Neural Computing and Applications
https://doi.org/10.1007/s00521-020-05347-y

S.I. : HEALTHCARE ANALYTICS

A novel binary chaotic genetic algorithm for feature selection and its utility in affective computing and healthcare

Madiha Tahir[1] · Abdallah Tubaishat[2] · Feras Al-Obeidat[2] · Babar Shah[2] · Zahid Halim[1] · Muhammad Waqas[1,3]

Received: 10 April 2020 / Accepted: 4 September 2020

Abstract

Genetic algorithm (GA) is a nature-inspired algorithm to produce best possible solution by selecting the fittest individual from a pool of possible solutions. Like most of the optimization techniques, the GA can also stuck in the local optima, producing a suboptimal solution. This work presents a novel metaheuristic optimizer named as the binary chaotic genetic algorithm (BCGA) to improve the GA performance. The chaotic maps are applied to the initial population, and the reproduction operations follow. To demonstrate its utility, the proposed BCGA is applied to a feature selection task from an affective database, namely AMIGOS (A Dataset for Affect, Personality and Mood Research on Individuals and Groups) and two healthcare datasets having large feature space. Performance of the BCGA is compared with the traditional GA and two state-of-the-art feature selection methods. The comparison is made based on classification accuracy and the number of selected features. Experimental results suggest promising capability of BCGA to find the optimal subset of features that achieves better fitness values. The obtained results also suggest that the chaotic maps, especially sinusoidal chaotic map, perform better as compared to other maps in enhancing the performance of raw GA. The proposed approach obtains, on average, a fitness value twice as better than the one achieved through the raw GA in the identification of the seven classes of emotions.

Keywords Affective computing · Genetic algorithms · Emotion identification · Feature selection · Optimization tasks · Healthcare computing

✉ Madiha Tahir
madiha.tahir@giki.edu.pk

Abdallah Tubaishat
abdallah.tubaishat@zu.ac.ae

Feras Al-Obeidat
feras.Al-Obeidat@zu.ac.ae

Babar Shah
babar.shah@zu.ac.ae

Zahid Halim
zahid.halim@giki.edu.pk

Muhammad Waqas
muhammad.waqas@giki.edu.pk

[1] The Machine Intelligence Research Group (MInG), Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan

[2] College of Technological Innovation, Zayed University, Abu Dhabi, UAE

[3] Faculty of Information Technology, Beijing University of Technology, Beijing, China

Published online: 18 September 2020

1 Introduction

The evolutionary algorithms (EAs) have recently shown promising results in solving multiple optimization problems. The EAs are powered by their stochastic search ability in multifaceted environments and are guided by one (or more) objective functions. This enables them to search the best possible solution for an optimization problem. With the advancement in information and communication technologies, coupled with the data processing and storage capacity of computing devices in the current era, the size of the average data generated against multiple activities has increased by manifolds. This is applicable to both the number of instances in the data and also the features count. Selecting the appropriate number and most informative features from such large datasets enables the various learning tasks (supervised or unsupervised) to produce better results. EAs have also been used in the past to select the informative features by eliminating irrelevant and redundant features in a high-dimensional space [1–3]. Some of the popular EAs include genetic algorithm (GA), many objective particle swarm optimization (MOPSO), particle swarm optimization (PSO), and differential evolution (DE). The common principals in all EAs include random population initialization, execution until some termination criteria is met (like number of iterations or required efficiency), and reproduction operations, to name a few. In many cases, EAs are stuck in the local optima. It can be because of inappropriate population size, too few iterations, or other parameters of the EAs. In most cases, hybridization is the way to avoid this problem. Several hybrid methods have been presented in the past literature to overcome the local optima and to increase convergence rate [2, 4, 5]. In addition to the utility of hybrid methods, there are a number of mathematical ways to find the optimum solution. Chaos is one of the mathematical approaches which can be used to increase the performance of the EAs in terms of convergence rate and finding the acceptable optima.

Feature selection is considered as a preprocessing step before applying any computational learning technique such as classification or clustering. This is because the selection of informative and relative features improves the performance of the learning methods. Feature selection methods, other than the EA-based techniques, that have been used in the past include Pearson correlation which is a filter-based method, recursive feature elimination a wrapper-based method, and tree-based embedded methods. The feature selection methods can primarily be categorized into three types, namely filter-based methods, wrapper-based methods, and embedded methods. In the filter-based methods, the feature set is filtered using a predefined metric. Examples of filter-based methods are correlation and Chi-square. In the wrapper-based methods, the set of features is considered as a search space; for example, recursive feature elimination is a wrapper-based method. Embedded methods use algorithms that have built-in feature selection methods like in the random forest (RF) technique.

EAs are also used, for instance, selection and data production problems due to their ability to solve global optimization problems and optimize several fitness criteria simultaneously. The GA has been used since decades to optimize various problem solutions. GA is also widely used for feature selection tasks. Performance of the GA depends on multiple parameters, to name a few: population size, maximum number of generations, probability of crossover and mutation, and elitism rate.

1.1 Our contribution

In the past, the algorithms like crow search algorithm (CSA) and grasshopper optimization algorithm (GOA) are combined with chaotic maps to accelerate their global convergence speed and to obtain global optima. However, there are some limitations in these algorithms such as limited classification accuracy and their application in real-world engineering problems needs to be investigated. To overcome these drawbacks, the present work proposes a novel feature selection algorithm which combines the GA [6] with the chaotic maps. The value of each gene in the chromosomes of the initial population is replaced by binary value, where 1 s corresponds to selected features and 0 s shows the otherwise. The binary value of a gene in the population initialization phase is extracted by using chaotic sequences. The selected features of solutions are passed to evaluate the fitness value using a fitness function. In this work, the fitness function is based on the number of selected features and classification accuracy. Thus, this work overcomes the classification error rate and also decreases the space complexity. Chromosomes with high fitness value are used as parents for the next generation. Crossover between selected parents produces child solutions which are evaluated using the fitness function. Mutation of the child solutions is performed to produce diversity. Chaotic sequences are used to mutate the genes of the chromosomes. The proposed binary chaotic genetic algorithm (BCGA) outperforms chaotic crow search algorithm (CCSA) and other evolutionary algorithms in terms of accuracy and reduced feature set. The key contributions of this work are as follows.

- A novel BCGA is proposed for feature selection. The proposed BCGA enhances the mutation and population initialization steps of the raw GA using chaotic maps.

- Random variables' selection process is replaced with the chaotic sequence value to improve the quality of the solution.
- The performances of 10 different chaotic maps are validated on the proposed BCGA for feature selection problem.
- The effectiveness of BCGA is demonstrated through a comparison with different evolutionary algorithms used for feature selection in the past.
- The significance of BCGA is proved by applying Wilcoxon's *t* test.

The primary novelty of the proposed work is innovating different chaotic variants of the GA as well as determination of the various GA parameters using chaos. Furthermore, solving the feature selection problem in the affective computing domain is validated and it can be applied to assist researchers in other areas of applications as well. Emotion detection from neurophysiological signals and healthcare data is a challenging and an evolving problem nowadays. Data collected in the form of electroencephalogram (EEG) signals, electrocardiogram (ECG) signals, and galvanic skin response (GSR) require feature analysis to detect the affective state and health status of a person. The AMIGOS (A Dataset for Affect, Personality and Mood Research on Individuals and Groups) dataset used in this work is an affective database to detect emotions of persons after watching short and long video clips. Additionally, this work utilizes two healthcare datasets to identify important features using the proposed method. As a case study, this work focuses on the application of the proposed algorithm in the fields of affective computing and healthcare; however, in the future, it can be applied in other domains as well.

1.2 Paper organization

The rest of paper is organized as follows. Section 2 gives a brief overview of fundamentals, GA, chaotic maps and presents the problem statement of the present work. The related work on feature selection using EAs and for affective computing and healthcare systems is covered in Sect. 3. The proposed method is explained in Sect. 4. Section 5 presents the conducted experiments and obtained results. This section also analyzes illustrating properties of the proposed algorithm, and its comparison with state-of-the-art methods is also listed. The discussion on the obtained results is presented in Sect. 6. Finally, Sect. 7 concludes this work with a few future directions mentioned.

2 Basics and problem formulation

This section gives an overview of the techniques used in this work, including the GA and chaotic maps. The section also formally formulates the problem at hand. Initially, the basics of GA are covered, followed by the details on chaotic maps, and then present work's problem statement.

2.1 Genetic algorithm

In order to design a multimodal emotion detection system, one needs to have multiple emotions related data from different types of sources. Ensemble of data to make an efficient system requires selection of a subset of data features which can classify different categories of emotions. The GA [7] is capable of being used as a feature selection technique. The GA makes it possible to explore potential solutions to a problem in a better way that cannot be achieved through a conventional method. Most organisms (solutions) are produced naturally or by the crossover/mutation process. The GA first decides which chromosome of the population has the potential to survive in the next generation. Afterward, it determines the mixing and recombination of genes of their parent chromosomes to produce new offspring. Selection of the parents is based on the solutions' fitness value, if it fails on the fitness criteria, it gets eliminated, otherwise the solutions are sorted according to their fitness values, and parents are selected by one of the available parent selection methods.

2.2 Chaotic maps

Chaotic map is an evolution function that shows chaotic behavior. Chaotic maps are parameterized by both discrete time and continuous time behavior. Any change of its initial condition may lead to nonlinear behavior for the future. Chaos is used to optimize performance of an algorithm. They provide fast convergence rate and are used to avoid the local minima. These solutions are beneficial for the performance of the evolutionary algorithms [2]. Chaos employs chaotic variables rather than the random ones. Chaos methods are found in nonlinear dynamical system which are non-periodic, non-converging, and bounded. Chaos is dependent on initial conditions and parameters. Additionally, an enormous number of different sequences can be generated by changing the initial conditions [4]. A variety of chaotic maps is available for optimization problems. In the present work, ten discrete chaotic maps are implemented. The mathematical description of these chaotic maps is given in the following.

Sine map The Sine map is defined using Eq. (1).

$$x_{n+1} = \frac{a}{4}\sin(\pi x_n), \quad a = 4 \tag{1}$$

where a is the control parameter whose value lies in the range [0, 4]. This map generates the chaotic number in the range [0, 1].

Circle map Circle map belongs to the dynamical systems on circle. The circle map is defined using Eq. (2).

$$x_{n+1} = \mathrm{mod}\left(x_n + d - \left(\frac{c}{2\pi}\right)\sin(2\pi x_n), 1\right), \quad c = 0.5, \quad d = 0.2 \tag{2}$$

This map will generate chaotic number in range [0, 1] by using control parameters c = 0.5 and d = 0.2.

Iterative map The iterative chaotic map is mathematically defined using Eq. (3).

$$x_{n+1} = \sin\left(\frac{c\pi}{x_n}\right), \quad c = 0.7 \tag{3}$$

This map generates values in the range of [− 1, 1], while c is a control parameter with 0.7 as its value.

Chebyshev map Chebyshev map is defined by Eq. (4).

$$x_{n+1} = \cos(n\cos^{-1}(x_n)) \tag{4}$$

This map generates value in the range [− 1, 1].

Logistic map Logistic map is defined by Eq. (5).

$$x_{n+1} = ax_n(1 - x_n), \quad a = 4 \tag{5}$$

where a is a controlling parameter with value of 4. This map generates value in range [0, 1].

Singer map Singer map is defined by Eq. (6).

$$x_{n+1} = \mu\left(7.86x_n - 23.31x_n^2 + 28.75x_n^3 - 13.302875x_n^4\right), \quad \mu = 1.07 \tag{6}$$

where $\mu = 1.07$ is a control parameter. Singer map generates value in the range [0, 1].

Sinusoidal map This map is formulated using Eq. (7).

$$x_{n+1} = cx_n^2\sin(\pi x_n), \quad c = 2.3 \tag{7}$$

where c = 2.3 is a control parameter of this map and sinusoidal map generates numbers in range [0, 1] and initial value of $x_0 = 0.7$.

Tent map The tent map is defined using Eq. (8).

$$x_{n+1} = \begin{cases} ax_n & \text{for } x_n < 0.7 \\ a(1 - x_n) & \text{for } x_n \geq 0.7 \end{cases} \tag{8}$$

where a is the controlling parameter. Tent map gives a value in the range of [0, 1]. The present work uses a = 1/0.7 for $x_n < 1/2$ and a = 10/3 for $x_n \geq \frac{1}{2}$.

Gauss map The Gauss map is described mathematically using Eq. (9).

$$x_{n+1} = \begin{cases} 1, & \text{for } x_n = 0 \\ \dfrac{1}{\mathrm{mod}(x_n, 1)}, & \text{otherwise} \end{cases} \tag{9}$$

This map generates chaotic sequences in range [0, 1].

Piecewise map Piecewise map is described mathematically using Eq. (10).

$$x_{n+1} = \begin{cases} \dfrac{x_n}{a} & 0 \leq x_n < a \\ \dfrac{x_n - a}{0.5 - a} & a \leq x_n < 0.5 \\ \dfrac{1 - a - x_n}{0.5 - a} & 0.5 \leq x_n < 1 - a \\ \dfrac{1 - x_n}{a} & 1 - a \leq x_n < 1 \end{cases} \tag{10}$$

where a = 0.4. Summary of ten chaotic maps used in this work is listed in Table 1.

2.3 Problem statement

The past literature reveals that a significant importance is given to the feature selection problem for various learning tasks irrespective of them being supervised or unsupervised approaches. This is because there are generally many features in the underlying data of a learning task and not every feature is important. Additionally, there may also exist multiple correlating features that simply add into the redundancy and biases of the learning process. There is always a need for an optimal solution for the selection of best suited subset of features to increase the classification (or clustering) accuracy (or heterogeneity) and to decrease the number of selected features. However, some of the feature selection algorithms in the evolutionary computing field suffer from weak diversity particularly, when handling high-dimensional tasks and when the search space is thin causing premature convergence. Addressing these limitations of the exiting evolutionary computing-based feature selection methods is the main motivation of this work. Additionally, this work is motivated by the use of different chaotic maps with the GA to solve the optimization problem of selecting a suitable feature subset. As a test bed, this work utilizes affective and healthcare datasets. The primary aim here is to select the optimum features from the affective dataset using the proposed method and later utilize the standard classification methods to predict the emotion hidden in them. Figure 1

Table 1 Description of the ten chaotic maps used in this work

Number	Name	Definition	Range
BCGA1	Sine map	$x_{n+1} = \frac{a}{4}\sin(\pi x_n), a = 4$	[0,1]
BCGA2	Circle map	$x_{n+1} = \mathrm{mod}\left(x_n + d - \left(\frac{c}{2\pi}\right)\sin(2\pi x_n), 1\right), c = 0.5, d = 0.2$	[0, 1]
BCGA3	Iterative map	$x_{n+1} = \sin\left(\frac{c\pi}{x_n}\right), c = 0.7$	[− 1, 1]
BCGA4	Chebyshev map	$x_{n+1} = \cos(n\cos^{-1}(x_n))$	[− 1, 1]
BCGA5	Logistic map	$x_{n+1} = ax_n(1 - x_n), a = 4$	[0, 1]
BCGA6	Singer map	$x_{n+1} = \mu\left(7.86x_n - 23.31x_n^2 + 28.75x_n^3 - 13.302875x_n^4\right), \mu = 1.07$	[0, 1]
BCGA7	Sinusoidal map	$x_{n+1} = cx_n^2\sin(\pi x_n), c = 2.3$	[0, 1]
BCGA8	Tent map	$x_{n+1} = \begin{cases} ax_n & \text{for } a < 1/2 \\ a(1 - x_n) & \text{for } a \geq 1/2 \end{cases}$	[0, 1]
BCGA9	Gauss map	$x_{n+1} = \begin{cases} 1, & x_n = 0 \\ \frac{1}{\mathrm{mod}(x_n, 1)}, & \text{otherwise} \end{cases}$	[0, 1]
BCGA10	Piecewise map	$x_{n+1} = \begin{cases} \frac{x_n}{a} 0 \leq x_n < a \\ \frac{x_n - a}{0.5 - a} a \leq x_n < 0.5 \\ \frac{1 - a - x_n}{0.5 - a} 0.5 \leq x_n < 1 - a \\ \frac{1 - x_n}{a} 1 - a \leq x_n < 1 \end{cases}$	[0, 1]

demonstrates the higher-level abstraction of the proposed work. Thus, the problem statement of this work is as follows.

> To develop a novel evolutionary computing-based feature selection method utilizing the binary chaotic maps and demonstrate its utility on the affective and healthcare data.

3 Related works

This section presents the related work done in the domain of feature selection. The section primarily focuses on the evolutionary computing-based feature selection methods, because the solution proposed in this work is an evolutionary algorithm. Additionally, the section also covers a few of the works related to the feature selection from affective datasets.

3.1 Feature selection methods

Feature selection is an important preprocessing step to extract informative features for the classification and clustering problems. However, the method is mostly utilized for classification, i.e., supervised learning tasks. To select the optimum subset of features from the original dataset is a non-deterministic polynomial-time hard (NP-hard) problem. Therefore, the metaheuristics perform better than the exact methods. In recent years, hybrid metaheuristics have been used in the field of optimization problems research. Hybrid methods show better performance in solving problems like the feature selection task. The first hybrid metaheuristic algorithm for feature selection was proposed in [8]. In this algorithm, local search techniques are merged with the GA to make it more efficient. The work in Majdi et al. [9] presents a hybrid whale optimization algorithm (WOA) with simulated annealing for feature selection. The purpose of using simulated annealing in their work is to exploit most promising regions located by the WOA algorithm. They evaluate the proposed algorithm on 18 standard benchmark datasets obtained from the UCI repository[1] and compare with three wrapper feature selection methods. The work in EzgiZorarpac et al. [10] presents a feature selection method based on combination of artificial bee colony (ABC) optimization technique with differential evolution algorithm. The purpose of their method is to address the overfitting problem which is caused as a result of high dimensionality of the data [11]. In [3], a modified cuckoo search algorithm with rough sets is presented for feature selection. The rough sets theory is included to modify fitness function which takes the number of selected features, and classification accuracy is utilized as the fitness criteria. A binary cuckoo search

[1] https://archive.ics.uci.edu/.

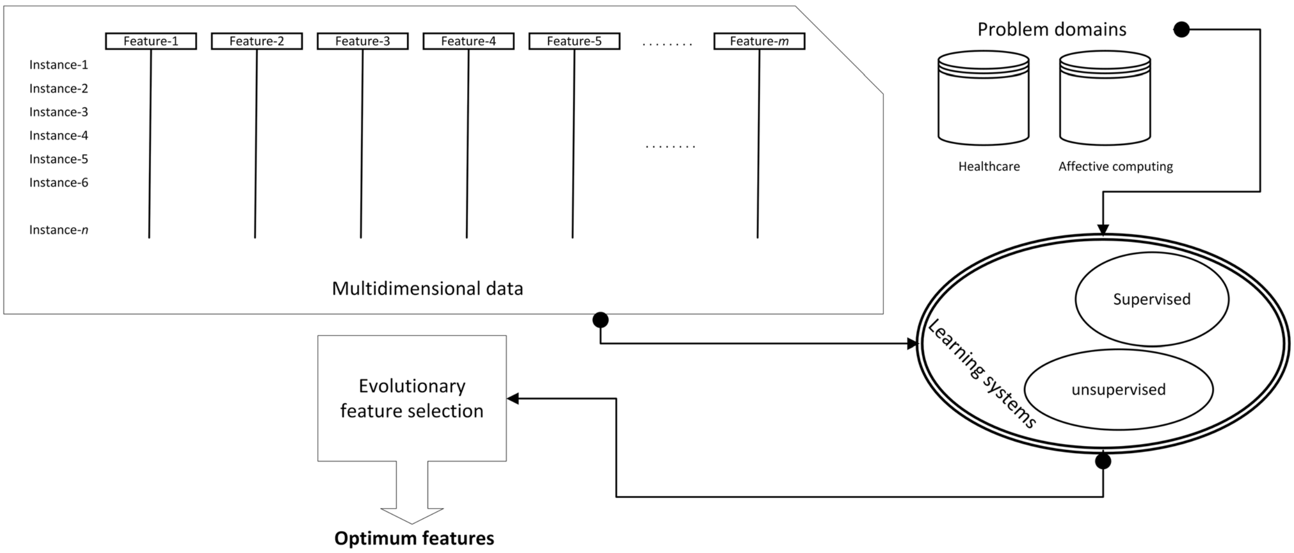

Fig. 1 A higher-level abstraction of the proposed work

algorithm is proposed by Douglas et al. [12] for feature selection. Their method is based on behavior of the cuckoo bird to find the important information from a given set of features in an optimization problem.

In [13], a nature-inspired feature selection technique based on bats behavior is proposed. Their wrapper approach combines the exploration quality of bats with the speed of the optimum-path forest classifier to find the optimum set of features. Experiments are conducted on five public datasets. Results show that their approach outperforms some well-known swarm-based techniques. A hybrid of binary gravitational search algorithm and mutual information is proposed by Bostani et al. [14]. Chaotic GA is used as a wrapper-based method for global search of the features. A mutual information (MI)-based approach is used as filter-based method to compute the gain of a feature with respect to the feature and also feature with respect to the class. This combination is used to find the least redundant features that are most relevant to the target class.

The work in [14] considers two objective functions: maximizing the detection rate and minimizing the false positive rate. An improved version of salp swarm algorithm (SSA) is proposed in [15] to select optimal subset of features using a wrapper-based technique. Opposition-based learning (OBL) is used in the initial phase of SSA to improve its population diversity in the search space. A new local search algorithm (LSA) is developed to improve exploitation capability of the SSA. The improved salp swarm algorithm (ISSA) is validated on 18 datasets from UCI repository. A hybrid feature selection approach based on GA is proposed in [1]. Enhanced GA (EGA) is developed by modifying the crossover and mutation operators. Crossover is performed based on chromosomes portioning with term and document frequencies of the features, while the mutation procedure is performed based on classifier performance of the original parents and feature importance. The work in [1] incorporates six filter-based feature selection methods with EGA to create hybrid feature selection approach. A hybrid genetic local search algorithm (HGA) is proposed in [16] with the *k*-nearest neighbor (*k*-NN) classifier to simultaneously select subset of relevant features and feature weighting, for particularly medium-sized datasets. Gravitational search algorithm (GSA) is a population-based metaheuristic algorithm inspired by the Newton's law of gravity. Mohammad et al. [17] propose the novel GSA algorithm which is based on evolutionary crossover and mutation rates. Majdi et al. [18] develop six variants of the ant lion optimization (ALO) method where each variant employs a transfer function to map a continuous search space to a discrete search space. Their method is compared with particle swarm optimization (PSO), GSA, and two existing ALO-based approaches.

3.2 Feature selection methods to predict emotion

A number of contributions have been made in the past for the task of feature selection from the affective computing perspective. Altun et al. [19] address the strategies for feature selection and multi-class classification for the problem of emotion detection. Identification of most informative and discriminative features is more critical factor according to them in comparison with the classification problem. In order to improve the performance of multi-class SVMs, 58 features are extracted in their work from Berlin Emotional Speech Database (EmoDB). They employ four feature selection techniques, namely sequential forward selection (SFS), least square bound feature selection (LSBOUND), mutual information-based feature selection (MUTINF), and R2W2. The work in [19] concludes that prosodic and sub-band energy features are the most selected ones by all the algorithms in each framework. The framework used in [19] is such that a multi-class classification problem is decomposed into binary classification problem, and then, they perform features selection. They perform intersection and unification operators to construct final feature set from each subset. In the final phase, multi-class classifiers are employed to determine the emotional state using the final feature set.

The work in [20] presents results that help to select methods which are efficient to enhance classification performance and reduce computational complexity for emotion detection from EEG data. They use state-of-the-art methods for feature extraction, feature selection, and classification algorithms. Database for Emotion Analysis using Physiological data (DEAP) is used for experiments in their work. SemEval-2019 Task 3—EmoContext, is presented in [21], and EmoContext is a contextual emotion detection system for text data. For this task, textual dialogues from user interaction with a conversational agent are taken and annotated with emotion classes. A training dataset of 30,160 dialogues and two evaluation datasets containing 2755 and 5509 dialogues is released to the participants. The system analyzes that bi-directional long short-term memory (BiLSTM) is the most used neural architecture and best detection of emotion is achieved for *sad* class, whereas the worst performance is observed for *happy* emotion class. Yan et al. [22] propose a feature selection method based on sparse learning (SL-FS). The focus of their work is to identify emotions from EEG signals. By comparing with the traditional feature selection methods, the SL-FS method improves the correct rate for classification of five classes of emotions when the number of the selected features is same.

Sarcasm is the form of sentiment in which people convey criticism and radicalism in a humorous way. Sundararajan et al. [23] detect sarcasm from text, and additionally, they propose an approach to identify the types of sarcasm. Finding the sarcasm type is to identify the level of hurt or the intention behind the sarcasm. Sarcasm in their work has been classified into four types, namely polite, rude, raging, and deadpan sarcasm. The overall accuracy of their proposed ensemble feature selection algorithm for sarcasm detection is around 92.7%, and multi-rule approach for the identification of types of sarcasm achieves an accuracy of 95.98%, 96.20%, 99.79%, and 86.61% for polite, rude, raging, and deadpan types of sarcasm, respectively. A review on gray wolf optimizer (GWO)-based feature selection for classification is given in [24]. GWO is a recent method belonging to the swarm intelligence family. The mathematical model of GWO consists of encircling, hunting, and attacking the prey. The main step of GWO in solving feature selection problem is to extract the subset of features in the representation of the solution. GWO is appropriate for continuous search problems. Original GWO is converted to its binary version for the feature selection problem in their work.

3.3 Addressed limitations of the past works

The selection of optimum set of features is an NP-hard problem. Any single algorithm is not capable of solving all the optimization problems [25]. Therefore, despite the advantages of hybrid methods mentioned above there is always a room to develop new methods. In developing algorithms for optimization problems, researchers focus to increase convergence rate and to achieve global optima. In the last decade, the domain of mathematics has developed chaos. It has been applied in different optimization researches [2, 4, 26, 27]. Like most the optimization problems, GA also has the tendency to get stuck in the local optima. Most of the past evolutionary computing-based solutions have also overlooked to explicitly address this aspect. The present work utilizes the chaos maps to enhance the capacity of the traditional (or raw) GA. Based on this, an evolutionary feature selection technique is developed here that helps in selecting the optimum feature subset for the classification task. Table 2 lists the key features of the closely related past works and the solution presented here.

4 The proposed binary chaotic genetic algorithm

This section presents the proposed binary chaotic genetic algorithm. In this work, the chaos is used to initialize GA population. This proposal replaces the random variables in mutation procedure of the GA with the chaos variables. The utilization of chaos in the population initialization phase and in the mutation procedure influences the search of an optimal solution. Here, the chaotic sequences are generated from chaotic maps. For this, ten different chaotic maps are used for the optimization process as mentioned in Table 1.

4.1 The binary chaotic GA

The BCGA is an efficient version of the basic GA. The GA is a nature-inspired algorithm, and it has the ability to solve

Table 2 Key features of the closely related past works and the proposed solution

Works	Fitness value		Optimization parameter	Chaos as evolution function		Computational intelligence method	Objectives of optimization	
	Length of selected features	Classification accuracy		Use of chaos	Number of chaos		Multi-objective	Number of objectives
Madiha et al. (proposed)	✔	✔		✔	10	BCGA	✔	2
Ghareb et al. [1]	✔	✔		×	.	EGA	✔	3
Mafarja et al. [18]	✔	✔		×	.	BALO	✔	2
Taradeh et al. [17]	✔	✔		×	.	Hybrid GSA	✔	3
Tubishat et al. [15]	✔	✔		×	.	ISSA	✔	4
Bostani et al. [14]	×	✔		×	.	BGSA	✔	2
Nakamura et al. [13]	×	✔		×	.	BBA	×	1
Sayed et al. [2]	✔	✔		✔	10	CCSA	✔	2

complex optimization problems because of its design. The traditional GA is categorized as global search heuristic that uses iterative process to find an optimal solution. Despite of its advantages, GA has the limitations of requiring a higher number of iterations and its implementation cost is high. It does not solve complex constraint problems conveniently. To account for some of the deficiencies in the raw GA and employ the uses of GA, an enhanced version of this basic algorithm is proposed in this work, namely the binary chaotic genetic algorithm (BCGA).

The steps followed to develop the BCGA are shown in Fig. 2. Parameter initialization of GA and BCGA is same, i.e., setting the values of number of chromosomes (M), maximum number of generations ($G_{\max}$), crossover probability (P_{c}), mutation probability (P_{m}), and elitism rate. Next step is the population initialization using chaos, and this is followed by the selection of parents to perform the crossover procedure between the selected parents that produces a child solution. Next phase is the mutation operation. For this, the chaotic variable is used. Fitness evaluation of the produced solutions is done using two fitness criteria. The first one is the number of selected features, and second is the classification accuracy. The present work uses classification of affective and healthcare data as a case study of the proposed approach. However, the proposed BCGA can be used to form multiple other tasks and the fitness function will vary accordingly. The elites are selected in this work on the bases of elitism rate. This work sets maximum number of generation as the stopping criteria for the proposed approach.

In the proposed solution, initialization of the population and mutation of chromosomes is done using chaotic maps. The chaotic maps used here are sine, circle, iterative, Chebyshev, logistic, singer, sinusoidal, tent, Gauss, and piecewise. Chaotic maps are used to improve performance of the GA in terms of computational time, number of selected features, and classification accuracy. In the current work, BCGA is implemented to select features that can improve classification accuracy and extract lesser number of features that are most significant. Initial population is modified using chaos (Eq. 11). This work generates t chaos, where t = (number of chromosomes) x (number of genes in a chromosome).

$$t = M * N$$

$$\text{for } i = 1 : t$$

$$m_i = C_i$$

$$C = \text{vector to matrix}(m)$$

$$\text{population.chromosomes.gene} = \begin{cases} 1, & C_{i,j} < \text{rand}() \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $C_{i,j}$ is the chaotic value of the ith chromosome and jth gene. If value of jth gene of the ith chromosome is less than a random number, then the jth gene of the ith chromosome is replaced by 1. This means that the feature will be used in the solution. This is because each gene represents a feature and each chromosome is the set of features. A value of 1 in the gene means that the feature represented by the gene is selected and 0 indicates otherwise. The population, in the BCGA, is initialized using random number through Eq. (12).

$$\text{population.chromosomes.gene} = \{\text{rand} \in [0, 1] \tag{12}$$

Mutation operator in the GA is used to mutate the genes (feature's index) of the chromosomes (solution). In case of feature selection task, mutation is used to change the status of the feature. This means if value of a random number is less than the mutation probability, then the value of the gene will be changed using Eq. (13).

$$\text{population.chromosomes.gene} = \begin{cases} \sim\text{Child.gene}, & \text{rand}() < P_\text{m} \\ \text{Child.gene}, & \text{otherwise} \end{cases} \tag{13}$$

Mutation operation is modified by the chaotic maps by replacing random variable of chaos value using Eq. (14).

$$\text{child.gene} = \begin{cases} \sim\text{Child.gene}, & C_{1,k} < P_\text{m} \\ \text{child.gene}, & \text{else} \end{cases} \tag{14}$$

where P_m is the mutation probability, and $C_{1,k}$ is chaotic value for kth gene.

4.2 Parameter initialization

The proposed solution provides the flexibility to adjust various parameters depending on the properties of the task at hand. Here, parameters of the BCGA are adjusted through the hit and trial strategy in the simulation stage. Parameter setting used in the experiments is given in Table 3. In the experiments, this work uses 30 chromosomes (M), maximum generations (G_max) is set to 50, crossover probability (P_c) is set to 0.8, mutation probability (P_m) is 0.01, and elitism rate (E_r) is 0.05. For the sake of fairness, parameter values are kept fixed for all our experiments to evaluate the performance on different datasets given under same conditions.

4.3 Fitness function

After each generation, a subset of features is selected for assessment. To evaluate the fitness of the selected subset of features, a fitness function is defined. Data are divided into two different groups, namely training and testing data using cross-validation through the k-fold method. In this case, the value of k is set to 10 to check the stability of the results. To evaluate the proposed algorithm, fitness function proposed in [2] is used. In [2], two objective criteria are used to evaluate the algorithm, namely classification accuracy and number of selected features. These two criteria are combined here using Eq. (15).

```
Algorithm: Binary chaotic genetic algorithm
Input: Dataset
Output: Optimum feature set

Set the initial values of M, $G_{max}$, $P_c$, $P_m$ and $E_r$
Initialize the initial population using chaos
Get value of chaotic map for i^th chromosome and j^th gene
    $population.chromosomes.gene = \begin{cases} 1, & C_{i,j} < rand() \\ 0, & otherwise \end{cases}$
Evaluate the fitness of initial population
for g=2 to $G_{max}$
    for i=1 to M
        Evaluate fitness of each individual (M)
    end
    for k=1:2:M
        Select parents from population
        Crossover between parents to produce children
        Mutation of each child using chaotic maps
            $child.gene = \begin{cases} \sim Child.gene, & C_{1,k} < P_m \\ child.gene, & else \end{cases}$
    end
    for i=1:M
        Evaluate fitness of each individual of new population
    end
    Select elites through elitism form previous and new population
End
Return Best chromosome and its fitness value
```

Algorithm 1. BCGA pseudocode

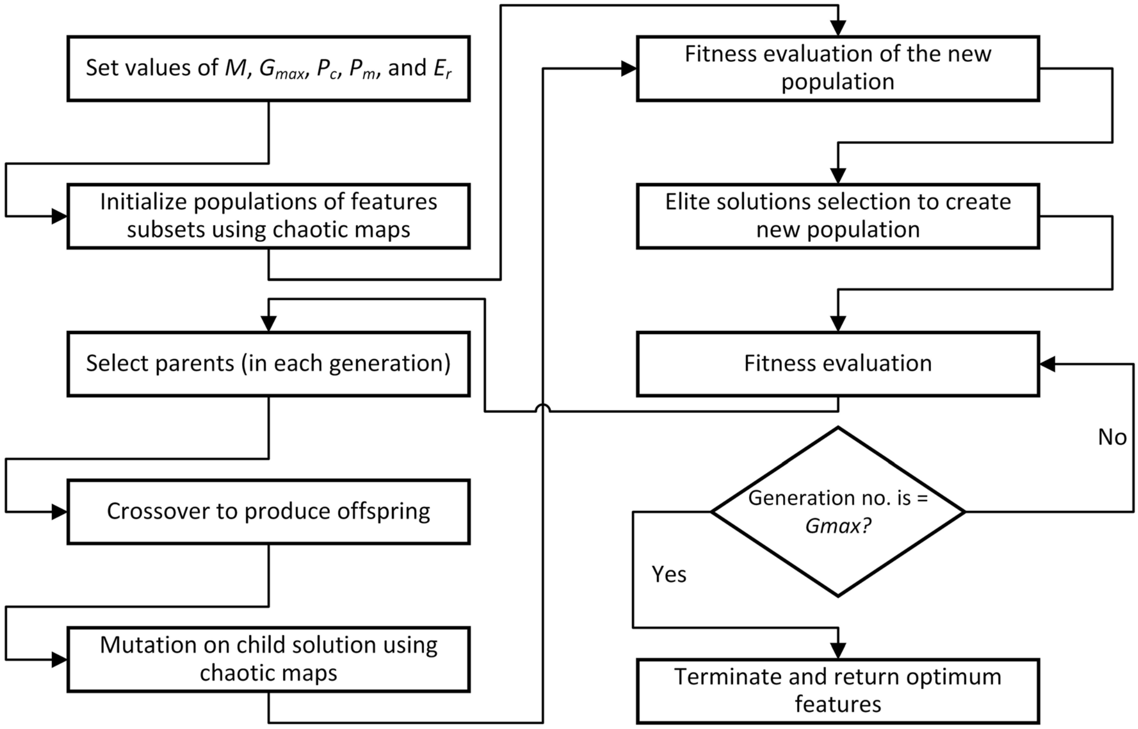
Set values of M, G_{max}, P_c, P_m, and E_r
Initialize populations of features subsets using chaotic maps
Select parents (in each generation)
Crossover to produce offspring
Mutation on child solution using chaotic maps
Fitness evaluation of the new population
Elite solutions selection to create new population
Fitness evaluation
Generation no. is = Gmax?
No
Yes
Terminate and return optimum features

Fig. 2 Flow of the BCGA method

$$\text{Fitness} = \text{accuracy} + w_G\left(1 - {}^{L_s}\!/_{L_t}\right) \quad (15)$$

where w_G is the weight factor to combine both objective functions, L_s is the length of the selected features, and L_t is the length of total features. Value of w_G lies in the range [0, 1]. It is used to combine two objective functions and to give importance to the number of selected features with accuracy.

Table 3 Parameters setting and number of features extracted from data

Parameter	Value
M (number of chromosomes)	30
Max. generations	50
P_c (crossover probability)	0.8
P_m (probability of mutation)	0.01
E_r (elitism rate)	0.05
Data	**No. of features**
ECG	24
EEG	70
GSR	24
ECG + EEG	94
ECG + GSR	48
GSR + EEG	94
All	118

As accuracy is more important for a classification problem, therefore, its weight value is set close to 1, i.e., 0.8. The best solution is that it has the higher accuracy and less number of selected features. Accuracy is calculated by dividing the number of correctly classified instances with the total number of test instances. To classify the test instances, k-nearest neighbor (k-NN) classifier is used. The value of k is set to 3 using the hit and trial procedure. The k-NN is a supervised learning classifier which assigns a class to the test sample from the group of training samples having the nearest distance.

4.4 Termination criteria

The termination condition is an important parameter of an evolutionary computing-based approach. It is usually set to the maximum number of iterations (generations), or the algorithm is set to stop when objectives are achieved with high efficiency. In this work, termination criteria are the number of generations which is set to 50 in all experiments. The pseudocode of proposed algorithm is listed in Algorithm 1.

Figure 3 shows the overall working of the proposed solution. The proposed BCGA method is implemented here as a feature selection algorithm of a wrapper-based method. In BCGA, a chaotic sequence is embedded in its population

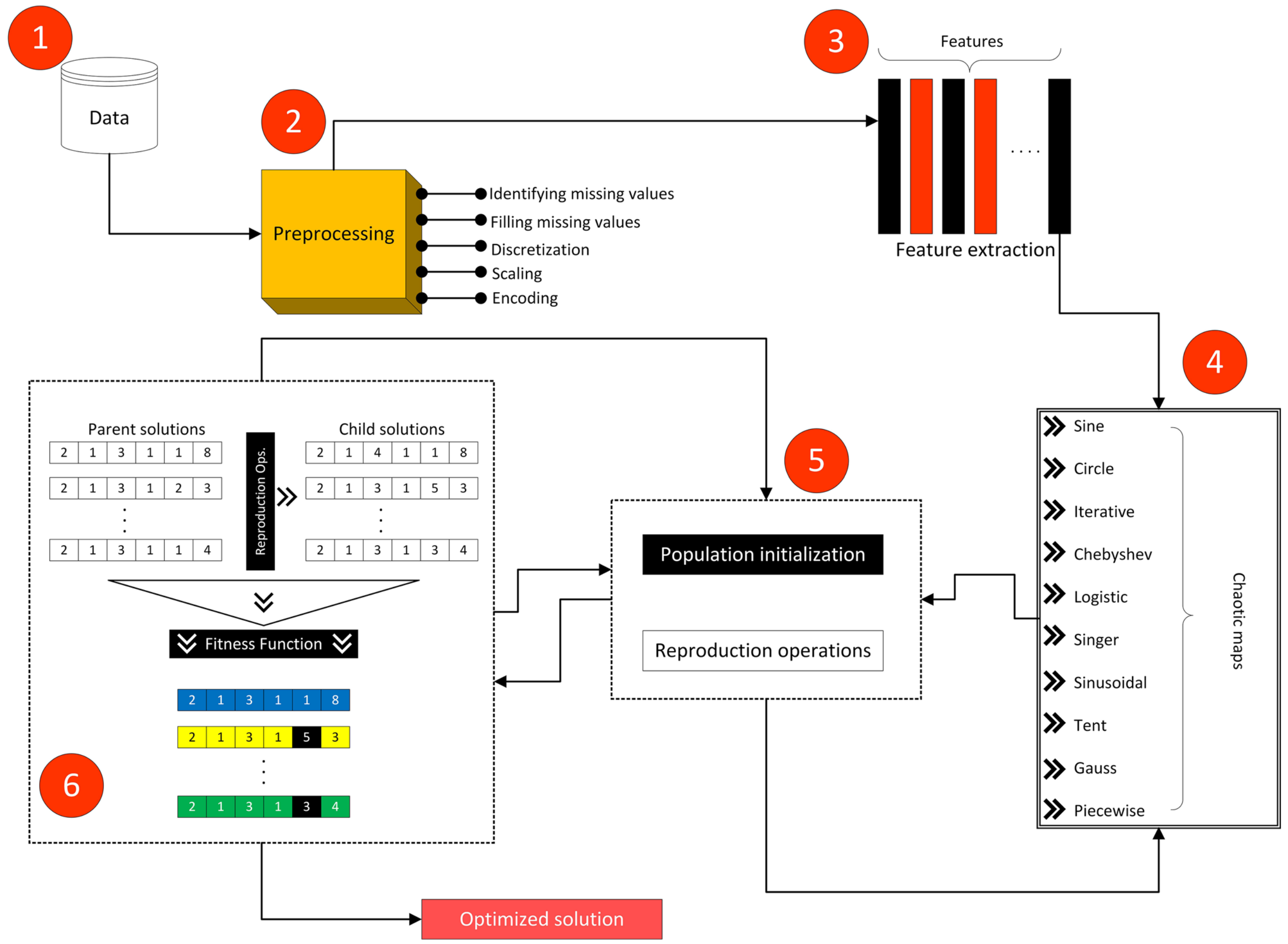

Fig. 3 Overall working of the proposed solution

initialization and mutation phases. The optimum subset of features describing the affective dataset is selected using the BCGA. The purpose of the proposed algorithm here is to extract the optimal features of neurophysiological signals of emotions and also from a healthcare data, to reduce the computational cost and to increase the classification accuracy. However, it is applicable to multiple other problems as well

5 Experiments and results

The proposed BCGA is evaluated on an affective dataset and also on healthcare data to evaluate its application in these domains. This work chose neurophysiological signals of AMIGOS dataset which consists of data for affect, personality, and mood research on individuals and groups. Description of datasets is given in Sect. 5.1. To compare the performance of BCGA with other evolutionary algorithms, this work executes GA, particle swarm optimization (PSO), ant colony optimization (ACO), simulated annealing (SA), and differential evolution (DE) on the neurophysiological signals of AMIGOS. A state-of-the-art method which is used for feature selection and works with chaos, i.e., chaotic crow search algorithm (CCSA) [2], is also implemented to compare its performance on the affective dataset. Results and details of the experiments are given in this section.

5.1 Dataset description

The proposed algorithm is validated on a large benchmark dataset consists of mood, affect, and personality research data obtained by individuals and groups (AMIGOS) [28]. This dataset consists of participants' profile, rating by the participants, external annotations, neurophysiological recordings, and video recordings. The data are resultant of two experiments based on emotion induction through short and long videos. In short videos experiment, 40 participants watched 16 short affective videos. Participants individually rated each video in valence, arousal, dominance, familiarity, and liking. Afterward, they label each video with one of the basic emotions (i.e., *neutral*, *happiness*, *surprise*, *sadness*, *fear*, *anger*, or *disgust*). In case of long videos experiment, 37 participants watched four long videos related to various emotions chosen from the movies. Seventeen participants perform this experiment individually, while the other 20 participants watched it in groups of four individuals. Each participant rates videos in arousal, valence, dominance, liking, and familiarity. As in case of the short videos, the participants select one of the basic emotions against each video. Finally, the videos are annotated by three annotators on the scales of valence and arousal. We evaluated our algorithm on data obtained by using short videos.

Two healthcare datasets are also used in this work for experiments. The first healthcare dataset is the lung cancer dataset[2] obtained from the UCI machine learning repository. In this data, pathological types of lung cancer are classified into three classes having 56 attributes. Fifth and 39th attributes of the dataset have missing values due to which they are discarded from data. The second healthcare dataset is the Parkinson disease (PD) classification data.[3] The data were gathered from 188 patients with PD out of which 107 were men and 81 were women with age ranging from 33 to 87 at the Istanbul University. In the data collection phase, the microphone is set to 44.1 kHz. The sustained phonation was collected from each subject with three repetitions. Features extracted from PD dataset include time–frequency features, Mel frequency cepstral coefficients, wavelet transform-based features, and vocal fold features. There are total 756 samples and 754 attributes in the dataset.

5.2 Preprocessing and feature extraction of EEG, ECG, and GSR data

Experiments in this work are performed on the preprocessed data of the EEG, ECG, and GSR signals. EEG data were recorded using EMOTIV epoch, with sampling frequency of 128 Hz, the data were averaged to the common reference, and band-pass filter was applied with the frequency range of 4.0–45.0 Hz. EEG data are extracted from 14 channels. ECG data were recorded using the shimmer platform with a sampling frequency of 256 Hz. ECG signals are captured from right and left arm. Low pass filter was applied with cutoff frequency of 60 Hz. GSR data were recorded and filtered with low pass filter with cutoff frequency of 60 Hz. This work considered only the data obtained by short videos for experiments. As 40 participants watched 16 short videos, this resulted in a total of 640 samples.

The neurophysiological data of EEG, ECG, and GSR are used here to record the emotions of the participants. In this section, features extracted from these modalities are described. The summary of the features is listed in Table 4. To extract features from the EEG data, EEG signals are rescaled between -1 and 1. NaN (not a number) values are replaced with zero. The signals are filtered using high pass filter with a frequency of 0.2 Hz. Welch's power spectral density method is applied to extract the power spectral density (PSD) of each channel of the signal. Theta, slow alpha, alpha, beta, and gamma bands are extracted from the preprocessed signal. PSDs were averaged on theta band over the frequency range of 4–7.9 Hz, slow alpha band over the frequency range of 7.9–10 Hz, alpha band over the frequency range of 10.1–12.9 Hz, beta band over the frequency range of 13–17.9 Hz, and gamma band over the frequency range of 18–27.9 Hz. Logarithm of averaged signals is used as features of 14 channels of EEG signal of each sample.

The ECG data are preprocessed by rescaling it between the range $[-1, 1]$. Low pass filter is applied with 0.2 Hz and 0.08 Hz frequency. ECG signals show patterns that are not intrinsic to the data, and thus, these trends hinder the data analysis. Filtered signal is determined to remove trends of the ECG. Statistical features (like, minimum value, maximum value, standard deviation, variance, median, and root mean square) are extracted from the preprocessed ECG data. Frequency domain features are extracted from the ECG data using Fourier transform. The two-sided spectrum is determined by dividing magnitude of Fourier transform by the total length. Afterward, its single-sided spectrum is computed to extract the statistical features (i.e., mean, minimum value, maximum value, median, variance, and root mean square). For wavelet analysis, discrete wavelet transform-based features are determined from the ECG data. Statistical features (mean, minimum, maximum, median, variance, root-mean-square values) are extracted from approximated coefficients and detailed coefficients of discrete wavelet transform of the data. Preprocessing and features of GSR data are the same as that of the ECG data.

5.3 Performance metrics

To evaluate the performance of the proposed algorithm, standard metrics and a few statistical measures are utilized here. These metrics are: best fitness value using Eq. (14), worst fitness value using Eq. (15), mean of fitness values using Eq. (16), standard deviation of fitness values using Eq. (17), computational time, number of selected features, and *P*-value from the Wilcoxon's rank test [29]. The rank

[2] http://archive.ics.uci.edu/ml/datasets/Lung+Cancer.

[3] http://archive.ics.uci.edu/ml/datasets/Parkinson%27s+Disease+Classification.

Table 4 Summary of features extracted from EEG, ECG, and GSR data

Modality	Features
EEG	Five bands (theta, slow alpha, alpha, beta, and gamma) are extracted from PSD of signals of 14 channels. Log of average of each band of each channel is used as a feature
ECG	Statistical features of ECG data, statistical features of Fourier transform of the data, and statistical features of discrete wavelet transform of the data
GSR	Statistical features of ECG data, statistical features of Fourier transform of the data, and statistical features of discrete wavelet transform of the data

test is necessary to compare the performance of the proposed algorithm with the traditional GA. The Wilcoxon's rank sum test is a nonparametric statistical test between two independent samples. It considers proportionality of differences between two samples. It gives the sum of the ranks of positive differences between the observations in the two samples. The best value of P is obtained when $P < 0.05$. The abovementioned performance metrics are defined mathematically in the following.

$$\text{Best Fitness value} = \max_{i=1\ \text{to} G_{\max}} \text{FV}_i \tag{14}$$

$$\text{Worst Fitness value} = \min_{i=1\ \text{to} G_{\max}} \text{FV}_i \tag{15}$$

$$\text{Mean Fitness value} = \frac{1}{G_{\max}} \sum_{i=1}^{G_{\max}} \text{FV}_i \tag{16}$$

$$\text{SD} = \sqrt{\frac{\sum_{i=1}^{G_{\max}} (\text{FV}_i - \mu_{\text{FV}})^2}{G_{\max}}} \tag{17}$$

The best fitness value is evaluated using (14), where FV_i is the fitness value of ith solution and i ranges from 1 to the maximum number of generations ($G_{\max}$). The worst fitness value is calculated through (15), where FV_i is the fitness value of the ith solution. The mean fitness value is calculated by dividing the sum of fitness values of i solutions with the total number of generations ($G_{\max}$). Standard deviation of fitness values for i number of generations is calculated by using (17), where FV_i is the fitness value of ith solution, and μ_{FV} is the mean fitness value.

5.4 Performance of BCGA with different chaotic maps for seven basic emotions

This section analyzes the results obtained by experiments on the AMIGOS dataset for seven basic emotions. Ten discrete chaotic maps of BCGA are evaluated on the EEG signals, ECG signals, and GSR data. Evaluation is performed on these individually and in all possible combinations as well (i.e., EEG, ECG, GSR, EEG + ECG, EEG + GSR, GSR + ECG, and all). Table 5 compares BCGA with 10 chaotic maps with the GA. Results obtained using the proposed BCGA and GA for seven basic classes of emotions are given in the table in terms of worst value, best value, mean fitness value, standard deviation in fitness values, P-value of BCGA, computational time, and the number of selected features. It can be seen that for multimodal signals (ECG + EEG + GSR), BCGA with sinusoidal chaotic map obtains highest fitness value which is also greater than the one obtained by GA, and its computational time is also better than what is obtained by GA. The P-value obtained for the sinusoidal chaotic map is 0.0056. It can be seen from the results that for all models, in most of the cases, GA with sinusoidal chaotic map gives best fitness values and with Gauss chaotic map, it has the worst results.

5.5 Performance of BCGA with different chaotic maps for HAHV, HALV, LAHV, LALV

The proposed chaotic genetic algorithm is implemented on AMIGOS for classification of high arousal high valence, high arousal low valence, low arousal high valence, and low arousal low valence. Table 6 compares BCGA for four classes with different chaotic maps in terms of worst, best, mean, standard deviation of fitness value. P-value is calculated between BCGA and the GA. The number of selected features and computational time is extracted from experiments. The experiments are evaluated for seven data models (i.e., ECG, EEG, GSR, EEG + ECG, ECG + GSR, EEG + GSR, and all) of physiological signals of AMIGOS. BCGA1 through BCGA10 are binary chaotic GAs with 10 chaotic maps as given in Table 1. The time here is reported in seconds. It can be observed from Table 6 that for hybrid data model of ECG + EEG + GSR, GA with sinusoidal chaotic map gave the highest fitness value in comparison with others. Its computational time is also less than others. It can be observed that in all data models, the proposed BCGA performed better than the traditional GA.

Table 5 Results obtained by BCGA and GA for 7 basic classes of emotions

	ECG + GSR + ECG (all)								ECG							
	Worst	Best	Mean	SD	*P*-value	Total features	Selected features	Time (s)	Worst	Best	Mean	SD	*P*-value	Total features	Selected features	Time (s)
GA	0.850	1.000	0.928	0.034	–	118.000	35.000	141.484	0.822	1.127	0.973	0.075	–	24.000	2.000	130.068
BCGA1	0.892	0.974	0.937	0.022	0.701	118.000	40.000	164.423	0.928	1.055	1.002	0.029	1.000	24.000	4.000	125.157
BCGA2	0.848	0.928	0.885	0.019	0.225	118.000	44.000	168.936	1.030	1.089	1.062	0.016	0.375	24.000	4.000	128.412
BCGA3	0.844	0.988	0.915	0.035	0.527	118.000	39.000	141.254	0.913	1.091	1.007	0.041	1.000	24.000	3.000	124.404
BCGA4	0.907	0.967	0.932	0.018	0.886	118.000	36.000	141.034	0.970	1.151	1.065	0.056	1.000	24.000	2.000	128.888
BCGA5	0.899	0.965	0.928	0.015	0.884	118.000	36.000	139.317	1.051	1.121	1.088	0.020	1.000	24.000	3.000	123.364
BCGA6	1.026	1.100	1.070	0.020	0.033	118.000	22.000	135.123	1.085	1.188	1.153	0.023	1.000	24.000	3.000	124.513
BCGA7	1.084	1.156	1.122	0.018	0.006	118.000	16.000	127.043	1.051	1.147	1.098	0.028	1.000	24.000	2.000	124.290
BCGA8	0.915	0.993	0.958	0.019	0.662	118.000	32.000	138.213	0.936	1.100	1.023	0.040	1.000	24.000	2.000	124.021
BCGA9	0.543	0.829	0.679	0.080	0.001	118.000	60.000	156.592	0.636	0.890	0.756	0.064	0.031	24.000	9.000	123.228
BCGA10	0.876	0.960	0.928	0.021	0.336	118.000	42.000	143.614	0.979	1.064	1.021	0.022	1.000	24.000	3.000	128.614
	EEG								ECG + GSR							
GA	0.950	1.092	1.017	0.035	–	70.000	14.000	130.816	0.811	1.002	0.926	0.044	–	48.000	9.000	127.624
BCGA1	0.956	1.044	0.993	0.023	0.034	70.000	20.000	130.292	0.949	1.075	1.013	0.035	1.000	48.000	7.000	140.193
BCGA2	0.986	1.091	1.031	0.030	0.059	70.000	16.000	136.420	0.831	0.938	0.886	0.026	0.581	48.000	13.000	134.908
BCGA3	0.920	1.003	0.958	0.020	0.162	70.000	21.000	127.985	0.922	1.033	0.991	0.027	1.000	48.000	9.000	124.026
BCGA4	0.905	0.986	0.948	0.020	0.083	70.000	23.000	133.373	0.907	1.019	0.956	0.027	0.581	48.000	12.000	125.745
BCGA5	0.949	1.036	0.994	0.022	0.144	70.000	21.000	127.317	0.909	1.024	0.974	0.027	1.000	48.000	8.000	124.698
BCGA6	1.022	1.111	1.075	0.022	0.842	70.000	13.000	126.039	0.991	1.095	1.054	0.023	0.227	48.000	4.000	124.637
BCGA7	1.056	1.150	1.102	0.023	0.513	70.000	11.000	131.467	0.996	1.131	1.083	0.035	0.146	48.000	3.000	123.711
BCGA8	1.004	1.081	1.045	0.021	0.819	70.000	15.000	127.042	0.971	1.057	0.999	0.018	1.000	48.000	8.000	125.007
BCGA9	0.553	0.802	0.656	0.062	0.000	70.000	40.000	139.873	0.564	0.860	0.654	0.071	0.022	48.000	20.000	129.649
BCGA10	0.921	1.000	0.951	0.018	0.117	70.000	22.000	132.354	0.994	1.111	1.065	0.030	0.388	48.000	5.000	127.128
ECG + EEG									GSR							
GA	0.885	1.086	0.997	0.059	–	94.000	24.000	244.850	0.735	1.139	0.975	0.108	–	24.000	3.000	128.933
BCGA1	0.863	1.000	0.920	0.036	0.631	94.000	29.000	141.640	0.956	1.145	1.060	0.046	1.000	24.000	2.000	124.674
BCGA2	0.841	0.967	0.907	0.033	0.250	94.000	33.000	185.108	1.009	1.136	1.078	0.032	1.000	24.000	2.000	137.908
BCGA3	0.868	0.962	0.923	0.023	0.150	94.000	33.000	135.651	0.882	1.081	1.009	0.062	1.000	24.000	2.000	123.515
BCGA4	0.890	0.949	0.913	0.016	0.505	94.000	35.000	159.476	1.029	1.120	1.076	0.027	1.000	24.000	2.000	143.563
BCGA5	0.925	0.993	0.967	0.017	0.262	94.000	31.000	133.921	1.001	1.099	1.048	0.025	0.625	24.000	1.000	123.499
BCGA6	1.026	1.088	1.054	0.018	0.028	94.000	18.000	127.475	1.047		1.096	0.024	1.000	24.000	2.000	141.284
BCGA7	1.112	1.174	1.143	0.015	0.019	94.000	12.000	123.697	1.044	1.113	1.070	0.019	1.000	24.000	3.000	151.086
BCGA8	0.931	1.002	0.971	0.016	0.384	94.000	29.000	133.733	1.003	1.095	1.049	0.025	1.000	24.000	3.000	124.065
BCGA9	0.558	0.776	0.683	0.050	0.000	94.000	52.000	149.082	0.651	0.913	0.800	0.073	0.227	24.000	8.000	124.929

Table 5 (continued)

	ECG + GSR + ECG (all)								ECG							
	Worst	Best	Mean	SD	*P*-value	Total features	Selected features	Time (s)	Worst	Best	Mean	SD	*P*-value	Total features	Selected features	Time (s)
BCGA10	0.905	1.011	0.960	0.024	0.752	94.000	26.000	137.012	0.959	1.019	0.987	0.014	1.000	24.000	4.000	126.006
	GSR + EEG															
GA	0.868	1.024	0.947	0.037	–	94.000	22.000	136.290								
BCGA1	0.872	0.980	0.932	0.025	1.000	94.000	30.000	207.051								
BCGA2	0.923	0.989	0.955	0.021	0.194	94.000	31.000	143.252								
BCGA3	0.846	0.951	0.892	0.024	0.869	94.000	32.000	134.162								
BCGA4	0.969	1.057	1.002	0.023	0.273	94.000	25.000	132.408								
BCGA5	0.918	0.989	0.952	0.017	0.655	94.000	28.000	130.395								
BCGA6	1.008	1.092	1.057	0.019	0.042	94.000	18.000	127.572								
BCGA7	1.026	1.108	1.069	0.020	0.078	94.000	20.000	126.748								
BCGA8	0.982	1.066	1.028	0.023	0.117	94.000	22.000	133.928								
BCGA9	0.531	0.786	0.633	0.066	0.014	94.000	49.000	149.400								
BCGA10	0.936	1.033	0.977	0.024	0.622	94.000	28.000	136.451								

5.6 Comparison with the state-of-the-art methods

This section provides a comparison of the proposed approach with two state-of-the-art methods, namely CCSA for feature selection [2] and evolutionary algorithm for features selection of EEG-based signals [30]. The state-of-the-art methods chosen from the literature are the optimization algorithms for feature selection [2, 30]. Hassanien et al. [2] present a feature selection method based on chaotic crow search algorithm (CCSA). They use 10 chaotic maps to optimize the crow search algorithm (CSA). Their work has low convergence rate and may stuck in local optima, while these problems are addressed in the CCSA. In this work, for the comparison purpose, their approach is applied on the AMIGOS dataset to compare the results. Evaluation criteria used in [2] are based on fitness value which measures the accuracy and the number of features [31–33]. Fitness score and computational time obtained by CCSA on the proposed models in this work are given in Table 7. Values of parameters in CCSA are: maximum number of iterations (*M*) which is set to 50 here, Awareness Probability (AP) that is set at 0.1, flight length fl which is assigned a value of 2, lower bound set at 0, upper bound set at 1, maximum iterations (t_{Max}) assigned a value of 50, and *D* that is same as total number of features in original database. Nakisa et al. [3] used evolutionary computation algorithms to select features of EEG signals for emotion recognition using mobile sensors. Here, the approach in [30] is implemented on EEG signals of AMIGOS dataset to compare their approach with the proposed method. The experimental setup used in [30] consists of feature extraction from time domain, frequency domain, and time–frequency domain. Evolutionary algorithms, GA [34, 35], particle swarm optimization (PSO), ant colony optimization (ACO), simulated annealing (SE), and differential evolution (DE), are used to select the optimal subset of features. For the ACO algorithm, the number of ants is 20, evaporation rate is 0.05, and initial pheromone and heuristic value is 1. For the SA algorithm, initial temperature is 10, cooling ratio is 0.99, and maximum number of iteration in each temperature state is 20. For the GA, crossover percentage is 0.7, mutation percentage is 0.3, mutation rate is 0.1 and selection pressure is 8. For the PSO algorithm, construction coefficient is 2.05, damping ratio is 0.9 and particle size is 20. For the DE algorithm, population size is 20, crossover probability is 0.2, lower bound of scaling factor is 0.2 and upper bound of scaling factor is 0.8. By using features and feature selection methods of [30], we get the results given in Table 8. It can be seen from the results obtained for [2, 30] that the current proposal performs better. It can be seen in Table 7 that

Table 6 Performance evaluation of BCGA for HAHV, HALV, LAHV, LALV

	ECG + GSR + ECG (all)								ECG							
	Worst	Best	Mean	SD	*P*-value	Total features	Selected features	Time (s)	Worst	Best	Mean	SD	*P*-value	Total features	Selected features	Time (s)
GA	0.827	0.965	0.913	0.035	–	118.000	30.000	46.698	0.834	1.061	0.963	0.067	–	24.000	2.000	43.158
BCGA1	0.820	0.962	0.898	0.033	1.000	118.000	38.000	42.722	0.886	1.053	0.983	0.040	1.000	24.000	3.000	39.461
BCGA2	0.792	0.888	0.848	0.024	0.199	118.000	47.000	44.123	0.940	1.033	0.993	0.027	0.688	24.000	4.000	39.217
BCGA3	0.839	0.947	0.887	0.027	0.484	118.000	43.000	49.346	0.957	1.110	1.040	0.046	1.000	24.000	3.000	39.764
BCGA4	0.852	0.954	0.899	0.027	0.515	118.000	43.000	49.978	0.891	1.064	0.975	0.040	1.000	24.000	2.000	43.605
BCGA5	0.894	1.026	0.957	0.031	0.217	118.000	30.000	45.116	0.925	1.048	1.000	0.028	0.688	24.000	4.000	39.684
BCGA6	0.913	1.010	0.957	0.026	0.555	118.000	34.000	41.282	0.978	1.083	1.030	0.032	1.000	24.000	2.000	38.768
BCGA7	1.018	1.130	1.073	0.025	0.002	118.000	18.000	40.483	1.029	1.105	1.069	0.021	1.000	24.000	2.000	39.535
BCGA8	0.820	0.913	0.872	0.021	0.782	118.000	40.000	42.066	0.978	1.066	1.027	0.021	1.000	24.000	3.000	39.177
BCGA9	0.551	0.738	0.620	0.047	0.001	118.000	60.000	46.776	0.574	0.856	0.760	0.069	0.022	24.000	10.000	38.070
BCGA10	0.801	0.915	0.867	0.030	0.484	118.000	43.000	54.429	0.950	1.120	1.035	0.036	1.000	24.000	2.000	49.835
	EEG								ECG + GSR							
GA	0.819	0.998	0.897	0.047	–	70.000	18.000	49.187	0.771	0.968	0.873	0.049	–	48.000	11.000	46.409
BCGA1	0.869	0.981	0.935	0.032	0.602	70.000	21.000	43.149	0.859	0.975	0.926	0.032	1.000	48.000	11.000	37.463
BCGA2	0.775	0.899	0.828	0.041	0.059	70.000	28.000	40.184	0.800	0.905	0.843	0.026	0.275	48.000	16.000	38.232
BCGA3	0.919	1.012	0.966	0.023	0.715	70.000	20.000	39.322	0.868	0.968	0.924	0.029	0.808	48.000	12.000	37.922
BCGA4	0.928	1.005	0.950	0.018	0.450	70.000	22.000	43.027	0.790	0.975	0.908	0.046	0.670	48.000	13.000	37.862
BCGA5	0.909	1.010	0.961	0.027	1.000	70.000	18.000	41.990	0.885	1.049	0.962	0.047	0.637	48.000	9.000	38.080
BCGA6	0.962	1.025	0.992	0.018	0.835	70.000	19.000	40.113	1.012	1.087	1.054	0.018	0.302	48.000	6.000	41.831
BCGA7	0.998	1.112	1.046	0.027	0.088	70.000	10.000	38.476	1.065	1.150	1.104	0.019	0.119	48.000	4.000	39.462
BCGA8	0.937	1.057	0.986	0.023	0.564	70.000	15.000	38.580	0.931	1.063	1.003	0.042	0.346	48.000	7.000	38.460
BCGA9	0.519	0.788	0.662	0.070	0.000	70.000	38.000	43.172	0.479	0.813	0.587	0.081	0.033	48.000	21.000	39.740
BCGA10	0.829	0.951	0.893	0.032	0.157	70.000	26.000	52.009	0.890	0.995	0.949	0.031	0.808	48.000	10.000	46.681
ECG + EEG									GSR							
GA	0.785	0.952	0.883	0.041	–	94.000	27.000	49.274	0.817	1.095	0.977	0.077	–	24.000	3.000	43.933
BCGA1	0.836	0.925	0.873	0.017	0.149	94.000	37.000	40.805	0.921	1.098	1.020	0.046	1.000	24.000	2.000	39.293
BCGA2	0.800	0.917	0.863	0.025	0.217	94.000	35.000	41.944	0.905	1.018	0.966	0.025	0.625	24.000	5.000	39.112
BCGA3	0.843	0.930	0.891	0.024	0.355	94.000	33.000	40.539	0.942	1.070	1.000	0.029	1.000	24.000	4.000	39.861
BCGA4	0.891	0.957	0.923	0.018	0.343	94.000	33.000	48.415	0.999	1.079	1.040	0.020	1.000	24.000	4.000	45.677
BCGA5	0.880	0.988	0.936	0.030	0.866	94.000	26.000	40.381	0.946	1.088	1.033	0.038	1.000	24.000	2.000	42.135
BCGA6	0.903	0.984	0.950	0.022	1.000	94.000	27.000	40.815	1.032	1.120	1.074	0.021	1.000	24.000	2.000	38.789
BCGA7	0.992	1.113	1.051	0.031	0.028	94.000	14.000	37.962	0.664	0.924	0.835	0.056	0.344	24.000	7.000	38.566
BCGA8	0.906	1.002	0.950	0.026	0.876	94.000	28.000	41.246	1.054	1.156	1.110	0.022	1.000	24.000	2.000	39.204
BCGA9	0.635	0.800	0.722	0.041	0.011	94.000	45.000	45.371	0.546	0.844	0.664	0.063	0.065	24.000	10.000	38.489

Table 6 (continued)

	ECG + GSR + ECG (all)								ECG							
	Worst	Best	Mean	SD	*P*-value	Total features	Selected features	Time (s)	Worst	Best	Mean	SD	*P*-value	Total features	Selected features	Time (s)
BCGA10	0.899	0.987	0.940	0.025	0.715	94.000	29.000	52.478	0.953	1.066	1.000	0.030	1.000	24.000	2.000	48.569
	GSR + EEG															
GA	0.838	0.961	0.898	0.033	–	94.000	30.000	50.510								
BCGA1	0.815	0.920	0.866	0.028	0.435	94.000	35.000	43.383								
BCGA2	0.751	0.870	0.795	0.027	0.019	94.000	45.000	42.546								
BCGA3	0.814	0.902	0.850	0.019	0.330	94.000	36.000	41.993								
BCGA4	0.884	0.982	0.930	0.022	1.000	94.000	30.000	40.967								
BCGA5	0.818	0.919	0.862	0.025	0.355	94.000	36.000	43.450								
BCGA6	0.960	1.063	1.015	0.025	0.194	94.000	22.000	38.968								
BCGA7	0.990	1.103	1.048	0.026	0.005	94.000	14.000	38.592								
BCGA8	0.931	1.036	0.990	0.029	0.480	94.000	26.000	39.929								
BCGA9	0.469	0.726	0.588	0.061	0.000	94.000	59.000	46.008								
BCGA10	0.844	0.925	0.887	0.022	0.882	94.000	31.000	47.189								

BCGA gives higher score than CCSA in all combinations of data. CCSA gives 1.0233 score value for hybrid combination of EEG, ECG, and GSR signals, while BCGA gives 1.1557 score value for the same combination. Score for all competing methods is extracted using the same fitness function used in this work. Although computational time of CCSA is less than BCGA, however, that is because of the basic architecture of GA. Similarly, Table 8 shows scores obtained by feature selection techniques used in [30], GA, PSO, ACO, SA, and DE. Each technique is validated for 15, 25, 45, and 100 iterations on AMIGOS dataset. Average fitness score is calculated for each feature selection technique. The average fitness score obtained by GA is 0.6784, by PSO is 0.8971, by ACO is 0.9067, by SA is 0.9017 and by DE is 0.8925. EEG signals are used in [30] for experiments; therefore, here only EEG data of AMIGOS dataset are utilized for a fair comparison. By using proposed BCGA on EEG data, a fitness score of 1.1496 is obtained which is higher than the methods presented in [30].

The proposed algorithm in this work is used in an optimization problem to overcome the limitations of the raw GA. BCGA increases the convergence rate, gives more significant and less number of features, and achieves the global optima. It can be seen from results given in Table 5 that for 7 basic classes of emotions, a score value of 1.1557 is obtained on hybrid data of all signals (i.e., EEG, ECG, and GSR) with 16 selected features out of 118 and computational time is 127.042870 s, standard deviation is 0.0177, worst value is 1.0843 and average value is 1.1216. Table 6 shows that for four dimensions of emotions, this work obtained 1.1295 best score value on hybrid data of EEG, ECG, and GSR with 18 selected features out of 118 and computational time is 40.482767 s, standard deviation in fitness values is 0.0253, worst value is 1.0181, and average value is 1.0730. In both the cases, BCGA7 (binary chaotic GA with sinusoidal chaotic map) performs better as compared to others, whereas BCGA9 produces worst results for both, 7 classes of emotions and 4 dimensions of emotions by giving 0.5429 score value for 7 classes and 0.5508 score value for four-dimensional data.

5.7 Experiments with healthcare data

Results obtained for BCGA on the lung cancer dataset are shown in Table 9. It can be seen from the results that the best fitness value is obtained using BCGA7, i.e., by using sinusoidal map, and worst results is observed for BCGA9, i.e., by using the Gauss map. Computational time taken by BCGA7 is 136.842090 s which is less than the time taken by the raw GA. BCGA7 selects 10 features, while simple GA selected 16 features. Table 9 also shows the results obtained for PD classification dataset. It can be seen from

Table 7 Performance comparison of proposed BCGA with CCSA [2]

Features	CCSA		BCGA	
	Score	Time	Score	Time
All	1.0223	36.766941	1.1557	127.04287
ECG	1.0484	24.186268	1.1876	124.5133
GSR	1.0484	27.996435	1.1445	124.673728
EEG	1.079	24.422253	1.1496	131.467387
ECG + GSR	1.0353	24.521543	1.1308	123.710654
EEG + GSR	1.0677	24.703019	1.1076	126.747508
EEG + ECG	1.0698	25.027313	1.1735	123.696552

the results that best fitness value is obtained for BCGA7, i.e., by suing sinusoidal map, while the worst value is obtained using BCGA9. Computational time taken by BCGA7 is 167.011444 s which is less than the time taken by the raw GA, i.e., 238.037346 s. BCGA7 selects 185 features, while simple GA selects 312 features. These results show better performance of the proposed method on healthcare datasets as well, thus suggesting the general applicability of the proposed approach.

6 Discussion

This work presented a novel feature selection algorithm by enhancing the raw GA through chaotic maps. Two datasets from affective computing [36] and healthcare domains [37–40], namely AMIGOS, lung cancer and Parkinson' disease, were selected to assess the proposed algorithm. Ten different chaotic maps (namely sine, circle, iterative, Chebyshev, logistic, singer, sinusoidal, tent, Gauss, and piecewise) were used to increase the efficiency of the traditional GA in terms of classification accuracy and number of selected features. The AMIGOS dataset consisted of data obtained by the ECG signals, EEG signal, and GSR signals for the detection of seven basic categories of emotions (namely *neutral*, *happiness*, *surprise*, *sadness*, *fear*, *anger,* and *disgust*) and 4 dimensions of emotions (high valence high arousal, low valence high arousal, high valence low arousal, and low valence low arousal). Results obtained suggested that for all kinds of data of AMIGOS (i.e. ECG + EEG + GSR), BCGA7 had best fitness value of 1.156, while GA gave 1.00 with *p*-value of 0.006. BCGA7 selected 16 features, while GA selected 35 features, which depicted the efficiency of BCGA7 over the raw GA. For ECG data, BCGA6 had a fitness value of 1.188, while GA had 1.127 fitness score, and BCGA6 selected 3, while GA selected 2 features. For EEG data, BCGA7 had 1.150 fitness value, while GA had a value of 1.092. The BCGA7 selected 11, while GA selected 14 features. For GSR data, BCGA1 gave 1.145 fitness value, while GA had a value of 1.139. The BCGA1 selected 2, while GA selected 3 features. For ECG + GSR data, BCGA7 fitness was 1.131, while GA had 1.002 as the fitness value. The BCGA7 selected 3, while GA selected 9 features. For ECG + EEG data, BCGA7 obtained 1.174 as

Table 8 Results achieved by other optimization methods [30]

Method	No. of iterations	Time (s)	Accuracy	Fitness score	Avg. fitness score
GA	15	991.42039	29.42	0.683	0.6784
	25	298.902322	28.01	0.6839	
	45	520.532768	28.01	0.6789	
	100	1172.74298	28.06	0.6679	
PSO	15	1717.50948	28.17	0.8939	0.8971
	25	3078.92019	28.01	0.8923	
	45	5568.09036	27.86	0.8908	
	100	14252.6113	29.91	0.9113	
ACO	15	994.622346	29.01	0.9011	0.9067
	25	1723.76188	29.13	0.9035	
	45	2889.45185	30.20	0.9142	
	100	6371.96398	29.60	0.9082	
SA	15	417.679116	29.13	0.9035	0.9017
	25	677.23834	28.48	0.897	
	45	1277.92451	28.95	0.9017	
	100	5251.96398	29.26	0.9049	
DE	15	72.541351	28.79	0.9002	0.8925
	25	120.773407	28.50	0.8973	
	45	216.571631	27.86	0.8908	
	100	482.449023	26.95	0.8817	

Table 9 Results obtained using BCGA on healthcare data

Dataset	Method	Worst	Best	Mean	SD	*P*-value	Total features	Selected features	Time (s)
Lung cancer	GA	0.877	1.225	1.048	0.098	–	54	16	139.554
	BCGA1	0.871	1.198	1.065	0.088	0.513	54	20	141.005
	BCGA2	0.923	1.268	1.094	0.091	0.297	54	22	144.501
	BCGA3	0.983	1.310	1.127	0.096	1.000	54	17	140.082
	BCGA4	0.831	1.265	1.074	0.104	0.607	54	20	146.089
	BCGA5	0.786	1.104	0.968	0.091	0.144	54	24	140.284
	BCGA6	0.998	1.359	1.172	0.075	0.808	54	16	97.020
	BCGA7	1.043	1.449	1.272	0.079	0.090	54	10	136.842
	BCGA8	0.847	1.335	1.097	0.110	0.314	54	22	141.900
	BCGA9	0.578	0.999	0.770	0.098	0.000	54	39	147.216
	BCGA10	0.801	1.165	0.962	0.090	0.336	54	22	153.124
Parkinson’s disease	GA	1.118	1.186	1.154	0.016	–	754	312	238.037
	BCGA1	1.104	1.163	1.137	0.014	0.015	754	354	247.362
	BCGA2	1.066	1.118	1.092	0.014	0.000	754	387	272.248
	BCGA3	1.138	1.179	1.160	0.013	0.103	754	339	251.588
	BCGA4	1.162	1.200	1.181	0.011	1.000	754	308	238.273
	BCGA5	1.107	1.254	1.193	0.041	0.055	754	345	247.086
	BCGA6	1.223	1.327	1.275	0.029	0.000	754	234	184.881
	BCGA7	1.333	1.393	1.369	0.015	0.000	754	185	167.011
	BCGA8	1.138	1.201	1.171	0.016	0.639	754	317	307.956
	BCGA9	0.864	0.919	0.887	0.012	0.000	754	574	653.906
	BCGA10	1.111	1.186	1.144	0.019	0.237	754	330	403.536

the fitness value, whereas GA obtained 1.086 fitness while selecting 12 and 24 features, respectively. In case of GSR + EEG, the fitness value obtained by GA was 1.024, while from BCGA7, it was 1.108, whereas the number of selected features by GA was 22 and BCGA7 selected 20 attributes. BCGA9 performed the worst for all data (EEG + ECG + GSR), ECG data, EEG data, GSR data, ECG + GSR data, ECG + EEG data, and GSR + EEG data by giving poor fitness values. Figure 4 visually shows the average fitness values achieved by the 10 variants of the proposed approach and the raw GA for the seven basic classes of emotions.

Results obtained by AMIGOS for four-dimensional categorization of emotions were also computed. It was observed from the results that for all kinds of data, BCGA7 gave best fitness value of 1.130, while GA obtained 0.965 fitness value with a *P*-value of 0.002. BCGA7 selected 18 features, while GA selected 30 features, which depicted the efficiency of BCGA7 over GA. For ECG data, BCGA10 attained a fitness of 1.120, while GA obtained 1.061 score. Both BCGA10 and GA selected 2 features. For EEG data, BCGA7 gives 1.112 fitness value, while GA obtained a fitness of 0.998. The BCGA7 selected 10 features, whereas the GA selected 18 features. For GSR data, BCGA8 achieved a fitness of 1.054 and the raw GA attained 0.817 fitness. The BCGA8 and GA attained these values while selecting 2 and 3 features, respectively. For ECG + GSR data, BCGA7 gave 1.065 fitness, while GA had 0.771 fitness. The BCGA7 and GA selected 4 and 11 features, respectively. For ECG + EEG data, BCGA7 attained 1.113 fitness while selecting 14 features only and the GA gave a fitness value of 0.952 with 27 features selected. In case of GSR + EEG, the fitness value obtained by GA was 1.103 with 30 features, while for the BCGA7, it was 0.961. The BCGA9 performed the worst for four-dimensional categories by yielding poor fitness values. Figure 5 shows the performance comparison of BCGA and CCSA.

In case of lung cancer data, the results obtained by the proposed algorithm suggest that BCGA7 gave best fitness value of 1.449, while GA had a fitness score of 1.225. Additionally, the BCGA7 selected 10 attributes and the raw GA selected 16 features. The BCGA9 performed the worst by obtaining 0.578 fitness value. In case of Parkinson’ disease data, the results showed that BCGA7 attained a fitness of 1.393, while the raw GA had 1.186 as the fitness value. The BCGA7 selected 185 features, whereas the raw GA selected 312 features. Based on this, it can be concluded from the results that BCGA7 performs better among

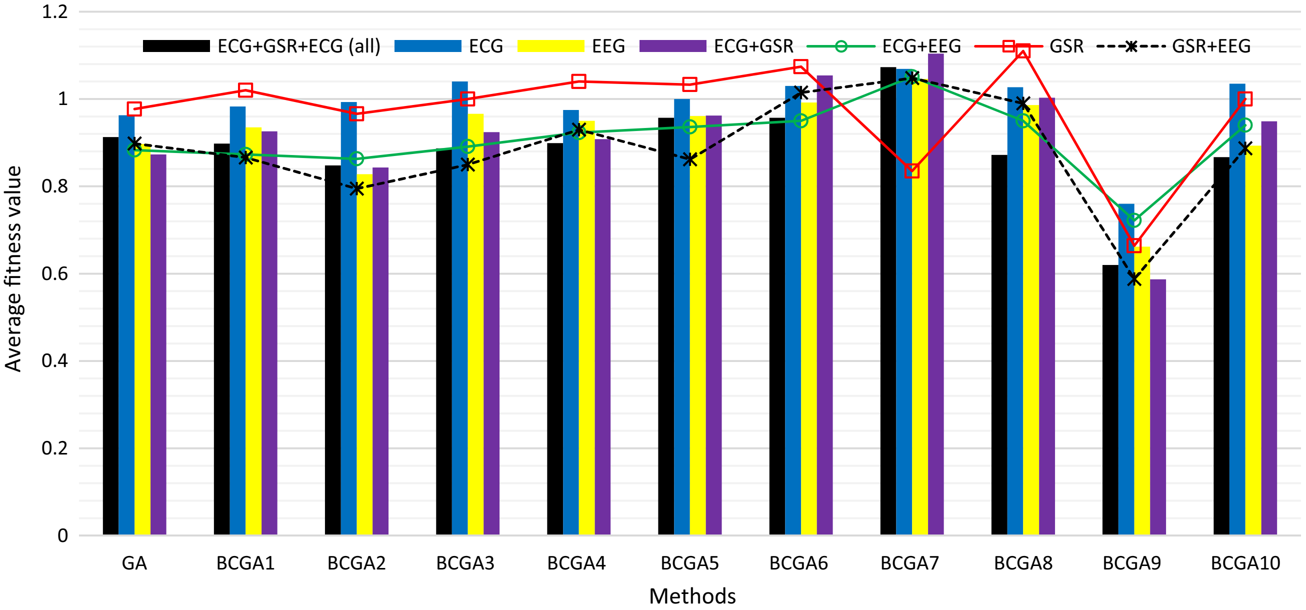

Fig. 4 Average fitness values achieved by the 10 variants of the proposed approach and the raw GA

the other given approaches. Other than the novelty and multiple strengths of the proposed work, there are a few limitations of this work as well. A limitation of the proposed work is that it may require a couple of additional computations in comparison with the raw GA. However, these are at the cost of producing better results.

7 Conclusion and future directions

This work presented a novel chaotic genetic algorithm with binary fitness evaluation criteria named as binary chaotic genetic algorithm (BCGA). Ten different chaotic maps were used to increase the performance of the traditional genetic algorithm addressing its computational time and number of selected features for a classification problem. Chaotic maps were selected based on their application in the scenarios under consideration. To evaluate and analyze the results, worst value, best value, mean, and standard deviation of the fitness values were calculated for 50 generations. The count of 50 generations was selected after hit and trial and was fixed to this number to compare with all models on the same number of generations. Wilcoxon test was applied to calculate the *P*-value of proposed algorithm compared with the genetic algorithm. The proposed algorithm was evaluated on the AMIGOS (A Dataset for Affect, Personality and Mood Research on Individuals and Groups) benchmark and healthcare datasets. Experiments were performed on AMIGOS data collected against short videos experiment and labeled with seven basic emotions rated by participants for arousal and valence values. Data of three physiological signals, i.e., electroencephalogram, electrocardiogram, and galvanic skin response, were considered in this work. The proposed work was also compared with the traditional GA and two state-of-the-art evolutionary computing-based methods for feature selection. In the proposed method, i.e., BCGA, chaotic values were used instead of random number in population initialization and mutation phase. BCGA was used to select significant features according to the classification accuracy. The results obtained suggested better performance of the proposed solution in comparison with the two state-of-the-art methods. From the results, it is concluded that the

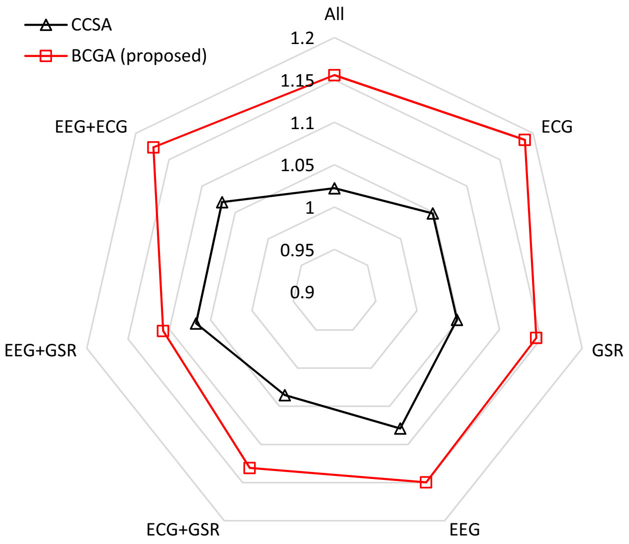

Fig. 5 Performance comparison of BCGA and CCSA

BCGA7 variation of the proposed approach performs best among the ten variants and the BCGA9 variant performs the worst.

There are several directions for future enhancement in the present work. In the future, some other discrete chaotic maps can be used in mutation and population initialization phases and chaotic maps can be used for different steps of the raw GA rather than for the two stages utilized here. Fitness function used in this work can be replaced with a different version to observe its overall performance. Parameters such as computational time could be included in the fitness function to optimize the efficiency trade-off for large dataset. This work focused on the affective computing as an application of the proposed method; in the future, it can be applied and validated in other fields of computing and engineering. Another future direction of this work is that new meta-hybrid approaches can also be considered to address optimization problems, based on the current work. The experiments and the obtained results show that the BCGA can become a promising method for solving multiple real-world problems.

Acknowledgements The authors wish to thank GIK Institute for providing research facilities. This work was sponsored by the GIK Institute graduate research fund under GA4 scheme Grant Number GCS1737. The authors are indebted to the editor and anonymous reviewers for their helpful comments and suggestions.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ghareb AS, Bakar AA, Hamdan AR (2016) Hybrid feature selection based on enhanced genetic algorithm for text categorization. Expert Syst Appl 49:31–47
2. Sayed GI, Hassanien AE, Azar AT (2019) Feature selection via a novel chaotic crow search algorithm. Neural Comput Appl 31(1):171–188
3. El Aziz MA, Hassanien AE (2018) Modified cuckoo search algorithm with rough sets for feature selection. Neural Comput Appl 29(4):925–934
4. Arora S, Anand P (2019) Chaotic grasshopper optimization algorithm for global optimization. Neural Comput Appl 31(8):4385–4405
5. Uzma Halim Z (2020) Optimizing the DNA fragment assembly using metaheuristic-based overlap layout consensus approach. Appl Soft Comput 92:106256
6. Iqbal S, Halim Z (2020) Orienting conflicted graph edges using genetic algorithms to discover pathways in protein–protein interaction networks. IEEE/ACM Trans Comput Biol Bioinf 1:1–26
7. Halim Z, Muhammad T (2017) Quantifying and optimizing visualization: an evolutionary computing-based approach. Inf Sci 385:284–313
8. Oh IS, Lee JS, Moon BR (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26(11):1424–1437
9. Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 260:302–312
10. Zorarpacı E, Özel SA (2016) A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Syst Appl 62:91–103
11. Halim Z, Ali O, Khan G (2020) On the efficient representation of datasets as graphs to mine maximal frequent itemsets. IEEE Trans Knowl Data Eng 1:1–18
12. Rodrigues D, Pereira LA, Almeida TNS, Papa JP, Souza AN, Ramos CC, Yang XS (2013) BCS: a binary cuckoo search algorithm for feature selection. In: IEEE international symposium on circuits and systems, pp 465–468
13. Nakamura RY, Pereira LA, Costa KA, Rodrigues D, Papa JP, Yang XS (2012) BBA: a binary bat algorithm for feature selection. In: 25th SIBGRAPI conference on graphics, patterns and images, pp 291–297
14. Bostani H, Sheikhan M (2017) Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems. Soft Comput 21(9):2307–2324
15. Tubishat M, Idris N, Shuib L, Abushariah MA, Mirjalili S (2020) Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. Expert Syst Appl 145:113122
16. Eroglu DY, Kilic K (2017) A novel hybrid genetic local search algorithm for feature selection and weighting with an application in strategic decision making in innovation management. Inf Sci 405:18–32
17. Taradeh M, Mafarja M, Heidari AA, Faris H, Aljarah I, Mirjalili S, Fujita H (2019) An evolutionary gravitational search-based feature selection. Inf Sci 497:219–239
18. Mafarja M, Eleyan D, Abdullah S, Mirjalili S (2017) S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. In: Proceedings of the international conference on future networks and distributed systems, pp 1–7
19. Altun H, Polat G (2009) Boosting selection of speech related features to improve performance of multi-class SVMs in emotion detection. Expert Syst Appl 36(4):8197–8203
20. Ackermann P, Kohlschein C, Bitsch JÁ, Wehrle K, Jeschke S (2016) EEG-based automatic emotion recognition: feature extraction, selection and classification methods. In: IEEE 18th international conference on e-health networking, applications and services, pp 1–6
21. Chatterjee A, Narahari KN, Joshi M, Agrawal P (2019) SemEval-2019 task 3: EmoContext contextual emotion detection in text. In: Proceedings of the 13th international workshop on semantic evaluation, pp 39–48
22. Yan Y, Li C, Meng S (2019) Emotion recognition based on sparse learning feature selection method for social communication. SIViP 13(7):1253–1257
23. Sundararajan K, Palanisamy A (2020) Multi-rule based ensemble feature selection model for sarcasm type detection in twitter. Comput Intell Neurosci 2020:1–17
24. Al-Tashi Q, Rais H M, Abdulkadir SJ, Mirjalili S, Alhussian H (2020) A review of grey wolf optimizer-based feature selection methods for classification. In: Evolutionary machine learning techniques, pp 273–286
25. Rehman S, Tu S, Waqas M, Rehman O, Ahmad B, Halim Z, Zhao W, Yang Z (2020) Optimization based training of evolutionary convolution neural network for visual classification applications. IET Comput Vis. https://doi.org/10.1049/iet-cvi.2019.0506

26. Tavazoei MS, Haeri M (2007) Comparison of different one dimensional maps as chaotic search pattern in chaos optimization algorithms. Appl Math Comput 187:1076–1085
27. Yang DX, Li G, Cheng GD (2007) On the efficiency of chaos optimization algorithms for global optimization. Chaos Solitons Fractals 34:1366–1375
28. Correa JAM, Abadi MK, Sebe N, Patras I (2020) Amigos: a dataset for affect, personality and mood research on individuals and groups. IEEE Trans Affect Comput 1:1–14
29. Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1:80–83
30. Nakisa B, Rastgoo MN, Tjondronegoro D, Chandran V (2018) Evolutionary computation algorithms for feature selection of EEG-based emotion recognition using mobile sensors. Expert Syst Appl 93:143–155
31. Razzak MI, Imran M, Xu G (2020) Big data analytics for preventive medicine. Neural Comput Appl 32:4417–4451
32. Razzak I, Saris RA, Blumenstein M, Xu G (2020) Integrating joint feature selection into subspace learning: a formulation of 2DPCA for outliers robust feature selection. Neural Netw 121:441–451
33. Razzak MI, Imran M, Xu G (2018) Efficient brain tumor segmentation with multiscale two-pathway-group conventional neural networks. IEEE J Biomed Health Inform 23(5):1911–1919
34. Uzma Al-Obeidat F, Tubaishat A, Shah B, Halim Z (2020) Gene encoder: a feature selection technique through unsupervised deep learning-based clustering for large gene expression data. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05101-4
35. Halim Z, Uzma (2018) Optimizing the minimum spanning tree-based extracted clusters using evolution strategy. Clust Comput 21(1):377–391
36. Halim Z, Atif M, Rashid A, Edwin CA (2019) Profiling players using real-world datasets: clustering the data and correlating the results with the big-five personality traits. IEEE Trans Affect Comput 10(4):568–584
37. Naseer A, Rani M, Naz S, Razzak MI, Imran M, Xu G (2020) Refining Parkinson's neurological disorder identification through deep transfer learning. Neural Comput Appl 32(3):839–854
38. Razzak MI, Naz S, Zaib A (2018) Deep learning for medical image processing: overview, challenges and the future. Classif BioApps 26:323–350
39. Rehman A, Naz S, Razzak MI, Akram F, Imran M (2020) A deep learning-based framework for automatic brain tumors classification using transfer learning. Circuits Syst Signal Process 39(2):757–775
40. Shah A, Halim Z (2019) On efficient mining of frequent itemsets from big uncertain databases. J Grid Comput 17(4):831–850

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.