A Reliable Weighted Feature Selection for Auto Medical Diagnosis

Golnaz Sahebi
Department of Future Technologies
University of Turku, Finland
golnaz.sahebi@utu.fi

Amin Majd
Department of Information Technology
Abo Akademi University, Finland
amin.majd@abo.fi

Masoumeh Ebrahimi
KTH Royal Institute of Technology, Sweden;
University of Turku, Finland
masebr@kth.se

Juha Plosila
Department of Future Technologies
University of Turku, Finland
juplos@utu.fi

Hannu Tenhunen
KTH Royal Institute of Technology, Sweden;
University of Turku, Finland
hannu@kth.se

Abstract*— Feature selection is a key step in data analysis. However, most of the existing feature selection techniques are serial and inefficient to be applied to massive data sets. We propose a feature selection method based on a multi-population weighted intelligent genetic algorithm to enhance the reliability of diagnoses in e-Health applications. The proposed approach, called PIGAS, utilizes a weighted intelligent genetic algorithm to select a proper subset of features that leads to a high classification accuracy. In addition, PIGAS takes advantage of multi-population implementation to further enhance accuracy. To evaluate the subsets of the selected features, the KNN classifier is utilized and assessed on UCI Arrhythmia dataset. To guarantee valid results, leave-one-out validation technique is employed. The experimental results show that the proposed approach outperforms other methods in terms of accuracy and efficiency. The results of the 16-class classification problem indicate an increase in the overall accuracy when using the optimal feature subset. Accuracy achieved being 99.70% indicating the potential of the algorithm to be utilized in a practical auto-diagnosis system. This accuracy was obtained using only half of features, as against an accuracy of 66.76% using all the features.

***Keywords*—Data Analysis; Feature Selection; K-Nearest Neighbor Classification; Optimization; Parallel Genetic Algorithm; E-Health.**

I. Introduction

Heart and blood vessel diseases (Cardiovascular Diseases - CVDs) are the first cause of death in the world. Potential life threatening conditions like heart failure can be successfully avoided if arrhythmias are detected at early phases. A most valuable diagnostic means that enhances the detection of CVDs is electrocardiogram (ECG), providing a successor representation of cardiac activity [1]. In recent years, one of the most significant innovations in early detection of diseases is wearable devices, which aim at providing real-time feedback information about the health condition of a person. Besides all their advantages, wearable systems face a number of challenges to become a reality. The most important hurdle is that their processors and architectures require a large amount of energy, demanding sizable batteries. This creates challenges for reducing the size of wearable devices. While minimization is done, another challenge arises that is the reliability of decision making. The detection accuracy depends on the data analysis process. From this perspective, data analysis and machine learning algorithms play an important role [2].

The process of knowledge discovery in databases (KDD) or data analysis involves some steps, such as dataset selection, data understanding, data preparation, data analysis, result interpretation, and result evaluation [3]. An important phase in data preparation, which is one of the significant issues in the construction of classification model, is feature selection. Feature selection can be determined as a process of choosing a minimum subset of features (N_{FS}) from the original set of features (N) so that the feature space is optimally reduced while the classification accuracy remains relatively the same [4].

Two general categories to solve the feature selection problem are filter and wrapper. In the filter approach, features are selected by statistical properties. By applying the filter approach, features can be quickly selected, but the performance of the learning models is not usually as high as that of the wrapper method as the selected feature may not be the best possible ones [5]. The wrapper technique, on the other hand, employs optimization algorithms in the learning machine techniques to find optimal subset of features. This utilization allows the use of standard optimization methods with the learning machine techniques. The wrapper approach is considered in this paper.

To solve the optimization problem, there are different methods such as deterministic solutions, heuristic searches, and meta-heuristic searches [6]. In large scale datasets, the meta-heuristic approaches are more efficient regarding the NP-complete aspect of the feature selection problem [7]. Evolutionary algorithms (EAs) are a well-known class of meta-heuristic searches [6]. A dominant advantage of EAs for feature selection problems, compared with deterministic algorithms, is their capability to escape from local optima that often encounters in feature selection problems [8]. A popular group of EAs are genetic algorithms (GAs). They are population-based search techniques, which mimic the process of natural selection and evolution. A GA is started with initializing a population and then running frequent operations such as selection, crossover, mutation, and replacement. All operations of a GA are repeated until reaching a competent result or a certain iteration. [9].

Although EAs are successful in solving various problems, there are some disadvantages associated with them in dealing with large search spaces [10], [11]. In these cases, it is possible for algorithms to converge to local optima. This problem can be mitigated if the initial population is increased, which is not feasible with a single processor. Utilizing more than one processor (parallelizing EAs) to enable an enormous diversity of population is a key point in feature selection for critical detection systems. Parallelizing EAs can improve the result quality and timing overhead [10], [11]. Among the parallelizing approaches for EAs, Multi-population techniques are useful for GAs when there are multiple processors with several memory units, by providing a larger population diversity to improve the accuracy of results [10]. In a multi-population method, there is a set of processors, such that each processor hosts an independent population and independently runs a serial GA on this population. One of the key features of this approach is the migration operation. After several iterations, some of the best chromosomes are selected by each processor and are then sent to the other processors. This operator shares the best solution of each processor with the others, enabling discovery of the best solution in lower iterations while providing a higher accuracy [10].

In this work, a parallel weighted intelligent genetic algorithm is proposed to solve the feature selection problem (Fig. 1). The KNN classifier is utilized to evaluate subsets of the selected features.

The proposed method, called PIGAS, is evaluated on cardiovascular diseases. The weighting is performed for increasing the priority of some dominant features. Since feature selection algorithms are generally run offline, especially the proposed algorithm is applied in this case, the accuracy is more important than the speed. The main contributions of this work are as follows. 1) We could obtain a high accurate detection of multi-class arrhythmia diseases based on KNN classification using only half of features 2) an intelligent crossover and mutation operations are presented in order to enable the algorithm to escape from potential local optima. 3) The multi-population strategy is utilized to improve the classification accuracy. 4) Features are weighted combining the human knowledge and auto weighting techniques. This offers a better accuracy and relevance for medical applications.

The rest of the paper is organized as follows: Section II covers the state of the arts in the area. Section III presents the proposed multi-population implementation of a weighted feature selection based on genetic algorithm (PIGAS). Section IV evaluates the proposed method and presents the experimental results. Finally, Section V concludes the paper.

II. Related Work

So far, lots of papers have presented the wrapper-based feature selection approach to reduce the dimensionality of datasets and improve the accuracy of classifiers. They have some differences in three used materials: classifiers, datasets, and selection methods.

Some hybrid GA-based feature selection techniques have been presented in [15], [34]. A parallel GA has been used for feature selection problem in [16]. A feature selection algorithm based on heuristic search technique has been utilized in [4]. Some simple GAs have been used for feature selection problem in [17], [35]. A multi-objective evolutionary algorithm has been employed for feature selection approach in [18]. A binary GA for feature selection has been used for dimensionality reduction to enhance the performance of classification in [36]. A hybrid GA has been used in [37] for feature and instance selection concurrently. A feature selection method based on normalized mutual information wrapped on a KNN classifier has been presented in [19]. A wrapper-filter approach has been utilized in [38] to remove irrelevant features for improving classification accuracy. A wrapper approach has been used in [20] to predict more accurately the presence of cardiovascular disease with reduced number of features. Moreover, feature selection is widely utilized to discover the best informative subset of tests in a disease diagnosis similar to our approach in this paper. For instance, a GA-based feature selection method has been proposed in [21] for detection of abnormal ECG recording. A feature selection using GA has been utilized in [22] for coronary artery diseases. A GA-based feature selection technique has been proposed in [23] on cardiac arrhythmia dataset. A GA-based feature selection has been employed in [24] to improve the classification accuracy for detection of heart diseases. Some different methods for feature selection have been used in [39] in order to optimize skin tumor diagnosis. Among all of the solutions in the mentioned paper, GA has obtained the best results. A GA-based feature selection has been employed in [40] to find the best patterns and features to recognize breast cancer. A GA-based feature selection has been utilized in [41] for detect of arrhythmia. A system for diagnosing the risk level for heart disease by applying fuzzy rules is proposed in [43].

III. Prpposed Work

In this section, we propose a feature selection technique applied on the UCI Arrhythmia database, it is worth mentioning that this algorithm can be applied and tuned on other datasets. The main objective of the proposed selection method is maximizing the accuracy of the KNN classification $(i.e., \alpha)$ while reducing the number of features. We design a weighted intelligent genetic algorithm (multi-population implementation) within the wrapper framework to solve the feature selection problem. The flowchart and algorithm are shown in Fig. 1 and Algorithm 1. In this approach, a KNN classifier is utilized to evaluate subsets of the selected features. Different phases of the algorithm are described in the following subsections.

A. The Feature Weighting Techniques

In medical applications, some features are known to be necessary for a detection of a disease that are based on clinical observations. However, if only an automatic feature selection method is applied, some of those features might be wrongly removed, e.g., due to the lack of sufficient data. For this purpose, we applied three feature weighting techniques, without any specific order, to keep the known clinically important features and those selected by the machine learning methods. The three feature selection approaches are as parallel coordinate plots, intelligent GA (a meta-heuristic search), and physiologist knowledge. Parallel coordinates plot (PCP) is used to find the features that identify useful predictors for separating classes [25]. Intelligent GA is applied to find the features that are appeared in at least 20 times running the GA. This subset of features is considered as a part of the main initial population in GA. Note that the intelligent GA is the same as the one we apply in the proposed method and is described in Section III.B To keep the clinically important features, the system weights to ECG features that are clinically known to be important such as P-wave, QRS- complex, and T-wave [26] . These features can be read from a file (the first phase of Fig. 1), and there is a numeric weight associated with each feature. This file can be set for other datasets.

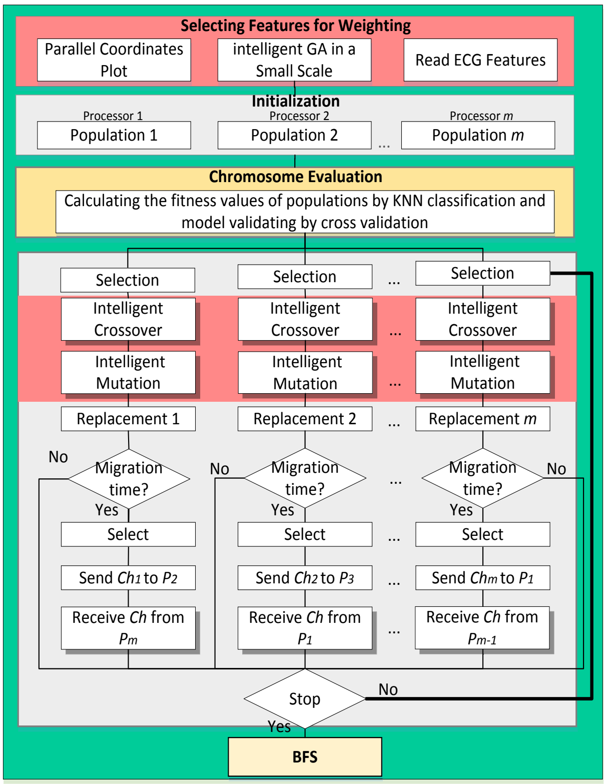

Fig. 1. Flowchart of the proposed work (PIGAS)

B. *The Proposed Parallel Intelligent Genetic Algorithm*

The GA is started with initializing a population and then running frequent operations such as selection, crossover, mutation, and replacement. All operations of a GA are repeated until reaching a competent result or a certain iteration. Several phases of the proposed parallel intelligent GA are described as follows.

1) *Initial Population*

In the genetic algorithm, the initial population consists of some individuals (chromosomes), each of them carrying a probable solution of the problem and is composed of some genes. Each gene represents an attribute of the intended individual. The key point of evolutionary algorithms is the formation and determination of these attributes. The proposed GA operates on binary search space as the chromosomes are bit strings. For binary chromosome 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 chromosomal evaluation. The gene width equals to 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 gene width. Basically each bit randomly takes the value zero or one while the bits associated with the important features takes the value one with a higher probability.

```
Input: The Arrhythmia Data Set
Output: Maximum accuracy of KNN while reducing the number of features
Processor $P_i$: $1 \le i \le Number\ of\ Processors$
1. for each processor do
2.    Run Weighting Function for increasing the priority of some dominant features.
3.    Generate initial population based on the weights
4.    Evaluate initial population by applying the KNN function.
5.    Select some chromosomes from the initial population by the Tournament operation.
6.    Run intelligent Crossover operator
7.    Run intelligent Mutation operator.
8.    Evaluate intermediate population by calling KNN function
9.    Run Steady-State replacement operator and replace the intermediate population on the initial population.
10.   If ( #Iterations % Migration Gap == 0)
11.       Select the best and the worst chromosomes
12.       Send the best chromosome to $P_{i+1}$
13.       Receive the worst chromosome from $P_{i-1}$
14.   If ( termination condition )
15.       go to step 18
16.   else
17.       go to step 4
18.   End
```

Algorithm 1. Algorithm of the proposed work (PIGAS)

2) *Chromosome Evaluation*

In this step, we employ the KNN-based fitness function to evaluate the chromosomes. Since accuracy is more important than the training time in this paper, a nonlinear classification has been chosen that is very simple to understand but works incredibly well in practice [3]. The usage of KNN has some other important advantages such as the capability of adapting to different types of data by selecting a suitable distance measure; and obtaining good predictive accuracy in large and sufficiently representative training datasets. In KNN classification, an object is classified by a majority vote of its neighbors and the output is a class of dataset. It solves the classification problem by seeking the shortest distance between the test data and training sets in the feature space. It assumes that the data is in a feature space; therefore, they have a notion of distance such as Euclidean, Manhattan, Minkowski, and Hamming distance. In this paper, the distance ($D(x_{test}, x_i)$) is based on the Euclidean distance (EDM). The distance between observations is given in Equation (1).

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

where n is the number of features, x_i is an observation in the training set and x_{test} is an observation in the test set. EDM is a function to calculate the Euclidean distance with the domain as the dataset and the range as the real numbers (EDM $:D \rightarrow \mathbb{R}$). This EDM function is shown in Equation (2).

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

where $ch[j]$ is the j-th chromosome in the initial population. The proposed classification model is validated by the leave-one-out method. Validation means measuring the predictor behavior on data points other than those in the training set.

The evaluation of a subset in GA must be performed by a fitness function. The classification accuracy (α) and the number of selected features (N_{FS}) are the two criteria that are utilized to design our fitness function. As given in Equation (3), a high fitness value is produced for the chromosome with a high classification accuracy and a small number of features. Accuracy (α) is a real number between zero and one. Furthermore, the inverse of the number of selected features (i.e.,$^1/_{N_{FS}}$) is a real number between zero and one. Therefore, the fitness value will be between zero and two ($0 \leq fitness \leq 2$).

$$fitness = \alpha + {}^1/_{N_{FS}} \tag{3}$$

Accuracy is calculated as the sum of correct classifications divided by the total number of classifications. For the multiple class datasets, the accuracy is demonstrated only by the average hit rate [33]. The performance of PIGAS has been investigated by the average accuracy for multiclass classification (Equation (4)).

$$\alpha = \frac{\sum_{i=1}^{l}\left(\frac{tp_i + tn_i}{tp_i + tn_i + fp_i + fn_i}\right)}{l} \tag{4}$$

where l is the number of classes, tp_i are true positive for an individual classC_i, fp_i are false positive, fn_i are false negative, and tn_i are true negative counts respectively.

3) Selection Operator

The fundamental idea of the selection operator is giving preference to better chromosomes and allowing them to pass their genes to the next generation. In PIGAS, the best chromosome is selected based on the tournament technique with three members, meaning that in every cycle three chromosomes are randomly chosen and then the best one is selected for the next generation.

4) Crossover Operator

In genetic algorithms, two leading operators that impact the fitness value are the crossover and mutation. After evaluating chromosomes by the fitness function and selecting them for reproduction, the crossover is applied on each pair of the chromosomes. Crossover is a critical genetic operator to explore new solution regions in the search space and to escape from being stuck in local optima. Crossover randomly exchanges genes between two chromosomes. The crossover operator is performed on a random set of chromosomes that are chosen based on a probability, called crossover rate (Pc) [27].

In this work, an intelligent crossover is utilized to escape from being stuck in local optima. The proposed crossover is based on the two-point type, 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,r_2 that are indexes of the genes, are generated such that $1\leq r_1$ *and* $r_2 \leq$ the number of features. Afterward, values of these two genes are exchanged. The outstanding advantage of our crossover operator is that the crossover rate changes whenever the convergence of the solutions is not improved for some successive generations. In other words, to escape from being stuck in local optima, the value of P_c dynamically changes. At first, P_c is set to 0.8 so that exploitation is performed for as long as the algorithm converge, P_c is reduced and the mutation rate is increased. This shifts the algorithm state from exploitation to exploration. In other words, exploitation, which is related to local search, leads to probe a promising limited region of the search space with the hope of improving the solution that we already have. On the other hand, exploration, which is related to global search, consists of probing a much larger region of the search space with the hope of finding other promising solutions that are yet to be refined [28].

5) Mutation Operator

The mutation operator is performed with a probability, called mutation rate (Pm) [27]. In mutation process, the genes may occasionally be inverted (i.e. from 0 to 1 or vice versa). As previously mentioned, this operator explores the search space to find a new search region to prevent being stuck in local optima.

In the proposed algorithm, the type of mutation operator and the number of its probability are adaptively tuned by the algorithm. The mutation operator is intelligently applied in various types based on different states of the algorithm.

It means that, first the mutation is bit flipping (one-point) and its rate is 0.2 (Pm=0.2) until the algorithm appropriately converges to the best solutions. In the bit flipping mutation, a number is randomly generated between 1 and the number of features and the value of the ${r_1}^{st}$ gene in the selected chromosome is inverted. But like our crossover operator, the mutation rate incrementally grows when the convergence of the algorithm does not change for several generations. In this case, the mutation type switches to a new kind of mutation somewhat similar to the boundary mutation (boundary mutation is applied on genomes, but our operator is applied on genes) where two numbers are randomly generated between one and the number of features (i.e.,$1\leq r_1$,$r_2 \leq$ #features) and all values of the genes between r_1 and r_2 are inverted. This intelligent mutation helps to discover a new search region and prevent all solutions in a population from falling into local optima. The proposed mutation returns to its initial state after the improvement in the process of convergence was completely done. This movement to initial state enhances the crossover to more explore the space.

1) Replacement Operator

Offspring 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 method is employed for the replacing process. The proposed operation compares each chromosome in the current population with its corresponding one in the last generation. If a chromosome in the current generation is better than its corresponding one in the last generation, the new chromosome is replaced with the old one.

2) Migration Operator

We use the multi-population strategy where each processor separately runs a GA with its own initial random population until the migration time. The migration operator enables processors to exchange their best genetic material. This operator occurs at fixed intervals (migration gap). The migration time happens on every migration gap. In this work, the migration gap is set to two generations. During the migration process, the best chromosomes in each processor are selected and sent to the next processor in a ring. Each processor replaces its worst chromosomes with the received best ones. This exchanging happens with a fixed migration rate. In PIGAS, the migration rate is set to one chromosome. Seven processors have been utilized in the implementation of this work.

3) *Stopping Strategy*

In general, the evolutionary process operates many iterations until the termination condition is met. The proposed algorithm stops after forty iterations, because it has the best convergence until this generation.

IV. Experimental Results

In this section, a series of experiments have been carried out to evaluate the effectiveness of the proposed methods, PIGAS (a parallel weighted intelligent genetic algorithm) and PAGAS (a parallel genetic algorithm). They are evaluated on Arrhythmia Dataset obtained from UCI Machine Learning Repository [29]. Biomedical datasets create a unique classification challenge to machine learning and data mining algorithms because of their high dimensionality, noisy data, missing values, and multiple classes. The UCI dataset consists of 452 observations, 279 features (from which 206 are linear values and the rest are nominal), and 16 classes of diseases. The missing values in the dataset (32%) were replaced with the mean during the preprocessing phases. The performance of PIGAS has been investigated by the average accuracy in the case of multiclass classification (Equations (4)). To guarantee valid results for making predictions, the data set was validated by the leave-one-out technique.

A. *Implementation Details*

PIGAS and PAGAS were implemented in Visual C++ using the MPI library for parallelization. They were run with MPICH2 on a shared memory structure although they work on both shared memory and message passing architectures. Ring topology has been utilized for connections. All implementations and experiments have been performed on an Intel Core i7-4770 CPU 3.40 GHz, RAM 16.0 GB, running Windows 7 Enterprise (64-bit). Seven cores have been leveraged in the parallel implementation. The parallel GA parameters are indicated in TABLE I.

TABLE I. Parallel genetic algorithm parameters

Initial population (each processor)	400
# Iterations	40
Crossover	Intelligent Crossover
Crossover rate P_c	Dynamic
Mutation	Intelligent Mutation
Mutation rate P_m	Dynamic
Replacement	steady-state
# Processors	7
Migration rate	1 chromosome
Migration gap	2 iterations

B. *Validation of Experimental Results*

PIGAS and PAGAS were compared with two prior feature selection techniques. All approaches were evaluated on the UCI dataset. Furthermore, they were compared with a number of classical feature selection methods implemented in Weka software for dimensionality reduction [30].

In this work, the test performance was also evaluated on the selected features on a number of Weka classifiers such as KNN (called IBK), Naive Bayes (NB), Multi-Layer Perceptron (MLP), and SVM (called SMO). The Weka Wrapper Subset Evaluator was used for features evaluation and the genetic and greedy stepwise approaches were used as the search techniques. The accuracy of PIGAS proved to be better than that of the other considered methods, as demonstrated in TABLE II. In this table, B1 refers to a feature selection technique based on normalized mutual information [19]. B2 refers to a method [31] that recognizes the best feature sets and accordingly ensembles the classifications to obtain a high accuracy [31]. As reported in this table, PAGAS and PIGAS outperforms the other methods under all classifications. Specifically, the accuracy of 98.48% and 99.70% are obtained when the KNN classifier is applied in which the methods are optimized for. Although PAGAS and PIGAS has not been optimized on the SVM, NB, and MLP classifiers, they still gain the highest accuracy as compared to the other approaches.

One of the disadvantages in evolutionary algorithms is that the convergence speed decreases when the number of iterations grows (Generally, difficult problems have no proper convergence to the best solution in low iterations). However, it can be easily observed from the convergence diagram of PIGAS and PAGAS (Fig. 2) that the distance between the diagrams increases when the number of iterations grows, demonstrating the strength of PIGAS. This improvement is because of increasing the selection pressure and population diversity, intelligent operators, and weighting system. This is the best proof to show the benefits of applying the parallel intelligent method, especially on biomedical datasets, where the number of features is large. In Fig. 2, the "best" diagrams represent the elitism chromosomes (the best chromosomes) in the population for PIGAS and PAGAS. The "mean" diagrams, in turn, represent the mean accuracies of the other chromosomes. The stability diagrams of PIGAS and PAGAS are shown in Fig. 3.We can see that the diagram of PIGAS fluctuates less, indicating that PIGAS is more stable and reliable than the non-intelligent algorithm.

The statistical results are reported in TABLE IV. There are four important parameters in this table, described as follows: The standard deviation (STD), a standard deviation close to zero indicates that the data points tend to be very close to the mean (also called the expected value) while a high standard deviation shows that the data points are spread over a wider range of values [32]. The mean is the average value of all the best results in all 20 runs. The best and the worse are the best and the worse values of all 20 runs. A method can be considered the best when it has the lowest values of STD and the mean and the highest values of the best and the worst. TABLE IV demonstrates that our algorithm is more accurate with fewer errors than the non-intelligent algorithm.

Two major performance measures in evaluating a parallel system are speedup and efficiency. The speedup is defined for each number of processors n as the ratio of the single processor execution time to the execution time when n processors are available, which is obtained by Equation (5):

$$S(n) = {}^{T_1}/_{T_n} \tag{5}$$

The efficiency is defined as the average utilization of the n allocated processors, which is obtained by Equation (6):

$$E(n) = S(n)/_{n} \qquad (6)$$

where n is the number of processors [42]. In TABLE III, serial and parallel implementations are compared regarding these two measures. This table shows that the efficiency and speedup of the proposed parallel implementation are prominent. According to the efficiency value, PIGAS is able to efficiently utilize 80% of resources, and based on the speedup value, it is 5.6 times faster than the serial method.

TABLE II. ACCURACY OF DIFFERENT FEATURE SELECTION METHODS UNDER VARIOUS CLASSIFICATION APPROACHES

FS Method	**Classifiers**			
	KNN	*SVM*	*NB*	*MLP*
All features	66.76	67.01	61.50	67.25
PIGAS	**99.70**	**70.57**	**70.40**	**72.56**
Weka	54.20	68.91	62.38	68.92
PAGAS	**98.48**	**69.24**	**69.24**	**69.02**
$\boldsymbol{B_1}$ [19]	65.47	67.92	-	-
$\boldsymbol{B_2}$ [31]	-	66.67	-	-

TABLE III. SPEEDUP AND EFFICIENCY VALUES FOR PARALLEL IMPLEMENTATION

	#Processors	Parallel Time (*m*)	Serial Time (*m*)	Efficiency	Speedup
PIGAS	7	467	2615.2	0.8	5.6

TABLE IV. STATISTICAL RESULTS FOR PAGAS AND PIGAS

	STD	Mean	Median	Best	Worst
PAGAS	0.0019	0.9832	0.9836	0.984895	0.9803
PIGAS	0.0012	0.9933	0.9970	0.9970	0.9890

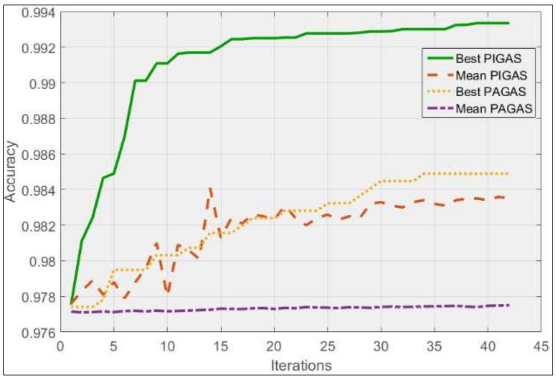

Fig. 2. Convergence diagrams for PIGAS and PAGAS

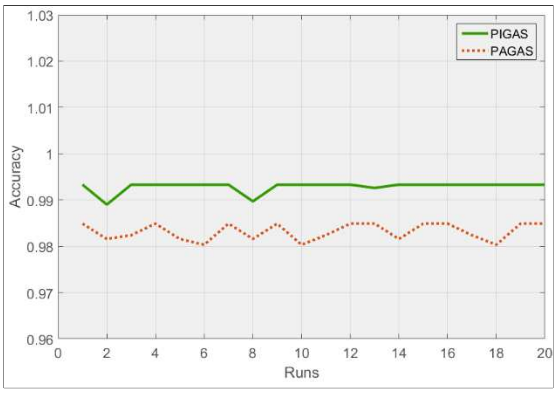

Fig. 3. Stability diagrams for PIGAS and PAGAS

I. CONCLUSION

In this paper, to enhance the reliability of diagnoses in e-Health applications, a wrapper approach including a parallel intelligent genetic algorithm for feature selection (PIGAS) was presented and evaluated on cardiovascular diseases. The main contributions of this work are the following. 1) Intelligent crossover and mutation operations were proposed in order to enable the algorithm to escape from potential local optima. 2) The multi-population strategy was utilized to improve the classification accuracy while improving the time and scalability. 3) Features were weighted combining human knowledge and auto weighting methods. This offered a better accuracy and relevance for medical applications. The application of the proposed multi-population genetic algorithm for feature selection enhanced the accuracy of KNN to 99.70% using only half of features, as against an accuracy of only 66.76% using all the features. PIGAS achieved this accuracy for the KNN classifier with three neighbors examined on the UCI Arrhythmia dataset. The classification model was validated with the leave-one-out technique. The performance of the proposed method was studied by comparing it with various types of approaches. The efficiency of the parallel implementation was 80%, and it was 5.6 times faster than the serial algorithm. The results clearly demonstrated the effectiveness of the proposed PIGAS wrapper approach.

REFERENCES

[1] P. Kligfield, L. S. Gettes, J. J. Bailey, R. Childers, B. J. Deal, E. W. Hancock, G. Herpen, J. A. Kors, P. Macfarlane, D. M. Mirvis, O. Pahlm, P. Rautaharju, and G. S. Wagner, "Recommendations for the Standardization and Interpretation of the Electrocardiogram", Journal of the American College of Cardiology, Vol. 49, No. 10, Elsevier Inc., 2007.

[2] A. Pantelopoulos and N. G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis", IEEE Transaction on Systems, Man, and Cybernetics, Vol. 40, No. 1, 2010.

[3] M.R. Berthold, C. Borgelt, and F. Höppner, "Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data", Series: Texts in Computer Science Springer Verlag, ISBN 978-1-84882-259-7, 2010.

[4] H.-D. Zhu and Y. Zhong, "New feature selection algorithm based on multiple Heuristics", Journal of Computer Applications, Vol. 29, No. 3,849-851, 2009.

[5] D. Koller and M. Sahami, "Toward optimal feature selection", ilpubs.stanford.edu, 1996.

[6] A. E. Eiben, J. E. Smith, "Introduction to Evolutionary Computing", Springer-Verlag Berlin Heidelberg, 2010.

[7] A. Ziarati, "A multilevel evolutionary algorithm for optimizing numerical functions", IJIEC 2, 2011.

[8] N. Milickovic, M. Lahanas, D. Baltas, and N. Zamboglou, " Comparison of Evolutionary and Deterministic Multiobjective Algorithms for Dose Optimization in Brachytherapy", Chapter Evolutionary Multi-Criterion Optimization, Vol 1993 of the series Lecture Notes in Computer Science, 167-180, springer, 2001.

[9] P.G. Espejo, S. Ventura, and F. Herrera, "A survey on the application of genetic programming to classification", IEEE Transactions on Systems, Many, and Cybernetics – Part C: Applications and Reviews 40, 121–144, 2010.

[10] A. Majd and 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,Vol. 9, pp. 97-105, 2014.

[11] E. Alba, F. Luna, A. J. Nebro, and J. M. Troya, "Parallel Heterogeneous Genetic Algorithms for Continuous Optimization," Parallel Computing, Vol. 30, pp. 699–719, ELSEVIER, 2004.

[12] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition", International Journal of Pattern Recognition and Artificial Intelligence, 17(6), 903–929, 2003.

[13] H. Zhang, “Visualization of public and private school choice using synchronized parallel coordinate plot (PCP) approach”, Papers of the Applied Geography Conferences 30, 177-186, 2007.

[14] S. Karpagachelvi, M. Arthanari, M. Sivakumar, “ECG Feature Extraction Techniques – A Survey Approach”, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 1, 2010.

[15] J. Huang, Y. Cai, and X. Xu, “A hybrid genetic algorithm for feature selection wrapper based on mutual information”, Journal of Pattern Recognition, Vol. 28, 1825-1844, 2007.

[16] J. Zhao, F. Zhang, “Feature Selection Based on Parallel Collaborative Evolutionary Genetic Algorithm”, Journal of Advances in information Sciences and Services Sciences (AISS), Vol. 4, 296-304, 2012.

[17] B. Oluleye, A. Leisa, J. Leng, and D. Dean, “A Genetic Algorithm-Based Feature Selection”, International Journal of Electronics Communication and Computer Engineering, Vol. 5, 2014.

[18] C. Emmanouilidis, A. Hunter, and J. MacIntyre: “A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator”, In Proceedings of the 2000 Congress on Evolutionary Computation, IEEE Press, 309–316, 2000.

[19] L. T. Vinh, S. Lee, Y. Park, and B. J. d’Auriol, “A novel feature selection method based on normalized mutual Information”, Journal of Applied Intelligence, Vol. 37, 100-120, 2012.

[20] S. Shilaskar, A. Ghatol, “Feature selection for medical diagnosis: Evaluation for cardiovascular diseases”, Journal of Expert Systems with Applications, Vol. 40, 4146–415, 2013.

[21] H. A. Guvenir and B. Acar “Feature Selection using a Genetic Algorithm for the Detection of Abnormal ECG Recordings”, Computers in Cardiology Conference, 2001.

[22] S. Mokeddem, B. Atmani, and M. Mokaddem, “Supervised Feature Selection for Diagnosis of Coronary Artery Disease Based on Genetic Algorithm”, CSE, CICS, DBDM, AIFL, SCOM, 41-51, 2013.

[23] R Yeniterzi, S Yeniterzi, A. Kücükural, and U Sezerman, “Feature selection with genetic algorithms on cardiac arrhythmia database”, the 2nd International Symposium on Health Informatics and Bioinformatics (HIBIT), 2007.

[24] S. Bhatia, P. Prakash, and G.N. Pillai, “SVM based decision support system for heart disease classification with integer-coded genetic algorithm to select critical features”, In Proceedings of the World congress on engineering and computer science, USA, 2008.

[25] H. C. Purchase, N. Andrienko, T. J. Jankun-Kelly, and M. Ward, “ Information Visualization”, Book: ISSN 0302-9743, Springer, 2008.

[26] Y.T. Tsao, T.-W. Shen, and T.F. Ko, “The Morphology of the Electrocardiogram for Eevaluating ECG Biometrics”, IEEE, 2007.

[27] D. E. Goldberg and J. H. Holland, “Genetic algorithms and MachineLearning,Journal of Machine Learning Volume 3, pp 95–99, 1988.

[28] M. Črepinšek, Sh. H. Liu, and M. Mernik,”Exploration and exploitation in evolutionary algorithms: A survey“, ACM Computing Surveys (CSUR) Surveys, Vol. 45, 2013.

[29] H. A. Guvenir, B. Acar, and H. Muderrisoglu,. UCI Repository of Machine Learning Datasets [https://archive.ics.uci.edu/ml /datasets/Arrhythmia], CA: University of California, School of Information and Computer Science, 1998.

[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update”, SIGKDD Explorations, Vol. 11, 2009.

[31] E. Namsrai, T. Munkhdalai, M. Li, JH. Shin, OE. Namsrai, and K. H. Ryu, “A Feature Selection-based Ensemble Method for Arrhythmia Classification”, Journal of Information Processing Systems, Vol. 9, 31-39, 2013.

[32] B.S. Everitt, “The Cambridge Dictionary of Statistics”, ISBN 0-521-81099-X, 2003.

[33] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks”, Information Processing and Management 45, 2009.

[34] I.S. Oh, J.S. Lee, and B. R. Moon, “Hybrid Genetic Algorithms for Feature Selection”, IEEE Transaction on pattern Analysis and Machine Intelligence, Vol.26, No.11, 1424-37, 2004.

[35] R. M. Jarvis and R. Goodacre, “Genetic algorithm optimization for pre-processing and variable selection of spectroscopic data,” Bioinformatics, vol. 21, no. 7, 860-868, 2005.

[36] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression", the American Statistician, 46 (3), 175–185, 1992.

[37] Ch. F. Tsai, W. Eberle, and C.Y. Chu, “Genetic algorithms in feature and instance selection”, journal of Knowledge-Based Systems, Vol. 39, 240–247, Elsevier, 2013.

[38] M. A. Esseghir, “Effective Wrapper-Filter hybridization through GRASP Schemata JMLR”, In The fourth workshop on feature selection in data mining, Workshop and Conference Proceedings, 45–54), 2010.

[39] H. Handels, T. Ross, J. Kreusch, H.H. Wolff, and S.J. Poppl, "Feature Selection for Optimized Skin Tumor Recognition Using Genetic Algorithms", Artificial Intelligence in Medicine, 16(3), 283-97, 1999.

[40] R. Jain and J. Mazumdar, "A Genetic Algorithm Based Nearest Neighbor Classification to Breast Cancer Diagnosis", Australasian Physical & Engineering Sciences in Medicine, Vol.6, No.1, 6-11, 2003.

[41] H.A. Guvenir, B. Acar, G. Demiroz, and A. Cekin, "A Supervised Machine Learning Algorithm for Arrhythmia Analysis", Proceedings of the Computers in Cardiology Conference, Lund, Sweden, 1997.

[42] D. L. Eager, J. Zahorian, and E. D. Lazowska, “Speedup Versus Efficiency in Parallel Systems”, IEEE Transaction on Computers, Vol. 38, No. 3, pp. 408-423, 1989.

[43] Anooj, P. K. Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules. Journal of King Saud University –Computer and Information Sciences 24, 27– 40, 2012.