Computers in Biology and Medicine 125 (2020) 103974

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: http://www.elsevier.com/locate/compbiomed

GeFeS: A generalized wrapper feature selection approach for optimizing classification performance

Golnaz Sahebi [a,*], Parisa Movahedi [a], Masoumeh Ebrahimi [a,b], Tapio Pahikkala [a], Juha Plosila [a], Hannu Tenhunen [a,b]

[a] Department of Future Technologies, University of Turku, Turku, FI-20014, Turun yliopisto, Finland
[b] School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden

ARTICLE INFO

Keywords:
Medical datasets
Evolutionary computing
Machine learning
Data mining
Feature selection
Overfitting
Parallel computing

ABSTRACT

In this paper, we propose a generalized wrapper-based feature selection, called GeFeS, which is based on a parallel new intelligent genetic algorithm (GA). The proposed GeFeS works properly under different numerical dataset dimensions and sizes, carefully tries to avoid overfitting and significantly enhances classification accuracy. To make the GA more accurate, robust and intelligent, we have proposed a new operator for features weighting, improved the mutation and crossover operators, and integrated nested cross-validation into the GA process to properly validate the learning model. The *k*-nearest neighbor (*k*NN) classifier is utilized to evaluate the goodness of selected features. We have evaluated the efficiency of GeFeS on various datasets selected from the UCI machine learning repository. The performance is compared with state-of-the-art classification and feature selection methods. The results demonstrate that GeFeS can significantly generalize the proposed multi-population intelligent genetic algorithm under different sizes of two-class and multi-class datasets. We have achieved the average classification accuracy of 95.83%, 97.62%, 99.02%, 98.51%, and 94.28% while reducing the number of features from 56 to 28, 34 to 18, 279 to 135, 30 to 16, and 19 to 9 under lung cancer, dermatology, arrhythmia, WDBC, and hepatitis, respectively.

1. Introduction

With the development of new technologies, such as industrial internet-of-things (IoT), mankind has been pushed to an era of great information [1]. Nowadays, due to ubiquitous health IoT and digital healthcare, enormous biomedical datasets have been generated. However, in the face of substantial digital information, an urgent challenge is how to acquire accurate information [2]. The biomedical datasets have the characteristics of high-dimensionality, different sizes, data noises, and missing values [3]. These complex raw data demote the performance of the machine-learning algorithms, reducing their accuracy, reliability and generalization capability to unseen test data. Therefore, developing accurate and reliable machine-learning algorithms is a critical issue in auto-detection systems, especially for disease detection systems in medical analysis.

Neural networks (NNs) or deep neural networks (DNNs) are proper candidates to solve classification problems. Despite their benefits in a wide range of applications in different domains, deep learning is not still a panacea for high-dimensional medium/small-sized data. NNs/DNNs have many advantages, but they need immense training datasets to establish the accuracy of learning classifier [4,5]. Furthermore, NNS/DNNs are not reliable when the data is noisy and small. NNs/DNNs are only as good as the data they learned from Ref. [6]. While DNNs are recently used to learn non-linear feature interactions, the deep structure makes them difficult to train [7]. NNs inherently perform feature selection, but the performance degrades when datasets are high dimensional and sparse [5]. Considering the mentioned barriers, one of the best weapons against the curse of dimensionality and leakage of samples in datasets is still the use of traditional machine learning and feature selection techniques. It can decrease the overall training time, enhance generalization, and avoid overfitting. In this case, a model trained on a lower-dimensional dataset is computationally efficient and more reliable.

Feature selection plays a remarkable role in raising the performance of machine learning in terms of reducing the curse of dimensionality, alleviating the current situation of information abundance and knowledge shortage understanding data, increasing the accuracy in the learning process, and reducing the complexity and time of building the

* Corresponding author.
E-mail addresses: golnaz.sahebi@utu.fi, sahebigolnaz@gmail.com (G. Sahebi).

https://doi.org/10.1016/j.compbiomed.2020.103974
Received 8 April 2020; Received in revised form 12 August 2020; Accepted 12 August 2020
Available online 20 August 2020
0010-4825/© 2020 Elsevier Ltd. All rights reserved.

learning model [8]. Feature selection techniques overcome this problem by removing irrelevant and redundant features. Feature selection can be expressed as a process of choosing a minimum subset of relevant features from the given set of features for use in model construction. Therefore, the optimal feature space reduces the cost of model construction while enhances the classification performance or remains it relatively the same [9]. Three general approaches to solve the feature selection problem are filter, wrapper, and embedded methods. Features are chosen by statistical properties in the filter category. By applying the filter approach, features can be immediately selected, but the selected features may not be the best possible ones. Hence, the performance of the learning models is not usually as high as that of the wrapper method [10]. The embedded techniques perform feature selection as part of the learning procedure. One of the most typical embedded technique is the decision tree algorithm. The wrapper methods evaluate subsets of features according to their usefulness to a given predictor. The wrapper technique employs optimization algorithms to find an optimal subset of features. A remarkable accomplishment of this method is providing the use of optimization techniques combined with machine learning. There are different approaches to solve optimization problems such as deterministic, heuristic, and meta-heuristic searches [11].

However, searching for an optimal feature subset in a complex high-dimensional feature space is a multi-objective and NP-complete problem [12]. Therefore, solving these complex problems cannot be efficiently accomplished by deterministic methods and traditional optimization algorithms. The wrapper-based meta-heuristic methods are known as proper approaches to address the feature selection problem in high-dimensional datasets [13,14]. Evolutionary algorithms (EAs) are a well-known class of meta-heuristic searches [11]. A dominant advantage of EAs, compared with deterministic algorithms, is that they may avoid getting stuck in the local optima, although providing a guarantee for this is not effortless [15]. A popular group of EAs is the genetic algorithm (GA). GA is a type of meta-heuristic optimization method with a long history in the artificial intelligence and robotics domain [16]. GA is a population-based search technique that mimics the process of natural evolution. A genetic algorithm is begun with initializing a population of individuals and then performing frequent operations such as selection, crossover, mutation, and replacement. All operations of the algorithm are repeated until gaining a competent result or a specified iteration [17].

Even though EAs are fruitful in solving different problems, some drawbacks are associated with them in dealing with large search spaces [18,19]. Converging to the local optima is more probable for algorithms in the large search spaces. In addition, the wrapper-based feature selection combined with EAs is computationally expensive. These problems can be mitigated by creating an excellent trade-off between exploration and exploitation of the search space [20] and parallelization. Parallelizing EAs can improve the quality of results while decreasing timing overhead [18,19]. Among the parallelization techniques of EAs, multi-population implementations are helpful for GAs where there are multiple processors with several memory units. This implementation provides a more significant population diversity to improve the accuracy of results while reducing the time overhead by distributing the computational effort [18].

Increasing the initial population size improves the population diversity, which is a key point in selecting relevant features. In a multi-population strategy, there is a collection of processors, such that each processor hosts an independent population of chromosomes and runs a serial GA on its population. After several iterations of running GA on the processors in the migration phase, which is one of the principal operators of multi-population GA, each processor selects some of the best chromosomes to send to other processors. The migration operator shares the best chromosome of each processor with the others, enabling the discovery of the best solution in lower iterations while providing higher accuracy. In this paper to effectively solve the feature selection problem, a novel intelligent GA with an efficient multi-population implementation is proposed. The proposed feature selection algorithm lead to significantly increase the classification accuracy and the reliability of decision-making. The proposed GA introduces an adaptive weighting operator to enhance the accuracy of the optimum solution. The weighting method runs during the process of the genetic algorithm. It improves the model generalization and enhances the effectiveness under different datasets. The method is evaluated on five medical datasets from UCI machine learning repository.

Despite the fact that wrapper-based feature selection methods are more effective in the face of high-dimensional data, they are computationally expensive and may overfit to the training data. To mitigate these challenges, in this paper, our focus is on four significant characteristics of a proper feature selection method as: having a good generalization capability to unseen data, prevention from overfitting, increasing the average accuracy of the classifier for both two-class and multi-class datasets, and cost reduction for measuring feature values. The first two properties can be satisfied by choosing a proper model validation and hyper-parameter set for the classifier in use, utilizing nested cross-validation for evaluating the selected model and the subset of features (chromosomes) [21]. Furthermore, increasing the accuracy and minimizing the cost are achieved by choosing a subset of essential and unique features with high predictive performance along with a fast and efficient multi-population novel genetic algorithm. The proposed method, called GeFeS, is evaluated on various medical datasets, representing different data dimensionality, sizes, and class distributions. The main contributions of this work are as follows:

- The proposed GA introduces an adaptive embedded weighting method based on a novel operator, called Inverse, to make the algorithm more accurate, robust and intelligent. This operator is used during the process of the GA to generalize the algorithm for different numeric datasets.
- GeFeS integrates the nested cross-validation into the GA process for model validation. An extensive analysis is performed on different medical datasets to evaluate the proposed method. The obtained results indicate that GeFeS is superior to other methods in terms of the optimum number of features and the classification average accuracy under different sizes of numerical two-class and multi-class datasets while carefully tries to avoid overfitting.
- The proposed method creates an excellent trade-off between exploration and exploitation of search space by creating intelligence mutation and crossover operators. These intelligent mutation and crossover operators are proposed, which offer a good trade-off between exploration and exploitation of the search space. This new genetic algorithm efficiently escapes from the local optima.
- An efficient parallel environment is employed, which significantly enhances the performance of GeFeS in terms of accuracy and computational time.

The rest of the paper is organized as follows: Section 2 covers the state-of-the-arts in the area. Section 3 presents GeFeS: the proposed GA-based feature selection. Section 4 describes the experimental designs of GeFeS. Section 5 evaluates the proposed method and presents the experimental results. Finally, Section 6 discusses the findings and Section 7 concludes the paper.

2. Related work

Currently, a massive amount of data with hundreds of features is produced by the leading experimental techniques in biology as well as the other domains. This vast amount of data comes with many irrelevant and redundant features. Moreover, either medium or small sample sizes with the high dimensionality of the features is still one of the main issues in medical datasets. On the other hand, data mining methods tend to over-fit the data, which leads to poor generalization. Therefore, there is a need to find an optimized and smaller set of features while avoiding overfitting to increase the accuracy of making decisions and reducing the cost [22]. As will be reported in this section, there are various studies focusing on the diagnosis tasks on the medical datasets by applying

different feature selection methods.

Feature selection methods are generally presented in three classes depending on how the selection algorithm is combined with the model building. The classes are divided as filter, wrapper, and embedded techniques. Wrappers are significantly useful when the curse of dimensionality becomes a considerable challenge. Although this method can be computationally expensive and may have a risk of model overfitting, these issues can be mitigated by a reliable optimization algorithm and an efficient implementation. This section briefly summarizes some classification and feature selection methods based on (deep) neural networks and some traditional wrapper, filter, and hybrid feature selection techniques which have been tested on UCI Wisconsin diagnostic breast cancer (WDBC) [23], arrhythmia [24], dermatology [25], hepatitis [26], and lung cancer [27] datasets.

Xu et al. [28] proposed a hybrid method to improve the performance of heart arrhythmia classification by selecting relevant features from ECG signals and applying deep neural networks for classification. The number of hidden nodes were set to 25 per layer, and the networks had a varied number (2–4) of hidden layers. They employed the arrhythmia dataset from UCI to test their method. Since the dataset is highly imbalanced, the fifteen classes representing different kinds of heart arrhythmia are merged into one class, called Abnormal. The obtained best accuracy across 10-fold cross-validation by NNs only, DNNs only, FDR + DNNs, and PCA + DNNs were 82.22%, 81.42%, 82.96%, and 75.22%, respectively. Dutta et al. [29] presented a classification method for medical data mining. This method applies the firework algorithm (an evolutionary algorithm) in the training of multi-layer perceptron. The classification was performed on five medical datasets from UCI machine learning repository such as WDBC, lung cancer, kidney, and heart. The obtained testing accuracy based on a 10-fold cross-validation for WDBC and lung cancer data were 95.53% (±1.45) and 66.71% (±4.72), respectively. A. Saygili [30] proposed a survey on different classification and diagnosis prediction methods for the breast cancer. Several classifiers from Weka were applied on UCI WDBC dataset. One of the best accuracies, across 10-fold cross-validation, was obtained by multi-layer perceptron (98.41%). Jadhav et al. [31] presented a method based on a modular neural network to classify arrhythmia diseases. The experiments were performed on the UCI arrhythmia dataset, and a varying number of hidden layers (1–3) are used to construct the proposed neural network. The obtained testing classification accuracy was 82.22% when the Modular neural network (MNN) network with one to three hidden layers was utilized. Kumar et al. [32] proposed a dual-stage approach that combines machine learning and computer vision to detect different types of dermatology diseases. The method has used maximum entropy model and artificial neural networks for the feature extraction phase. Then it has utilized some classifiers such as *k*NN, decision tree, and neural network to classify the subjects. The UCI dermatology dataset has been used, and the system obtained an accuracy of up to 95.00%. Sun et al. [33] presented a collaborative deep learning method to make a clinical decision on medical datasets. The approach has presented neighbor-based and latent feature-based CF methods to diagnose diseases, and in order to extract the latent features a discriminative restricted Boltzmann machine has been proposed, where the deep learning was adopted to analyze the clinical data. The best-obtained accuracy and F1-score on UCI dermatology dataset were 96.89% and 0.44, respectively. Panthong et al. [34] proposed a wrapper feature selection method for dimension reduction that works based on an ensemble learning algorithm. This work utilizes sequential forward selection (SFS), sequential backward selection (SBS), and optimize selection (evolutionary) based on ensemble algorithms. The experiments were conducted with several UCI datasets, and the best accuracy is obtained on lung cancer (87.50%) and dermatology (98.36%) datasets. 10-fold cross-validation is applied to evaluate the model. Zainudin et al. [35] proposed a hybrid feature selection algorithm as a combination of ReliefF with the differential evolution methods. In this study, the generation size and population size were adaptively determined from the number of features from ReliefF. The achieved *k*NN classification accuracies for the UCI lung cancer and dermatology datasets were 66.70% and 97.60%, respectively. 10-fold cross-validation was used to validate the performance of the method. Wan et al. [36] proposed a feature selection algorithm based on a modified binary-coded ant colony optimization algorithm (MBACO) in cooperation with the genetic algorithm. The method includes two models, which are the visibility density model (VMBACO) and the pheromone density model (PMBACO). In VMBACO, the solution obtained by GA is utilized as visibility information. On the other hand, in PMBACO, the solution achieved by GA is used as initial pheromone information. On the other hand, in PMBACO, the solution obtained by GA is used as initial pheromone information. The obtained classification accuracy for the UCI dermatology dataset was 95.16% and 94.64% for the two proposed approaches. The performance estimation of classifiers was done by choosing half of the data as training data, and the second half as testing data. Zhao et al. [37] proposed a feature selection method based on potential entropy evaluation criteria (FMPE). This method takes the distribution of the data into consideration when measuring the importance of the feature. The *k*NN classification accuracy of 98.08% is obtained on UCI dermatology dataset while utilizing 10-fold cross-validation. Gu et al. [38] used a very recent particle swarm optimization variant, known as a competitive swarm optimizer (CSO) to solve high-dimensional feature selection problems. This work adapted the CSO, which was originally developed for continuous optimization, to perform feature selection. It introduced an archive technique to reduce computational cost. However, it used the *k*NN classifier with a constant hyper-parameter ($k = 5$) to test the effectiveness of the proposed algorithm. It also used the average error rates of 10-fold cross-validation on training data as the fitness function. The obtained average error rate was 0.3240 on UCI arrhythmia dataset. Vinh et al. [39] proposed a feature selection method based on the normalization of the mutual information measurement. Their method was derived from the max-relevance and min-redundancy (mRMR) approach. Based on their claim, they could eliminate the domination of relevance or redundancy. However, they obtained *k*NN classification accuracy of 65.47% on the UCI arrhythmia dataset and 97.18% on WDBC data. The results show that the classification accuracy is significantly reduced by increasing the data dimensionality. Rao et al. [40] presented a feature selection algorithm based on the bee colony and gradient boosting decision tree. The approach utilized the bee colony optimization technique to identify the informative features, and achieved global optimization of the inputs that obtains from the decision tree. The proposed method was applied on several public datasets such as UCI WDBC, and the model performance was evaluated by 10-fold cross-validation. They claimed that their method effectively decreased the dimensions of the dataset and obtained superior classification accuracy using the selected features (average 92.38% and max 97.90%). Lim et al. [41] extended Bandler–Kohout subproduct to interval-valued fuzzy sets, introduced a weight parameter in the BK subproduct-based inference engine, and developed a fuzzification method that is able to fuzzify the input data and train the inference engines. The proposed algorithm was evaluated on the WDBC dataset and obtained 95.26% classification accuracy based on 5-fold cross-validation. Zheng et al. [42] developed a system for breast cancer diagnosis based on feature extraction using k-means and SVM (K-SVM). The proposed methodology improves the accuracy to 97.38% based on 10-fold cross-validation when tested on the WDBC dataset, and six features are extracted from the 32 original features for the training phase. Chen et al. [43] proposed a system for concurrent parameter optimization and feature selection for SVM based on the parallel time-variant particle swarm optimization (PTVPSO). The classification accuracy of 98.44% is obtained when tested on the WDBC dataset. Saez et al. [44] proposed mutual information (MI) between features as a weighting factor for the *k*-nearest neighbor classifier. The obtained classification accuracy for the WDBC dataset was 96.14%, and the performance estimation of classifiers was done by means of 3 runs of a 10-fold distribution, optimally balances stratified cross-validation, averaging its test accuracy result. Oh et al. [45] proposed a hybrid genetic algorithm with embedded devised local search operations. They claimed that the hybridization method creates two excellent effects: a notable improvement in the final performance; and the acquisition of subset-size control. Experiments conducted on various standard datasets indicated that the proposed hybrid GA is superior to both a sequential search

algorithm and a simple GA. The method obtained the classification accuracy of 94.40% on WDBC dataset. Islam et al. [46] presented a novel modality for the prediction of breast cancer. They introduced the Support Vector Machine and *k*-nearest neighbor, which are the supervised machine-learning techniques for breast cancer detection by training its attributes. The proposed system used 10-fold cross-validation to an accurate outcome. The techniques achieved the accuracy of 97.14% by applying *k*-nearest neighbors on the WDBC dataset.

3. GeFeS: generalized feature selection algorithm based on parallel genetic algorithm

Previous literatures addressed the importance of selecting relevant features in biomedical and biological problems and illustrated the state-of-the-art methods [9,18,19]. The process of wrapper-based feature selection is tied to the performance of the classification model. Feature selection is based on the use of optimization methods and various search strategies such as genetic algorithm, hill climbing, and simulated annealing. A well-studied variant of randomized optimization methods and search strategies are evolutionary search algorithms (e.g., GA) that may avoid falling into locally optimal solutions. While this is an advantage, on the downside, the wrapper-based feature selection combined with randomized search strategies is computationally expensive when the data volume increases. Furthermore, the wrapper-based feature selection may lead to the selection of features that are biased and create an overfitting issue.

In this section, a reliable feature selection method based on a parallel new intelligent genetic algorithm is proposed to enhance the classification performance under different sizes of medical multi-class and two-class datasets. The main objective of the proposed approach is to tackle the wrapper-based feature selection challenges as: the tendency to overfitting and expensive computations while increasing the average accuracy of the classifier. We design an intelligent weighted genetic algorithm with the multi-population implementation that efficiently reduces the computation costs and escapes from the local optima to reduce the number of features while maximizing the accuracy of the classifier. Furthermore, it efficiently adapts to different dimensionality, sizes, and class distributions of numeric datasets.

3.1. Overview of GeFeS

We design an improved parallel genetic algorithm within the wrapper framework to solve the feature selection problem. Algorithm 1, Algorithm 2, and Fig. 1 demonstrate the progress of the proposed method. GeFeS increases the classifiers' accuracy by carefully selecting the best subset of features (chromosome). Through the cooperation of a proper optimization algorithm, a wrapper framework can achieve two main factors that affect the generalization ability of a classifier: selecting an optimal feature subset and finding the appropriate model learned from the selected features. The proposed framework makes the feature selection algorithm more accurate, efficient, and generalized. The multi-population strategy is utilized to parallelize computation while increasing accuracy. In the multi-population implementation of genetic algorithm, as shown in Fig. 1 (b), there are several processors that each processor separately runs a modified GA with its initial random population. Our proposed GA starts with creating the initial population of chromosomes, where each chromosome represents a subset of features in the dataset. Then the "goodness" of each chromosome (i.e., fitness value) is evaluated using *k*NN classifier, known as chromosome evaluation operator. It is worth mentioning that *k*NN is only a component to evaluate the quality of the candidate chromosomes. The parameters of the *k*NN model are carefully tuned using nested cross-validation. *k*NN can be replaced with other classifiers in the GeFeS architecture. In the current study, *k*NN is utilized due to the simplicity of the model construction and the lower complexity of its linear implementation to reduce the execution time. It helps to make fast decisions, especially for applications with the immediate need for model construction.

The biggest challenge here is to select a proper GeFeS model with a best chromosome set, evaluated using *k*NN classifier. The aim of the proposed model selection is to obtain a model that is simple and fits a given dataset with competent average accuracy on two-class and multi-class datasets, meanwhile preserving a good generalization capability to unseen data, which is described with more details in Subsection 3.2.2. Afterwards, the algorithm selects some chromosomes using the selection operator to improve and pass to the next generation. The improvements are made by using intelligent crossover, intelligent mutation, and a novel operator called inverse. In our intelligent crossover (IC), we create various combinations of chromosomes, and in our intelligent mutation (IM), we make random changes in some of them. The proposed operators provide an excellent trade-off between exploration and exploitation of the search space. While these changes are monitored, the inverse operator investigates the monitored genes more accurately. This operator creates the weighting method, which is based on the behavior of the problem during the process of two previous operations. The inverse operator efficiently updates the weights of features during the program. After the inverse operator, the replacement operator is applied to replace the improved chromosomes with the corresponding chromosomes in the first generation. Since we use the parallel method, the algorithm examines if it is the time for migration of the chromosomes between the processors. In migration time, the best chromosome of each processor is replaced with the worst one of the next processor, and this cycle is repeated until reaching a competent result or a determined iteration. Several phases of the proposed algorithm are described with more details in Subsection 3.1.

Algorithm 1. The proposed feature selection (GeFeS)

Input: Numerical dataset
Output: A reduced subset of features with maximum of average classification accuracy while carefully trying to avoid overfitting

Initialize:
  processor P_0**:** //master processor
  K = 10 //K-fold cross-validation
  $N_i \leftarrow$ number of instances
  $i = 0$
while $i < K$ **do**
  $train_1, test_1 \leftarrow$ create training and testing sets
  $ch_{best} \leftarrow$ IGA $(train_1)$ //to find the best chromosome of the subset
  $prediction[i] \leftarrow k$NN $(test_1, ch_{best})$ //to predict all instances in $test_1$
  $i = i + 1$
end while
$a_{avg.} \leftarrow$ compute the average accuracy of *prediction*
return $a_{avg.}$
end

Algorithm 2. The proposed genetic algorithm (IGA)

Input: $train_1$
Output: The best chromosome //The optimal subset of features

Initialize:
  processor P_p // $1 \le p \le N_{proc.}$
  $i = 0$ // number of iterations
  N_p ← number of chromosomes
  N_g ← number of genes // $N_g = number\ of\ features\ in\ dataset\ (N_f)$
  pop_i ← randomly create the initial population of size $N_p \times N_g$
  F_i ← Evaluate $(pop_i, train_1)$
while stopping criteria not met **do**
  pop_{i+1} ← Select parents (pop_i)
  pop_{i+1} ← ICrossover (pop_{i+1})
  pop_{i+1}, $W_{best}(1{:}N_F)$ ← IMutation (pop_{i+1}) // find the features that have the most affects (W_{best}) while mutating
  pop_{i+1} ← Inverse $\left(pop_{i+1}, W_{best}(1{:}N_f)\right)$ // inverse value of chromosomes based on W_{best}
  F_{i+1} ← Evaluate $(pop_{i+1}, train\ 1)$
  pop_{i+2} ← Replace (pop_i, pop_{i+1})
  $ch_{best}, ch_{worst} \leftarrow Find(best\&worst\ (pop_{i+2}))$
  if migration time is met **then**
    send ch_{best} to P_{p+1}
    $ch_{worst} \leftarrow receive\ ch_{best}\ from\ P_{p-1}$
  end if
  $i = i + 1$
end while
$Total_{best} = FindBest(ch_{best})$ // Find the best chromosome between all processors
return: $Total_{best}$
End

3.2. Performance evaluation of GeFeS

To evaluate the performance prediction of the proposed GeFeS framework, we use the K-fold cross-validation (K-FCV) strategy on top of the GeFeS. Fig. 1 (a) gives a demonstration of the performance evaluation. In each experiment set, the dataset is divided into ten separate subsets, and in each round of 10-fold cross-validation, one subset is held out for testing. The remaining sets are used for the training. In each round of K-FCV, the best chromosome chosen by the GeFeS model based on the training subset is utilized to build a *k*NN classifier and the constructed model is tested using the test dataset to evaluate the GeFeS feature selection method coupled with *k*NN classifier.

The number of the neighbors (k), for the *k*NN classifier, in each round of the 10-FCV is set to be the most frequently chosen k for the best-chosen chromosome retrieved from the fitness evaluation function. The prediction performance of the model is then evaluated based on the held-out set in each round of cross-validation using accuracy criteria. The final performance is achieved by averaging the accuracy of all the 10-FCV.

3.3. Search strategy description

The search strategy of GeFeS is based on the genetic algorithm (GA). GA alternatively works on two types of spaces: genotype (coding) and phenotype (solution). One of the pivotal issues of GAs is to map the phenotype space to the genotype one, i.e., representing the genes in a chromosome. A chromosome points to a certain-length string where all the genetic information of an individual is kept. This is the encoding phase in GA. An improper encoding breeds a poor GA performance. In addition, one dominant problem associated with encoding is that some chromosomes correspond to illegal or infeasible solutions for a given problem. This may become adequately intense for constrained and combinatorial optimization problems. Illegality takes place when a chromosome does not represent a solution in the related space. The problem can be mitigated by repair techniques that usually converts an illegal chromosome to a legal one. Infeasibility refers to the situation that a solution decoded from a chromosome lies outside the feasible area for the given problem. This issue can be handled by several methods that one of them is forcing genetic algorithms to approach an optimal solution from both feasible and infeasible areas [47]. Choosing the right scheme for encoding genes is a crucial task. Encoding mainly relies on the type of problem. Among the available encoding methods such as binary, integer, real-valued, ordinary, and permutation, GeFeS utilizes the binary encoding. The reason is that the feature selection should select a number of features from all available features in the complete dataset, and this can be perfectly represented by a binary encoding. Based on this encoding method, GeFeS is efficiently able to convert a complicated phenotype space to a one-dimensional vector that can represent a solution in legal and feasible areas. As it will be explained, GeFeS seeks all possible solutions in the search space to find the optimal solution. Several phases of the proposed GeFeS is described in more details as follows.

3.3.1. Representation of candidate solutions

To specify data structures in GA, the space of feasible solutions must be mapped onto the space of encoded solutions that is a representation of a GA candidate solution to the optimization problem. In GA, chromosomes represent the candidate solutions, and several chromosomes form a population. Several encoding schemes currently used in GA such as binary, integer, and real-valued encodings. The selection of encoding is depended to the optimization problem. Regarding the essence of the feature selection problem, binary encoding is used in this paper. The initial population consists of some chromosomes; each of them carries a possible solution to the problem and is formed of several genes. Each gene shows an attribute of the intended chromosomes, and the formation and determination of these attributes is the key task in evolutionary algorithms. The proposed GA operates on a binary search space, and the chromosomes are bit strings. Fig. 2 (a) visualizes an initial population of this study. As can be seen in Fig. 2 (b.i), for a binary chromosome

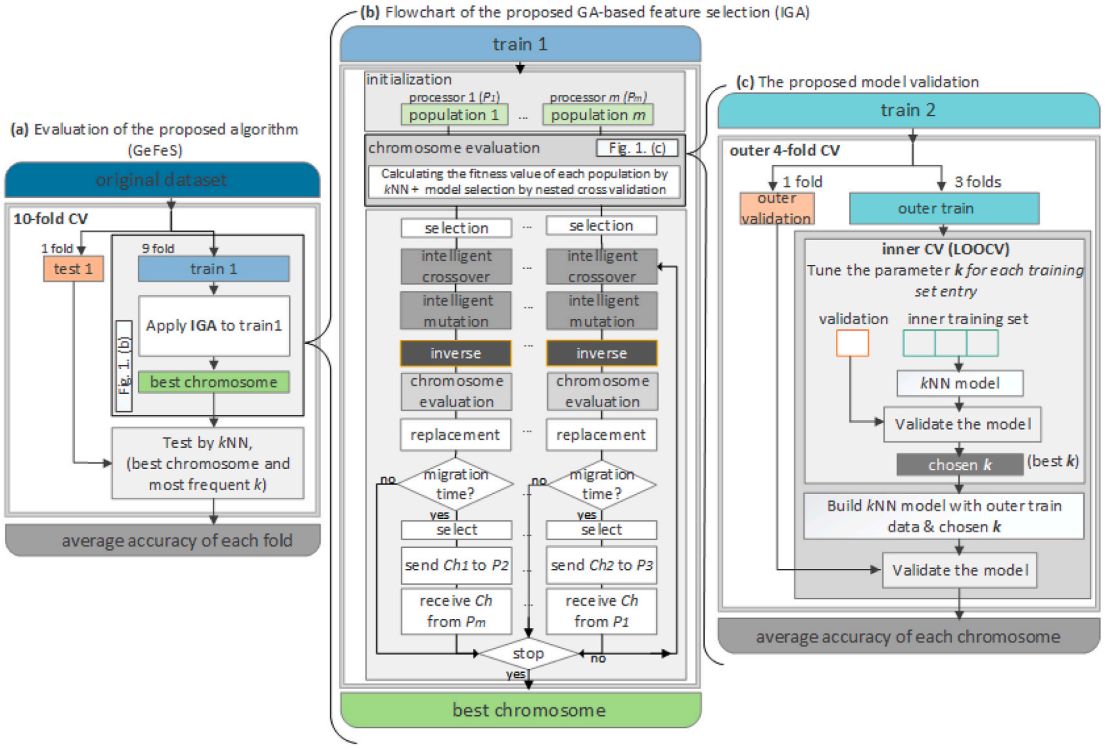


Fig. 1. Overall procedure of the proposed method (GeFeS).

employed in this work, the gene value '1' indicates that the corresponding feature is selected while the value '0' means that the feature is not selected for passing to the chromosomal evaluation phase. The gene width equals the number of features in the dataset. Thus, the initial population is a $N_p \times N_f$ matrix ($IP_{N_p \times N_f}$) where N_p is the population size and N_f is the chromosome width. Typically, each bit randomly takes the value one or zero while the bits associated with the more relevant features take the value one with a higher probability. Considering an example can be helpful to get a clear vision of the population and datasets in this study. Suppose that the original dataset has four columns of features and five rows of instances (Fig. 2 (b.ii)). Therefore, the length of each chromosome in the population equals four. Now, suppose that the selected chromosome is as same as shown in Fig. 2 (b.i), it also shows the indexes of the selected features. Hence based on the proposed chromosome, the first and the last features are selected from the original dataset to shape the new datasets (Fig. 2 (b.iii), which is an example of train 2 in Fig. 1 (c).

3.3.2. *Chromosome evaluation*

A fitness function is used to evaluate solutions (chromosomes) produced by a round in GeFeS. The probability of obtaining an optimal feature subset for classification can be increased by employing a proper fitness function for the feature selection.

3.3.2.1. Calculating the fitness evaluation function. One of the challenging stages in evolutionary algorithms is to define a proper fitness evaluation function since it provides the main link between a particular problem and the evolutionary algorithm. The fitness function of a chromosome is calculated in each iteration to decide if any improvement is observed in the solutions or not using the chromosome in question. In other words, the fitness function determines how to fit a chromosome is (the ability of a chromosome to compete with the others). Finally, in each step, the chromosome with the highest fitness value is considered as an optimal solution. We employ *k*NN classifier in a nested cross-validation routine to evaluate the effectiveness of chromosomes while tuning the *k*NN model parameters. *k*NN is very simple in implementation while works incredibly well in practice [22]. In addition, *k*NN is capable of adapting to various types of data by choosing a proper distance measure and builds the model in a short time with a low prediction bias. In the *k*NN classifier, an object is classified by a majority

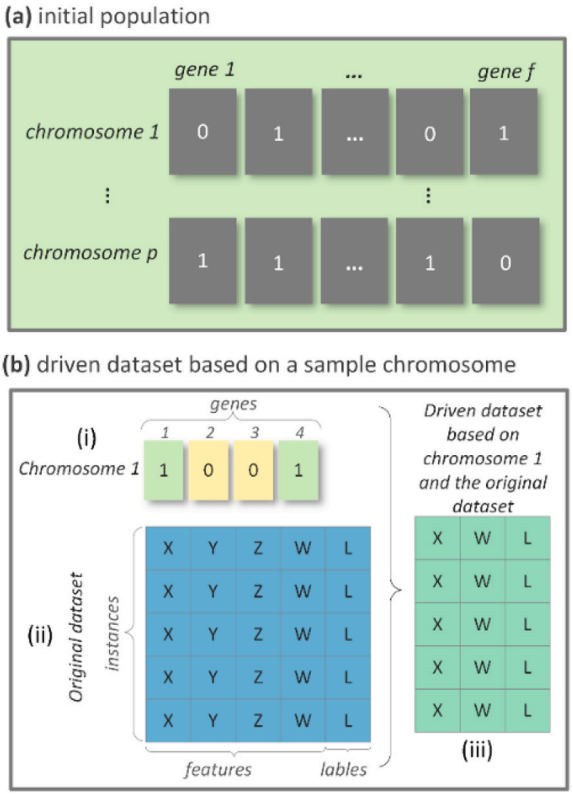


Fig. 2. A sample of an initial population (a) and a driven dataset based on a chromosome (b).

vote of its neighbors. The *k*NN solves the classification problem by searching the shortest distance between the test data and the training set in the feature space. In this paper, the distance ($D(x_{test}\ ,\ x_i)$) measure is based on the Euclidean distance, given in Equation (1):

$$D(x_{test}\ ,\ x_i) = \sqrt{\sum_{j=1}^{N_f} EDM(test,i,j)} \tag{1}$$

where x_{test} is an instance from the test set, x_i is an instance from the training set, and N_f is the number of features. The EDM function computes the Euclidean distance with the domain as the dataset and the range as the real numbers (EDM: $D \rightarrow \mathbb{R}$). This EDM function can be seen in Equation (2).

$$EDM\ (test,i,j) = \begin{cases} 0 & if\ ch[s][j] = 0 \\ (x_{test}[j] - x_i[j])^2 & if\ ch[s][j] = 1 \end{cases} \tag{2}$$

where $ch[s][j]$ is the *j-th* genom of the *s-th* chromosome in the initial population.

The evaluation of the chromosome *j* must be performed by an objective function (*f(j)*). The classification average accuracy ($a_{avg.}$) and the number of selected features (N_{sf}) are the two main criteria in our fitness function. As given in Equation (3), the chromosomes with a smaller number of features ($^1/_{N_{sf}}$) and higher $a_{avg.}$ will have a larger fitness value. Considering that both $a_{avg.}$ and $^1/_{N_{sf}}$ are real numbers between zero and one, the fitness value will be between zero and two ($0 \leq fitness \leq 2$).

$$f(j) = a_{avg.} + {}^1/_{N_{sf}} \tag{3}$$

Accuracy is calculated as the sum of correct classifications divided by the total number of classified objects [48]. The overall GeFeS performance and the chromosome fitness accuracy in the fitness function has been evaluated by the average classification accuracy ($a_{avg.}$) as shown in Equation (4).

$$a_{avg.} = \sum_{i=1}^{l} \left(\frac{tp_i + tn_i}{tp_i + tn_i + fp_i + fn_i} \right) \Big/ l \tag{4}$$

where *l* is the number of classes and tp_i, tn_i, fp_i, *and* fn_i are true positive, true negative, false positive, and false negative, respectively.

3.4. Model validation for chromosome evaluation

One of the main goals of the proposed GeFeS algorithm is to select a proper feature set (chromosome) to construct a prediction model, where selected features are evaluated by the *k*NN classifier. The model selection aims to obtain a model that is simple and fits a given dataset with satisfactory accuracy, meanwhile preserving a good generalization capability to unseen data [22]. When *k*NN is used inside the fitness evaluation function, its model parameters must be chosen carefully in order to build an effective classifier for evaluating the chromosome in question. An important issue in the model selection is a bias-variance tradeoff. Bias associates with the ability of the model to approximate the data while variance indicates the model stability in response to new training examples. A "biased" model generalizes well but does not fit the data perfectly ("under-fitting") while a "high-variance" model fits the training data well to the detriment of generalization ("overfitting"). There is normally a tradeoff between bias and variance, and many learning algorithms control this tradeoff, for instance, by a regularization parameter that penalizes complex models in many types of linear modelling approaches, or the number of neighbors (*k*) value in *k*NN [49]. Since an algorithm like *k*NN does not precisely assume anything particular about the distribution of the data points, it has a low bias. In other hands, *k*NN has a high variance because it can easily change its prediction in response to the composition of the training set. *k*NN can properly fit the training data if k is selected correctly, but may not generalize truly to new examples. One of the eminent solutions that enhances this issue is choosing a proper validation method to validate the performance of the classification model. One of the outstanding validation methods is nested cross-validation [21,22]. This technique splits data to K folds in the outer loop and assumes one fold as the validation set and the remaining as the training set. The training set is passed to the inner loop. The inner loop uses the LOOCV to evaluate the received dataset for different *k* values. Thereby in the inner loop, one of the instances is chosen as the validation set, and the rest are considered as the inner training set. By this technique, in addition to utilizing the maximum training set, we also use the maximum validation set without having a common member between the two sets. After the inner loop was executed for all desired neighbors (*k*), the algorithm passes the best *k* to the outer loop. Then, the next segment is chosen as the new validation set, and all the above actions are repeated again, and these procedures are executed K times. Fig. 3 describes an example of a 4-fold nested cross-validation function in one round.

In this study, as can be seen in Fig. 1 (c), the predictive performance of the model built by the *k*NN algorithm for fitness evaluation function is estimated by a nested cross-validation strategy, which consisted of an outer K-fold (K-FCV) and an inner leave-one-out cross-validation (LOOCV) for hyper-parameter selection. In K-FCV, the dataset is divided into K separate subsets, and one subset is held out at a time as the validation set, while the remaining data forms the training set used to build the model for predicting the held-out samples. The hyper-parameter selection is performed for each round on LOOCV using the training set. In this regard, the *k* value of the classifier is dynamically tuned in the inner cross-validation loop. Selecting an appropriate hyper-parameter can avoid the undesirable optimistic bias and high variance in the model selection. In this work, we choose the stable, most frequent *k, chosen by the fitness function,* in each fold by investigating the number of successful neighborhoods. It is illustrated that the nested CV is stable and properly fits our algorithm.

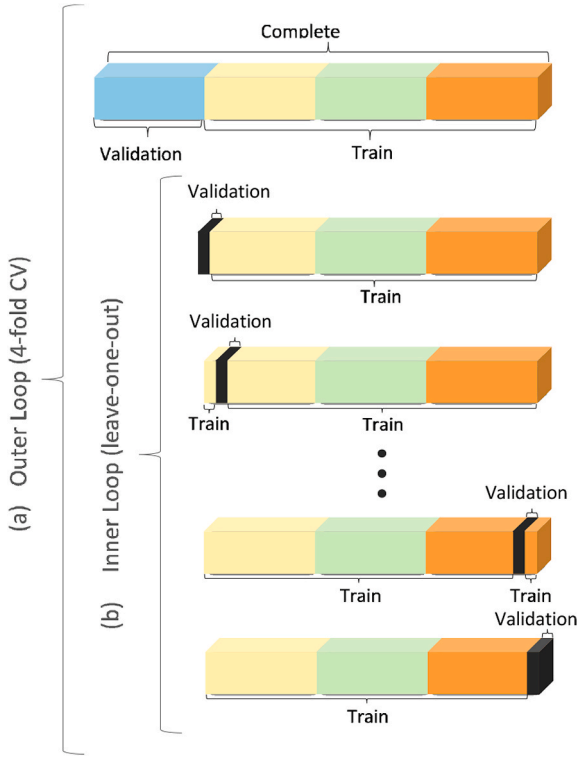


Fig. 3. An example of *4-fold* nested cross-validation, for one round.

3.4.1. Selection operator

The basic idea of the selection operator is to give preference to better chromosomes and allow these chromosomes to pass their genes to the next generation. Two pairs of chromosomes (parents) are chosen based on their fitness scores. Chromosomes with high score have more chance to be selected for reproduction (producing offspring). The selection operator in GeFeS is the tournament technique with three members. It means that in every cycle, three chromosomes are randomly selected, and then the best one is chosen for the next generation.

3.4.2. Exploitation strategy (intelligent crossover operator)

Crossover and mutation are two leading operators of GA that impact the fitness value. Crossover is applied to each pair of selected parents with the responsibility of recombination. Crossover is the most potent recombination operator in GA. It escapes from being stuck in the local optima by exploring new solution regions in the search space. Crossover randomly exchanges genes between two chromosomes. The operator is applied on a random set of chromosomes that are selected based on a probability, called the crossover rate (*Pc*) [50]. Fig. 4 (a) shows an example of a typical crossover operation in GA.

For binary encoding, several types of the crossover have been presented in the literatures such as single-point, multi-point, and uniform. In this study, the exploitation operator is created based on a multi-point crossover that intelligently escapes from the local optima. The proposed operator is a two-point crossover, acting as follows: first, two parents are chosen from the population based on the crossover rate (P_c). Second, two random numbers, like r_1 and r_2 (indexes of the genes), are generated such that r_1 is greater or equal to one and r_2 is lesser or equal to the number of features ($r_1 \geq 1, r_2 \leq N_f$). Afterwards, the values of these two genes are swapped. The notable advantage of our crossover operator is that the crossover rate adaptively changes during the process of GA. This change occurs whenever the convergence of the solutions is not improved for some successive iterations. In other words, the value of P_c adaptively changes to escape from the local optima. To fulfil this goal, P_c is set to 0.80 at the beginning so that exploitation is performed for as long as the algorithm converges. When there is not any improvement of convergence in the last five iterations, the proposed algorithm reduces the crossover rate by 0.2 and increases the mutation rate (P_m) by 0.2. This parameter tuning continues until reaching an improvement of convergence. From this point, the algorithm increases the crossover rate by 0.2 and reduces the mutation rate by 0.2 in every five iterations until reaching the initial rates of crossover (i.e., 0.8) and mutation (i.e., 0.2). It is worth mentioning that the crossover and mutation rates never become less than 0.2 and more than 0.8. Increment of P_m and synchronously decrement of P_c shifts the algorithm state from exploitation to exploration. From the perspective of local search, exploitation probes a promising limited region of the search space with the aim of improving the current solution. Whereas from the aspect of global search, exploration probes a much larger region of the search space with the aim of finding other promising solutions that are yet to be found [51].

3.4.3. Exploration strategy (intelligent mutation operator)

The main idea of mutation is to insert random genes in offspring to keep the diversity in the population. It is performed to prevent the premature convergence. Mutation randomly modifies the value of genes to obtain a significantly distinct phenotype of the offsprings compared to its parents. This operator occurs based on a probability, called mutation rate (P_m) [50]. Mutation may occasionally reverse the genes (i.e., from 0 to 1 or vice versa). This operator explores the search space and discovers a new exploration region to escape from the local optima. Fig. 4 (b) shows an example of applying a typical mutation on a selected chromosome (ch_i).

In the proposed approach, the type of mutation and $P_m$are adaptively adjusted by the algorithm. Depending on the state of the algorithm, the proposed mutation is intelligently applied in different forms. First, the bit flipped (one-point) mutation is performed in our algorithm with the rate of 0.20 until the algorithm appropriately converges to the optimal solutions. In this type of mutation, a number is randomly generated between 1 and the number of features (N_f) and assigned in the variable r_1. Then, the corresponding value of the r_1-*th* gene in the selected chromosome is reversed. Similar to our crossover process, P_m increases when the algorithm does not converge further for several iterations. After arising the first improvement in the convergence process, the proposed mutation returns to its initial state. The movement to the initial state enhances the crossover to exploit the space better.

In the mutation operation, an N_f-tuple vector (W_{best}) is created and initialized to zero (Fig. 4 (c.i)). Then the fitness function is computed for the chromosome that the mutation operator has been run on its r_1-*th* gene. The goodness of the chromosome is re-evaluated with the new value of fitness function. If the new value of the fitness function has a significant improvement, the value of the corresponding cell r_1-*th* gene in W_{best} is converted to one (Fig. 4 (c.ii)). This procedure continues until the end of the mutation operation, and W_{best} is updated as needed.

3.4.4. Inverse operator

One of the effective factors in the process of improving convergence in evolutionary algorithms is the feature weighting system. In our method, we increase/decrease the probability of the desired features and try to do a precise search on the possible composed cases of these features. As a result, the intelligence of our work increases, and the exploration and exploitation actions are conducted with a better knowledge on the search environment. Creating a comprehensive method that can perform a weighting operation for various datasets is a vital and valuable task. In this work, we create a comprehensive

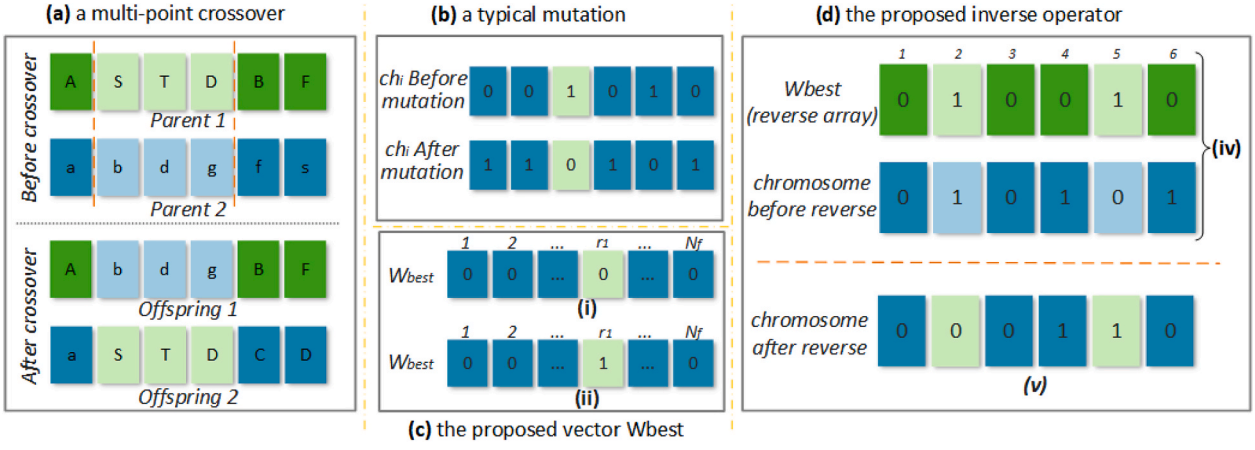


Fig. 4. Examples of a typical crossover (a), a typical mutation (b), the proposed vector W_{best} (c), and the proposed inverse (d).

weighting method that is embedded in the genetic algorithm process. The weighting operation in our algorithm consists of two main phases: 1) recognizing the effective positives or negatives features; 2) expressing the presence or absence of the recognized features. These actions are performed continuously throughout the execution of the program and applied to all generations. Therefore, the features' weights adaptively change depending on their impact on the chromosomes. This weighting system enhances the promotion of the final solution.

The first part, recognizing the effective feature, is done in the proposed mutation operator. As mentioned in the mutation phase, one of the gens (s^{th}) is randomly modified, and our algorithm investigates the effect of removing or adding this feature to the desired chromosome. If this improvement (γ) is more than Equation (5), the desired gene is marked in a vector which identifies the effective features ($W_{best}[s] = 1$).

$$(a_{best} - a_{worst})/100 \quad (5)$$

In the second phase, some of the chromosomes are randomly selected in the inverse function and then the corresponding gens with the effective features in W_{best} are reversed. Fig. 4 (d) shows an example of the inverse operator. W_{best} and the selected chromosome for performing the inverse operation are shown in Fig. 4 (d.iv). The result of performing the inverse on the selected chromosome is shown in Fig. 4 (d.v).

3.4.5. Replacement operator

Replacement operator performs the process of creating the next generation of individuals by removing or replacing some offsprings or parent individuals. After applying the exploitation and exploration operators, a new generation of chromosomes is produced. First, the fitness value of offspring, which have been reproduced so far, is calculated by the evaluation function. Then, the replacement is done based on the fitness value. In GAs, offspring usually replaces the old population using the steady-state or elitism replacement strategy and establishes a new population in the next generation. In our algorithm, the steady-state technique is used for the replacing process. The responsibility of the proposed operator is to compare each chromosome in the current population with its corresponding one in the last generation. If the accuracy of a chromosome in the current generation becomes better than its corresponding one in the last generation, the new chromosome is replaced with the old one.

3.4.6. Migration operator

The mission of the migration operator is to enable processors to exchange their best genetic materials (migration rate) at fixed intervals (migration gap). The migration occurs on every migration gap. In this paper, the migration gap is set to two generations. During the migration process, each processor selects and sends its best chromosomes to the next processor in a ring. Meanwhile, the worst chromosomes in each processor are replaced with the best ones received. This exchange happens with a fixed migration rate. In GeFeS, the migration rate is set to one chromosome.

3.4.7. Termination strategy

In general, the evolutionary process operates many iterations until the termination condition is met. Our termination criteria are a combination of the maximum number of iterations (n_{itr}) and the ceiling accuracy (a_{ceil}). Our algorithm stops when either the iteration reaches its maximum number or the ceiling accuracy is achieved. GeFeS calculates the ceiling accuracy during the execution process based on Equation (6). Based on our experimental observations, our method reaches the best convergence before this maximum number of iterations.

$$a_{ceil} = a_{wf} + (100 - a_{wf}) * \beta \quad (6)$$

where a_{wf} is the accuracy of GeFeS before performing feature selection that is evaluated, in the first step of the algorithm, by kNN and the full dataset. β is the improvement rate ($0 \leq \beta \leq 1$). GeFeS considers the improvement of the average accuracy as much as β percent of a_{wf}. In this algorithm, β is set to 0.80.

4. Experimental designs

4.1. Data description

Five different biomedical datasets, extracted from UCI Machine Learning Repository [23–27], are used to evaluate the system proposed in this study. These datasets are commonly used to evaluate machine-learning methods for feature selections and classifications in medical studies. In this study, the employed datasets can be divided into three categories based on the number of features and the number of instances presented as small, medium and large. Besides, both two-class and multi-class datasets are used to evaluate the performance of the proposed method. The most focus of this study is the detection of healthy and non-healthy patients (not the highly imbalanced data) in each disease type. Hence, most of the datasets are set to have two classes, and a binary classification scheme is applied throughout the GeFeS model.

It should be noted that for the UCI cardiac arrhythmia, unhealthy classes representing the fifteen classes of different types of heart arrhythmia are highly imbalanced. So, it is assumed that all unhealthy classes are fit to one class of unhealthy against the normal (healthy) class. Therefore, the dataset is set to have two classes.

In this work, all datasets with different ranges in feature values are normalized. Normalization is a part of data preparation for machine learning. The goal of normalization is to change the values of numeric features in the dataset to a common scale, without distorting differences in the ranges of values. GeFeS employs the min-max scaling technique as the normalization method in the preprocessing step. GeFeS utilizes the min-max scaling to rescale the value of features in the range of [0, 1]. Missing values in the datasets are replaced with the mean value during the preprocessing phases. Table 1 summarizes the details for each dataset used in this study.

4.2. Experimental setup

The proposed feature selection algorithm (GeFeS) was implemented in Visual C++ using the MPI library for parallelization. Although GeFeS works on both shared memory and message passing architectures, it was run with MPICH2 on a shared memory structure. The ring topology has been used for connections. All implementations and experiments have been performed on an Intel Core i7-4770 CPU 3.40 GHz, RAM 16.00 GB, running Windows 7 Enterprise (64-bit). Seven cores have been leveraged in the parallel implementation. To obtain the average accuracy of the classifier for all datasets, the prediction power of the model is evaluated by 10-fold cross-validation.

According to the operational process of parallel evolutionary computation algorithm, the results of GA and parallel implementation depend on parameter setting to some extent. Fine-tuning of the parameters could improve the results. The parallel GA parameters are indicated in Table 2.

5. Experimental results

In this study, we proposed a new parallel genetic algorithm for

Table 1
General cases of UCI public employed datasets.

Name	Sample size	no. of features	no. of classes
Lung	32	56	3
Dermatology	366	34	6
Arrhythmia	452	279	2
WDBC	569	30	2
Hepatitis	155	19	2

Table 2
Parameters used in GeFeS.

Parameter	Explanation	Value
$N_{proc.}$	no. of processors	7
Mg	migration gap	1 chromosome
Mr	migration rate	2 iterations
N_p	population size	100
n_{itr}	maximum no. of iterations	100
IC	crossover operator	intelligent & adaptive
P_c	crossover rate	$0.20 \leq P_c \leq 0.80$
IM	mutation operator	intelligent & adaptive
P_m	mutation rate	$0.20 \leq P_m \leq 0.80$
P_i	inverse rate	0.30
β	improvement rate	0.80
rpl	replacement	steady-state
k	no. of the neighbors in kNN	dynamic
K	no. of the folds in CV	10
N_f	no. of features in dataset	19–279
N_i	no. of instances in dataset	32–569

feature selection to improve the accuracy and reliability of kNN classifier, as a sample, by selecting relevant features, minimizing the risk of data overfitting. A series of experiments have been carried out to evaluate the effectiveness of the proposed method, GeFeS (a generalized GA-based feature selection algorithm). We used five different UCI medical datasets to test the GeFeS performance. The proposed method has been compared with several state-of-the-art studies in the area of feature selection and classification of medical datasets. These studies, which can be seen with more details in Section 2, include deep learning [28,33], neural networks [29,32], filters [37], hybrids [35,40], and wrappers [36,46] feature selection and classification methods for medical datasets. Comparisons against the state-of-the-art methods have been performed base on the reported detection accuracy from the literature on the same datasets. Besides, the comparison has been performed with three well-known traditional feature selection techniques that have been implemented in Weka 3.8, an open-source machine learning software [52]. The first algorithm is ReliefF, which evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different classes [53]. The second one is Cfs, correlation feature selection, which evaluates the worth of a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them [53]. The third algorithm is PCA, principal component algorithm, which performs a principal component analysis and transformation of the data [53].

In this work, we attempted to improve the classification performance using feature selection. The prediction power of the model is evaluated based on 10-fold cross-validation. The final performance was achieved by averaging the accuracy of all the 10-FCV. Table 3 and Fig. 5 illustrate that feature selection has enhanced further the classification accuracy.

To further compare the prediction power of GeFeS algorithm, the average accuracy and F-score of all folds have been compared to several state-of-the-art feature selection methods. The results are shown in Table 4 and Table 5. Classification accuracy is widely used as a single measure for summarizing the model performance. Precision is the number of correct positive predictions, divided by the total number of positive predictions returned by the classifier and the recall counts for the number of correctly predicted positives out of all real positives (True positives + False negative). F-score is the harmonic mean of the precision and the recall. F-score tries to capture both properties of the precision and recall into one measure for assessing the classifiers' prediction power.

Table 3
Classification average accuracy (%) of GeFeS with/without feature selection (FS).

Dataset	Original		Selected	
	no. of all features	Accuracy without FS	no. of selected features	Accuracy with FS
Lung Cancer	56	71.87	28	95.83
Dermatology	34	82.41	18	97.62
Arrhythmia	279	95.80	135	99.02
WDBC	30	92.25	16	98.51
Hepatitis	19	85.06	9	94.28

As shown in Fig. 6 and Table 6, we have computed the running time spent by GeFeS in serial and parallel implementations on all datasets for comparing of the computational time.

To explore how many features and which features were selected, we further conducted an experiment on all five datasets to investigate the detailed feature selection mechanism of the GeFeS algorithm. For the simplicity in Table 7, we show the selected features through one time run of 10-fold CV for hepatitis and dermatology datasets, as two samples. The original numbers of features of each dataset were 34 and 19, respectively. Besides, the frequency of selected features in one run 10-fold CV on the hepatitis data can be seen in Table 8.

In meta-heuristic optimization approaches, determining whether a measure quantity converged acceptably in each iteration is an important task. In this regard, Fig. 8 shows the convergence of GeFeS for ten folds on all datasets.

6. Discussion

The proposed wrapper-based feature selection method, GeFeS, has given promising results by employing a new intelligent genetic algorithm. In the classification of different diseases under various sizes of numeric two-class and multi-class datasets, GeFeS obtained accuracies of up to 99.02% and F-measure up to 98.97%. The performance of GeFeS was compared with different feature selection methods in order to demonstrate its superiority using five medical datasets extracted from the UCI machine learning repository. The experimental results on the five medical datasets show that GeFeS with the multi-objective optimization approach outperforms the other methods in terms of the average accuracy, F-measure, and the lower number of selected features. We compared the performance of GeFeS with several state-of-the-art methods whose evaluation protocols are similar to ours.

Two-class cases: [28] proposed a hybrid feature selection in a combination of a DNN, and the accuracy of 82.96% under arrhythmia data was reported. We achieved 16% more detection accuracy on healthy and unhealthy cases. We obtained 3% more accurate result than

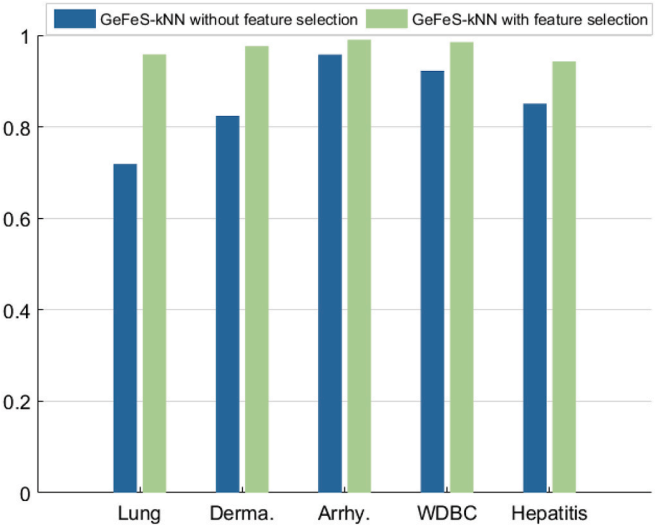


Fig. 5. Predictive average accuracy of GeFeS with & without performing FS.

Table 4
The average accuracy (%) of GeFeS vs. state-of-the-art methods.

Datasets	Methods												
	GeFeS	ReliefF	Cfs	PCA	[28]	[29]	[32]	[33]	[35]	[36]	[37]	[40]	[46]
Lung	95.83	87.50	87.50	78.12		66.71			72.2				
Dermatology	97.62	95.62	96.72	92.13			95.00	96.89	97.60	95.16	98.08		
Arrhythmia	99.02	67.66	67.92	65.05	82.96								
WDBC	98.51	93.67	95.78	83.85		95.53						92.80	97.14
Hepatitis	94.28	85.80	83.22	83.01									

Table 5
The experimental results of classification for GeFeS on two-class and multi-class datasets.

Dataset	# of instances	# of healthy instances	# of unhealthy instances	Recall	Precision	F-Measure	Accuracy (%)
Lung	32	0	23 (9, 13, 10)	0.937	0.944	0.938	95.83
Dermatology	366	0	112 (112, 61, 72, 49, 52, 20)	0.921	0.905	0.911	97.62
Arrhythmia	452	245	207	0.984	0.994	0.989	99.02
WDBC	569	357	212	0.981	0.993	0.987	98.51
Hepatitis	155	32	123	0.789	0.918	0.848	94.28

[29], which proposed a firework algorithm to train an MLP and reported the accuracy of 95.53% under the WDBC dataset [40]. presented a feature selection based on BCO and gradient boosting decision tree and reported the accuracy of 92.80% under WDBC data while we achieved 5.71% higher accuracy. We gained 1.37% larger accuracy than [46], which proposed a feature selection method based on a hybrid GA with embedded devised local search operations and reported the accuracy of 97.14% under the WDBC dataset.

Multi-class cases: [32] presented a feature selection method based on the maximum entropy model and an ANN and reported the accuracy of 95.00% under the dermatology data. On the other hand, we obtained 2.62% more detection accuracy on six classes of diseases. We obtained 2.46% more accurate result than [36], which proposed a feature selection based on a modified ACO with a GA and reported the accuracy of 95.16% under the dermatology dataset [33]. proposed a collaborative DNN with a neighbor-based and latent feature-based CF and reported the accuracy of 96.89% under dermatology data. We achieved 0.73% more accurate detection of different dermatology diseases. GeFeS reached the average accuracy of 97.62% under the dermatology dataset while a combination of ReliefF with differential evolution method under the same dataset [35] reported the best accuracy of 97.60% [37].

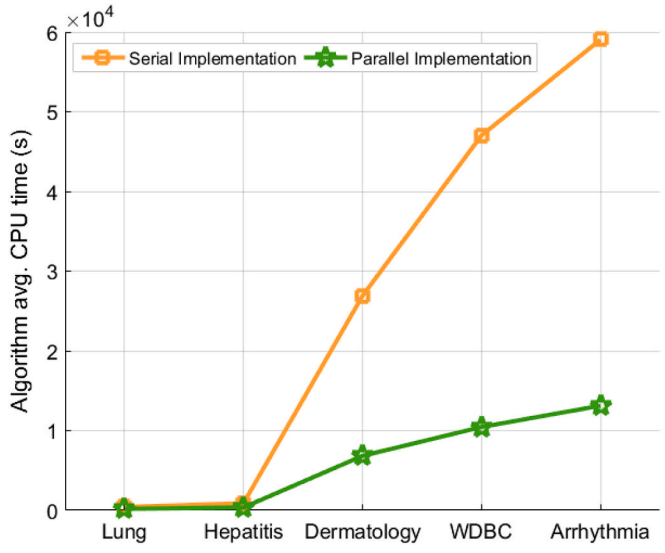


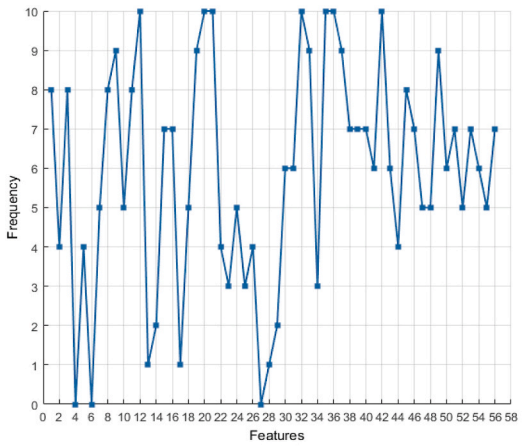
Fig. 6. The average CPU time of the proposed serial and parallel implementations on each dataset.

presented a feature selection method based on FMPE and reported the best accuracy of 98.08%. This method reduced the number of features from 34 to 29 under the dermatology dataset. We achieved an average accuracy of 97.62% while reducing the number of features from 34 to 18 under the same dataset.

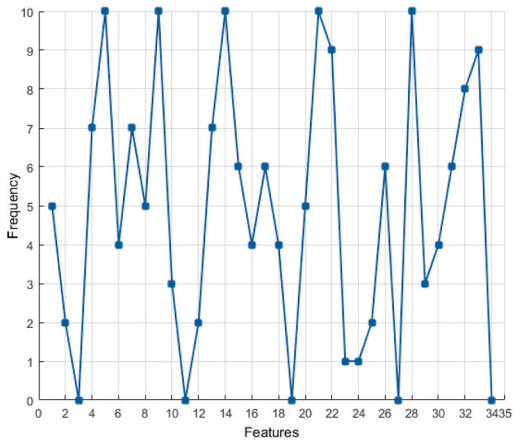
As can be seen in Table 4, in all five datasets, the accuracy has improved dramatically using GeFeS with the feature selection capability. The average classification accuracies in GeFeS with feature selection compared with GeFeS without feature selection capability have been improved by 23.96%, 14.61%, 3.22%, 6.26%, and 9.22% under lung cancer, dermatology, arrhythmia, WDBC, and hepatitis, while the number of features has been reduced to 28, 18, 135, 16, and 9, respectively. The results reported in Table 4 demonstrated that our feature selection method is more robust than the others with regard to classification average accuracy and the size of applied datasets. As can be seen in Table 5, we have also calculated the recall, precision and F-measure to further investigate the possibility of class unbalance and its effect on accuracy. As shown in Fig. 6 and Table 6, parallel GeFeS needs much less CPU time when compared to serial GeFeS. Moreover, as mentioned before, the multi-population strategy in parallel implementation has increased the accuracy of the algorithm while reducing the CPU consumption time. As shown in Fig. 7, not all features are selected for classification after the feature selection on all datasets. For Hepatitis dataset, the average number of selected features by GeFeS is 8, and the most important features are F_5, F_6, F_8, and F_{18} which can be found in the frequency of selected features of 10-fold CV as shown in Table 8. For Dermatology dataset, the average number of selected features by GeFeS is 18, and the most important features are F_5, F_8, F_9, F_{14}, F_{21}, F_{22}, F_{28}, and F_{33}, which can be found in the frequency of selected features of 10-fold CV as shown in Fig. 7 (b). This figure guides the health applications designers to find the most essential features in each dataset. The curves of Fig. 8 thoroughly show that the convergence of GeFeS to the optimum solutions, in terms of accuracy of *k*NN, is ascending and robust for all

Table 6
Comparison of serial and parallel implementations of GeFeS in terms of average accuracy (%) and consumption time (s) on all datasets.

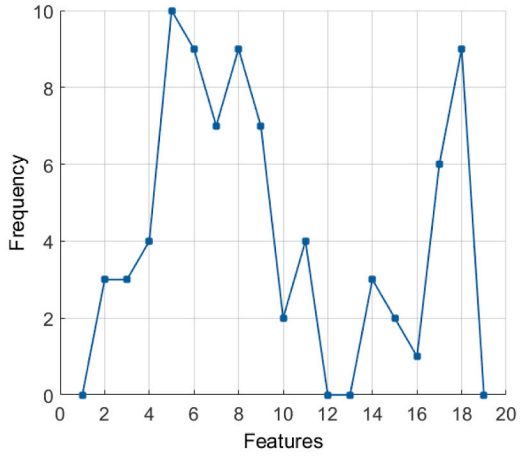
Dataset	Serial		Parallel	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
Lung	93.78	420	95.83	181.41
Dermatology	95.96	26898	97.62	6830
Arrhythmia	98.00	59049	99.02	13122
WDBC	96.05	47029	98.51	10451
Hepatitis	91.50	880	94.28	358



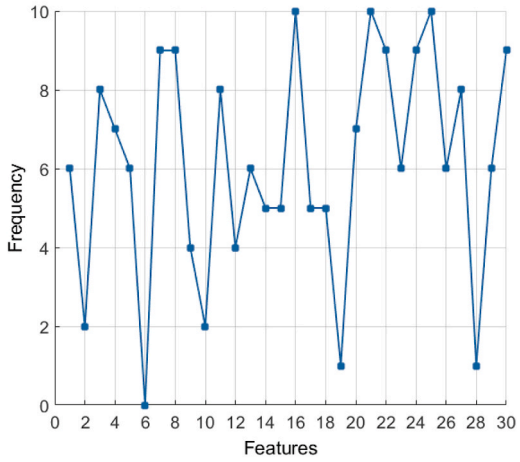
(a) Lung cancer



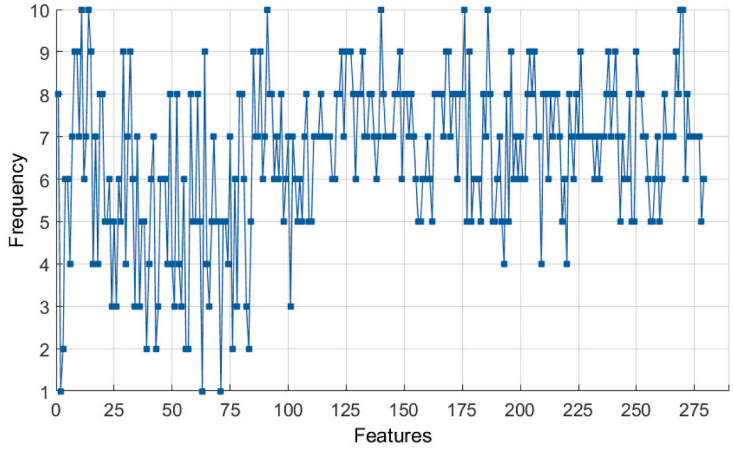
(b) Dermatology



(c) Hepatitis

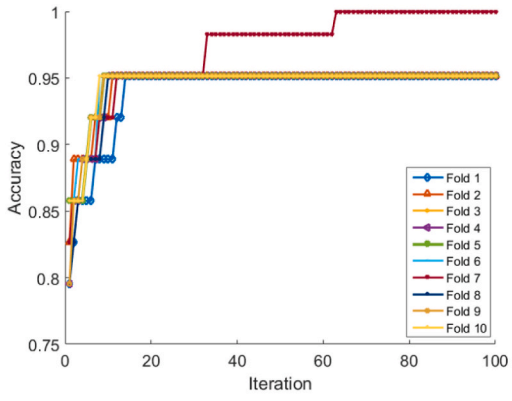


(d) WDBC

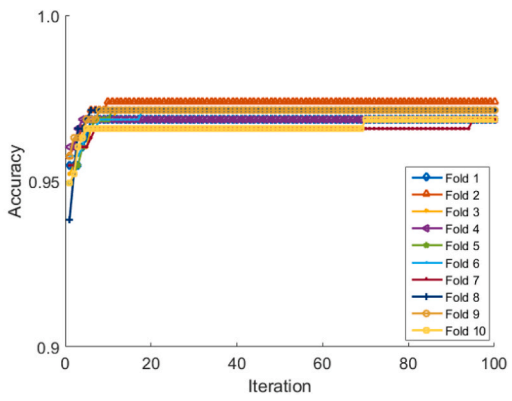


(e) Arrhythmia

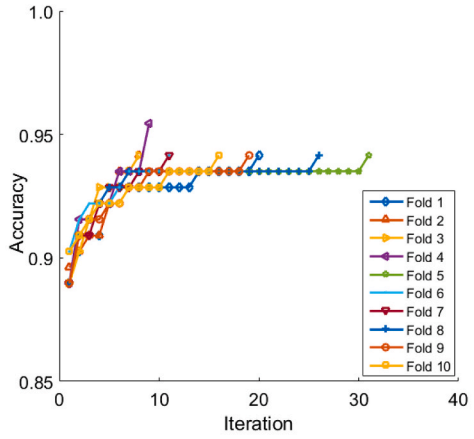
Fig. 7. The frequency of the selected features in one run 10-fold CV process on the five datasets.



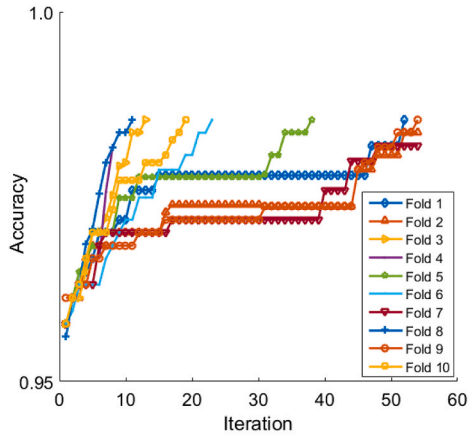
(a) Lung cancer



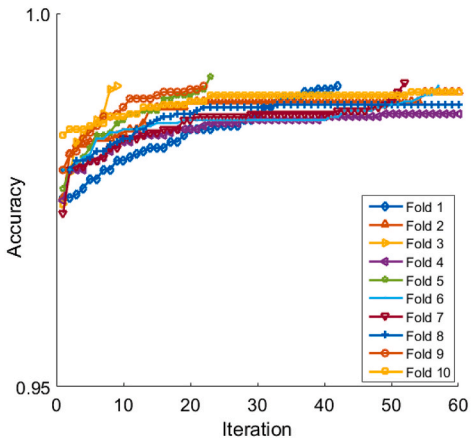
(b) Dermatology



(c) Hepatitis



(d) WDBC



(e) Arrhythmia

Fig. 8. Convergence of GeFeS on the five datasets.

Table 7
Feature selected per folds for Hepatitis and Dermatology datasets by the GeFeS method.

Fold	Hepatitis	Dermatology
#1	$F_3F_5F_6F_9F_{10}F_{14}F_{17}F_{18}$	$F_2F_4F_5F_6F_7F_9F_{12}F_{13}F_{14}F_{15}F_{16}F_{17}F_{18}F_{21}F_{22}F_{23}F_{26}F_{27}F_{28}F_{30}F_{33}$
#2	$F_2F_4F_5F_6F_7F_8F_9F_{11}F_{17}F_{18}$	$F_4F_5F_9F_{13}F_{14}F_{15}F_{20}F_{21}F_{22}F_{25}F_{28}F_{29}F_{30}F_{31}F_{33}$
#3	$F_2F_4F_5F_6F_7F_8F_9F_{11}F_{17}F_{18}$	$F_1F_2F_4F_5F_6F_8F_9F_{10}F_{13}F_{14}F_{17}F_{20}F_{21}F_{22}F_{26}F_{28}F_{31}F_{32}F_{33}$
#4	$F_3F_4F_5F_6F_7F_8F_{11}F_{14}F_{16}$	$F_2F_5F_7F_8F_9F_{14}F_{15}F_{16}F_{18}F_{20}F_{21}F_{22}F_{27}F_{28}F_{32}F_{33}$
#5	$F_5F_6F_7F_8F_{15}F_{18}$	$F_1F_2F_4F_5F_7F_9F_{10}F_{13}F_{14}F_{15}F_{17}F_{21}F_{22}F_{26}F_{28}F_{29}F_{31}F_{32}$
#6	$F_3F_5F_8F_9F_{10}F_{14}F_{17}F_{18}$	$F_1F_2F_4F_5F_8F_9F_{13}F_{14}F_{17}F_{21}F_{22}F_{26}F_{27}F_{28}F_{29}F_{31}F_{32}F_{33}$
#7	$F_2F_4F_5F_6F_7F_8F_9F_{11}F_{17}F_{18}$	$F_2F_5F_6F_7F_9F_{10}F_{14}F_{15}F_{16}F_{18}F_{20}F_{21}F_{25}F_{28}F_{30}F_{32}F_{33}$
#8	$F_5F_6F_7F_8F_9F_{18}$	$F_1F_2F_4F_5F_7F_8F_9F_{13}F_{14}F_{17}F_{21}F_{22}F_{26}F_{27}F_{28}F_{31}F_{32}F_{33}$
#9	$F_5F_6F_8F_9F_{17}F_{18}$	$F_1F_2F_4F_5F_7F_8F_9F_{13}F_{14}F_{17}F_{20}F_{21}F_{22}F_{26}F_{28}F_{30}F_{31}F_{32}F_{33}$
#10	$F_5F_6F_7F_8F_{15}F_{18}$	$F_2F_5F_6F_7F_9F_{12}F_{14}F_{15}F_{16}F_{18}F_{21}F_{22}F_{24}F_{28}F_{32}F_{33}$

Table 8
Frequency of selected features in one run 10-fold CV on the Hepatitis dataset.

Features	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}
Frequencies	0	3	3	4	10	9	7	9	7	2	4	0	0	3	2	1	6	9	0

folds, or different search spaces. It should be mentioned that the differences in the ending points of the curves are deduced from the second termination criteria of GeFeS. Our algorithm stops when either the maximum iteration number is reached or the ceiling accuracy (a_{ceil}) is achieved.

The obtained results show that the performance of GeFeS is comparable with state-of-the-art articles. The obtained results have demonstrated that the proposed method is effective in feature selection, thereby in diseases detection on the five UCI medical datasets. GeFeS thoroughly handles the small to large, two-class to multi-class, and high-dimensional datasets that we have utilized in this paper.

7. Conclusion and future work

Selecting relevant and removing redundant features in high-dimensional biomedical datasets is essential for increasing the performance of machine-learning algorithms in terms of raising the prediction accuracy of detection and reducing the time of building the algorithm. In this work, we proposed GeFeS, a wrapper-based feature selection and parameter optimization method for *k*NN based on a novel parallel genetic algorithm to increase the accuracy, reliability, and generalization of the classifier. It is worth mentioning that *k*NN is an embedded component in the proposed GA that can be easily replaced by other classifiers. The main novelties of this method, in terms of genetic algorithm, lie in the adopted weighting operator (a new operator for genetic algorithm, called inverse), intelligent crossover, and mutation operators. In addition, we utilized the nested cross-validation in chromosome evaluation step to select the best chromosome, suggest a good generalization capability to unseen data, and minimize the risk of data overfitting. To evaluate the performance prediction of the proposed algorithm, we used a K-fold cross-validation strategy on top of GeFeS. Moreover, the proposed method was implemented in an efficient parallel environment, which further enhanced the performance of the GeFeS in terms of accuracy and computational time.

The aim of the improvements and novelties was at maximizing the generalization capability and the accuracy of the proposed algorithm under both two-class and multi-class data from small-scale to large-scale while reducing the number of features and the consumption time. These objectives led the algorithm to make more reliable diagnosis decisions. Based on our experiential analysis, it can be safely concluded that the GeFeS method can serve as a promising tool for parameter optimization and feature selection in *k*NN by avoiding overfitting of the classifier model. The experiments on a number of datasets with different sizes proved that GeFeS achieved remarkably good results when faces with large, small, high-dimensional, two-class, and multi-class datasets.

As future works, we plan to develop GeFeS for unsupervised problems, which is optimized with more hyper-parameters and highly imbalanced datasets. Furthermore, we plan to make our method more efficient with a hybrid parallel implementation. Finally, we plan to develop an open-source tool of GeFeS for different classifiers and clusters.

Declaration of competing interest

We the under signed declare that this manuscript entitled “GeFeS: A Generalized Wrapper Feature Selection Approach for Optimizing Classification Performance” is original, has not been published before and is not currently being considered for publication elsewhere.

Acknowledgments

This work is supported by the Department of Future Technologies and MATTI foundation of University of Turku, Finland, and the K. Albin Johanssons foundation. The experimental datasets used in this paper are achieved from the UCI Machine Learning Repository. (Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: the University of California, School of Information and Computer Science).

References

[1] A. Yang, et al., Optimum surface roughness prediction for titanium alloy by adopting response surface methodology, Results Phys 7 (2017) 1046–1050.
[2] K. Cui, W. Yang, H. Gou, Experimental research and finite element analysis on the dynamic characteristics of concrete steel bridges with multi-cracks, J. Vibroeng. 19 (2017) 4198–4209.
[3] J.P. Arrais, P. Lopes, J.L. Oliveira, Challenges Storing and Representing Biomedical Data, Information quality in e-Health, 2011, pp. 53–62.
[4] M.I. Razzak, S. Naz, A. Zaib, Deep learning for medical image processing: overview, challenges and the future, calssification in BioApps, Lecture Notes in Computational Vision and Biomechanics 26 (2017) 323–350.
[5] Y. Jiang, N. Bosch, R.S. Baker, L. Paquette, J. Ocumpaugh, J.M.A.L. Andres, A. L. moore, G. Biswas, Expert Feature-Engineering vs. Deep Neural Networks: Which Is Better for Sensor-free Affect Detection? 10947 Springer International Publishing AG, part of Springer Nature, 2018, pp. 198–211.
[6] D. Heaven, Deep trouble for deep learning, Nature 574 (2019) 163–166. Springer.
[7] X. He, T.S. Chua, Neural factorization machines for sparse predictive analytics, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 355–364.
[8] J.-Q. Li, F.R. Yu, G. Deng, C. Luo, Z. Ming, Q. Yan, Industrial Internet: a survey on the enabling technologies, applications, and challenges, IEEE Commun. Surveys Tuts. 19 (2017) 1504–1526.

[9] H.-D. Zhu, Y. Zhong, New feature selection algorithm based on multiple Heuristics, J. Comput. Appl. 29 (2009) 849–851.
[10] D. Koller, M. Sahami, Toward Optimal Feature Selection, ilpubs.stanford.edu, 1996, pp. 1–14.
[11] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer-Verlag Berlin Heidelberg, 2010.
[12] I.A. Gheyas, L.S. Smith, Feature subset selection in largedimensionality domains, Pattern Recogn. 43 (2010) 5–13.
[13] L.F. Chen, C.T. Su, K.H. Chen, P.C. Wang, Particle swarmoptimization for feature selection with application inobstructive sleep apnea diagnosis, International Journal ofNeural Computing and Applications 21 (8) (2012) 2087–2096.
[14] A. Ziarati, A multilevel evolutionary algorithm for optimizing numerical functions, IJIEC 2 (2011).
[15] N. Milickovic, M. Lahanas, D. Baltas, N. Zamboglou, Comparison of Evolutionary and Deterministic Multiobjective Algorithms for Dose Optimization in Brachytherapy, Chapter Evolutionary Multi-Criterion Optimization, 1993, pp. 167–180, 2001.
[16] D. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, Addison -Wesley, Reading, MA, 1989.
[17] P.G. Espejo, S. Ventura, F. Herrera, A survey on the application of genetic programming to classification, IEEE Trans. Syst. Man Cybern. C Appl. Rev. 40 (2010) 121–144.
[18] A. Majd, G. Sahebi, A Survey on Parallel Evolutionary computing and introduce four general frameworks to parallelize all EC algorithms and create new operation for migration, Journal of Information and computing Science 9 (2014) 97–105.
[19] E. Alba, F. Luna, A.J. Nebro, J.M. Troya, Parallel heterogeneous genetic algorithms for continuous optimization, Parallel Computing 30 (2004) 699–719. ELSEVIER.
[20] R. Guha, M. Ghosh, S. Kapri, Sh Mutsuddi, V. Bhateja, R. Sarkar, Deluge Based Genetic Algorithm for Feature Selection, in: Evolutionary Intelligence, vol. 2, Springer-Verlag GmbH Germany, part of Springer Nature, 2019.
[21] A. Airola, S. Pihlasalo, I.M. Perez, T. Salakoski, T. Pahikkala, Assessment of metal ion concentration in water with structured feature selection Pekka Naula, Journal of Chemosphere 185 (2017) 1063–1071. Elsevier.
[22] Gavin C. Cawley, Nicola L.C. Talbot, On over-fitting in model selection and subsequent selection bias in performance evaluation, J. Mach. Learn. Res. 11 (2010) 2079–2107.
[23] Diagnostic Wisconsin Breast Cancer Dataset, W.H. Wolberg, W. NickStreet, O. L. Mangasarian, available in, https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+, 1995 (Diagnostic).
[24] Arrhythmia dataset, H. Altay, B. Acar, H. Muderrisoglu, Available in, https://archive.ics.uci.edu/ml/datasets/Arrhythmia, 1998.
[25] Dermatology dataset (N. Ilter and H. Altay (1998)). Available in: http://archive.ics.uci.edu/ml/machine-learning-databases/dermatology/dermatology.data.
[26] Hepatitis dataset, available in, https://archive.ics.uci.edu/ml/datasets/hepatitis, 1988.
[27] Lung cancer dataset, available in UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. http://cml.ics.uci.edu.
[28] S. Sh Xu, M. Mak, Ch Cheung, Deep neural networks versus support vector machines for ECG arrhythmia classification, Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW) (2017) 127–132.
[29] R.K. Dutta, N.K. Karmakar, T. Si, Artificial neural network training using fireworks algorithm in medical data mining, Int. J. Comput. Appl. 137 (2016).
[30] A. Saygilia, Classification and diagnostic prediction of breast cancers via different classifiers, International scientific and vocational journal (2018) 48–56.
[31] Sh Jadhav, S.L. Nalbalwar, A.A. Ghatol, ECG arrhythmia classification using modular neural network model, in: IEEE EMBS Conference on Biomedical Engineering & Sciences, 2010, pp. 62–66.
[32] V.B. Kumar, S.S. Kumar, V. Saboo, Dermatological disease detection using image processing and machine learning, in: Third International Conference on Artificial Intelligence and Pattern Recognition, AIPR, 2016, ISBN 978-1-4673-9187-0, pp. 88–93. IEEE.
[33] M. Sun, T. Min, T. Zang, Y. Wang, CDL4CDRP: a collaborative deep learning approach for clinical decision and risk prediction, Journal of Processes 7 (5) (2019), 265, https://doi.org/10.3390/pr7050265.
[34] R. Panthonga, A. Srivihokb, Wrapper feature subset selection for dimension reduction based on ensemble learning algorithm, in: Proceeding of the Third Information Systems International Conference, 2015, pp. 162–169.
[35] M.N.Sh Zainudin, M.N. Sulaiman, N. Mustapha, Th Perumal, A. Sh A. Nazri, R. Mohamed, S.A. Manaf, Feature selection optimization using hybrid relief-f with self-adaptive differential evolution, International Journal of Intelligent Engineering & Systems 10 (2017) 21–29.
[36] Y. Wan, M. Wang, Z. Ye, X. Lai, A feature selection method based on modified binary coded ant colony optimization algorithm, Journal of Applied Soft Computing 49 (2016) 248–258.
[37] L. Zhao, X. Dong, An industrial internet of things feature selection method based on potential entropy evaluation criteria, Journal of IEEE Access 6 (2018) 4608–4617.
[38] Sh Gu, R. Cheng, Y. Jin, Feature selection for high-dimensional classification using a competitive swarm optimizer, Journal of Soft Comput (2018) 811–822.
[39] La T. Vinh, S. Lee, Y.T. Park, B.J. d'Auriol, A novel feature selection method based on normalized mutual information, Journal of Applied Intelligence 37 (2012) 100–120.
[40] H. Rao, X. Shi, A.K. Rodrigue, J. Feng, Y. Xia, M. Elhoseny, X. Yuan, L. Gu, Feature selection based on artificial bee colony and gradient boosting decision tree, Journal of Applied Soft Computing 74 (2019) 634–642.
[41] C.K. Lim, C.S. Chan, A weighted inference engine based on interval-valued fuzzy relational theory, Expert Syst. Appl. 42 (2015) 3410–3419.
[42] B. Zheng, S.W. Yoon, S.S. Lam, Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms, Expert Syst. Appl. 41 (2014) 1476–1482.
[43] H.L. Chen, B. Yang, S.J. Wang, D.Y. Liu, H.Z. Li, B.L. Wen, Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy, Journal of Applied Mathematics and Computation 239 (2014) 180–197.
[44] J.A. Saez, J. Derrac, J. Luengo, F. Herrera, Statistical computation of feature weighting schemes through data estimation for nearest neighbor classifiers, Journal of Pattern Recogn 47 (12) (2014) 3941–3948.
[45] I.S. Oh, J.S. Lee, B.R. Moon, Hybrid genetic algorithms for feature selection, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 1424–1437.
[46] M. Islam, H. Iqbal, R. Haque, K. Hasan, Prediction of breast cancer using support vector machine and k-nearest neighbors, IEEE Region 10 Humanitarian Technology Conference, 2017, pp. 226–229.
[47] M. Gen, R. Cheng, L. Lin, Network Models and Optimization: Multiobjective Genetic Algorithm Approach, Decision Engineering, Springer, 2008.
[48] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (2009).
[49] M.R. Berthold, C. Borgelt, F. Höppner, F. Klawonn, Guide to Intelligent Data Analysis, vol. 9, Springer-Verlag London Limited, 2010, p. 26.
[50] D.E. Goldberg, J.H. Holland, Genetic algorithms and MachineLearning, Journal of Machine Learning 3 (1988) 95–99.
[51] M. Črepinšek, Sh H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Surv. 45 (2013).
[52] Weka 3: Data Mining with Open Source Machine Learning Software in Java, Weka 3, data mining with open source machine learning software in java, Retrieved 15 Mar 2012, from, http://www.cs.waikato.ac.nz/*ml/weka/, 2012.
[53] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explorations 11 (2009) 10–18.

Golnaz Sahebi received her bachelor's degree in Computer Science at the University of Shahid Bahonar, Kerman, Iran, in 2005, and her master's degree in Computer Science and Artificial intelligence at the University of Tabriz, Iran, in 2014. She is currently a last year Ph.D. candidate at the Department of Future Technologies, University of Turku, Finland. Her current research interests focus on optimization of machine learning algorithms and lightweight computing for power-hungry devices. She is the author of more than 16 peer-reviewed publications.

Parisa Movahedi received her B·Sc. (Tech.) degree in Computer Software Engineering from the Department of Electrical and Computer Engineering, Tehran Azad University, Tehran, Iran in 2010, and M.Sc. (Tech.) degree in Information Technology from the Department of Information Technology and Communication Systems, University of Turku, Finland in 2014. She is currently working towards her Ph.D. degree at the Department of Future Technologies, University of Turku, Finland. Her current research interests include algorithms and applications of machine learning and data analysis in medical field.

Masoumeh (Azin) Ebrahimi received a PhD degree with honors from University of Turku, Finland in 2013 and MBA in 2015. She is currently an Adjunct professor at University of Turku, Finland and a senior researcher (docent) at KTH Royal Institute of Technology, Sweden. Her scientific work contains more than 100 publications including journal articles, conference papers, book chapters, edited proceedings, and edited special issue of journal. The majority of work has been performed on on-chip interconnection networks, fault-tolerant methods, and deep learning accelerators. She actively acts as a guest editor, organizer, and program chair in different workshops and conferences.

Tapio Pahikkala currently holds an associate professorship of machinelearning with the University of Turku, Finland, from which he also received his doctoral degree in 2008. He has authored more than 140 peer-reviewed scientific articles and participated in the winning teams of several international scientific competitions/challenges. He has led many research projects, supervised eight doctoral theses, held several positions of trust in academia and served in the program committees of numerous international conferences. His current research interests include theory and algorithmics of machine learning, data analysis, andcomputational intelligence, as well as their applications on various different fields.

Juha Plosila received the Ph.D. degree in electronics and communication technology from the University of Turku (UTU), Finland, in 1999, where he is currently a Professor (Full) of autonomous systems and robotics with the Department of Future Technologies. He is also the Head of the EIT Digital Master Programme in Embedded Systems at the EIT Digital Master School (European Institute of Innovation and Technology) and represents UTU in the Node Strategy Committee of the EIT Digital Helsinki/Finland node. He has a strong research background in adaptive multi-processing systems and platforms and their design.

Hannu Tenhunen received the Diploma degree from the Helsinki University of Technology, Finland, in 1982, and the Ph.D. degree from Cornell University, Ithaca, NY, USA, in 1986. He received an Honorary Doctorate from Tallinn Technical University. He was a Full Professor, an Invited Professor, or a Visiting Honorary Professor in Finland (TUT, UTU), Sweden (KTH), USA (Cornel U), France (INPG), China (Fudan and Beijing Jiatong Universities), and Hong Kong (The Chinese University of Hong Kong). He is currently a Professor with the Electronic Systems Laboratory, Royal Institute of Technology (KTH), and a Professor with the University of Turku (UTU). He has contributed over 850 international publications with an H-index 41.