbest[] and gbest[] are the personal and the swarm best values. rand() is a random number in the range (0,1). c1, c2 are learning factors and they are defined as c1 = c2 = 2.

Ant Colony Optimization (ACO). ACO, proposed by Marco Dorigo in 1992, is a computational technique inspired by natural ants that intelligently find the shortest path between their nest and food source [97]. Each ant is a solution to an objective function and establishes its communication through a special liquid called pheromone. When the ants start the journey from their nests to search for foods, they move randomly in a particular direction and dropping pheromone in their path. On reaching the destination (food source), ants return back with food, leaving pheromone again on the same path. Thus, the path with a large amount of pheromone represents the best (shortest) route from an ant's nest to the food source. In the ACO parameter settings, the pheromones are the special chemicals which the ants drop on the path while building the solution and could be evaporated at a specific rate. The heuristics are the distance or visibility from the current node of the ant to the next visited node. In our experiments, the pheromone rate is set as 2.0, evaporate rate as 0.9, and the heuristic rate is set as 0.7. The process of feature selection with ACO begins by generating a number of ants, namely k, which are then placed randomly on the graph (i.e., each ant starts with one random feature). Alternatively, the number of ants to place on the graph may be set equals to the number of features within the data and each ant starts path construction at a different feature. From these initial positions, they traverse edges probabilistically until a traversal stopping criterion is satisfied (i.e., best MAE value is achieved). The resulting subsets are gathered and then evaluated. If an optimal subset has been found or the algorithm has executed a certain number of times, then the process ends and outputs the best feature subset selected. If neither condition holds, then the pheromone is updated, a new set of ants are created and the process iterates once more. In particular, the pheromones are updated in a way that the shortest paths get more pheromone compared to the paths that are longer in distance. The transition of an ant from a feature i to another j at a time t could be obtained as follows:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in J_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}} & if\ j \in J_i^k \\ 0 & otherwise \end{cases}$$

where τ_{ij} is the intensity/amount of pheromone trail between the current feature i and next feature j. α is the parameter that controls the influence of τ_{ij}. η_{ij} (optional) shows the distance from feature i to j while β controls the influence of η_{ij}.

The pheromone on each edge is updated according to the following formula:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}$$
$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

where $\Delta\tau_{ij}$ is the increase in the intensity of the pheromone trail between the features i and j.

Tabu Search (TS). TS is another meta-heuristic algorithm which is proposed by Fred Glover to cope with some of the problems of Local Search (LS) heuristics [98]. TS has successfully been used to get the optimal and near-optimal solutions

in different application domains, e.g., scheduling, telecommunications, character recognition, and artificial neural networks [98]. It starts with a random solution and applies the local transformations to the current solution i, to make the neighboring solutions, N(i), and when no N(i) solutions do exist, TS performs a temporary worsening move. When the maximum number of iterations is reached, the search terminates. One of the important parameters of TS is the diversification, which (unlike intensification that only examines the neighbors of elite solutions) encourages to evaluate the unvisited regions/neighbors and produces solutions that are significantly different than those seen before. We set the diversification probability as 1.0, which is a default value. In this case feature selection starts with the initial solution, i.e., all features are included. Neighbors are then generated by adding or excluding a feature randomly in the feature vector n. For example, if 11001 is the current feature vector, then the possible neighbors with a candidate list size of 3 can be 10001, 11101, 01001. Among the neighbors, the one with the best cost (i.e., the one which leads to better prediction accuracy) is selected and considered as a new current solution for the next iteration. A list (called Tabu list) is maintained to prevent returning to previously visited solutions (i.e., moves). In this way, if a feature is added or excluded at iteration i, then the same feature is added to the Tabu list, along with the information if the feature is added or excluded. The algorithm terminates with a predefined number of iterations reached.

Firefly Algorithm (FA). Another nature-inspired algorithm is the Firefly algorithm, which is a multimodal optimization algorithm and inspired by the behavior of fireflies or lightning bugs [99]. First introduced in 2007 by Xin-She, FA has performed well in different domains when compared to the other metaheuristic algorithms [99]. FA has three basic rules: (a) each firefly is attracted by the others regardless of the sex; (b) the attractiveness is proportional to the brightness such that the fireflies with more brightness attract the one with low and if there is no brighter fly, the movement becomes random; (c) the fireflies' brightness is determined by the landscape of the objective function. As for the FA parameter settings in Weka, the beta is the brightness of the bugs and the absorption coefficient represents γ, which is the fixed light absorption coefficient. We applied this setting: beta $= 0.33$ and Absorption Coefficient $= 0.001$. As for the selection process, at the beginning each firefly is randomly generated as the weight of features, i.e., Xi={Wi1, Wi2, Wi3, ..., Wid}, where d is the number of all features and $i = 1, 2, \ldots, n$. Light intensity is determined by the attractiveness of the fireflies and it is calculated using the following equation:

$$\beta(r) = \beta_0 e^{-\gamma r^2}$$

where β_0 shows the attractiveness of the fireflies at distance $(r) = 0$ and γ indicates the intensity of the light absorption. r represents the (Eucledian) distance between two fireflies (features) i and j that are in different positions. The attractiveness of the two firefies depends on the distance between them. The movement of a firefly i that is attracted by another firefly j is determined by the following equation:

$$x_i = x_i + \beta_0 e^{-\gamma r_{j,j}{}^2}(x_i - x_j) + \alpha\left(rand - \frac{1}{2}\right)$$

where the second term of the above equation represents the attraction of the fireflies and the third term describes the randomization parameter (α). Note that Yang [25] proposed $\beta_0 = 1$, $\alpha \in [0, 1]$, and $\gamma \in [0.01, 100]$, so FA selects values in these ranges.

The light intensity li for the feature Xi is determined by the objective function and if $lj > li$ the firefly i is moved towards j in d-dimension. Subsets of features are selected if Wij is greater than threshold τ. The prediction accuracy is calculated (with an estimation technique) and the light intensity is updated. If no one of the fireflies is brighter than li, then li moves randomly. The fireflies are ranked and the current best is found. The process continues until maximum iterations are reached.

Harmony Search (HS). HS is the emerging bio-inspired algorithm that mimics the process of musician's improvisation [100]. HS has been successfully applied in many domains like traveling salesman and chaotic systems. It allows us to obtain three components: employment of harmony memory, pitch adjusting, and randomization. The HS's optimization operators can be specified as "harmony memory" which keeps the solution vectors within the search space. The steps of HS are: initialize the problem and parameters, initialize the harmony memory, improvise and update the harmony and finally check the stopping criteria. In the parameter settings of the HS, the HMCR (Harmony Memory Considering Rate) is the probability of choosing a component from the harmony memory and its value is set to 0.9 by default. The Pitching Adjust Rates (PAR) values describe the probability of a candidate from the harmony memory to be mutated and its values range between 0.4 (minimum) to 0.9 (maximum). As for the selection process, HS starts with the initialization of parameters such as HMCR, PAR, Harmony Memory (HM), and maximum iterations. Each feature is called 'Harmony' and is represented by the n-dimension real vector. The initial population of harmony vectors is generated randomly and stored in the harmony memory (HM). After that, a new candidate harmony (feature) is generated from all of the solutions of the HM by using an HMCR, a PAR, and a stopping criterion or the number of improvisations. The HM is finally updated by comparing and evaluating the new harmony with the worst harmony vector in the HM. If the new harmony vector has better fitness function (i.e., in terms of MAE values) than the worst harmony in the HM, then the new harmony is added in the HM and the existing worst harmony is excluded from the HM. If the stopping criteria is reached the algorithm terminates, otherwise new harmony is generated and the HM is updated.

As for the non-bio-inspired feature selection algorithms considered in our study to accomplish a comparison with the bio-inspired ones, we have employed four traditional feature selection algorithms available in the Weka tool, namely Random Search (RS), Best-First Search (BFS), Subset Size Forward Selection (SSFS), Greedy Stepwise Search (GSS). In the following, we provide a brief description of each algorithm.

Random Search (RS): RS is an optimization method that can be used on functions that are neither continuous or differentiable and neither require the gradient of the problem to be optimized [101]. In the search space, RS iteratively moves towards the best solution, sampled from a hypersphere surrounding the current position. Several flavors are available of RS, such as Optimum step size RS, Adaptive step size RS, Fixed step size RS etc.

Best-First Search (BFS): BFS selects a subset of features randomly using the greedy hill climbing and supplemented with the backtracking [102]. In BFS, initially, an empty set of features is selected for forward search, a full set of features are selected for backward search or, in other cases, it begins in the middle and starts search both ways to examine the addition/deletion of distinct features at any locations.

Subset Size Forward Selection (SSFS): SSFS is an extension of the Linear Forward Selection, which in turn is an extension of the Best First search algorithm [102]. The subset forward selection executes the k-fold cross validation specified by the users and subset size is selected by running the Linear Forward Selection on every fold. Finally, the overall dataset is used to execute a Linear Forward Selection until the prime subset-size.

Greedy Stepwise Search (GSS): GSS implements a greedy forward or backward search using the features subset space [103]. It stops the search when the addition or deletion of the remaining features causes a decrease in evaluation. It produces a list of attributes by moving from one side of the space to the other and record the order that attributes are selected.

3.4. Baselines

We have different baselines for all of our three RQs. For instance, for RQ1 (which indeed is a sanity check), the baselines are the estimation techniques employed without a feature selection algorithm, which are evaluated against the bio-inspired algorithms by testing the following null hypotheses:

Hn1_X_D_T: The effort predictions obtained with estimation technique T alone (i.e., without employing any feature selection algorithms) are (significantly) better than those achieved using X as a bio-inspired feature selection algorithm, for the datasets D.

whereas X $\in$ {Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, Tabu Search, Harmony Search, and Firefly Algorithm}, D $\in$ {Albrecht, China, COCOMO, Finnish, Kemerer, Maxwell, Miyazaki and Nasa} and T $\in$ {MultiLayer Perceptron, Support Vector Regression, Random Forest, Linear Regression, and M5P algorithm}. In case the null hypothesis can be rejected, the alternative hypothesis holds:

Ha1_X_D_T: The effort predictions obtained with estimation technique T alone (i.e., without employing any feature selection algorithms) are not (significantly) better than those achieved using X as a bio-inspired feature selection algorithm, for the datasets D.

Similarly, for RQ2, we investigated and compared the performance of bio-inspired algorithms to the non-bio-inspired feature selection algorithms and hence, thus the non-bio-inspired algorithms are considered as the baseline. The null hypotheses are:

Hn2_X_Y_D_T: The effort predictions obtained using the bio-inspired feature selection algorithm X are not significantly better than those achieved employing the non-bio-inspired feature selection algorithm Y, with the estimation technique T and for the dataset D.

whereas X is as above, Y $\in$ {Best-First Search, Greedy Stepwise Search, Subset Forward Selection, Random Search}, X1 $\in$ {Genetic Algorithm, Particle Swarm Optimization}, and X2 $\in$ {Ant Colony Optimization, Tabu Search, Harmony Search, and Firefly Algorithm}. In case the null hypothesis can be rejected, the alternative hypothesis holds:

Ha2_X_Y_D_T: The effort predictions obtained using the bio-inspired feature selection algorithm X are significantly better than those achieved employing the non-bio-inspired feature selection algorithm Y, with the estimation technique T and for the dataset D.

For RQ3, we have evaluated and compared the performance of earlier-employed bio-inspired algorithms with those of bio-inspired algorithms employed more recently and hence the earlier-employed bio-inspired algorithms (GA and PSO) are marked as a baseline. The null hypotheses are:

Hn3_X1_X2_D_T: The effort predictions obtained using the bio-inspired feature selection algorithm X1 are not significantly better than those achieved employing the bio-inspired feature selection algorithm X2, with the estimation technique T and for the dataset D.

whereas X1 ∈ {Genetic Algorithm, Particle Swarm Optimization}, and X2 ∈ {Ant Colony Optimization, Tabu Search, Harmony Search, and Firefly Algorithm}. In case the null hypothesis can be rejected, the alternative hypothesis holds:

Ha3_X1_X2_D_T: The effort predictions obtained using the bio-inspired feature selection algorithm X1 are significantly better than those achieved employing the bio-inspired feature selection algorithm X2, with the estimation technique T and for the dataset D.

3.5. Validation method

Cross-validation is the process in which we estimate the predictions' error to evaluate the performance of our estimation model and has a single parameter, k, which denotes the number of groups we intend to split in [104]. In particular, k-fold cross-validation, or simply rotation estimation, allows to divide a dataset into equally-sized k subsets. The estimation model is trained and tested k times, i.e., one subset (portion) is used for testing and the remaining $k-1$ subsets are used for building and training the estimation model [105]. In our investigation, we applied a 10 fold-cross-validation, so for ten times each considered dataset is divided into 10-subsets, one of which is used for testing and the remaining 9-subsets are used to build and train the estimation model. It is worth noting that for those datasets having few observations, Weka allows to perform the process of stratification, which means the rearrangement of data to make sure each fold is a good representative of the whole. But stratification is performed when there is a classification problem. For regression data, Weka just shuffles and split the data into 10 folds. For Kemerer dataset, we have only 15 observations, so we evaluated it using 3-fold cross-validation.

We used the 10-fold cross-validation because it is the widely employed validation method in software engineering (used in 49% studies, followed by holdout validation in 45%) according to the literature by Chakkrit et al. [106].

3.6. Evaluation criteria

Different studies used different evaluation measures to evaluate their prediction accuracy, most of them are based on the Absolute Error, i.e., the difference between actual and predicted efforts. Widely used metrics are MMRE and Pred(25) [107], as also highlighted in our recent SLR [28]. However, both of these have been criticized because of being biased towards underestimation and behaving very differently when working with prediction models [108–110]. In our study, we have decided to employ Mean Absolute Error (MAE), which is widely recommended (see e.g., [111,112]). MAE measures how our estimated value differs from the actual value. MAE is easy to compute because it directly calculates the mean of the absolute errors:

$$MAE = \frac{\sum_{i=1}^{n} |p_i - a_i|}{n}$$

Moreover, we verified if the absolute errors/residuals obtained by applying different estimation models come from the same population to assess whether the differences observed using the above evaluation criteria were legitimate or due to chance. In particular, we used a statistical test, namely paired samples Wilcoxon Signed Rank test (with $\alpha = 0.05$), to verify statistical significance differences in the absolute residuals achieved by two different estimation models [113]. Since our data are not normally distributed (we test it using Shapiro Test [114]), the T-test is not suitable in our case. The Wilcoxon test could be regarded as a safe test to use (even for normally distributed data), since it raises the bar for significance, by making no assumptions about underlying data distributions. A p-value of less than 0.05 indicates that the null hypothesis (there is no difference between the absolute residuals obtained with the two considered models) can be rejected. If the p-value is greater than 0.05, we have no proof that a model has better performance than the other model. Since the (Wilcoxon) statistical test gives us information about a possible significant difference between the two samples, it is recommended to calculate if the effect size is notable. So, to this aim, we used the Vargha and Delaney's A12 statistics [115]. Indeed, according to [116,117], when the data is not normally distributed, we need to use the Vargha and Delaney effect size test rather than the pooled Cohen'd. The formula of Vargha and Delaney's is as: A12=(R1/m-(m+1)/2)/n, where R1=rank sum of the first group of samples and m and n are observations of both samples. The effect size can be regarded as small if A12 is from 0.57 to 0.64, medium if A12 is from 0.65 to 0.71, and can be considered as large if it is over 0.71. The effect size equals or less than 0.56 could be considered as negligible. We are interested in any predictive performance improvement, no transformation of A12 is required as suggested by [115].

4. Results and discussion

In this section, we address all of our research questions and evaluate the performance of all feature selection algorithms for each dataset and estimation technique. As mentioned above, we have used the tool Weka [118] to apply the estimation techniques considered in our investigation, while all of the bio-inspired algorithms employed are included in a package

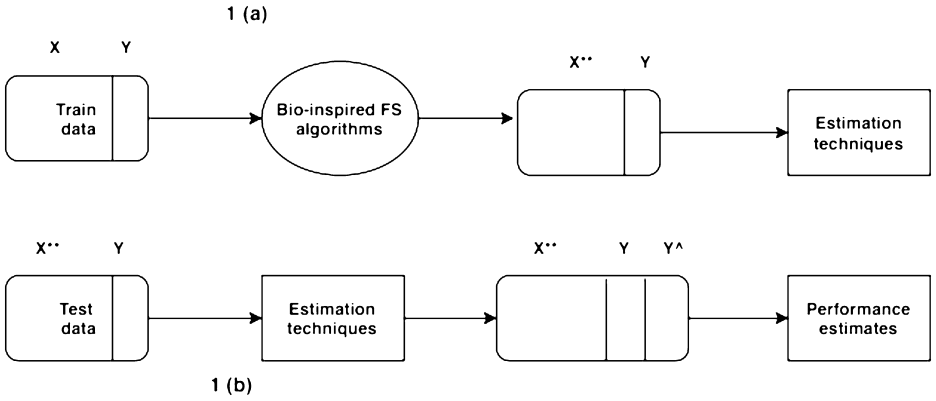

Fig. 1. Steps used for feature selection and performance estimation.

called Metaphor Search [87], created by the researchers of Data Analytics and Collaborative Computing Laboratory (University of Macau), which is upward compatible with Weka 3.7.4 and above. All the experiments are gauged and assessed in terms of MAE values. Furthermore, it is worth noting that we have used the 'Attribute Selected Classifier' option because it allows feature selection based on training data only. In this case, in a preprocessing step (see Fig. 1), the bio-inspired algorithm selects the subset of features (original dataset features are represented by X while reduced features are represented by X**, Y is the dependent variable) and the reduced data is used to train an estimation technique (classifier). Then, the test phase comes in Fig. 1(b) where the trained estimation technique is evaluated through an independent test set with the reduced features X** obtained in Fig. 1(a). The estimation technique predicts Yˆ for each observation. Similarly, various measures can be obtained by comparing the prediction Yˆ and the ground-truth Y. Researchers [118,119] as well as the Weka development team [117] recommend this procedure, rather than using the 'Select Attributes' option in Weka where the whole dataset (both train and test data) is used for feature selection.

4.1. RQ1: Do the use of (bio-inspired) feature selection algorithms provide better results than the estimation techniques applied alone?

We have performed a sanity check to evaluate if the (subset) selection of features from our eight employed datasets has any significance, which ultimately leads to the improvement of the accuracy of the classifiers. To this end, to answer RQ1, we verify whether, for each employed dataset, bio-inspired algorithm, and estimation technique, the predictions obtained using the bio-inspired feature selection algorithm are statistically (significantly) better than the ones achieved using the estimation technique alone (i.e., without employing any feature selection algorithms). Thus, we have about 240 different experiments (or cases) taking into account all the combinations of 6 bio-inspired algorithms, 5 estimation techniques, and 8 datasets.

As described in the study design section, to accomplish this, we have employed the Wilcoxon test, which is a non-parametric test to find the degree of significant difference. In particular, the result in terms of the p-value explains the significant difference between the absolute residuals achieved using the bio-inspired algorithm and those achieved using the baseline (i.e., the estimation technique used alone). In particular, when comparing the predictions of two models, a p-value less than 0.05 indicates a particular model had a significantly better performance than another. If the p-value is greater than 0.05 then we have no proof and cannot claim anything about their performance.

Analyzing all the experiment results, we have found that in about 157 (65%) experiments, the bio-inspired feature selection algorithms provided better results than the baseline techniques. In particular, in 92 experiments the bio-inspired algorithms significantly outperformed the baseline, while in 106 experiments there is a notable effect size of either small, medium, or large. Table 2 shows the experiments for which we can reject the null hypotheses defined in the study design section (i.e., **Hn1_X_D_T**) or a notable effect size is obtained, where the first column shows the estimation technique (baseline) and the dataset employed, while the second column the bio-inspired algorithms that allow to obtain significantly better predictions. The values between brackets are the p-values of the Wilcoxon test performed to accomplish the comparisons and the capital letter indicates the corresponding effect size (N for Negligible, S for Small, M for Medium, and L for Large). Note that the cases for which a p-value is greater than 0.05 (i.e., no significant difference) but with a notable effect size are a few, i.e., LR/Maxwell, MLP/Nasa, SVR/Maxwell, and M5P/China). In other 65 experiments, the bio-inspired algorithms provided better results but not significantly. In the other 52 experiments, the baseline techniques outperformed the bio-inspired algorithms, however, only in 12 experiments, they provided significantly better predictions while in 24 experiments with a notable effect size (see Table 3). In a few experiments with small datasets (i.e., Albrecht with LR, Kemerer with SVR, and Finnish with RF), the predictions obtained are comparable.

Fig. 2 graphically shows the number of experiments where bio-inspired algorithms provided better results than the baseline estimation techniques (in blue) and where baseline performed better than bio-inspired, considering when the

Table 2
Bio-inspired algorithms that performed (significantly) better w.r.t baseline estimation techniques, with the p-value of the performed statistical test and effect size between brackets.

Baseline/Dataset	(Bio-inspired Algorithms) Performing Significantly Better
LR/Albrecht	ACO (0.03 S), GA (0.03 S), FA (0.03 S), PSO (0.03 S), TS (0.03 S), HS (0.03 S)
LR/Maxwell	ACO (0.004 L), TS (0.04 L), HS (0.04 L)
LR/Nasa	ACO (<0.001 L), PSO (<0.001 L), TS (<0.001 L), FA (<0.01 L), GA (<0.01 L), HS (<0.001 L)
LR/Maxwel	GA (0.6 L), PSO (0.6 L), FA (0.6 L)
MLP/China	GA (0.003 L), TS (0.003 L), HS (<0.001 L)
MLP/Finnish	ACO (0.003 S), GA (0.003 S), FA (0.003 S), PSO (0.003 S), TS (0.003 S), HS (0.002 S)
MLP/Maxwell	GA (0.03 L), PSO (0.03 L), FA (0.03 L), TS (0.03 L), HS (0.001 L)
MLP/Nasa	GA (<0.001 L), FA (0.005 L), PSO (0.005 L), TS (0.005 L), ACO (0.3 S), HS (0.7 S)
RF/COCOMO	ACO (0.02 L), GA (<0.001 L), FA (0.002 L), PSO (0.002 L), TS (0.002 L), HS (0.04 L)
RF/China	GA (0.04 L), TS (0.04 L), HS (<0.001 L),
RF/Maxwell	ACO (0.01 M), GA (0.04 M), PSO (0.01 M), FA (0.01 M), TS (0.01 M), HS (<0.001 L)
RF/Miyazaki	ACO (0.02 L), GA (0.02 L), FA (0.02 L), PSO (0.02 L), TS (0.02 L), HS (0.02 L)
RF/Nasa	HS (0.02 L)
SVR/COCOMO	ACO (0.03 S), GA (0.005 S), FA (0.05 S), PSO (0.05 S), TS (0.05 S), HS (0.03 S)
SVR/Maxwell	ACO (0.03 S), FA (0.03 S), PSO (0.03 S), TS (0.03 S), GA (0.4 M), HS (0.7 S)
SVR/Miyazaki	ACO (0.05 M), GA (0.05 M), FA (0.05 M), PSO (0.05 M), TS (0.05 M), HS (0.05 M)
SVR/Nasa	ACO (0.03 L), GA (0.03 L), FA (0.03 L), PSO (0.03 L), TS (0.03 L), HS (<0.001 L)
M5P/Albrecht	ACO (0.04 M), GA (0.04 M), FA (0.04 M), PSO (0.04 M), TS (0.04 M), HS (0.04 M)
M5P/China	ACO (0.04 L), GA (0.04 L), HS (0.004 L), PSO (0.2 S), TS (0.2 S), FA (0.2 S)
M5P/Miyazaki	HS (0.01 L)
M5P/Nasa	ACO (0.03 M), PSO (0.03 M), TS (0.03 M), FA (0.03 M), HS (0.001 M)

Table 3
Baseline estimation techniques that performed (significantly) better w.r.t bio-inspired algorithms, with the p-value of the performed statistical test and effect size between brackets.

Baseline/Dataset	(Bio-inspired Algorithms) Performing Significantly worse
MLP/Miyazaki	ACO (0.03 L), GA (0.03 L), FA (0.03 L), PSO (0.03 L), TS (0.03 L), HS (0.03 L)
MLP/COCOMO	ACO (0.002 L), GA (0.007 L), FA (0.002 L), PSO (0.002 L), TS (0.002 L), HS (<0.001 L)
LR/Kemerer	ACO (0.1 L), GA (0.1 L), FA (0.1 L), PSO (0.1 L), TS (0.1 L), HS (0.1 L)
MLP/Kemerer	ACO (0.4 S), GA (0.4 S), FA (0.4 S), PSO (0.4 S), TS (0.4 S), HS (0.4 S)

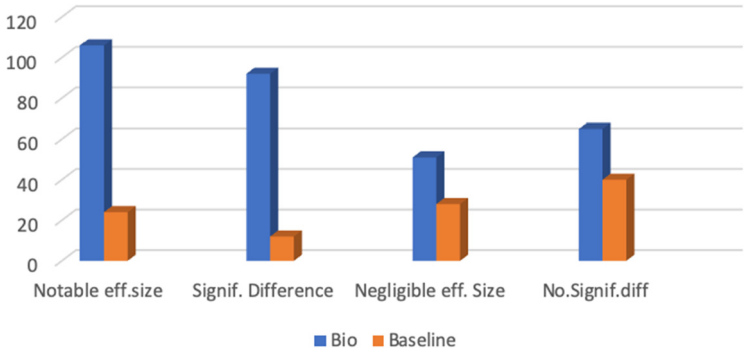

Fig. 2. Number of experiments for which bio-inspired algorithms or baselines provided better predictions. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

differences are significantly (i.e., Signif. Difference), not significantly (i.e., No Signif.diff) and with notable effect size or negligible. This visual summary allows to easily and quickly understand the achieved results.

We can conclude that in the majority of the experiments the effort predictions obtained using X as a bio-inspired feature selection algorithm are (significantly) better than those achieved with the baseline estimation techniques T alone when employed with the dataset D, and hence we can reject the corresponding null hypothesis and accept the alternative hypothesis. Thus, ***we can positively answer our first research question.***

To highlight which bio-inspired algorithms among those considered in our study performed better, i.e., provided accuracy improvement in a greater number of experiments, we show in Fig. 3 the results for individual bio-inspired algorithms. We can see that HS is the leading algorithm that provided better prediction accuracy (significantly) in more number of experiments compared to the other bio-inspired algorithms, followed by the TS and GA.

To analyze in more details our results, we want to focus on those feature selection algorithms that allowed to significantly outperform the baselines with a medium or large effect size (i.e., excluding here the discussion of the algorithms which significantly outperformed the estimation techniques applied alone with a negligible or small effect size). In this

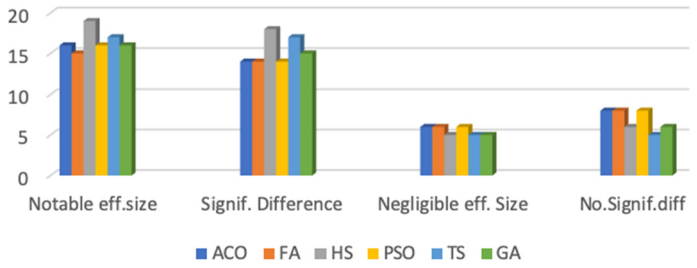

Fig. 3. Number of experiments for which bio-inspired algorithms performed *better* than baseline techniques.

case, the bio-inspired algorithms which figured-out as best feature selection algorithms are HS (in 15 experiments), TS (14 experiments), GA (12 experiments), and ACO, PSO, and FA (all in 10 experiments each). Thus, HS and TS can be considered as the bio-inspired algorithms that performed better, when it comes to explicitly addressing RQ1. Moreover, focusing on the datasets, the ones with which the bio-inspired algorithms provided better performance (significantly) than the baseline techniques are Nasa (22 experiments), Maxwell (19 experiments), and COCOMO (17 experiments). The worst performed dataset, in this case, is Kemerer, with which the bio-inspired algorithms performed better in no experiment. Similarly, the best-performed estimation techniques with the bio-inspired algorithms are SVR and RF which significantly outperformed the baseline techniques in 29 and 22 different experiments, respectively. Linear Regression, in this case, is the worst performed estimation technique when used in combination with the bio-inspired algorithms (which significantly performed better in 15 experiments).

Based on the results above, we can infer that feature selection algorithms have a major contribution in the accuracy of software effort estimations and, hence, need to be evaluated further to get the idea of which of these feature selection algorithms provides better performance with various datasets and estimation techniques. About the datasets related to software efforts estimation, according to [84], quite a reduced number of features are sufficient to build an effective software estimation model. Also, though specifically, it refers to the datasets in software fault predictions, [81] states that only 10% of features of a particular dataset are sufficient to provide a better predictive model. Before exploring the role and contribution of various feature selection algorithms, we analyze the individual features of the datasets to determine which features are most important. More specifically, we intend to analyze the features of the eight datasets which are not excluded by any feature selection algorithms, and hence could be regarded as very important features for the software efforts estimation. For instance, for Albrecht dataset, these features are 3/8 (Output, Inquiry, RawFPcounts), China 3/19 (Output, Duration, N-effort), Finnish 4/9 (FP, Prod, Insize), Kemerer 3/8 (Language, KSLOC, AdjFP), Maxwell 5/27 (Syear, T02, T03, T07, T08, T14), Miyazaki 5/9 (KLOC, SCRN, FORM, EFORM, EFILE), Nasa 11/24 (Recordnumber, Cat2, Forg, Center, Year, Mode, Data, Stor, aexp, Vexp, equivphyskloc), COCOMO 9/17 (Rely, Data, Stor, Acap, Pcap, Lexp, Modp, loc). These are the type of features of the datasets which no feature selection algorithms have managed to exclude. It also means that almost all of the employed datasets have features that do not contribute to the accuracy of estimation and, hence, the use of feature selection algorithms can help in the domain of software effort estimation.

4.2. RQ2: Which bio-inspired algorithms perform better than non-bio-inspired algorithms?

To address RQ2 we have compared each bio-inspired algorithm (ACO, PSO, TS, FS, HS, and GA) to each non-bio-inspired algorithm (BFS, GSS, RS, and SSFS). So, we have considered 40 combinations of datasets (8) and estimation techniques (5) and for each of these combinations we have applied both bio-inspired (6) and non-bio-inspired (4) features selection algorithms, obtaining 960 single comparisons. Note that when we indicate that a bio-inspired algorithm outperforms a non-bio-inspired algorithm, it means that the bio-inspired algorithm has allowed obtaining better effort predictions (i.e., summary measure results from Weka).

Table 4 shows the experiments for which we can reject the null hypotheses defined in the study design section (i.e., **Hn2_X_Y_D_T**) or a notable effect size is obtained, where the first column shows the bio-inspired algorithms that allow obtaining significantly better predictions than the non-bio-inspired algorithms in the second column, with the setting reported in the third column. Moreover, to provide a visual summary that allows to easily and quickly understand the achieved results, Fig. 4 shows the number of experiments for which bio-inspired algorithms provide better predictions than non-bio-inspired algorithms, considering when the differences are significantly (i.e., Signif. Difference), not significantly (i.e., No Signif.diff), and with notable effect size or negligible effect size.

We observed that ACO has performed better (in terms of MAE) than non-bio-inspired algorithms in 23 comparisons when employed with the following settings: SVR/Nasa, SVR/Maxwell, Linear Regression/Maxwell, MLP/China, M5P/Maxwell, SVR/Albrecht, and SVR/COCOMO. In only 4 comparisons, we have found a statistically significant difference in the absolute residuals since p-values less than 0.05 are obtained running the Wilcoxon test, in the combination of Linear Regression/Maxwell against GSS, SSFS and RS and SVR/Nasa against GSS. In the rest of the experiments, though the ACO got better results in terms of MAE, we cannot prove it statistically. Similarly, in 9 different experiments, there is either small, medium or large effect size while in the other 15 experiments, the effect size recorded is negligible.

Table 4
Bio-inspired algorithms that performed (significantly) better (Better) w.r.t. non-Bio-inspired algorithms (Worse), with the p-value of the performed statistical test and effect size between brackets.

Better	Worse	Setting
ACO	GSS (0.03 M), RS (0.7 M)	SVR/Nasa
	BFS (0.7 S), GSS (0.03 L), SSFS (<0.001L), RS (0.03 L)	LR/Maxwell
	BFS (0.4 S), SSFS (0.4 S), RS (0.4 S)	M5P/Maxwell
TS	BFS (0.3 S),	MLP/China
	BFS (0.05 L), RS (0.05 L)	Random Forest/Maxwell
	BFS (0.003 L), GSS (0.01 L), SSFS (<0.001 L), RS (0.01 L)	Random Forest/China
	BFS (0.3 S), SSFS (0.04 M), RS (0.04 M)	LR/Maxwell
PSO	SSFS (0.2 S), RS (0.2 S)	Linear Regression/Maxwell
	BFS (0.2 M), GSS (0.2 M), SSFS (0.7 M), RS (0.7 M)	Linear Regression/China
	BFS (0.009 M), GSS (0.01 S), SSFS (0.008 M), RS (0.008 M)	Linear Regression/Nasa
HS	GSS (0.4 S)	M5P/Miyazaki
	RS (0.02 S)	Linear Regression/Maxwell
	BFS (<0.001 L), SSFS (<0.001 L), RS (<0.001 L)	MLP/China
	BFS (0.4 S), GSS (0.6 S), SSFS (0.4 S), RS (0.4 S)	M5P/Maxwell
	BFS (0.3 M),	Random Forest/China
	BFS (0.04 M), GSS (<0.001 L), SSFS (0.04 M), RS (0.004 M)	Linear Regression/NASA
	GSS (0.02 N), SSFS (<0.001 M), RS (<0.0001 M)	SVR/NASA
	BFS (0.05 N), GSS (0.005 S), SSFS (0.006 L), RS (0.006 L)	Random Forest/COCOMO
	BFS (0.4 S), GSS (0.03 S), SSFS (<0.001 M), RS (<0.001 M)	RF/Nasa
	SSFS (0.1 S), RS (0.1 S)	SVR/Albrecht
	BFS (0.02 L), SSFS (0.03 L), RS (0.02 L)	RF/Maxwell
	BFS (0.1 S), GSS (0.1 S), SSFS (0.1 S), RS (0.1 S)	RF/Albrecht
FA	SSFS (0.05 S), RA (0.05 S)	Random Forest/Maxwell
	BFS (0.7 L), GSS (0.03 L), SSFS (<0.001 L), RS (<0.001 L)	Linear Regression/China
	SSFS (0.1 S), RS (0.6 S),	MLP/China
	BFS (0.03 M), GSS (0.03 M), SSFS (0.05 S), RS (0.05 S)	SVR/Nasa
	BFS (0.01 M),	MLP/Nasa
	BFS (0.1 L), GSS (0.1 L), SSFS (0.1 L), RS (0.1 L)	Random Forest/Albrecht
GA	BFS (<0.001 L), GSS (<0.001 L), SSFS (<0.001 L), RS (<0.001 L)	SVR/China
	BFS (0.003 M), SSFS (<0.001 M), GSS (0.01 M), RS (<0.001 M)	Random Forest/China
	BFS (0.04 N), SSFS (0.04 N), GSS (0.05 N), RS (0.04 N)	Random Forest/Maxwell
	BFS (0.03 M), GSS (0.03 M), RS (0.03 M)	M5P/NASA
	BFS (0.03 L), GSS (0.03 L), SSFS (0.05 M), RS (0.05 M)	SVR/Nasa
	GSS (0.3 M), SSFS (0.4 M), RS (0.4 M)	Linear Regression/Maxwell
	BFS (0.004 S), GSS (0.3 S), SSFS (0.3 S), RS (0.008 M)	Linear Regression/Nasa
	SSFS (0.05 M), RS (0.05 M)	MLP/China

TS has also allowed obtaining better effort predictions (in terms of MAE) with respect to non-bio-inspired feature selection algorithms in 22 different cases in particular when using SVR/China, MLP/China, Random Forest/Maxwell, Random Forest/China, SVR/COCOMO, Linear Regression/Maxwell, and M5P/Maxwell as settings. In eight different cases, we have found a statistically significant difference in absolute residuals, which are: Random Forest /Maxwell against BFS and RS, Random Forest/China against BFS, GSS, SSFS, and RS, and Linear Regression/Maxwell against SSFS and RS. In 10 different experiments, there is either small, medium, or large effect size while in 12 different experiments, though MAE values obtained are lower, there is no effect size (negligible) observed.

As for PSO, it has outperformed non-bio-inspired feature selection algorithms in 16 cases (in terms of MAE) and for 4 comparisons we have found a statistically significant difference in the absolute residuals. The experiments where it has performed significantly better is with Linear Regression/Nasa, against BFS, GSS, SSFS, and RS. In 6 experiments, even if MAE values obtained are lower, there is no effect size while in the other 10 experiments, there are either small, medium, or large effect sizes.

Similarly, FA has performed better (in terms of MAE) in 33 different experiments when compared with non-bio-inspired algorithms, while a statistically significant difference is found in 10 cases with the combination Random Forest/Maxwell (against SSFS and RS), Linear Regression/China (against GSS, SSFS, and RS), SVR/Nasa (against BFS, GSS, SSFS and RS), and MLP/Nasa (against BFS). A notable effect size (i.e., small, medium, or large), in this case, is recorded in 17 different experiments.

GA has performed better (in terms of MAE) than non-bio-inspired algorithms in 35 cases and in 23 comparisons we have found a statistically significant difference in the absolute residuals, in particular SVR/China (against BFS, GSS, SSFS, and RS), Random Forest/China (against BFS, GSS, SSFS, and RS), Random Forest/Maxwell (against BFS, GSS, SSFS, and RS), M5P/Nasa (against BFS, GSS, and RS), SVR/Nasa (against BFS, GSS, SSFS, and RS), Linear Regression/Nasa (against BFS and RS), and

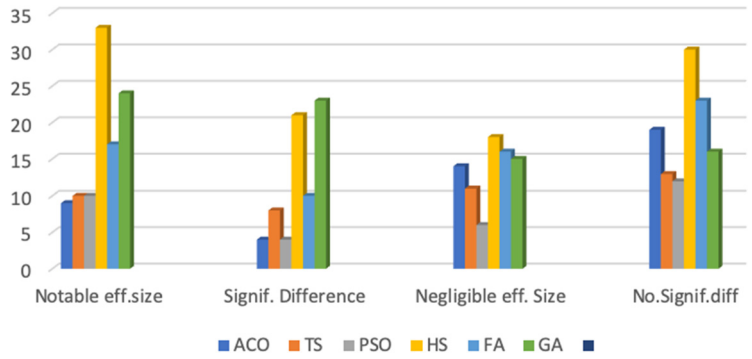

Fig. 4. Number of experiments for which bio-inspired algorithms performed *better* than non-bio-inspired algorithms.

MLP/China (against SSFS and RS). Out of 35 cases, there is an effect size (small, medium, or large) in 24 experiments while in 11 experiments there is no effect size.

HS results to be the algorithm that has outperformed non-bio-inspired feature selection algorithms in more comparisons (in terms of MAE), namely 51 involving the following settings (Linear Regression/Maxwell, SVR/COCOMO, SVR/Maxwell, MLP/China, M5P/Maxwell, Random Forest/China, Linear Regression/Nasa, SVR/Nasa, Linear Regression/COCOMO, Random Forest/COCOMO, RF/Nasa, SVR/Albrecht, Linear Regression/Miyazaki, Random Forest/Maxwell and Random Forest/Albrecht). The experiments where HS has significantly outperformed the non-bio-inspired algorithms are MLP/China (against BFS, GSS, SSFS, and RS), Linear Regression Nasa (against BFS, GSS, SSFS, and RS), SVR/Nasa (against GSS, SSFS, and RS), Random Forest/COCOMO (against BFS, GSS, SSFS, and RS), Random Forest/Nasa (against GSS, SSFS, and RS), and Linear Regression/Maxwell (against RS). Furthermore, we have found a statistically significant difference in the absolute residuals of 21 different experiments (i.e., Wilcoxon test p-values less than 0.05). HS is the type of bio-inspired algorithm which has provided better results compared to the other, however, only in 33 out of 51 experiments, we have noted an effect size of either small, medium, or large.

To sum up and provide an answer to RQ2, we can observe that in 36% of the total experiments where the bio-inspired algorithms performed better than the non-bio-inspired algorithms, the performance of the bio-inspired algorithms are significantly better, among which 66% of these significantly better performances are achieved only via GA and HS. Hence, we can conclude that GA and HS are the kind of bio-inspired algorithms that can significantly outperform the non-bio-inspired algorithms for a variety of datasets and estimation techniques when employed as feature selection algorithms in the domain of SDEE. A small, medium or large effect size has been recorded for all where the bio-inspired algorithms performed better (56% of them regards the use of GA and HS).

In this regard, the performance of ACO and PSO is not very satisfactory as they combinedly significantly outperformed the non-bio-inspired algorithms only in 12% experiments. Though FA and TS have also provided better results in various experiments, we do not hesitate to recommend only GA and HS as bio-inspired feature selection algorithms in the domain of SDEE to achieve better prediction accuracy.

If we would have just employed GA and HS for the comparison with the non-bio-inspired algorithms, the percentage of better performance of bio-inspired algorithms over non-bio-inspired would be much better. In a nutshell, GA and HS can be highlighted as the bio-inspired feature selection algorithms which are figured-out as the best performed, when we compare them to the non-bio-inspired algorithms.

As GA and HS performed better than the other bio-inspired algorithms, in Fig. 5 we show for each dataset the number of experiments for which GA and HS performed better than non-bio-inspired algorithms. As the figure depicts, both GA and HS provided better performance in a greater number of experiments in the cases of big datasets (i.e., China, Maxwell, and Nasa), while in the cases of small (e.g., Finnish, Kemerer, and Miyazaki) they did not perform better.

We have also some experiments where the non-bio-inspired features selection algorithms have performed better than the bio-inspired algorithms. In this case, the non-bio-inspired algorithm which provides significantly better results compared to the other is BFS, which has outperformed the bio-inspired algorithms in eight different settings (estimation techniques and datasets). However, in four of these eight settings, the dataset employed is Miyazaki which is the small size dataset and bio-inspired algorithms have never performed well with this dataset. BFS has performed better for the following settings: SVR/NASA, SVR/Miyazaki, MLP/NASA, Random Forest/NASA, Linear Regression/Miyazaki, SVR/Maxwell, M5P/Miyazaki, and Random Forest/Miyazaki. However, we have found a statistically significant difference in the experiments of SVR/NASA (against ACO, PSO, FS) and M5P/Miyazaki (against ACO, PSO, FS, GA, and HS).

The other better performed non-bio-inspired algorithms are SSFS (performed better in five settings) and GSS and RS (four settings each). When compared with bio-inspired algorithms, SSFS has performed better in the experiments SVR/Miyazaki (against ACO, PSO, FS, HS, TS), Random Forest/China (against ACO, PSO, FS, TS, GA), Random Forest/Miyazaki (against FS, PSO, GA, ACO, HS, TS), SVR/NASA (against ACO, PSO, TS, GA, FS), and Linear Regression/Miyazaki (against ACO, PSO, FS, HS, TS). With MLP/Maxwell and SVR/Maxwell, it has only outperformed HS. With Random Forest/China, SVR/Nasa, and Linear Regression/Miyazaki, SSFS has significantly outperformed the bio-inspired algorithms. GSS performed better in the

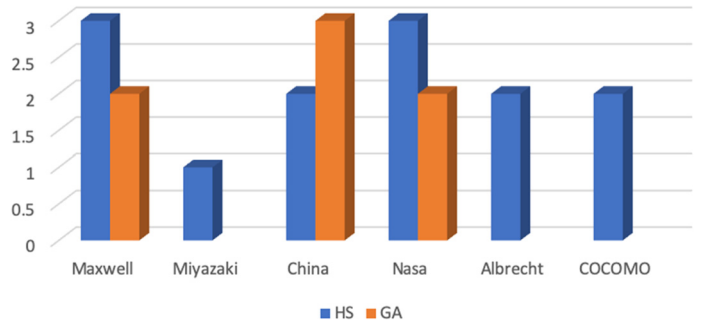

Fig. 5. Performance of GA/HS with different datasets vs non-bio-inspired algorithms.

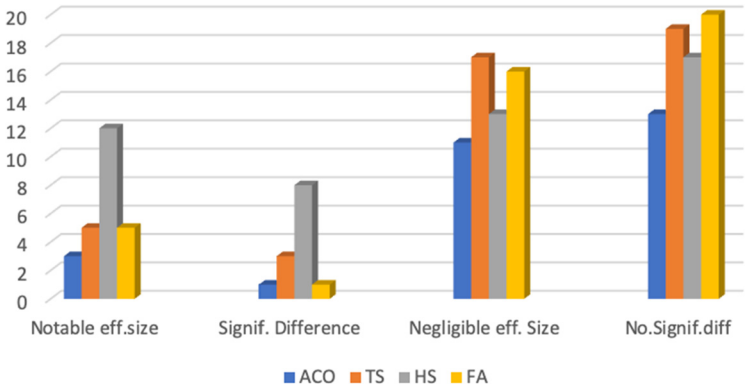

Fig. 6. Experiments where recently-employed outperformed earlier-employed bio-inspired algorithms.

experiments SVR/China (against FS, PSO, HS, GA), MLP/Maxwell (against HS), MLP/China (against ACO, PSO, FS, TS), and M5P/Maxwell (against GA) but for none of them, we have found a statistically significant difference in the absolute residuals.

Finally, RS is characterized by better performances in the case of SVR/Miyazaki (against ACO, PSO, FS, HS, TS), Random Forest/China (against ACO, PSO, FS, TS, GA), Random Forest/Miyazaki (against FS, PSO, GA, ACO, HS, TS), and SVR/Nasa (against ACO, PSO, FS, HS, TS). However, only with Random Forest/China and SVR/Nasa we have found a statistically significant difference in the absolute residuals.

4.3. RQ3: Do the recently employed bio-inspired algorithms (i.e., ACO, TS, HS, and FS) allow to obtain effort predictions better than those achieved with bio-inspired algorithms employed earlier (i.e., GA and PSO)?

In our recent Systematic Literature Reviews (SLRs) about bio-inspired algorithms in Software Development Effort Estimation (SDEE) [28], we found that GA and PSO are the two bio-inspired (feature selection) algorithms which are employed in the earlier (goes back to late 2000) and hence we found a greater amount of studies employed them. On the other hand, the bio-inspired algorithms such as ACO, FS, HS, and TS are observed to be investigated in the studies published recently. To this aim, to address our RQ3, we have considered the earlier employed bio-inspired algorithms (GA and PSO) as a baseline and compared them with other four recently-employed algorithms (ACO, FS, HS, and TS) in different cases (i.e., the combination of different datasets and estimation techniques).

Table 5 shows the settings (combinations of datasets and estimation techniques) for which the recently employed bioinspired feature selection algorithms have provided significantly better accuracy than the baselines GA and PSO or with a notable effect size (i.e., small, medium, or large). In particular, the first column (Better) shows the recently-employed bio-inspired algorithms, the middle column (Worse) indicates the baseline (GA and/or PSO), while the third column (Experimental setting) reports the employed dataset and estimation technique. Moreover, to provide a visual summary that allows to easily and quickly understand the achieved results, Fig. 6 shows the number of experiments for which recently employed bio-inspired algorithms provide better predictions than baselines PSO and GA, considering when the differences are significantly (i.e., Signif. Difference), not significantly (i.e., No Signif.diff), and with notable effect size or negligible.

The analysis of results in terms of MAE reveals that ACO performed better than both the baseline algorithms in 14 individual experiments (with the settings: SVR/Albrecht, SVR/COCOMO, MLP/COCOMO, and Random Forest/COCOMO. However, we have only one individual experiment where ACO performed significantly better than PSO (with the Random Forest/Maxwell combination) and 3 cases with at least a notable effect size (see Table 5). Similarly, TS has provided better results in terms of MAE in 22 individual experiments than the baselines (both when employed with SVR/COCOMO, MLP/China, MLP/COCOMO, MLP/Maxwell, MLP/Nasa, Random Forest/COCOMO, Random Forest/Maxwell, Random Forest/Nasa,

Table 5
Recently employed Bio-inspired algorithms that performed (significantly) better (Better) w.r.t. GA and PSO (Worse), with the p-value of the performed statistical test and effect size between brackets.

Better	Worse	Experimental Setting
ACO	PSO (0.2 S)	SVR/Nasa
	GA (0.08 L),	MLP/COCOMO
	PSO (0.03 M)	RF/Maxwell
TS	PSO (0.7 S)	SVR/China
	GA (0.05 S),	MLP/China
	PSO (0.04 L)	MLP/Maxwell
	PSO (0.05 M)	RF/China
	GA (0.06 L),	RF/Maxwell
HS	GA (0.3 L)	Linear Regression/Maxwell
	GA (<0.001 L), PSO (0.01 M)	SVR/Nasa
	GA (0.1 L), PSO (0.05 L)	Linear Regression/COCOMO
	PSO (0.4 L)	Linear Regression/Nasa
	PSO (<0.001 L)	MLP/China
	PSO (<0.001 L)	RF/China
	GA (0.001 M), PSO (0.03 S)	RF/Nasa
	PSO (0.05 S)	M5P/China
	PSO (0.4 M)	M5P/Maxwell
FS	PSO (0.6 S)	Linear Regression/Maxwell
	GA (0.08 M), PSO (0.4 S)	MLP/COCOMO
	GA (0.04 S),	RF/Nasa
	GA (0.1 M)	M5P/Maxwell

and M5P/China). However, TS has provided significantly better results than the baselines only in 3 individual experiments and with a notable effect size in 5 experiments (see Table 5). HS, on the other hand, has provided better results in terms of MAE in 25 individual experiments than the baselines (both when employed with SVR/Albrecht, SVR/COCOMO, SVR/Nasa, Linear Regression/COCOMO, Linear Regression/Miyazaki, Random Forest/Albrecht, Random Forest/Maxwell, and Random Forest/Nasa). However, HS has significantly outperformed the baselines in 8 different experiments and with a notable effect size in 12 experiments (see Table 5). About FS, it has provided better results in terms of MAE in 21 individual experiments than baselines (both when used with SVR/Albrecht, SVR/COCOMO, MLP/COCOMO, MLP/Maxwell, MLP/Nasa, Random Forest/Albrecht, Random Forest/COCOMO, and Random Forest/Nasa). In only one experiment, FA has performed significantly better than the baseline algorithms and 5 experiments with a notable effect size.

Focusing on the algorithms that allowed to outperform both baselines, the better performed bio-inspired algorithms are TS (9 times), HS (8 times), and FA (8 times) but we highlight only HS as a better algorithm because it has significantly outperformed the baselines in more number of individual experiments (i.e., 8). Maxwell and China are the datasets for which HS performed better (4 of 5 times each), while Linear Regression is the estimation technique that allows HS to perform better (5 out of 8 times). SVR and Random Forest are the second-best estimation techniques employed with HS. In the majority of experiments where the recently-employed algorithms have outperformed the baselines, the effect size is negligible. However, there are also experiments where we can observe the dominance of recently-employed algorithms with a significant difference/notable effect size.

For completeness, we have also analyzed the cases (combinations of datasets and estimation techniques) for which the baselines GA and PSO have provided significantly better accuracy than recently employed bio-inspired feature selection algorithms, or with a notable effect size (see Table 6). Moreover, to provide a visual summary that allows to easily and quickly understand the achieved results, Fig. 7 shows the number of experiments for which the baselines PSO and GA provide better predictions than the recently employed bio-inspired algorithms, considering when the differences are significantly (i.e., Signif. Difference), not significantly (i.e., No Signif.diff), and with notable effect size or negligible effect size. In this case, PSO has provided better results in 9 different experiments with notable effect size, however, only in 3 cases, it has significantly outperformed the recently-employed algorithms. GA is the baseline algorithm that has provided better results than almost all of the recent bio-inspired algorithms, not only outperforming them in a greater number of experiments but also providing significantly better results in the majority of the cases. In particular, GA has outperformed in terms of MAE the recently-employed algorithms in about 58 individual experiments. Among them, GA provided better results with a significant difference in 17 experiments and with a notable effect size in 20 different experiments (see Table 6). Furthermore, there are about 15 different settings (datasets/estimation techniques) for which it has provided better results than more than two recently-employed bio-inspired algorithms.

To sum up and provide an answer to RQ3, we can state that HS and GA have outperformed the rest of the bio-inspired algorithms when investigated with a greater number of combinations of datasets and estimation techniques. There are also a few cases where we can observe comparable results, particularly when employed with the datasets Finnish, Kemerer, and Miyazaki. In a nutshell, since GA has performed better than all the other bio-inspired algorithms (recently-employed and baseline) in terms of not only providing better accuracy in a greater number of experiments but also significantly

Table 6
Cases where baseline bio-inspired algorithms performed better.

Better	Worst	Experimental Settings
PSO	ACO (0.8 S), FA (0.4 S), HS (0.05 L)	SVR/China
	ACO (0.05 M), HS (0.6 N)	MLP/Maxwell
	HS (0.03 S)	MLP/Nasa
	ACO (0.1 L)	RF/Nasa
	TS (0.5 S)	M5P/Maxwell
	HS (0.2 S)	M5P/Nasa
GA	HS (<0.001 S)	MLP/China
	HS (0.9 N), TS (0.5 N), FA (0.9 N), ACO (0.2 S)	Linear Regression/China
	ACO (0.7 S), TS (0.005 M), FA (1.0 N), HS (0.004 M)	M5P/Nasa
	FA (0.03 M)	SVR/China
	ACO (0.03 L), FA (0.6 S), HS (<0.001 L),	M5P/China
	ACO (0.001 S), FA (0.008 L), HS (<0.001 L)	RF/China
	HS (0.2 S)	MLP/COCOMO
	ACO (0.05 L), HS (0.6 L)	MLP/Maxwell
	HS (0.02 M)	RF/COCOMO
	ACO (0.1 M), FA (0.03 M), HS (0.1 M), TS (0.03 M)	M5P/COCOMO

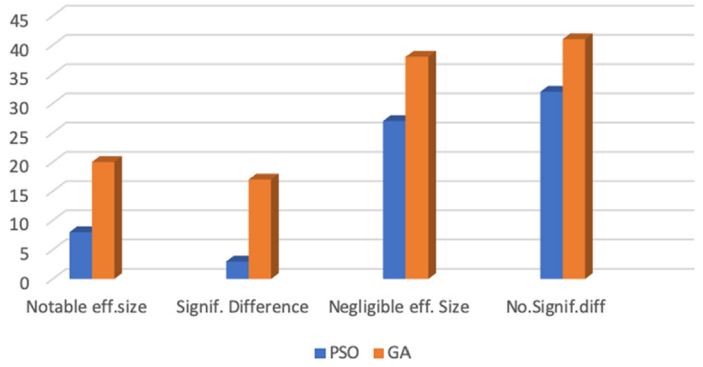

Fig. 7. Experiments where earlier-employed outperformed recently-employed bio-inspired algorithms.

outperforming the recently-employed algorithms in the majority of experiments, ***we cannot positively answer RQ3***. We can only highlight that the recently-employed bio-inspired algorithms have provided better performance than just PSO. Furthermore, HS is the only type of recently-employed algorithms that not only has outperformed PSO but also provided results comparable to those achieved with GA.

4.4. Main findings and suggestions

In about 90% of the all the experiments we considered, the performances of all feature selection (bio-inspired and non-bio-inspired) algorithms are comparable when employed with the datasets having a few numbers of features, i.e., Kemerer (8 features), Albrecht (8 features), Finnish (9 features), and Miyazaki (9 features). For instance, for the Finnish and Albrecht datasets, we have observed that all the feature selection algorithms selected the same set of features and the accuracy measure values achieved employing bio-inspired and non-bio-inspired feature selection algorithms are comparable. Furthermore, when considering the Kemerer dataset and the use of M5P, Random Forest, Linear Regression, and MLP as estimation techniques, the baseline estimation techniques have performed better than the all the bio-inspired and non-bio-inspired algorithms. For the Miyazaki dataset, the effort predictions of bio-inspired algorithms are not satisfactory. When employed with Linear Regression and M5P, the effort predictions of non-bio-inspired algorithms are better than the bio-inspired algorithms, except the GSS which has the worst performance. However, the effort predictions of bio-inspired algorithms with M5P and Miyazaki combinations are (not significantly) better than the baseline estimation techniques. Similarly, when employed with the MLP technique, the baseline estimation techniques have significantly provided better results than the feature selection algorithms. In a nutshell, it seems that the effect of feature selection algorithms (especially bio-inspired algorithms) becomes less effective when the features in the datasets are small in numbers.

For the remaining four datasets (NASA, Maxwell, COCOMO, and China), which are larger in terms of the number of features, we had a mix of results for the various estimation techniques and feature selection algorithms. In the following, we focus on the datasets and employed estimation techniques.

With dataset China, the results of bio-inspired algorithms are better in the majority of the cases, particularly when employed with the MLP, M5P, and Random Forest estimation techniques. The best performed bio-inspired algorithms, in this case, are TS and GA which have outperformed the non-bio-inspired algorithms in the majority of cases. GA has also

performed better with China dataset when comparing the baseline bio-inspired algorithms with the recently-employed algorithms.

ACO and FS, in this case, are the type of bio-inspired algorithms which has provided better results in less number of experiments when investigated with the China datasets.

As for the COCOMO dataset, we have found that HS has provided a better prediction accuracy with respect to non-bio-inspired algorithms when used in combination with SVR, Linear Regression, and Random Forest). HS has also provided better results with COCOMO as a recently-employed bio-inspired algorithm when compared with the baseline bio-inspired algorithms. When employed with the MLP, the baseline estimation techniques have provided better results with COCOMO datasets compared to all the feature selection algorithms.

With the dataset Maxwell, almost all the bio-inspired feature selection algorithms have performed better than non-bio-inspired algorithms as well as the baseline estimation techniques. In some of the cases, especially when used with the MLP, non-bio-inspired algorithms have provided better results with the Maxwell dataset.

For the Nasa dataset when employed with the SVR estimation technique, the performance of the non-bio-inspired algorithms is better than the bio-inspired algorithms; however, for the other estimation techniques, the bio-inspired algorithms have outperformed the non-bio-inspired algorithms. HS is the leading bio-inspired algorithm, having better results with the Nasa dataset.

From the analysis of the results achieved with the different datasets, we have also learned that it is not always true that feature selection algorithms removing more features than other algorithms provide better accuracy. Indeed, algorithms like SSFS and HS excluded more features of some datasets compared to others but still, their performance in those experiments was not recorded as best. We have the confirmation that the accuracy of the obtained predictions increases when irrelevant features are excluded rather than a greater number of features are removed.

In particular, we have found that MLP is the technique that has provided better predictions alone in more experiments. As we mentioned above, also our SLR results have revealed that ANN (which can be considered a general form of MLP) is the estimation technique that has provided better predictions alone in more experiments with respect to other estimation techniques. From our SLR results, we have also obtained an indication that SVR alone (i.e., without any feature selection algorithm) has outperformed two bio-inspired algorithms (PSO and TS) as well as CART against PSO. So, we encourage to further investigate these estimation techniques in the future as well as new ones, in combinations with various bio-inspired (and non-bio-inspired) algorithms.

To summarize and provide some suggestions to other researchers, we could not figure out any (bio- and non-bio-inspired) feature selection algorithms which resulted to be the best in all cases. Different algorithms have achieved better results with different datasets and estimation techniques and for other combinations have provided worse effort predictions. However, if we have to select one feature selection algorithm (from both bio- and non-bio-inspired), then HS can be considered the one that has provided better effort predictions in more combinations of datasets and estimation techniques with respect to the other bio- and non-bio-inspired algorithms. It is worth noting that from our SLR about the bio-inspired feature selection algorithms employed for effort estimation, HS results to be investigated in only 7% of the selected studies. This means that researchers have not focused a lot in the past on HS in the domain of SDEE. Therefore, we suggest other researchers to evaluate (further) the impact of HS in SDEE, with different combinations of datasets and estimation techniques. Similarly, the prediction accuracy of GA resulted to be not very impressive in our SLR results when compared with other bio-inspired algorithms, while in our empirical study GA has a satisfactory performance resulting to be a better option than other (bio and non-bio-inspired) feature selection algorithms. Thus, we also recommend employing GA further in the future and assess its support for feature selection in the domain of effort estimation.

We want also to highlight that we have employed FA, never used in the past, and it did not provide good effort predictions. Similarly, PSO has provided good effort predictions in a few cases in our analysis. This seems to confirm the findings of our SLR results, where PSO performed worse than other bio-inspired feature selection algorithms in the majority of the cases. So, we think we cannot suggest both FA and PSO for future investigations in the context of effort estimation, particularly when there is a variety of other bio-inspired algorithms available. In summary, from our results, we can recommend the use of HS, GA, and TS from the list of bio-inspired algorithms and BFS from the list of non-bio-inspired algorithms.

Furthermore, as future investigation, we would also suggest to other researchers to consider some other bio-inspired algorithms which are yet to be investigated in the domain of effort estimation, e.g., Bat Search, Cuckoo Search, Elephant Search, Wolf Search as well as some non-bio-inspired algorithms, e.g., Grey Relational Analysis, Gain Information, Mutual Information, Exhaustive Search. Combinations or hybrid approaches should also be investigated to exploit and assess the balance achieved by applying algorithms having different characteristics.

As for the selected features by the different algorithms, in each of the employed datasets, there are just few features that contribute to the prediction accuracy of an estimation technique. For instance, Gao et al. [120] state that accuracy can be increased with only 10% features subset. Hence, if we claim one bio-inspired algorithm outperforms the other, it means that the algorithm has extracted the relevant subset of features which allows the estimation technique to produce accurate predictions. There are only a few parameters (i.e., population size and the number of iterations) of the employed bio-inspired algorithms that are common to all the algorithms while the other parameters or heuristics are specific to each bio-inspired algorithm. If GA and HS provide better results than let's say ACO and PSO, then we can observe that the specific heuristics of GA and HS are more capable to extract the relevant features from a particular dataset. For instance, the mutation and crossover operators of GA are responsible to examine a feature in a search space and decide if it needs

to be selected or discarded. The same is the case of HS, where the heuristic Harmony Memory Considering Rate succeeds in choosing the best features of the dataset. On the other hand, the parameters of ACO (i.e., pheromone trail and heuristic rate) and PSO (i.e., inertia weight) that are responsible for searching the features in a search space did not manage to select the relevant features.

To conclude this discussion, we want to highlight that we have also analyzed the stability of feature selection algorithms. Let us recall that if a feature selection algorithm is insensitive to the small changes in the data and the settings of algorithmic parameters, it can be called a stable (feature selection) algorithm [121]. The stability of a feature selection algorithm can be defined as low sensitivity to small perturbations in the training data [122]. The change in training data can be the random excluding of some instances/observations from the datasets. In the following, we discuss the stability of the bio-inspired algorithms employed, by excluding the outliers from the datasets. We have discarded all the observations from a dataset that is after the third inter-quartile range. We have intentionally preferred to exclude the observations rather than replacing them with the mean/median values, to observe a significant change in the training data. The procedure works like a feature selection algorithm (i.e., GA) selecting a set of features from a dataset before the changes are made (original full dataset) and then the same feature selection algorithm selects the set of features after the changes in the training data (i.e., when outliers are excluded), and then the two sets of features are evaluated to identify which features are common in both sets, i.e., selected by the feature selection algorithm before as well as after the changes are made in the training data. Since this is not the main goal of the paper, we just report the stability ratio of the four better-performed algorithms (HS, GA, PSO, and TS) for the four large size datasets (China, COCOMO, Maxwell, and Nasa). The number of features in the considered SDEE datasets are not too large, thus we have decided to use the Jaccard stability index, which is a set-based stability metrics [122]

$$\text{Stab}(A, B) = |A \cap B|/|A \cup B| * 100 \tag{1}$$

where A and B are the two sets of features that are selected by a feature selection algorithm (i.e., GA) before and after changes to the training data. $A \cap B$ means the common features in the two sets and AUB means the features selected by either of the two sets (before and after to the changes). For instance, if GA selects the following features of the China dataset before changes in the training data.

$$\text{GA (before)} = \{\text{ID, AFP, Output, Enquiry, File, Interface}\}$$

and after changes in the data, select the following set of features:

$$\text{GA (after)} = \{\text{ID, Input, Enquiry, File}\}$$

then

$$\text{Stab (GA)} = |\text{GA (before)} \cap \text{GA (after)}|/|\text{GA (before)} \cup \text{GA (after}| * 100 = 3/7 * 100 = 42\%.$$

Table 7 shows the (average) stability index of different feature selection algorithms for the four large size datasets according to the equation (1). In particular, the first column of Table 7 indicates the settings of the experiments, the second column shows the stability index for each setting (i.e., feature selection algorithm and dataset) and the third column indicates the average/mean stability index of each feature selection algorithm for all the four datasets. In this case, GA, which is also one of the leading performed bio-inspired algorithms in our experiments, results to be the most stable algorithm with an average stability index of 68.25%, followed by the TS (65.75%) and PSO (64.75%). On the other hand, HS, which is also one of the better-performed algorithms, is observed as the least stable bio-inspired algorithm, having an average stability ratio of just 40.75%.

Moreover, from Table 7, we can also observe that the bio-inspired algorithms are more stable with Nasa dataset, with an average stability index of 78.5%, followed by the Maxwell dataset (61.75%), COCOMO dataset (59%), and China dataset (40%).

5. Threats to validity

In the following, we discuss the factors that can bias the validity of the performed empirical analysis by considering three types of validity threats: Construct validity, Conclusion validity, and External validity.

As suggested in [123], to satisfy construct validity a study has “to establish correct operational measures for the concepts being studied”. Thus, the choice of the features and how to collect them represents the crucial aspects. In our analysis, we have mitigated this possible threat by evaluating the employed estimation techniques and feature selection algorithms on publicly available datasets. Furthermore, as highlighted by our SLR, the selected datasets have been previously used in many empirical studies carried out in the past to evaluate software development effort estimation approaches. To this end, we read and evaluate all the studies which described the original data (and not just rely on the information available in public repositories) as it might cause misleading. The detail about it is mentioned in the Datasets section.

Regarding the conclusion validity, we carefully applied the performed statistical tests. Moreover, the observations in the SDEE datasets are no so high so it could lead to the threat to conclusion validity. Since, in our analysis we did not

Table 7
Stability index of different feature selection algorithms with different datasets.

Settings	Stability index for each dataset	Average stability index
HS/China	25%	**40.75**
HS/Maxwell	50%	
HS/Nasa	44%	
HS/COCOMO	44%	
GA/China	60%	**68.25**
GA/Maxwell	57%	
GA/Nasa	90%	
GA/COCOMO	66%	
PSO/China	44%	**64.75**
PSO/Maxwell	68%	
PSO/Nasa	85%	
PSO/COCOMO	63%	
TS/China	33%	**65.75**
TS/Maxwell	72%	
TS/Nasa	95%	
TS/COCOMO	63%	

manage to reject some null hypothesis which could be considered as the low power of the statistical tests as the number of observations are not too high. However, for most of the software effort estimation studies in the literature, the results are obtained using small sized datasets (e.g., [29]). However, to mitigate this threat, apart from the datasets having low observations, we also employed datasets like China (499 observations) which have a relatively high number of observations. We want also to observe that in our analysis we have used the default parameter settings of the bio-inspired algorithms (i.e., population size etc.) provided by the Weka tool. Though the use of default settings by Weka has been investigated in previous studies (e.g., [89,90]), we are aware that there might be some experiments where our findings might be different if user-defined parameter settings (i.e., applying some strategies to select own parameter values [124,125]) are used, which could be considered as a threat to the conclusion validity. However, as suggested by the no-free-lunch theorem, there are no parameter settings, which will provide better results in all the situation. Furthermore, we want to clarify that the motivation of sticking with the default parameter settings is that we just wanted to let our readers know which bio-inspired algorithms (and non-bio-inspired algorithms) provided better prediction accuracy with which datasets and estimation techniques, rather than to focus in general on improving the prediction accuracy. Despite this motivation, to just give our readers an idea of how the results will change if we use different parameter settings, we report here the results we achieved by running some experiments with different values of the parameters population size and the number of iterations [124,125]. For instance, we have used 50, 100, and 200 as values for the population size and iteration numbers as well as larger numbers such as 500, 1000, and 10,000 only for population size, and employed LR, SVR, and RF as estimation techniques because they resulted to be the three best-performed estimation techniques in our analysis. We have compared the obtained predictions with those achieved with the default settings employed in Section 3.3 (i.e., population size and number of iterations equal to 20). In Figs. 8, 9, and 10 we have reported for each setting (i.e., default, 50, 100, 200, 500, 1000, and 10000) and each bio-inspired algorithms (i.e., GA, HS, ACO, PSO, TS, FF) the number of experiments where the specific combination provided better predictions, when using LR, SVR, and RF, respectively, as estimation technique. The figures do not show the cases where the settings provided the same results (e.g., for LR apart from the cases shown in Fig. 8, there are other 12 experiments where bio-inspired algorithms achieved the same predictions for all the parameter setting values employed). We can observe that larger population size (and number of iterations) did not add anything useful apart from the extra computational time with respect to application of default settings. For example, in the case of LR, larger population size values (i.e. 500, 1000, and 10000) have provided results similar to lower population size values (i.e., 200), so increasing the value of population size seems to not help. One of the reasons is that the number of features in SDEE datasets is not very large (i.e., less than 50), so increasing these parameter values from a specific range does not provide a positive impact. Furthermore, similar to the default settings (see Section 4.1), we can conclude that no specific parameter settings has performed better in all the cases and, hence, the parameter settings should be configured based on individual requirements (i.e., according to the employed data and other algorithms). The above observations can also be provided for the results achieved with SVR and RF. In the case of RF, we can note that the larger population size values (i.e. 500, 1000, and 10000) have even provided results worse than lower population size values (i.e., 200).

Regarding the external validity, the project data considered in our analysis are from single company datasets (i.e., a dataset is obtained collecting information from a single company) as well as from cross-company datasets (i.e., a dataset is obtained collecting project data from different companies). In particular, we have employed eight different freely and publicly available datasets, widely used in several SDEE empirical studies (e.g., [19,32,14,15,74]). However, the findings of our study might be different when employed with other datasets, for example, ISBSG which is considered as the largest dataset in the domain of SDEE. We want to clarify that we did not employ ISBSG dataset because it is not freely available as those in the PROMISE repository [51]. It is our opinion that ISBSG has been ignored in several other comprehensive

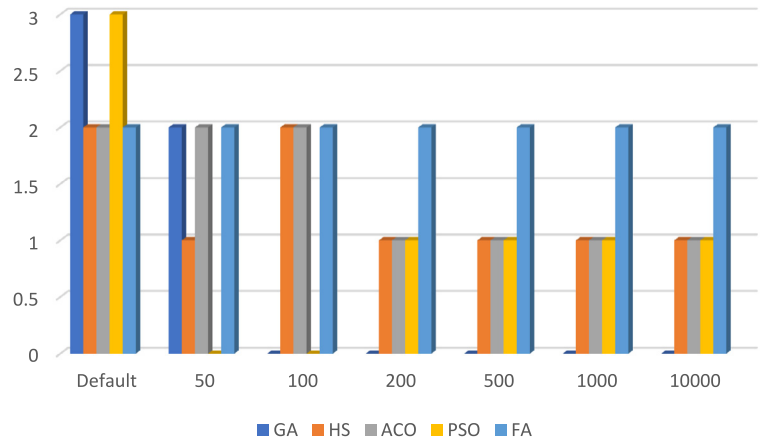

Fig. 8. Number of experiments for which the considered bio-inspired algorithms provided better predictions with different parameter settings, using LR as the estimation technique.

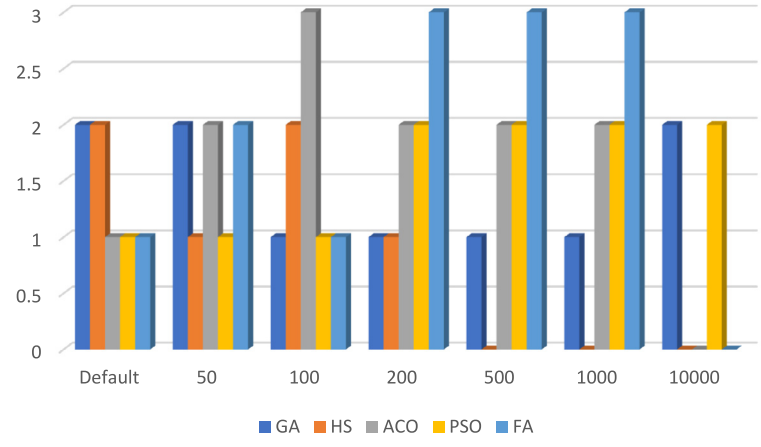

Fig. 9. Number of experiments for which the considered bio-inspired algorithms provided better predictions with different parameter settings, using SVR as the estimation technique.

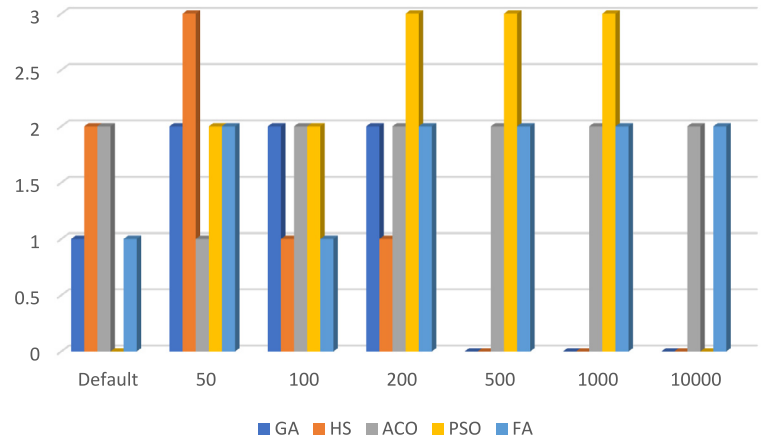

Fig. 10. Number of experiments for which the considered bio-inspired algorithms provided better predictions with different parameter settings, using RF as an estimation technique.

empirical studies (which have employed more than five datasets) published in relevant venues (e.g., [14,15,74]) because it is not freely available. However, despite the above reasons, we are aware that the findings of our study might not persist if we use ISBSG dataset and, hence, it could be considered as a threat to external validity. Thus, to mitigate this threat we could investigate the impact of various bio-inspired feature selection algorithms with the ISBSG dataset in the future. In particular, it would be of more interest to evaluate if HS still perform better with ISBSG as it did with the other employed SDEE datasets.

6. Conclusion

In this paper, we have provided an empirical study to evaluate the impact of bio-inspired (and some non-bio-inspired) feature selection algorithms on the prediction accuracy, when used with a variety of SDEE datasets. Almost all the bio-inspired algorithms and datasets selected for our study are inspired by the findings of an SLR we have recently performed to analyze the use of bio-inspired algorithms for feature selection in the context of effort estimation. In particular, we have evaluated the impact of bio-inspired algorithms against (a) the baseline estimation techniques (i.e., estimation techniques applied alone, without feature selection) and (b) non-bio-inspired feature selection algorithms. Furthermore, (c) we have compared earlier-introduced and recently-introduced bio-inspired algorithms for feature selection.

In our study, we have performed a variety of experiments considering six bio-inspired feature selection algorithms (GA, PSO, ACO, TS, HS, and FA) and four traditional non-bio-inspired algorithms (Best-First Search, Greedy Stepwise, Subset Forward Selection, and Random Search), used in combination with five widely used estimation techniques (MultiLayer Perceptron, Support Vector Regression, Random Forest, Linear Regression, and M5P algorithm) and applied to eight publicly available datasets widely used in the SDEE community (Albrecht, China, COCOMO, Finnish, Kemerer, Maxwell, Miyazaki, and NASA). And, we found that almost all the bio-inspired algorithms performed better with respect to the use baselines, i.e., estimation techniques applied alone without feature selection. In particular, HS and GA are the bio-inspired feature selection algorithms which provided significantly better effort predictions in the majority of the experiments when compared with the considered non-bio-inspired algorithms as well as with the other employed bio-inspired algorithms. BFS is one that provided better effort predictions among the used non-bio-inspired algorithms. However, we could not figure out any bio- and non-bio-inspired feature selection algorithms which resulted to be the best in all cases. Different algorithms have provided better effort estimations with different datasets and estimation techniques while for other combinations of datasets and estimation techniques have instead achieved worst effort predictions. Also, our analysis has highlighted that a feature selection algorithm can perform as the best only if it filters the irrelevant features of a dataset rather than a greater number of features.

To conclude, we believe that our study have highlighted that feature selection techniques can be an important tool for SQA team to improve the prediction accuracy of estimation techniques employed to predict software development effort. We also believe that the performed comprehensive analysis has allowed us to provide suggestions to researchers and SQA teams. In particular, a) the employed bio-inspired feature selection algorithms have allowed to obtain better effort predictions with respect to the considered non-bio-inspired algorithms in the majority of the experiments; b) HS and GA are the bio-inspired algorithms that have allowed to achieve better effort predictions with respect to the other employed bio-inspired algorithms in the majority of the experiments.

In the future, we intend to perform further analyses trying to increase the performance of various estimation techniques, not only excluding the irrelevant features but also removing the non-representative instances (i.e., applying a case selection). Similarly, since we used the default parameter settings of the bio-inspired algorithms in the future we could focus our attention on the assessment of strategies to select estimation technique parameter values. Furthermore, we intend to perform a similar type of empirical study in the context of software fault/bug predictions (SFP). We will first perform an SLR about the bio-inspired algorithms in the domain of fault predictions and then based on the results of the SLR, we will choose the type of algorithms to be investigated. Finally, by exploiting both the results achieved in the contexts of SDEE and SFP, it could be interesting to investigate the possibility of providing a sort of model/framework to apply feature extraction on different problems other than software cost estimation and software defect prediction.

CRediT authorship contribution statement

Asad Ali: Conceptualization, Formal analysis, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Carmine Gravino:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

References

[1] Robert N. Charette, Why software fails [software failure], IEEE Spectr. 42 (9) (2005) 42–49.

[2] E.I. Emam, A. Khaled, G. Koru, A replicated survey of IT software project failures, IEEE Softw. 5 (2008) 84–90.
[3] Barbara Kitchenham, O. Brereton, D. Budgen, Mark Turner, John Bailey, Systematic literature reviews in software engineering–a systematic literature review, Inf. Softw. Technol. 51 (1) (2009) 7–15.
[4] Miyoung Shin, Amrit L. Goel, Empirical data modeling in software engineering using radial basis functions, IEEE Trans. Softw. Eng. 26 (6) (2000) 567–576.
[5] Mahmoud O. Elish, Improved estimation of software project effort using multiple additive regression trees, Expert Syst. Appl. 36 (7) (2009) 10774–10778.
[6] Shashank Mouli Satapathy, Aditi Panda, Santanu Kumar Rath, Story point approach based agile software effort estimation using various SVR kernel methods, 2014.
[7] Petronio L. Braga, Adriano L.I. Oliveira, Gustavo H.T. Ribeiro, Silvio R.L. Meira, Bagging predictors for estimation of software project effort, in: International Joint Conference on Neural Networks, 2007, pp. 1595–1600.
[8] S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, Web effort estimation: function point analysis vs. COSMIC, Inf. Softw. Technol. 72 (2016) 90–109.
[9] L. Briand, J. Wüst, The impact of design properties on development cost in object-oriented systems, in: IEEE METRICS Symposium, 2001, pp. 260–271.
[10] L. De Marco, F. Ferrucci, C. Gravino, Approximate COSMIC size to early estimate Web application development effort, in: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, 2013, pp. 349–356.
[11] E. Mendes, N. Mosley, S. Counsell, Investigating Web size metrics for early Web cost estimation, J. Syst. Softw. 77 (2) (2005) 157–172.
[12] V. Bianco, L. Lavazza, G. Liu, S. Morasca, Model-based early and rapid estimation of COSMIC functional size - an experimental evaluation, Inf. Softw. Technol. 56 (10) (2014) 1253–1267.
[13] Zhihao Chen, Tim Menzies, Daniel Port, Barry W. Boehm, Feature subset selection can improve software cost estimation accuracy, ACM SIGSOFT Softw. Eng. Notes 30 (4) (2005) 1–6.
[14] A. Oliveira, P. Braga, Ricardo Massa Ferreira Lima, Márcio Cornéliob, GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation, Inf. Softw. Technol. 52 (11) (2010) 1155–1166.
[15] Q. Liu, J. Xiao, H. Zhu, Feature selection for software effort estimation with localized neighborhood mutual information, Clust. Comput. (2018).
[16] M. Hosni, A. Idri, Software development effort estimation using feature selection techniques, in: SoMeT, 2018, September, pp. 439–452.
[17] T. Menzies, D. Port, Z. Chen, J. Hihn, Specialization and extrapolation of software cost models, in: Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering, November 2005, pp. 384–387.
[18] V.R. Balasaraswathi, M. Sugumaran, Y. Hamid, Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms, J. Commun. Inform. Netw. 2 (4) (2017) 107–119.
[19] Zhihao Chen, Tim Menzies, Daniel Port, Barry W. Boehm, Feature subset selection can improve software cost estimation accuracy, ACM SIGSOFT Soft. Eng. Notes 30 (4) (2005) 1–6.
[20] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, Data Knowl. Eng. 17 (4) (2005) 491–502.
[21] Alan Jović, K. Brkić, N. Bogunović, A review of feature selection methods with applications, in: International Convention on Information and Communication Technology, Electronics and Microelectronics, 2015, pp. 1200–1205.
[22] Huang Yuan, Shian-Shyong Tseng, Wu Gangshan, Zhang Fuyan, A two-phase feature selection method using both filter and wrapper, in: International Conference on Systems, Man, and Cybernetics, Vol. 2, 1999, pp. 132–136.
[23] Z. Li, J. Zheng, X. Li, F. Wang, B. Ai, J. Qian, A genetic algorithm based wrapper feature selection method for classification of hyperspectral images using support vector machine, in: Joint Conference on GIS and Built Environment: Classification of Remote Sensing Images, Vol. 7147, 2008, 71471J.
[24] Chien-Pang Lee, Yungho Leu, A novel hybrid feature selection method for microarray data analysis, Appl. Soft Comput. 11 (1) (2011) 208–213.
[25] H. Osman, M. Ghafari, O. Nierstrasz, The impact of feature selection on predicting the number of bugs, preprint, arXiv:1807.04486, 2018.
[26] M. Sharma, P. Kaur, A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem, Arch. Comput. Methods Eng. (2020) 1–25.
[27] X.S. Yang, Nature-Inspired Algorithms and Applied Optimization, Vol. 744, Springer, 2017.
[28] A. Ali, C. Gravino, Using bio-inspired features selection algorithms in software effort estimation: a systematic literature review, in: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications, 2019.
[29] W. Ling, H. Ni, R. Yang, V. Pappu, M. Fenn, P. Pardalos, Feature selection based on meta-heuristics for biomedicine, Optim. Methods Softw. 29 (4) (2014) 703–719.
[30] A. Galinina, O. Burceva, S. Parshutin, The optimization of COCOMO model coefficients using genetic algorithm, Inf. Technol. Manag. Sci. (2012) 45–51.
[31] S. Sharma, A. Kaushik, Enhancement in software cost estimation using ant colony optimization, Int. J. Adv. Res. Comput. Sci. Soft. Eng. 6 (5) (2016).
[32] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, Estimating software development effort using tabu search, in: International Conference on Enterprise Information Systems, vol. 1, 2010, pp. 236–241.
[33] S.M. Sabbagh Jafari, F. Ziaaddini, Optimization of software cost estimation using harmony search algorithm, in: Conference on Swarm Intelligence and Evolutionary Computation, 2016, pp. 131–135.
[34] Amir Pourali, Amin Babazadeh Sangar, A new approach in software cost estimation with hybrid of imperialist competitive algorithm and ant colony algorithm, Bull. Séances Acad. R. Sci. O.-M. 4 (3) (2015) 106–113.
[35] M. Morera, C. Quesada-López, C. Castro-Herrera, M. Jenkins, A genetic algorithm based framework for software effort prediction, J. Soft. Eng. Res. Develop. 5 (1) (2017) 4.
[36] O. Adriano, P. Braga, R. Lima, M. Cornélio, GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation, Inf. Softw. Technol. 52 (11) (2010) 1155–1166.
[37] Pichai Jodpimai, P. Sophatsathit, C. Lursinsap, Ensemble effort estimation using selection and genetic algorithms, Int. J. Comput. Appl. Technol. 58 (1) (2018) 17–28.
[38] Hassani M. Saadi, V.K. Bardsiri, F. Ziaaddini, The application of meta-heuristic algorithms to improve the performance of software development effort estimation models, Int. J. Appl. Evolution. Comput. (IJAEC) 6 (4) (2015) 39–68.
[39] Z. Dan, Improving the accuracy in software effort estimation: using artificial neural network model based on particle swarm optimization, in: International Conference on Service Operations and Logistics, and Informatics (SOLI), IEEE, 2013, pp. 180–185.
[40] Mandeep Kaur, K. Sehra, Particle swarm optimization based effort estimation using function point analysis, in: International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014, pp. 140–145.
[41] Rao T.Benala, R. Mall, DABE: differential evolution in analogy-based software development effort estimation, Swarm Evol. Comput. 38 (2018) 158–172.
[42] A. Oliveira, P. Braga, Ricardo Massa Ferreira Lima, Márcio Cornélio, GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation, Inf. Softw. Technol. 52 (11) (2010) 1155–1166.
[43] Mohamed Hosni, Ali Idri, Alain Abran, Investigating heterogeneous ensembles with filter feature selection for software effort estimation, in: Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement, ACM, 2017, pp. 207–220.
[44] F. Sarro, S. Di Martino, F. Ferrucci, C. Gravino, A further analysis on the use of genetic algorithm to configure support vector machines for inter-release fault prediction, Sympos. Appl. Comput. (2012) 1215–1220.

[45] James H. Andrews, Tim Menzies, Felix Chun Hang Li, Genetic algorithms for randomized unit testing, IEEE Trans. Softw. Eng. 37 (1) (2011) 80–94.
[46] Kholed Langsari, Riyanarto Sarno, Optimizing effort and time parameters of COCOMO II estimation using fuzzy multi-objective PSO, in: International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2017, pp. 1–6.
[47] Zhang Dan, Improving the accuracy in software effort estimation: using artificial neural network model based on particle swarm optimization, in: International Conference on Service Operations and Logistics, and Informatics (SOLI), IEEE, 2013, pp. 180–185.
[48] Farhad Soleimanian Gharehchopogh, Isa Maleki, Seyyed Reza Khaze, A novel particle swarm optimization approach for software effort, Int. J. Acad. Res. 6 (2) (2014).
[49] Tirimula Rao Benala, Rajib Mall, DABE: differential evolution in analogy-based software development effort estimation, Swarm Evol. Comput. 38 (2018) 158–172.
[50] Seyyed Hamid Samareh Moosavi, Vahid Khatibi Bardsiri, Satin bowerbird optimizer: a new optimization algorithm to optimize ANFIS for software development effort estimation, Eng. Appl. Artif. Intell. 60 (2017) 1–15.
[51] Jin-cherng Lin, Han-yuan Tzeng, Yueh-ting Lin, Automatically estimating software effort and cost using computing intelligence technique, 2012.
[52] Vahid Khatibi Bardsiri, Dayang Norhayati Abang Jawawi, Siti Zaiton Mohd Hashim, Elham Khatibi, A PSO-based model to increase the accuracy of software development effort estimation, Softw. Qual. J. 21 (3) (2013) 501–526.
[53] Sultan Aljahdali, Alaa F. Sheta, Software effort estimation by tuning COOCMO model parameters using differential evolution, in: International Conference on Computer Systems and Applications (AICCSA), 2010, pp. 1–6.
[54] Farhad Soleimanian Gharehchopogh, Laya Ebrahimi, Isa Maleki, Saman Joudati Gourabi, A novel PSO based approach with hybrid of fuzzy C-means and learning automata in software cost estimation, Indian J. Sci. Technol. 7 (6) (2014) 795–803.
[55] Jin-Cherng Lin, Chu-Ting Chang, Sheng-Yu Huang, Research on software effort estimation combined with genetic algorithm and support vector regression, in: International Symposium on Computer Science and Society (ISCCS), 2011, pp. 349–352.
[56] Mohammed Algabri, Fahman Saeed, Hassan Mathkour, Nejmeddine Tagoug, Optimization of soft cost estimation using genetic algorithm for NASA software projects, in: National Symposium on Information Technology: Towards New Smart World (NSITNSW), 2015, pp. 1–4.
[57] Tirimula Rao Benala, Rajib Mall, DABE: differential evolution in analogy-based software development effort estimation, Swarm Evol. Comput. 38 (2018) 158–172.
[58] L.I. Adriano Oliveira, Petronio L. Braga, Ricardo MF Lima, Márcio L. Cornélio, GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation, Inf. Softw. Technol. 52 (11) (2010) 1155–1166.
[59] Shailendra Pratap Singh, Vibhav Prakash Singh, Ashok Kumar Mehta, Differential evolution using homeostasis adaption based mutation operator and its application for software cost estimation, J. King Saud Univ, Comput. Inf. Sci. (2018).
[60] Rohit Kumar Sachan, Ayush Nigam, Avinash Singh, Sharad Singh, Manjeet Choudhary, Avinash Tiwari, Dharmender Singh Kushwaha, Optimizing basic COCOMO model using simplified genetic algorithm, Proc. Comput. Sci. 89 (2016) 492–498.
[61] Y.F. Li, M. Xie, T.N. Goh, A study of genetic algorithm for project selection for analogy based software cost estimation, in: International Conference on Industrial Engineering and Engineering Management, 2007, pp. 1256–1260.
[62] Isa Maleki, Ali Ghaffari, Mohammad Masdari, A new approach for software cost estimation with hybrid genetic algorithm and ant colony optimization, Int. J. Innovat. Appl. Stud. 5 (1) (2014) 72.
[63] R. Kishore, D.L. Gupta, Software effort estimation using satin bowerbird algorithm, Int. J. Res. Appl. Sci. Eng. Technol. 6 (7) (2017).
[64] V. Venkataiah, Ramakanta Mohanty, J.S. Pahariya, M. Nagaratna, Application of ant colony optimization techniques to predict software cost estimation, in: Computer Communication, Networking and Internet Security, 2017, pp. 315–325.
[65] J. Keung, E. Kocaguneli, T. Menzies, Finding conclusion stability for selecting the best effort predictor in software effort estimation, Autom. Softw. Eng. 20 (4) (2013) 543–567.
[66] A.J. Albrecht, J.E. Gaffney, Software function, source lines of code, and development effort prediction: a software science validation, IEEE Trans. Softw. Eng. 9 (6) (1983) 639–648.
[67] Yun F. China, Effort Estimation Dataset, 2010.
[68] B.W. Boehm, Software Engineering Economics, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[69] B. Sigweni, M. Shepperd, Finnish Software Effort Dataset, 2015.
[70] C.F. Kemerer, An empirical validation of software cost estimation models, Commun. ACM 30 (5) (1987) 416–429.
[71] Y. Miyazaki, M. Terakado, K. Ozaki, H. Nozaki, Robust regression for developing software estimation models, J. Syst. Softw. 27 (1) (1994) 3–16.
[72] K.D. Maxwell, Applied Statistics for Software Managers, Software Quality Institute Series, Prentice Hall, 2002.
[73] S. Shirabad, T.J.J. Menzies, The PROMISE Repository of Software Engineering Databases, School of Information Technology and Engineering, University of Ottawa, Canada, 2005, available http://promise.site.uottawa.ca/SERepositor.
[74] A. Petrozziello Sarro, M. Harman, Multi-objective software effort estimation, in: Proc. of ICSE'16, 2016, pp. 619–630.
[75] F. Ferrucci Sarro, C. Gravino, Single and multi objective genetic programming for software development effort estimation, in: Proc. of SAC'12, ACM, 2012, pp. 1221–1226.
[76] C. Gravino Ferrucci, F. Sarro, Exploiting prior-phase effort data to estimate the effort for the subsequent phases: a further assessment, in: Proc. of PROMISE'14, ACM, 2014, pp. 42–51.
[77] M. Shepperd Sigweni, T. Turchi, Realistic assessment of software effort estimation models, in: Proc. of EASE'16, ACM, 2016, pp. 41:1–41:6.
[78] Adriano L.I. Oliveira, Estimation of software project effort with support vector regression, Neurocomputing 69 (13–15) (2006) 1749–1753.
[79] Y. Liu, Y. Wang, J. Zhang, New machine learning algorithm: random forest, Inform. Comput. Appl. (2012) 246–252.
[80] P. Subitsha, J. Kowski, Artificial neural network models for software effort estimation, Int. J. Technol. Enhancem. Emerg. Eng. Res. 2 (4).
[81] A. Schneider, G. Hommel, M. Blettner, Linear regression analysis, Dtsch Arzteblatt 107 (44) (2010).
[82] Ali Behnood, Venous Behnood, Mahsa Modiri Gharehveran, Kursat Esat Alyamac, Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm, Constr. Build. Mater. 142 (2017) 199–207.
[83] Mark Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, Ian H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.
[84] M. Shepperd, Q. Song, C. Mair, Data quality: some comments on the nasa software defect datasets, IEEE Trans. Softw. Eng. 39 (9) (2013) 1208–1215.
[85] S. Binitha, S. Siva Sathya, A survey of bio inspired optimization algorithms, Int. J. Soft Comput. Eng. 2 (2) (2012) 137–151.
[86] E.U. Haq, I. Ahmad, A. Hussain, I.M. Almanjahie, A novel selection approach for genetic algorithms for global optimization of multimodal continuous functions, Comput. Intell. Neurosci. 2019 (2019).
[87] Simon Fong, Robert P. Biuk-Aghai, Richard C. Millham, Swarm search methods in weka for data mining, in: Proceedings of the 2018 10th International Conference on Machine Learning and Computing, 2018, pp. 122–127.
[88] X.S. Yang, Harmony search as a metaheuristic algorithm, in: Music-Inspired Harmony Search Algorithm, Springer, Berlin, Heidelberg, 2009, pp. 1–14.
[89] Mohammad Aizat Basir, Yuhanis Yusof, Mohamed Saifullah, Optimization of attribute selection model using bio-inspired, J. ICT 18 (1) (2019) 35–55.
[90] D. Boughaci, A.A.S. Alkhawaldeh, Three local search-based methods for feature selection in credit scoring, Vietnam J. Comput. Sci. 5 (2) (2018) 107–121.
[91] J. Murillo-Morera, C. Quesada-López, C. Castro-Herrera, M. Jenkins, A genetic algorithm based framework for software effort prediction, J. Soft. Eng. Res. Dev. 5 (1) (2017) 4.

[92] Eibe Frank Retrieved from, https://weka.8497.n7.nabble.com/About-default-parameter-values-of-weka-td29652.html.
[93] Amir Hossein Gandomi, Amir Hossein Alavi, Krill herd, a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845.
[94] Pedram Ghamisi, Jon Atli Benediktsson, Feature selection based on hybridization of genetic algorithm and particle swarm optimization, IEEE Geosci. Remote Sens. Lett. 12 (2) (2015) 309–313.
[95] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2014) 261–276.
[96] Marco Dorigo, M. Birattari, Ant colony optimization, in: Encyclopedia of Machine Learning, 2011, pp. 36–39.
[97] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, Boston, 1997.
[98] Xin-She Yang, Xingshi He, Firefly algorithm: recent advances and applications, preprint, arXiv:1308.3898, 2013.
[99] Edgar Alfredo Portilla-Flores, ÁLvaro Sánchez-Márquez, Leticia Flores-Pulido, Eduardo Vega-Alvarado, Maria Bárbara Calva Yáñez, Jorge Alexander Aponte-Rodríguez, Paola Andrea Niño-Suárez, Enhancing the harmony search algorithm performance on constrained numerical optimization, IEEE Access 5 (2017) 25759–25780.
[100] James Bergstra, Yoshua Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.
[101] Yolanda S. Baker, Rajeev Agrawal, James A. Foster, Daniel Beck, Gerry Dozier, Applying machine learning techniques in detecting Bacterial Vaginosis, in: International Conference on Machine Learning and Cybernetics, Vol. 1, 2014, pp. 241–246.
[102] http://weka.sourceforge.net/doc.dev/weka/attributeSelection/GreedyStepwise.html, Waikato University.
[103] Lionel C. Briand, Khaled El Emam, Dagmar Surmann, Isabella Wieczorek, Katrina D. Maxwell, An assessment and comparison of common software cost estimation modeling techniques, in: Proceedings of the International Conference on Software Engineering, 1999, pp. 313–322.
[104] Trevor S. Wiens, Brenda C. Dale, Mark S. Boyce, G. Peter Kershaw, Three way k-fold cross-validation of resource selection functions, Ecol. Model. 212 (3–4) (2008) 244–255.
[105] T. Chakkrit, S. McIntosh, A. Hassan, K. Matsumoto, An empirical comparison of model validation techniques for defect prediction models, IEEE Trans. Softw. Eng. 43 (1) (2016) 1–18.
[106] Asad Ali, Carmine Gravino, A systematic literature review of software effort prediction using machine learning methods, J. Softw. Evol. Process (2019) e2211.
[107] L.M. Pickard Kitchenham, S.G. MacDonell, M.J. Shepperd, What accuracy statistics really measure, IEEE Proc. Softw. 148 (3) (2001) 81–85.
[108] M. Korte, Confidence in software cost estimation results based on MMRE and pred, in: Proc. of PROMISE'08, 2008, pp. 63–70.
[109] M. Shepperd, C. Schofield, Estimating software project effort using analogies, IEEE TSE 23 (11) (2000) 736–743.
[110] M. Shepperd, S. MacDonell, Evaluating prediction systems in software project estimation, IST 54 (8) (2012) 820–827.
[111] J. Dolado Langdon, F. Sarro, M. Harman, Exact mean absolute error of baseline predictor, MARP0, IST 73 (2016) 16–18.
[112] B. Kitchenham, L.M. Pickard, S.G. MacDonell, M.J. Shepperd, What accuracy statistics really measure, IEE Proc., Softw. 148 (3) (2001) 81–85.
[113] P. Royston, An extension of Shapiro and Wilk's W test for normality to large samples, Appl. Stat. 31 (2) (1982) 115–124.
[114] G. Neumann, M. Harman, S. Poulding, Transformed vargha-delaney effect size, in: International Symposium on Search Based Software Engineering, Springer, Cham, 2015, pp. 318–324.
[115] G. Neumann, H. Harman, S. Poulding, Transformed Vargha-Delaney effect size, in: Proc. of SSBSE'15, 2015, pp. 318–324.
[116] A. Arcuri, L. Briand, A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering, Softw. Test. Verif. Reliab. 24 (3) (2014) 219–250.
[117] Mark Hall, Advanced Data Mining with Weka, https://www.cs.waikato.ac.nz/ml/weka/mooc/moredataminingwithweka/slides/Class4-MoreDataMiningWithWeka-2014.pdf.
[118] Frishman D. Smialowski, S. Kramer, Pitfalls of supervised feature selection, Bioinformatics 26 (3) (2010) 440–443.
[119] E. Frank, et al., Data mining in bioinformatics using Weka, Bioinformatics 20 (2004) 2479–2481.
[120] H. Wang, T.M. Khoshgoftaar, K. Gao, N. Seliya, High-dimensional software engineering data and feature selection, in: 2009 21st IEEE International Conference on Tools with Artificial Intelligence IEEE, 2009, pp. 83–90.
[121] Sarah Nogueira, G. Brown, Measuring the stability of feature selection, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Cham, 2016, pp. 442–457.
[122] B.A. Kitchenham, L. Pickard, S. Peeger, Case studies for method and tool evaluation, IEEE Softw. 12 (4) (1995) 52–62.
[123] Cristian Ioan Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Inf. Process. Lett. 85 (6) (2003) 317–325.
[124] Adam P. Piotrowski, Jaroslaw J. Napiorkowski, Agnieszka E. Piotrowska, Population size in particle swarm optimization, Swarm Evol. Comput. 58 (2020) 100718.
[125] F. Sarro, Search-based approaches for software development effort estimation, in: Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement, 2011, June, pp. 38–43.