How to Cite:
Fofanah, A. J., Bundu, H. R., & Kargbo, J. G. (2022). A generic heart diseases prediction and application of genetic algorithms in healthcare systems: Genetic algorithm and machine learning algorithm approaches. *International Journal of Health Sciences*, *6*(S3), 12264–12290. https://doi.org/10.53730/ijhs.v6nS3.9024

A generic heart diseases prediction and application of genetic algorithms in healthcare systems: Genetic algorithm and machine learning algorithm approaches

Abdul Joseph Fofanah
Milton Margai Technical University, Department of Mathematics and Computer Science, Sierra Leone,
Email: abduljoseph.fofanah@mmtu.edu.sl

Habibu Rasin Bundu
Milton Margai Technical University, Department of Mathematics and Computer Science, Sierra Leone,
Email: habibu.bundu@mmtu.edu.sl

Jonathan Gibrill Kargbo
Milton Margai Technical University, Department of ICT and Technology
Email: jonathankargbo@mmtu.edu.sl

***Abstract*---**In this paper we present a cardiovascular diseases prediction which is referred to as heart diseases. A detail review and application of genetic algorithms in healthcare systems including machine learning algorithms were evaluated. The cardiovascular disease involves narrowed or blocked blood vessels that can lead to a heart attack, angina or stroke, and other heart failures such as muscle, valves or rhythm, which is one of the largest causes of morbidity and mortality among the world population. According to our analysis and results obtained between 85-89 percent of ages greater than 40 years were seriously affected by cardiovascular diseases which provides an imperative result in regards to the previous Ebola outbreak in West Africa in 2014-2016 and the current COVID-19 pandemic of which the aging population were more affected. Our results also indicate, the higher the generation the better prediction and performance and more complex the algorithm. However, with GA and ML approaches are useful to predict the output from the existing data and to match-up the probability computation against the cardiovascular diseases' dataset.

Keywords---genetic algorithm, generic prediction, healthcare, architecture, mutation, crossover, operators, healthcare.

Introduction

The region of the mid-brain that results from the death of dopamine generating cells in the substantia nigra is referred to as Parkinson Disease (PD) which is a degenerative disorder of the central nervous system [1]. According to Betarbet et al. [1], Parkinson's disease affects approximately 1 percent of the world population over the age of 55. Dementia and dysautonomia occur frequently in advanced stages of the disease, with nonmotor features. In the case of the presence of two or more cardinal motor features for example rest tremor, bradykinesia, or rigidity by Parkinson's disease [2]. The promise of improved diagnosis and enabled assessment in early disease is held by functional neuroimaging. Bradykinesia, tremor, rigidity, and postural instability are the major symptoms of Parkinson's disease. With Parkinson's disease and when all of these symptoms are seen, then the person can be diagnosed by doctors. One of the most challenging aspects of Parkinson's disease by many patients and their families is considered dysphonia.

Closely 9 out of 10 people with Parkinson's disease have a speech or voice disorder. Symptoms such as breathiness, reduced loudness, roughness decreased energy in the higher portion of the harmonic spectrum are symptoms of dysphonic overstated vocal tremor and these symptoms can be detected using many different vocal tests [3]. Due to the most frequent or common symptoms, voice data is more suitable in the design of an automatic diagnosis system for Parkinson's disease. Several studies were conducted to determine the speech signals and are recorded, and then these signals are detected by means of different methods certain features of these signal [4-6]. A classifier is used to diagnose patients with Parkinson's disease from certain features of the signal. The automatic diagnosis system depends on the classifier. As high accuracy as possible even through there are many unrestrained variations.

The results are very clear that computers have revolutionized our everyday life. Ranging from aerospace and astronomy, archeology, engineering and many other domains. Electronic chips and computers are the mainstay of imaging, diagnostic, monitoring, and therapeutic devices which have contributed greatly in medicine. The devices that are composed of many various hardware constituents, are managed and controlled by software applications and subsequently based on algorithms. A set of well described rules and instructions that describe or outline a sequence of operations or functions is referred to as algorithm. According to Osman et al. [7], algorithms that can more quickly resolve complex problems, alternatively find an estimate solution when classical techniques are not able to find an exact one is referred to as metaheuristic techniques.

Researchers have developed and constructed many metaheuristic algorithms to optimize solution that exist and are inspired by natural biological behaviour and patterns. Some of these algorithms include the following: ant colony inspired by ants' behaviour [8], artificial bee colony inspired by bees behaviour [9], Grey Wolf

optimizer inspired by grey wolves behaviour [10], artificial neural networks emanated from the neural systems [11], simulated annealing [12], river formation dynamics inspired on the process of river formation [13], artificial immune systems inspired on immune system functions [14], genetic algorithm inspired by genetic mechanisms [15], and many other algorithms. To solve complex problems or to find an optimal decision, metaheuristic approaches have been rapidly utilized in other field of science for decision making process. The power of these potent algorithms for offering solutions to the uncountable complex issues especially physicians encounter everyday which has not been fully exploited in medicine, however, valuable work has been done in this domain with the use of classical computers and it hoped that when artificial intelligence and machine learning is fully utilized these algorithms will solve a lot of problems in various field especially the expectation for full implementation of quantum computing. In this paper, we introduce the combined genetic algorithm with machine learning techniques in order to help predict challenges encountered in the healthcare systems. A case study for the evaluation of the proposed algorithm is also present for generations after generations as propounded by Darwinian theory. The metaheuristic and machine learning review some of its application in medicine.

Concepts of genetic algorithms

Considering other algorithms based on metaheuristic techniques, GA is a metaheuristic technique inspired by the laws of Darwinian in genetics, ensuring to find valuable solutions to complex problems. Some random solutions we referred to as individuals are generated each containing many features, we called chromosomes in this technique. Inspired by the law of genetics, crossover and mutations occur in chromosomes (features) to produce a second generation of individuals with more diverse features. The two most central techniques for diversifying individual are crossover and mutation. During crossover two chromosomes are selected. Then a crossover points along each chromosome is selected followed by interchange of the two chromosomes (features). A typical scenario of the phenomena is illustrated in Figure 1. The technique to persuade diversity in the population of individuals called candidate solutions. The diagram (a) indicates during crossover, one part of a chromosome (feature) is interchange by another fragment another chromosome and (b) during mutations one or more datasets on a chromosome are converted to various ones. These modifications or changes will generate new individuals whose fittest or more optimal solution will survive.

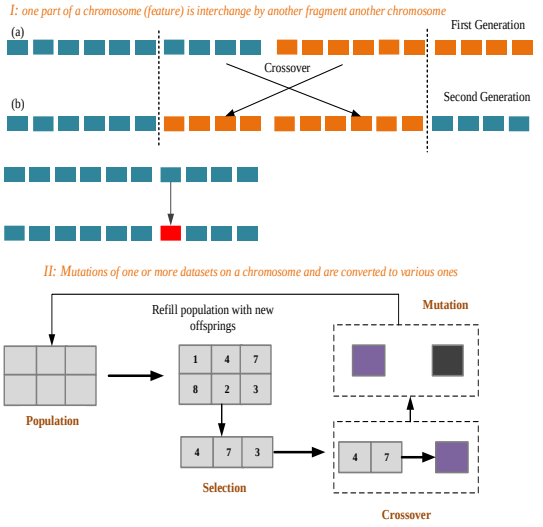


Figure 1. Crossover, one part of a chromosome is interchange by another fragment another chromosome and II: Mutations one or more datasets on a chromosome are converted to various ones

In genetic algorithms, the possibility of reproduction relies on the fitness functions of the individuals. The better the chromosomes (features) they have, the more likely they are to be chosen for breeding the next generation and better chances the subsequent generation are to be reproduced. By applying random changes in different chromosomes, the mutation then generates new configurations [16]. Figure 1 represents two scenarios of simplest mutation techniques. There are several selection techniques; consequently, the objective of all is to map-out the fitness values to individuals based on a fitness function and the selected fittest offspring's genetic alterations in chromosomes or features will occur through crossover and mutations to produce another generation. An optimal solution would have to be obtained due to it iterative process will continue until the fittest individual is formulated or the maximum number of generations is achieved [17-18]. Additionally, GA is very important to predict certain parameters for a given generation and determine which variable will attain the best performance especially in disease predictions with combinatory algorithm of GA and ML in healthcare domain.

This paper major contributions are to determine a given dataset and with variables sets in the GA for every generation and compare their performance matrix. Considering other optimization algorithms, GA is worth noting that from derivative -based optimization algorithms. Firstly, Gas searches a population of points in the solution space in each iteration while classical derivative-based techniques search a population using probabilistic transition rules and random number generators and on the other hand derivative-based algorithms using deterministic transition rules for choosing the nest point in the other or arrangement. Additionally, by predicting the successiveness of generation after generation in relation to their performance and precision of data in solving business and biologically related problems. This enhances the facilitation of

making GA research-based challenges especially during the Pandemic or COVID-19 crisis.

An evolutionary process is utilized in the structure of genetic algorithms to solve a problem [19]. A genetic algorithm begins with a set of solution which are represented by individuals. The set of solutions is referred to as a population. New solutions are selected according to their fitness values and each population is a solution set. Furthermore, in genetic algorithm, the iterative process is recurrent as long as the new population is better than the old one. For example, a computational recurrent for 20 generation is better that the recurrent values of the 10 generation. The higher the fitness values (functions) of an individual are, the more likely this individual is reproduced for the following population. The iterative process is completed when some conditions for example, the number of individuals in the population are fulfilled [20]. There are 8 phases of the genetic algorithm as highlighted below:

- ***Phase 1***: A random variable of individual $n\,population$ is created. These individuals are suitable solution to the problem. In this case, the value of $n\,is\,50$ for experimental purposes.
- ***Phase 2***: The fitness function $f(x)$ is an illustration of each individual (x) and is computed in the population [21]. In our experimental illustrations and studies, each of the individuals in the population is randomly constructed.
- ***Phase 3***: From among the individuals that are selected, two parental individuals are selected. These individuals have the higher fitness value in the population. This parental individual is realized to the then crossover operator.
- ***Phase 4***: A crossover probability estimation is utilized in this phase for crossover operator to form the new individuals. If the crossover is undone, the individual will be the exact copy of the parents.
- ***Phase 5***: Each new individual is obtained by mutating with a mutation probability, in this phase. This mutation process is realized by using any one or more bits of the individual generational computation
- ***Phase 6***: The new individuals are obtained from the new population in this phase.
- ***Phase 7***: Here certain conditions hold, if the end conditions are satisfied, the GA is stopped. It is returned to the best solution in the current population in this phase.
- ***Phase 8***: In this phase, it is returned to phase 2. Then, the new generated population is used for further algorithm.

Application of genetic algorithms and machine learning in healthcare systems

Radiology

Radiology generates a large amount of data that requires to be analyzed and interpreted by radiologists in a relatively short time through imaging techniques. Recently, various devices and applications have been developed to combat the detection and diagnosis of growing interdisciplinary technologies that aim to support radiologists in faster and more accurate image analysis through the

process of segmentation, detection, and classification of normal and pathological patterns found on different imaging modalities. Ultrasound, CT scan (compute tomography), magnetic resonance imaging (MRI), and X-rays are typical examples [22]. An image of scenery in machine vision such as organs of the human body in radiology images is acquired, processed, and interpreted. To access the objects in detail, the boundaries of shape and sizes of objects within the images are required to be determined. Consequently, the process of edge detection results in one of the integral portions of automatic image processing methods [23]. For various imaging modalities such as MRI, CT, and ultrasound, Gas for edge detection of images is acquired and by utilized many researchers.

Oncology

GA was used to analyze the biomolecular information generated vis the Raman spectroscopy using partial least square discriminant analysis system to distinguish between a normal and dysplastic cervix. This method is a statistical technique (partial least square) whose goal is to find a linear regression model between a dependent variable and some predictor variables. The results of the system differentiated dysplasia from a normal cervix with sensitivity and specification of 72 and 90 percent respectively [24].

Cardiology

The hallmarks of most myocardial infarctions and strokes are caused by atherosclerotic plaques. In different fields of cardiovascular medicine, Gas has been utilized for determining plague mechanical features such as elasticity that would enhance physician locate better and map vulnerability plaque or situation that is unstable [25]. A system involving Gas was used for parameter estimation which is essential for predicting accurate elasticity quantification and determining tissue elasticity [25]. For inhomogeneous solution spaces containing several local minima and the requirement for substantial computation time limits their application, this system is higher than gradient-based techniques. The rapid development in medical diagnosis, prognosis, and disease follow-up especially in the field of biomarker discovery and clinical proteomics. Mass spectrometry is an example of advanced technology which can generate readouts of thousands of patients from patients' samples. Nevertheless, the complexity and cost of each method on the one hand, and computational and statistical techniques for analysis, on the other hand, require the selection of few, significant markers for clinical research development. An improved version of GA that supports the local float enhancement method to predict the risk of a major adverse cardiac event (MACE) through recursiveness was employed by Zhou et al [26]. A panel of seven proteins including myeloperoxidase to predict the risk of MACE with 77 percent accuracy and thus outperformed several current techniques were selected.

Obstetrics and gynecology

During childbirth (if necessary), the difference between normal and prolonged delivery enables obstetricians to determine the optimal timing for interventions. The time to reach full cervical dilation, delivery time, and segregate normally against prolonged labour is one of the parameters that can support prediction

purposes. In a study, Hoh et al. [27] employed a three-parameter logistic model by utilizing GA or the Newtons Raphson (NR) technique to forecast the time to arrive at full cervical dilation. More cervical dilation was implemented based on the GA algorithm and outperformed the NR technique.

Surgery

ANN is one of the powerful mathematical algorithms with the capability of predicting the behaviour of systems. A GA-based on ANN (GANN) was developed to predict the outcomes of surgery for patients with non-small cell lung cancer (NSCLC) because of the predictive value of ANNs. For optimization purposes, the GA was applied not to fall into local minima. The outcome of NSCLC patients was more accurate and significantly better than logistic regression when the GANN model was applied to predict the outcome. Nevertheless, the inclusion of tumor size in calculating significantly improved the prediction outcomes [28]. The number of geriatric patients needing cardiac surgeries increases as population age increases. Proper prognostication of postoperative morbidity and mortality would be informative, precluding overestimation of risk and denial of surgery for patients deserving it, which could occur with some prediction models due to the high prevalence of comorbid conditions in an elderly individual. A short length of stay after cardiac surgery was correlated with younger age, no preoperative use of beta-blockers, shorter cross-clamp time, and absence of congestive heart failure showed that GA by applying GA can produce better predictivity in medicine [29].

Evolution of ga enhancement

The enhancement of one consequently converts performance limit evaluation to a general and optimal problem given as:

$$\underset{O_{pt} \in R}{min} \quad b(O_{pt}) \qquad [1]$$

Where O_{pt} is the optimization variables that is for any simulation exercise; $b(O_{pt})$ is the objective function that determine the desired performance; R is the set comprises of all possible situation. For the purpose of this scenario, equation [1] is the performance limit assumed to be the minimum value of the objective function. Also, the optimization problem can be converted to equation 1 equivalently.

Evolution of Desired Setting

Similar to the traditional assessment, the evolution test by the enhanced GA comprises of five basic steps as indicated in Figure 2 [30] where the crossover or mutation operators are adapted to enhance the evaluation efficiency. The new ones are referred to as crossover and manifold mutation, in that magnitude. The full crossover and manifold mutation operators are developed to increase the probability of producing more complex situations, in respect to the performance limit which is more likely to be activated with complex simulations. The initial population, say $\mathbb{x}_1 = \{X_1, X_2 \ldots . . X_{2n}\}$ evaluation situations, is generated randomly. The assess situation, X_i is composed of values chosen randomly from the $m - simulations$ elements. Each simulation is associated as an individual and

represented by a chromosome with L-genes. The performance is measured by an objectives function. When it iterates and reach it desired value, it shows that the performance limit is found and the simulation process terminate. The sort-based fitness function is utilized for natural selection [31], to circumvent premature caused by over-selection of individuals, and ensure the selection possibility of each individual is greater than zero. This is represented by a mathematical function as:

$$f(X_i) = \frac{P - 2(P-1)(Q_i - 1)}{(2n-1)(2n)} \qquad [2]$$

Where $f(X_i)$ is the fitness function determining the probability of the selection of $X_i, P \in [1, 2n]$ is the ranking of X_i number. The best individual found during the test is recorded before natural selection, which is referred to as elitist selection strategy to ensure the global convergence [32-33].

Complexity of genetic algorithm architecture

The core parameters of the GA are crossover and mutation operators. By increasing the probability of producing more complex simulations for better efficiency an index is required to measure the effectiveness of the evaluated situation without conducting the assessment which objective is to enhance them. Nevertheless, it is difficult to set up an analytical function in different scenarios: between the performance of automate driver system (ADS) and the evaluation scenarios and so on. The analytical hierarchy process (AHP) is adopted to develop the situation complexity index [34-35]. This is different from the objective function derived from equation [1].

The leverage factors of any complex problems are numerous and complicated depending on the multiple of scenarios being conducted for the assessment. It is very difficult to determine the significance of each factor when designing the evaluation situation quantitatively. Comparative analysis among the factors belonging to the same parent node is required by this tree structure. The AHP is the obtained by the significance degree of the node at the bottom layer [36-37]. This is illustrated mathematically by:

$$T_k = \prod_{(i,j)\in\mathbb{R}} R_{i,j} \qquad [3]$$

Where T_k is the significance degree, $R_{i,j}$ is the relative significance of node, $b_{i,j}$ and $\mathbb{R}$ is set composed of the index of the path from $b_{f,k}$ to the root.

An evaluation scenario, X, is denoted by the values of situation elements, that is $\mathbb{x}_1 = \{X_1, X_2 \ldots\ldots X_{2n}\}$, where X_i denotes the value of the $i-th$ situation element. To determine the exact performance limit, X_i is generated randomly in it continues range, consequently $X_i$is not equal to the discrete value, $b_{f,j}$. The linear interpolation is used to calculate the significance degree of $X_i \epsilon (b_{f,k-1}, b_{f,k})$ and the situation complexity index is obtained by the full function of:

$$C(X) = \sum_{i=1}^{m} T_i = \frac{T_{k-1} + X_i - V_{f,k-1}}{V_{f,k} - V_{f,k-1}} * (T_k - T_{k-1}) \quad [4]$$

where $C(X)$ is the situation complexity index.

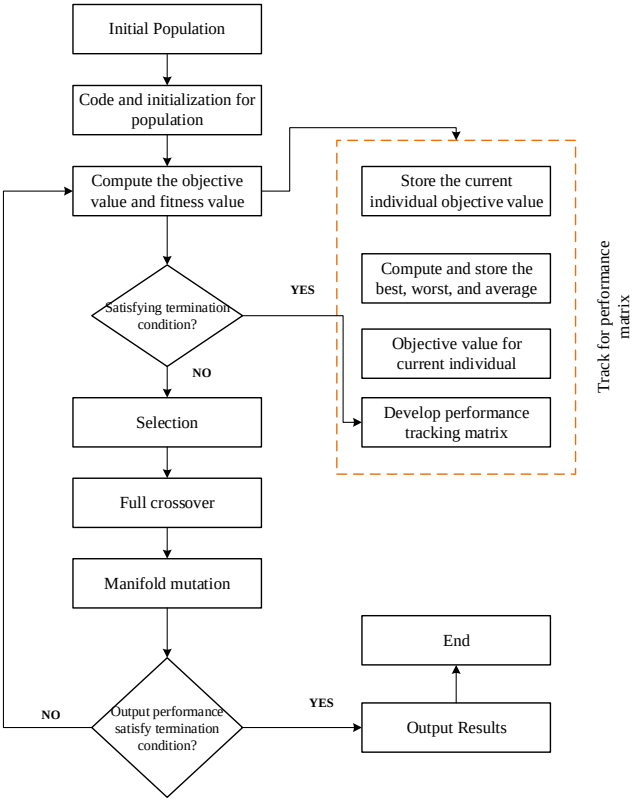


Figure 2. Evolutionary algorithm performance evaluation

Crossover Operator

Since the crossover point is selected randomly, the traditional GA has no prior knowledge about the efficiency of the offspring. The possibility of missing the best one is high; consequently, the good offspring exist in the candidate ones definitely, the whole crossover operator is developed in Figure 3 (for singleton and manifold operators). This performs the single point crossover at every positions. The number of offspring will eventually improve or increase by the geometric progression provided all candidate offspring are selected to mutate. The neighbouring competition mechanism is enhanced to choose the final offspring pair in relation to their situation complexity index, which has a positive correlation with probability of selection.

For manifold mutation operator, the traditional operator only generates one population, among which it has a low possibility to include the good individuals. The manifold mutation operators are developed as indicated in Figure 3, in order to increase the possibility. On the population for N *times*, it performs traditional

canonical mutation [38]. The neighbouring competition mechanism is adopted to choose the final offspring in relation to their overall simulations' complexity index given by $X_i^m = Mut(X^C, W_m)$. By randomly selecting a candidate mutation offspring population as the final on (X^m) with the probability that:

$$W_i^m = \frac{e^{\alpha x C(X_i^m)}}{\sum_{k=1}^{N} e^{\alpha x C(X_k^m)}} \qquad [5]$$

where $Mut(X^C, W_m)$ *donotes the canonical mutation on* X^C *with the mutation probability,* W_m [56]

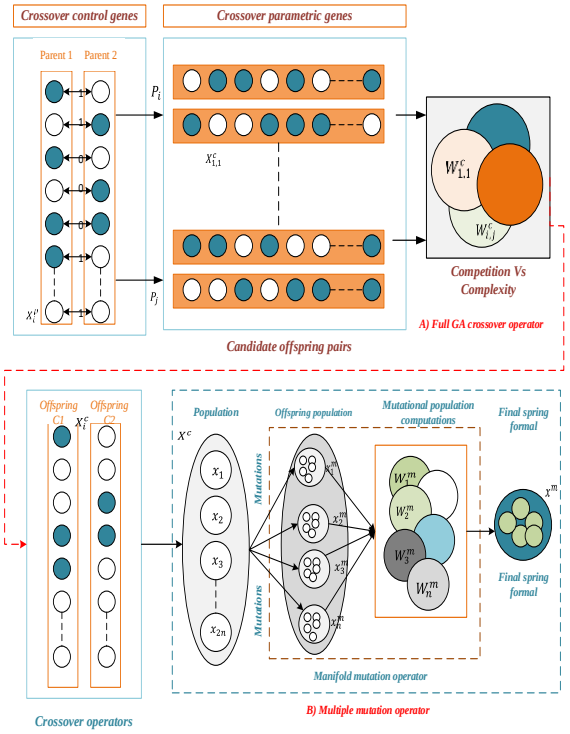


Figure 3. Full GA crossover operator (top) and multiple mutation operator (bottom)

Classical GA

Genetic algorithm is an optimization algorithm that is biologically inspired from the concept of natural selection. Since it is a population-based search algorithm, it uses the concept of survival of the fittest [39] by the Darwinian theory of evolution. Through iterative strategy the use of genetic operators on individual present in the population to produce new population. They are represented by key elements such as chromosome representation, selection, crossover, mutation, and fitness function. The algorithm of GA is as follows: A population (P) *of* n chromosomes are initialized randomly. The fitness of each chromosome P is calculated. Two chromosomes (example $C1$ *and* $C2$) are selected from the population P in relation to the fitness value. The single-point crossover operator with crossover probability (Cp) is applied on $C1$ *and* $C2$ to produce an offspring

say Os. Subsequently, the uniform mutation operator is employed on producing an offspring (Os) with probability of mutation (Mp) to generate $O's$ (new offspring). The new offspring $O's$ is placed in new population until the new population is complete. A typical classical GA is represented by the following pseudocode (algorithm).

Genetic Operator

There are a variety of Gas operators used to develop machine learning algorithm or hybrid of GA and ML. these operators include encoding schemes, crossover, mutation, and selection process. Figure 4 represents the GAs operator used in various health systems. The encoding scheme convert in a specific form and plays a significant role in most of the computational problems. The provided information has to be encoded in a particular bit string [40]. The encoding schemes are distinguished according to the problem domain. Currently, the most understood encoding schemes are binary, octal, hexadecimal, permutation, value-based, and tree. In binary encoding, each chromosome or gene is denoted as a string of $1\,or\,0$ [41], which is commonly used encoding scheme. Each bit represents the features of the solution and it enables the process quicker implementation of crossover and mutation operators. Nevertheless, it requires extra effort to convert into binary form and accuracy of algorithm depends upon the binary conversion. The binary encoding scheme is not adequate for some engineering design problems due to epistasis and natural representation due to the bit stream is changed according to the problem. In octal encoding scheme, the chromosome is denoted in the form of octal numbers. Similarly, in hexadecimal encoding scheme, the chromosome is denoted in the form of hexadecimal numbers [42]. The chromosome is denoted by the string of numbers that represents the position in a sequence in this encoding scheme. This is more significant in solving more complicated problems. In neural networks, it is used to find the optimal weights.

In crossover operators, the operators are used to generate the offspring by compounding the genetic information of two or more parents. The famous crossover operators are single-point, two-point, k-point, uniform, partially matched, order precedence preserving crossover, shuffle, reduced surrogate, and cycle. A random crossover point is chosen in a single point crossover and the genetic information of two parents which is beyond that point will be swapped with each other [43]. Whereas, in a two point and k-point crossover, two or more random crossover points are chosen and the genetic information will be swapped based on the segments that have been developed. Figure 4 shows the genetic information after swapping for single- and two-point crossover. It replaced the tail array bits of both the parent to generate the new offspring while in two-point crossover the middle segment of the parents is replaced to get the new offspring.

Mutation Operator

The genetic diversity from one population to the next is maintain by an operator called mutation. The eminent mutation operators are displacement, simple inversion, and scramble mutation. The operator that displaces a substring of a given individual solution within itself is referred to as displacement mutation. A

part of an individual solution is either exchanged with another part or inserted in another location, respectively in exchange mutation and inversive mutation operators. The simple inversion mutation operators (SIM) reverse the substring between any two specified locations in an individual solution. The randomly selected string and places it at a random location of which the SIM is an inversion operator that reverses. The scramble mutation operator places the elements in a specified range of the individual solution in a random order and determines whether the fitness value of the currently generated solution is improved or not.

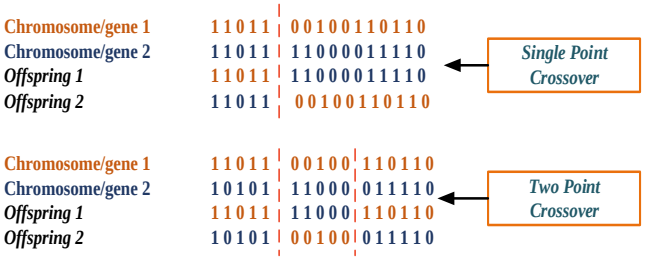


Figure 4. A typical scenario of a single point crossover (top) and two-point crossover (bottom)

Architecture for the proposed hybrid genetic algorithm and machine learning approaches

Currently, MATLAB is an extensively used programming environmental that enable large number of computer platform. Due to its simplicity and ease of use and learning ability, it is yet the fastest and powerful mathematical calculus and its extensively and straightforward data visualization tools make it a very appealing programming environment. The MATLAB setting provide a solid foundation on which to build and optimized toolboxes collection, and application specific functionalities. The toolbox for MATLAB is referred to as GPLAB (genetic programming). The proposed hybrid genetic algorithm and machine learning algorithms (HGAMLA) approaches encapsulate dynamic functionalities: *reset parameters function (which resets the parameter variables for the GPLAB algorithm using parameter variables with the default values), set parameters function, set operators function, desired obtained function (variables, indices, x parameters, black and white, size of x and y variables), accuracy and complexity function (variables, offsets, black and white, and size x*y), pareto function), and GP tree function.*

This demonstration of this architecture is a functional GPLAB toolbox which runs a symbolic regression problem called the polynomial quartic. The Figure 5 illustrates a block diagram used to develop the proposed HGAMLA architecture of the GA and ML algorithms for health-related problems and generic predictions and estimation criteria settings. In our experimental simulation exercises, 100 individuals for 40 or more generations with automatic adaptation of operator probabilities, drawing several plots in runtime, and finished with two additional post-run plots. The experimental results return all the variables of the algorithm needed to plot charts, continue runs, draw trees etc. and it also returns the best individuals found during the simulation exercises.

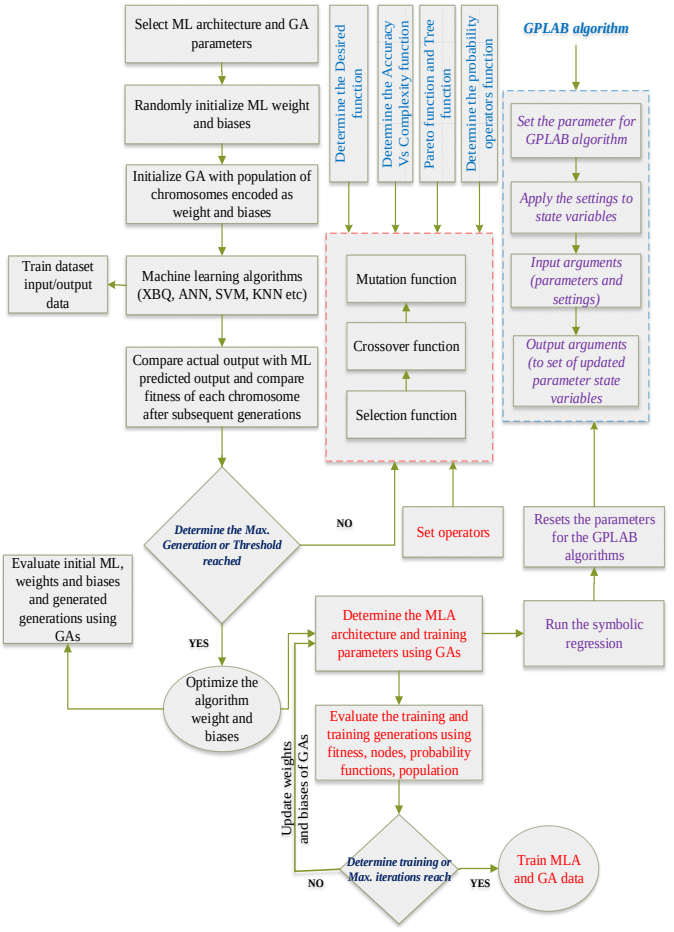


Figure 5. A block-diagram for the proposed HGAMLA architecture

Description of Datasets

The dataset used in this experimental analysis was presented in three folds: (a) 11-multiplexer represented in x and y variables, (b) the probability expressions represented in x and y variables, and (c) the heart disease dataset represented in x and y variables. The multiplexer was tested with a training set size of 2048 to determine the block and thread intertwined spirals (x variable) against the y variable. The probability expression in the x and y plot, which is represented between $[0,-1]\,and\,[0,1]$ determined by the $x-axis$, and $[0.3679, 1.0000]$ and $[1.1052,\ 2.7183]$ are determined by the y-axis. The heart disease dataset is presented as follows: A dataset is formed by taking into consideration some of the information of 779 individuals. The problem is based on the given information about each individual we have to calculate whether that individual will suffer from heart disease. There are 14 parameters in the dataset which suitable for the learning of GA and ML effectively and it denote variable x that encapsulate the following parameters:

Age- shows the age of the individual; *Sex*- shows the gender of the individual using the following format: 1 = male and 0 = female; *Chest-pain Type*- shows the type of chest-pain experienced by the individual using the following format: 1 = typical angina, 2 = atypical angina, 3 = non - anginal pain, 4 = asymptotic; *Resting Blood Pressure*- shows the resting blood pressure value of an individual in mmHg (unit); *Serum Cholesterol*- shows the serum cholesterol in mg/dl (unit), *Fasting Blood Sugar*- compares the fasting blood sugar value of an individual with 120mg/dl, If fasting blood sugar > 120mg/dl then : 1 (true), else : 0 (false); *Resting ECG*-0 = normal, 1 = having ST-T wave abnormality, 2 = left ventricular hypertrophy; *Max heart rate achieved*- shows the max heart rate achieved by an individual; *Exercise Induced Angina* : 1 = yes, 0 = no; *ST depression induced by exercise relative to rest* : shows the value which is integer or float; *Peak exercise ST segment* : 1 = upsloping, 2 = flat, 3 = down-sloping; *Number of major vessels (0-3) colored by fluoroscopy* : displays the value as integer or float; *Thalassemia* : shows the thalassemia : 3 = normal, 6 = fixed defect, 7 = reversable defect; *Diagnosis of heart disease* : shows whether the individual is suffering from heart disease or not : 0 = absence and 1,2,3,4 = present.

Model Training and Prediction

We can train our prediction model by analyzing existing data because we already know whether each patient has heart disease, probability expressions, and 11-multiplexer to determine the performance and accuracy of the GA model. This process is also known as supervised and learning modularization. The trained model is then used to predict if users suffer from heart disease, to determine which generation performs better, complexity and diversity of the GA model.

Experimental analysis and results

The experiment was conducted on MATLAB programming language platform, Intel(R) Core (TM)i7-7500U CPU running @ 2.70GHz and 2.90GHz, 16GB of RAM, and x64-based processor. The results were presented in table and figures to determine the following GPLAB functions: accuracy versus complexity, pareto front, GPLAB tree functionality, desired obtained function, GA operators, fitness, structural complexity, and population diversity. According to the analysis obtained, for 25 generations with 100 individual population were inserted and result for the generation were presented in Table 6. Astonishingly, for any set of the computation (25 generations and 100 individual), the GA produces different values based on the following parameters: generation type, used resources, fitness, test fitness, depth and umber of nodes.

The number of generations refers to the number of cycles prior the termination of the GA. In certain conditions, hundreds of loops are sufficient, however, in other situations we might need more and this is dependent on the type of problem and complexity. Also, depending on the GA design, sometimes this parameter is not utilized, particularly if the termination of the GA depends on deterministic criteria. The GA parameters are strongminded whether the GA finds an optimal or closer optimal solution, and efficient solution. The changes (increases or decreases) in the value of these parameters affects the result of the GA positively or negatively and subsequently selecting the right parameters is a nontrivial task.

During the simulation exercises, the test was conducted for 25 generations (top), 40 generations (bottom), and 100 individuals to determine the depth and nodes of the GA. The results shows that there were dynamic generation of different depth and nodes using the same generation and individual population for 1st, 2nd, ... Nth computation of the GPLAB algorithm as represented in Figure 6.1

Table 1
Experimental results showing 25 and 50 generations with 100 individual population

Experimental Computation	Number of Generation	Number of Individuals	Used Resources	Best obtained so far	Fitness	Test Fitness	Depth	Number of Nodes
1st Results	Initial Gen.	100 individual population	1024	67	10.103316	10.808240	6	16
2nd Results			1024	27	12.766600	25.066600	1	1
1st Results	1		1321	67	10.103316	10.808240	6	16
2nd Results			1116	146	10.842765	12.108143	5	6
1st Results	2		1220	232	8.895924	18.178717	4	7
2nd Results			988	243	10.368153	18.178717	6	15
1st Results	3		958	232	8.895924	18.178717	4	7
2nd Results			979	243	10.368153	18.178717	6	15
1st Results	4		856	448	6.897281	19.165462	10	24
2nd Results			1134	450	9.4726114	19.907328	14	39
1st Results	5		793	448	6.897281	19.165462	10	24
2nd Results			942	530	6.050000	17.366600	3	5
1st Results	6		582	581	6.667338	18.446664	6	11
2nd Results			976	530	6.050000	17.366600	3	5
1st Results	7		528	581	6.667338	18.446464	6	8
2nd Results			1489	530	6.050000	17.366600	3	14
1st Results	8		672	753	6.387716	12.654220	4	8
2nd Results			1930	817	4.799050	13.788685	6	14
1st Results	9		766	753	6.387716	12.654220	4	8
2nd Results			2877	817	4.799050	13.788685	6	14
1st Results	10		954	753	6.387716	12.387716	4	8
2nd Results			3096	1015	4.576328	15.511013	18	47
1st Results	11		889	753	6.387716	12.654220	4	8
2nd Results			3319	1059	4.335451	15.273556	18	43
1st Results	12		998	753	6.387716	12.654220	4	8
2nd Results			3608	1153	3.419405	15.799956	14	41
1st Results	13		1180	1304	5.901219	14.531697	6	15
2nd Results			3511	1153	3.419405	15.799956	14	41
1st Results	14		1470	1304	5.901219	14.531697	6	15
2nd Results			3511	1153	3.419405	15.799956	14	41
1st Results	15		1430	1444	3.521069	14.474716	8	17
2nd Results			3483	1153	3.419405	15.799956	14	41
1st Results	16		1530	1444	3.521069	14.474716	8	17
2nd Results			3451	1416	3.302804	13.774998	14	31
1st Results	17		1721	1647	2.468834	14.523477	12	23
2nd Results			3616	1519	2.712197	14.537375	14	34
1st Results	18		1657	1647	2.468834	14.523477	12	23
2nd Results			3848	1631	2.646642	14.537375	18	55
1st Results	19		1787	1647	2.468834	14.523477	12	23
2nd Results			3787	1631	2.646642	14.537375	18	55
1st Results	20		2060	1647	2.468834	14.523477	12	23
2nd Results			3598	1631	2.646442	14.537375	18	55
1st Results	21		2275	2064	1.902202	16.358613	11	22
2nd Results			3533	1631	2.646642	14.537375	18	55
1st Results	22		2352	2064	1.902202	16.358613	11	22
2nd Results			3478	1631	2.646642	14.537325	18	55
1st Results	23		2368	2064	1.902202	16.358613	11	22
2nd Results			3697	1631	2.646642	14.537378	18	55
1st Results	24		2368	2064	1.902202	16.358613	11	22
2nd Results			3908	1631	2.646642	14.537375	18	55
1st Results	25		2212	2064	1.902202	16.358613	11	22
2nd Results			3903	1631	2.646642	14.537375	18	55

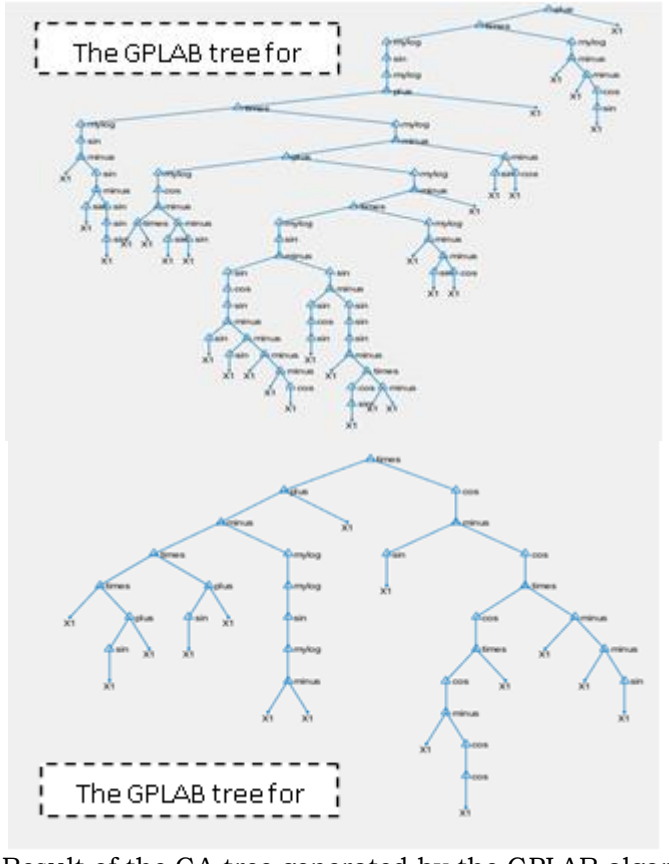
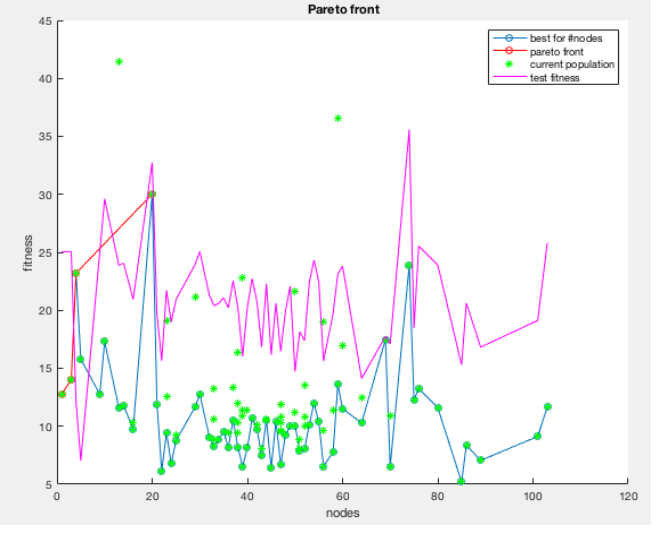


Figure 6.1. Result of the GA tree generated by the GPLAB algorithm for 25 generations (top) and 40 generations (bottom) for 100 individual population



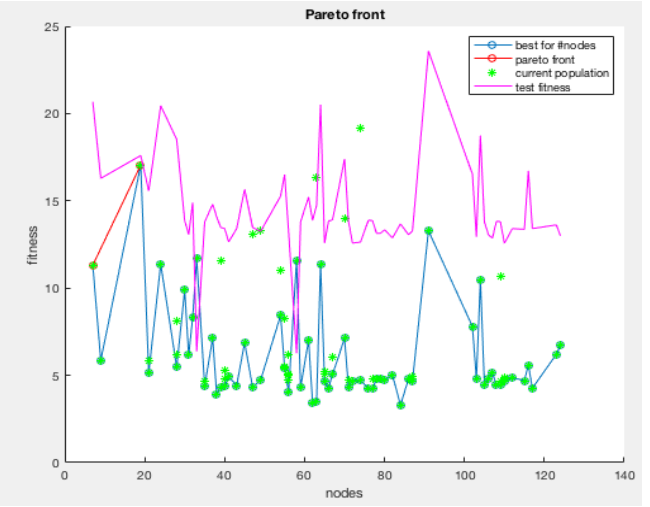


Figure 6.2. The pareto front derived for 25 generations (top) and 40 generations (bottom) and 100 individual population

According to the results obtained in Figure 6.2, over 100 nodes and 40 fitness was computed to determine the Pareto front algorithm (functional) and for 40 generations and 100 individual population (top). The line with blue indicates the best for the number of nodes, red line denotes the pareto front, green data point represents the current population, and pink line denotes the test for the fitness. Consequently, as the number of nodes increases the test fitness performance improves and the better the prediction obtained. Also, where the current population of data points are massive the better the fitness functions and more accuracy in regards to the heart diseases experimented against the GA/GP in our simulation exercises. However, it can be noted that as we continue to execute the algorithm more diverse pareto front is achieved but improves the prediction of the algorithm.

The GA or the GPLAB algorithm was used to determine the heart prediction for 25 generations (left) and 40 generations (right) and ages ranging from 0-90 years of people affected by heart diseases. The line with blue is called the fitness, yellow line the number of nodes (representing the ages against the generations), and red line indicates the level of complexity between the two parameters (fitness and nodes). According to the analysis, at generation 5 and between 10-35 years of age are affected by heart diseases than at generation 15 (less than 10 years old). However, at generation 20 and 25, ranging between 40-90 years are affected by heart diseases. In the other hand, the complexity of the algorithm increases as more generation are inserted in the algorithm and provides an intuitive prediction that aging population would be more affected by heart diseases and making it very difficult to survive infectious diseases as in the case of COVID-19 Pandemic. By comparing top and bottom figures, we found that the results are similar unlike at generation 40 where the aging population drops for ages greater than 40 years and so on (Figure 6.3).

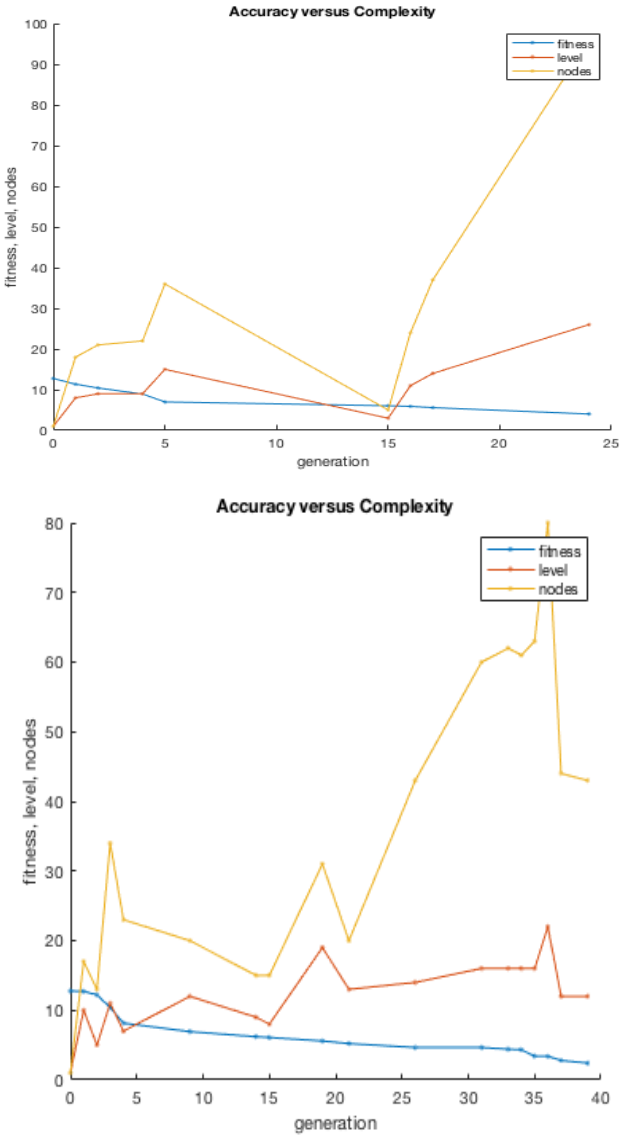


Figure 6.3. The accuracy versus the complexity for 25 generations and 100 individual population

This function was set to determine the probability predictions between -1 and +1 for 25 and 40 generations (Figure 6.4). By comparing generation from another generations, we found that the higher the generation the better the desired obtained algorithm. This further explained and provide an imperative result propounded by Darwinian theory. At generation 24 (top) and generation 39 (bottom), the probability of producing offspring that can provide more requisite accuracy and performance (80-100) % than compared to lower generation for 100 individual population.

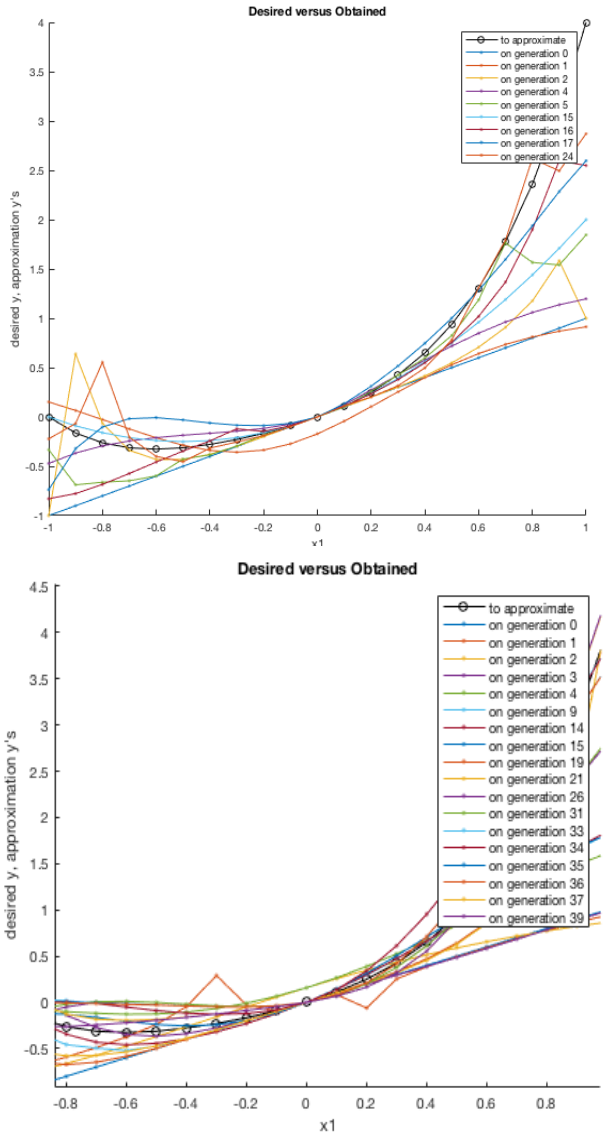
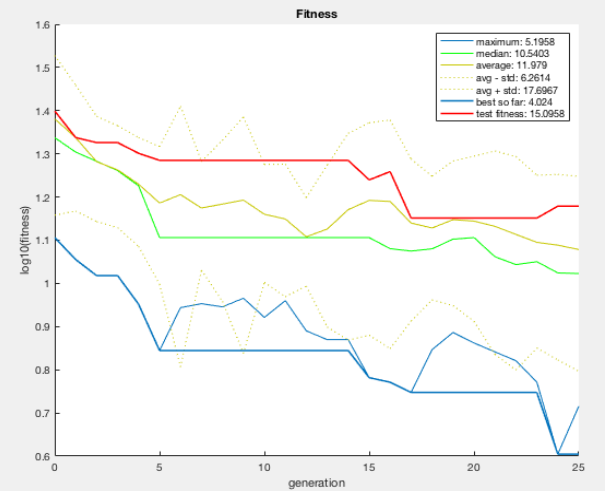


Figure 6.4. The desired versus obtained GPLAB algorithms for 25 generation (top) and 40 generation (bottom) with 100 individual population.



The probability computation executed as follows using 25 and 40 generations: light blue line indicates the maximum fitness (5.1958 and 3.2896, respectively), green line shows median (10.5403 and 4.9774, respectively), orange line shows the average (11.979 and 6.9774, respectively), negative broken orange line shows the average standard deviation (6.2613 and 6.4129, respectively), positive broken line orange line shows the average standard deviation (17.6967 and 9.6493, respectively), and the line with red colour indicates the test fitness (15.0958 and 17.2177, respectively). At generation 40, the algorithm performs better than at generation 25 when compared against test fitness, standard deviation, maximum depth, and best probability estimation so far (Figure 6.5). Conversely, at 40 generations the data are clustered around the mean (3.1764 and 9.6493) and at 25 generations, the data are more spread out or evenly distribution at certain point (6.2614 and 17.6967).

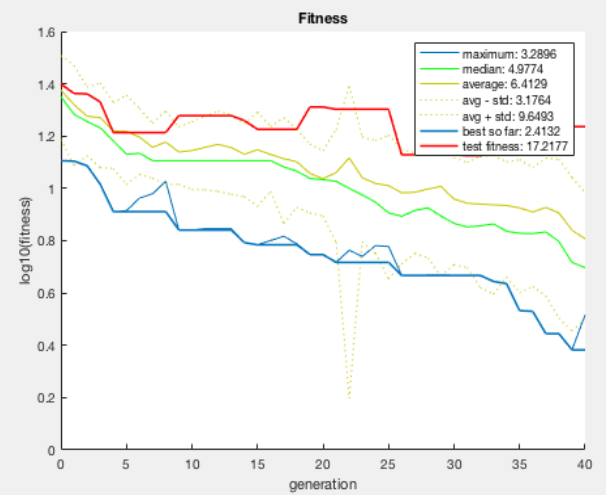


Figure 6.5. Fitness function for 25 generations (top) and 40 generations (bottom) using 100 individual population

The results obtained during the simulation depict that the higher the generational computation for a set of individual population the more complex the architecture of the GA algorithm. The experiment was conducted by comparing 25 and 40 generations against mathematical functions represented by different colour lines: maximum size plotted by thick orange line, best size result so far plotted by a dotted orange line with an asterisk average size, best result so far introns plotted by mauve line, best depth results so far plotted by blue line with asterisk, average depth plotted by blue line, and average tree fill plotted by black line with a dot. At generation 40, the maximum size complexity is 108 whereas at generation 25 the maximum size complexity is 96 (Figure 6.6).

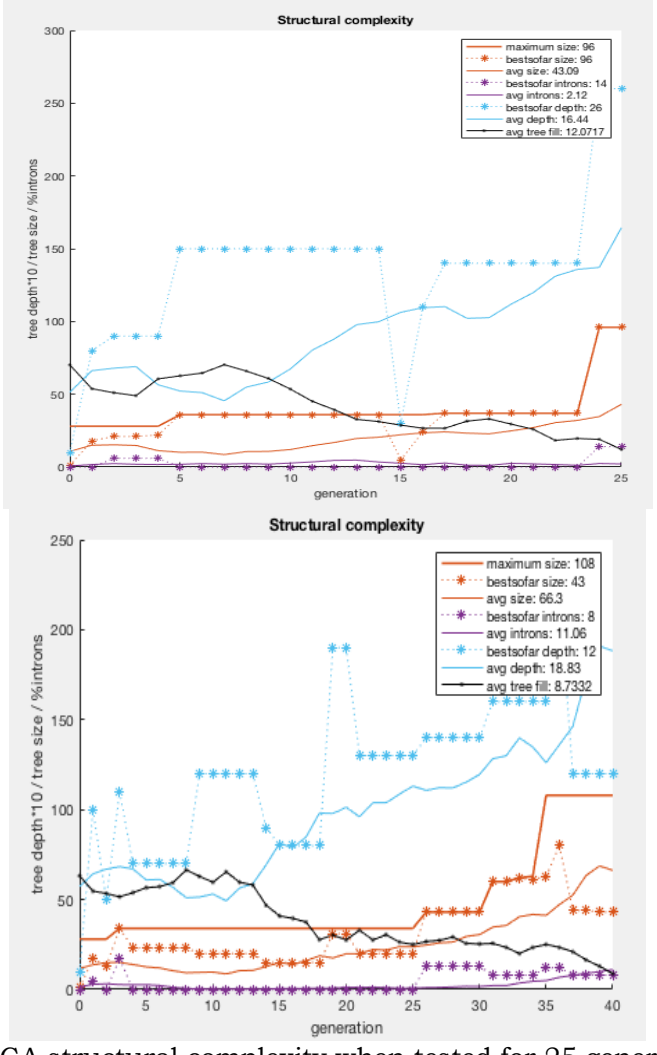
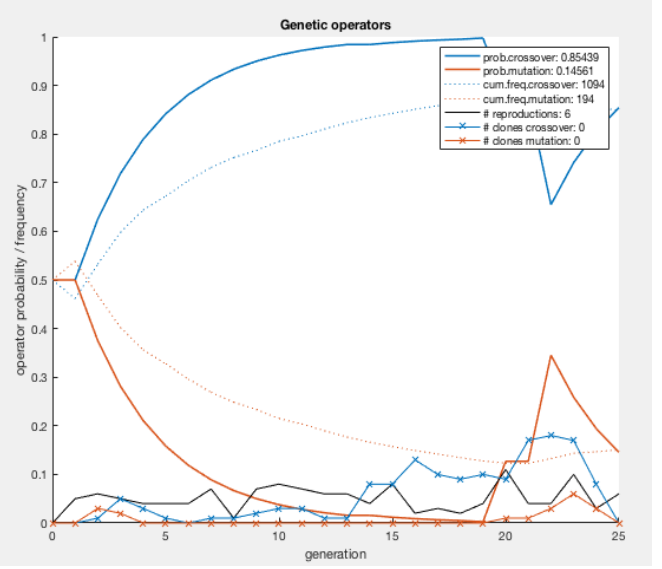


Figure 6.6. The GA structural complexity when tested for 25 generations (top) and 40 generations (bottom) using 100 individual population

The genetic operators used in this simulation performs using the heart diseases dataset against 25 and 40 generations. Here are the predictions obtained at 40th generation: the line in blue indicates the probability of crossover (89.2%), line in

orange shows the probability of mutation (10.8%), dotted blue line shows the cumulative frequency of crossover (1736), dotted orange line shows the cumulative frequency of mutation (303), thick green line shows the number of reproduction (4), line in blue with x shows the number of done crossovers (6), and line in orange with x shows the number of done mutation (2); which is superior over the results obtained at 25 generations with 85.4% and 14.5% probability of crossover and mutation, respectively. This means that between (85-89) % crossover rate that majority of the offspring are made by crossover (range of [0, 1]) and whereas between (10-14) % shows the probability of how many chromosomes should be mutated in one generation. The purpose of the mutation is to prevent the GA from converging to local optima.



The population diversity shows the total number of the population's inhabitants. By choosing the population size and sensitive diversity provided the population search space is small and thus means that little search space is available and consequently reach a local optimum (Figure 6.7). While if the population size is very large, the area of search is increased and the computational load becomes the higher and therefore, the size of the population must be reasonable. Comparing 25 and 40 generations, the analysis shows that between age 85-95 there are more clustered data indicating people that were more affected by heart diseases at generation 40 similarly at generation 25. However, less than 5 generations the population diversity falls from people's ages ranging 88 to peoples age diversity between 50-55 years computed by generation 25 with a unique generation of 98 (bottom figure). Similarly, between 5-10 and less than 5 generation the population diversity falls from population nearly 95 years to ages between 55-60 years with a unique generation of 92 (top figure).

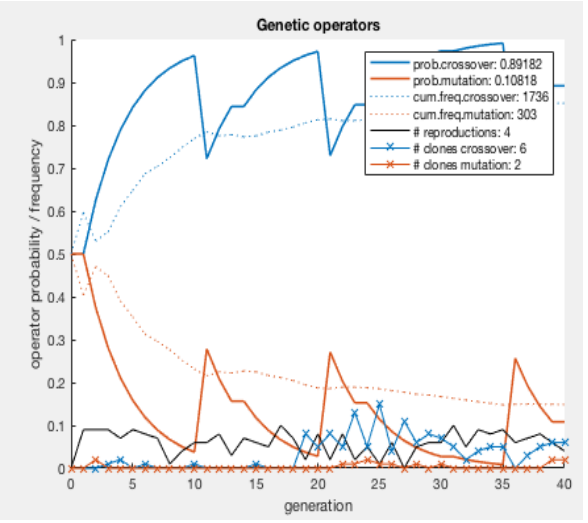


Figure 6.7. The genetic operators for 25 generations (top) and 40 generations (bottom) using 100 individual population

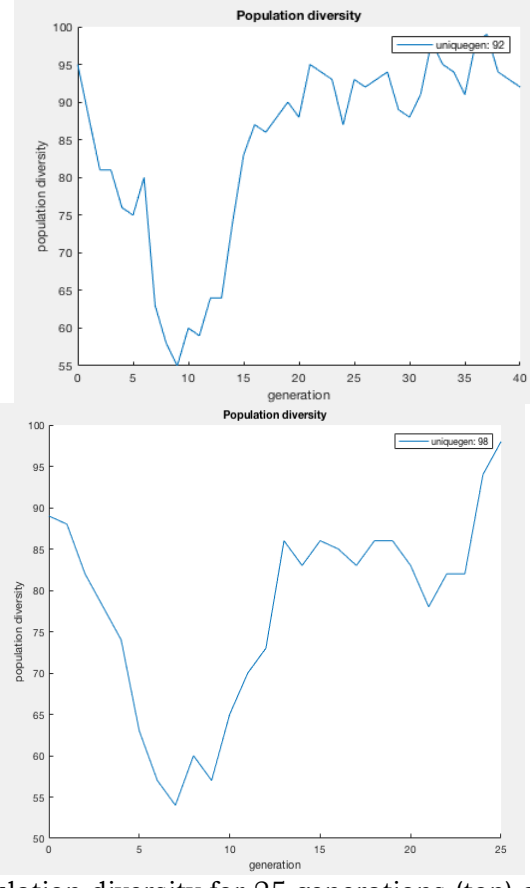


Figure 6.7. Population diversity for 25 generations (top) and 40 generations (bottom) using 100 individual population

Conclusion

A detail review has been conducted on genetic algorithm and machine learning and its applications to healthcare systems including the core concepts of GAs, its architecture, management evolution of GA enhancement, and evolution of desired settings. A theoretical and mathematical modelling were also presented in order to understand its architecture and modularization using GA operators. In this research work we presented a hybrid architecture and implemented the algorithm using MATLAB programming language platform. The GPLAB algorithm enable us to proposed a mode that can provide adequate and precise intuition on heart diseases (cardiovascular) prediction for various parameters. This paper provides the following contributions:

- An architecture by combining GA and ML algorithms in order to run the symbolic regression and eventually compute the GPLAB algorithm. The architecture includes desired obtained functionality, accuracy and complexity, pareto front functionality, structural complexity, and GP tree functionality.
- The test generation with fitness, test fitness, depth, and number of nodes to determine which generation performance better that the other.
- To determine the structural complexity, computational cost and flexibility when compared to PSO and ANFIS-algorithms using ANN approaches

References

1. Betarbet R., Sherer T. B., and Greenamyre J. T., “Animalmodels of Parkinson’s disease,” BioEssays, vol. 24, no. 4, pp. 308–318, 2002.
2. Hughes A. J., S. Daniel E., Kilford L. Lees J., “Accuracy of clinical diagnosis of diopathic Parkinson’s disease: a clinicopathological study of 100 cases,” BritishMedical Journal, vol. 55, pp. 181–184, 1992.
3. Little M. A., McSharry P. E., Hunter E. J., Spielman J., and Ramig L.O., “Suitability of dysphonia measurements for telemonitoring of Parkinson’s disease,” IEEE Transactions on Biomedical Engineering, vol. 56, no. 4, pp. 1015–1022, 2009.
4. Boyanov B. and Hadjitodorov S., “Acoustic analysis of pathological voices: a voice analysis system for the screening and laryngeal diseases,” IEEE Engineering in Medicine and Biology Magazine, vol. 16, no. 4, pp. 74–82, 1997.
5. Godino-Llorente J. I. and G´omez-Vilda P., “Automatic detection of voice impairments by means of short-term cepstral parameters and neural network-based detectors,” IEEE Transactions on Biomedical Engineering, vol. 51, no. 2, pp. 380–384, 2004.
6. Huang W., Li N., Lin Z. et al., “Liver tumor detection and segmentation using kernel-based extreme learning machine,” in Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC ’13), pp. 3662–3665, Osaka, Japan, July 2013.
7. Osman I.H. and Kelly J.P., 1996. Meta-heuristics: an overview. Meta-heuristics, pp.1-21.

8. Dorigo M. and Stützle T., 2003. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In Handbook of metaheuristics (pp. 250-285). Springer, Boston, MA.
9. Alrosan A., Alomoush W., Norwawi N., Alswaitti M. and Makhadmeh S.N., 2021. An improved artificial bee colony algorithm based on mean best-guided approach for continuous optimization problems and real brain MRI images segmentation. Neural Computing and Applications, 33(5), pp.1671-1697.
10. Rodrigues, L.R., 2021. A chaotic grey wolf optimizer for constrained optimization problems. Expert Systems, p.e12719.
11. Banupriya C.V. and Aruna D., 2021. D. Robust Optimization of electroencephalograph (EEG) Signals for Epilepsy Seizure Prediction by utilizing VSPO Genetic Algorithms with SVM and Machine Learning Methods. Indian J. Sci. Technol, 14, pp.1250-1260.
12. Hassani Z.I.M., El Barkany A., Jabri, A. Abbassi I.E. and Darcherif A.M., 2021. A Comparative Analysis of Metaheuristic Approaches (Genetic Algorithm/Hybridization of Genetic Algorithms and Simulated Annealing) for Planning and Scheduling Problem with Energy Aspect. SAE International Journal of Materials & Manufacturing, 14(4), p.363.
13. Game P.S. and Vaze D., 2020. Bio-inspired Optimization: metaheuristic algorithms for optimization. arXiv preprint arXiv:2003.11637.
14. Duma M. and Twala B., 2019. Sparseness reduction in collaborative filtering using a nearest neighbour artificial immune system with genetic algorithms. Expert Systems with Applications, 132, pp.110-125.
15. Jennings P.C., Lysgaard S., Hummelshøj J.S., Vegge T. and Bligaard T., 2019. Genetic algorithms for computational materials discovery accelerated by machine learning. NPJ Computational Materials, 5(1), pp.1-6.
16. Hussain A., Muhammad Y.S. and Sajid M.N., 2018. An efficient genetic algorithm for numerical function optimization with two new crossover operators. *International Journal of Mathematical Sciences and Computing*, *4*(4), pp.41-55.
17. Díaz, D., Valledor, P., Ena, B., Iglesias, M. and Menéndez, C., 2020. Improved Method for Parallelization of Evolutionary Metaheuristics. *Mathematics*, *8*(9), p.1476.
18. Sohail A., 2021. Genetic algorithms in the fields of artificial intelligence and data sciences. *Annals of Data Science*, pp.1-12.
19. Melanie M., An Introduction to Genetic Algorithms, A Bradford Book, The MIT Press, Cambridge, Mass, USA, 5th edition, 1999.
20. D. Whitley, “An executable model of a simple genetic algorithm,” Foundations of Genetic Algorithms, vol. 2, no. 1519, pp.45–62, 2014.
21. El-Sawy A.A., Hussein M.A., Zaki E.M. and Mousa A.A. An Introduction to Genetic Algorithms: A survey. A practical Issues. Int J Sci Eng Res, 5(1), pp.252-262, 2014
22. Wong S.H., Al-Hasani H., Alam Z. and Alam A., 2019. Artificial intelligence in radiology: how will we be affected?. *European radiology*, *29*(1), pp.141-143.
23. Muniyappan S. and Rajendran P., 2019. Contrast enhancement of medical images through adaptive genetic algorithm (AGA) over genetic algorithm (GA) and particle swarm optimization (PSO). *Multimedia Tools and Applications*, *78*(6), pp.6487-6511.
24. Heng H.P.S., Shu C., Zheng W., Lin, K. and Huang Z., 2021. Advances in real-time fiber-optic Raman spectroscopy for early cancer diagnosis: Pushing

the frontier into clinical endoscopic applications. Translational Biophotonics, 3(1), p.e202000018.
25. Silva E., Parente M., Brandão S., Mascarenhas T. and Natal Jorge R., 2019. Characterizing the biomechanical properties of the pubovisceralis muscle using a genetic algorithm and the finite element method. Journal of Biomechanical Engineering, 141(1), p.011009.
26. Zhou J, Krishnan S, Chong V, Huang J, eds. Extraction of tongue carcinoma using genetic algorithm-induced fuzzy clustering and artificial neural network from MR images. Engineering in Medicine and Biology Society, 2004 IEMBS'04 26th Annual International Conference of the IEEE; 2004: IEEE.
27. Hoh JK, Cha KJ, Park MI, Ting Lee ML, Park YS. Estimating time to full uterine cervical dilation using genetic algorithm. Kaohsiung J Med Sci 2012 Aug;28(8):423-428.
28. Gao J., Lyu T., Xiong F., Wang J., Ke W. and Li Z., 2021. Predicting the survival of cancer patients with multimodal graph neural network. IEEE/ACM Transactions on Computational Biology and Bioinformatics.
29. Rellum S.R., Schuurmans J., van der Ven W.H., Eberl S., Driessen A.H., Vlaar A.P. and Veelo D.P., 2021. Machine learning methods for perioperative anesthetic management in cardiac surgery patients: a scoping review. Journal of Thoracic Disease, 13(12), p.6976.
30. Gao F., Zhang Q., Han Z. and Yang Y., 2021. Evolution test by improved genetic algorithm with application to performance limit evaluation of automatic parallel parking system. IET Intelligent Transport Systems, 15(6), pp.754-764.
31. Di Francescomarino C., Dumas M., Federici M., Ghidini C., Maggi F.M., Rizzi W. and Simonetto L., 2018. Genetic algorithms for hyperparameter optimization in predictive business process monitoring. Information Systems, 74, pp.67-83.
32. Jun-hua L. and Ming L., 2013. An analysis on convergence and convergence rate estimate of elitist genetic algorithms in noisy environments. *Optik*, *124*(24), pp.6780-6785.
33. Peng Y., Luo X. and Wei W., 2014. A new fuzzy adaptive simulated annealing genetic algorithm and its convergence analysis and convergence rate estimation. *International Journal of Control, Automation and Systems*, *12*(3), pp.670-679.
34. Xia, Q., et at.: Test scenario design for intelligent driving system ensuring coverage and effectiveness. Int. J. Automot. Technol. 19(4), 751–758 (2018)
35. Gao F., et al.: Test scenario automatic generation strategy for intelligent driving systems. Math. Probl. Eng. 2019, 3737486 (2019)
36. Saaty T.L., 1986. Axiomatic foundation of the analytic hierarchy process. *Management science*, *32*(7), pp.841-855.
37. Ishizaka A. and Labib A., 2011. Review of the main developments in the analytic hierarchy process. *Expert systems with applications*, *38*(11), pp.14336-14345.
38. Rudolph G., 1994. Convergence analysis of canonical genetic algorithms. *IEEE transactions on neural networks*, *5*(1), pp.96-101.
39. Michalewicz Z., 1999. Genetic Algorithms+ Data Structures= Evolution Programs. Springer-Verlag, 1999. *Google Scholar Google Scholar Digital Library Digital Library*.

40. Katoch S., Chauhan S.S. and Kumar, V., 2021. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, *80*(5), pp.8091-8126.
41. Arifin H.H., Ong H.K.R., Daengdej J. and Novita D., 2019, July. Encoding technique of genetic algorithms for block definition diagram using OMG SysML™ notations. In *INCOSE International Symposium* (Vol. 29, No. 1, pp. 218-232).
42. Shanmugasundaram N., Sushita K., Kumar S.P. and Ganesh E.N., 2019. Genetic algorithm-based road network design for optimising the vehicle travel distance. *International Journal of Vehicle Information and Communication Systems*, *4*(4), pp.344-354.
43. Agarkar A.A., Ashraf M., Anandakumar Haldorai D., Hussain M.Z., Ram Subbiah D. and Bharathababu K., 2021. New Paths Generation Using Crossover And Mutation Genetic Operations For Mobile Robot To Find The Optimal Path In 2-D Static Environment. *NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal| NVEO*, pp.1675-1688.
44. Nyandra, M., Suryasa, W. (2018). Holistic approach to help sexual dysfunction. Eurasian Journal of Analytical Chemistry, 13(3), pp. 207–212.
45. Rinartha, K., & Suryasa, W. (2017). Comparative study for better result on query suggestion of article searching with MySQL pattern matching and Jaccard similarity. In 2017 5th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-4). IEEE.
46. Susilo, C. B., Jayanto, I., & Kusumawaty, I. (2021). Understanding digital technology trends in healthcare and preventive strategy. International Journal of Health & Medical Sciences, 4(3), 347-354. https://doi.org/10.31295/ijhms.v4n3.1769