Genetic Algorithms as a Feature Selection Tool in Heart Failure Disease

Asmaa Alabed[1], Chandrasekhar Kambhampati[2]and Neil Gordon[3]
[1] Research Visitor, Faculty of Science and Engineering, Computer Science Department, University of Hull, UK
[2] Reader, Faculty of Science and Engineering, University of Hull, UK
[3] Senior Lecturer, Faculty of Science and Engineering, University of Hull, UK

Abstract. A great wealth of information is hidden in clinical datasets, which could be analyzed to support decision-making processes or to better diagnose patients. Feature selection is one of the data pre-processing that selects a set of input features by removing unneeded or irrelevant features. Various algorithms have been used in healthcare to solve such problems involving complex medical data. This paper demonstrates how Genetic Algorithms offer a natural way to solve feature selection amongst data sets, where the fittest individual choice of variables is preserved over different generations. In this paper, a Genetic Algorithms is introduced as a feature selection method and shown to be effective in aiding understanding of such data.
Keywords: Feature selection, decision-making, algorithms, Genetic Algorithm.

1 Introduction

The performance of pattern modeling and classification is greatly affected if the dataset has a very high dimensionality. At the same time, the computational complexity, both numerically and in terms of space, increases ([1], [2], [3] and [4]). The rapid development of technology and the corresponding ability to gather data, has led to an explosion of the size of datasets. This does not imply that all of the features/attributes in a dataset are necessary and sufficient, in terms of the information required to determine patterns accurately and provide predictions. Feature selection methods can be used to identify and remove redundant or irrelevant features from a given dataset without loss of accuracy in predictions. At the same time, feature selection can provide an insight into the features in terms of their importance [1, 3].

Feature selection can be defined as the process of choosing a minimum subset of features from the original dataset where [3]:

- The classification accuracy does not significantly decrease
- The resulting class distribution, given only the values for the selected features, is a close as possible to the original class distribution, given all the features.

Feature selection algorithms consist of four key steps: subset generation, evaluation subset, stopping criteria and result validation ([4], [5]). Subset generation is a heuristic search that generates a subset of features for evaluation procedures. Each subset

generated is evaluated by certain evaluation criteria to determine the 'goodness' of the generated subset of the features. The generated subset is validated by carrying out different tests and comparisons with the previous best subset. If a new subset is found not to be better, then the previous best subset is replaced by the new subset. This process is repeated until stopping criteria is reached as shown in Fig. (1).

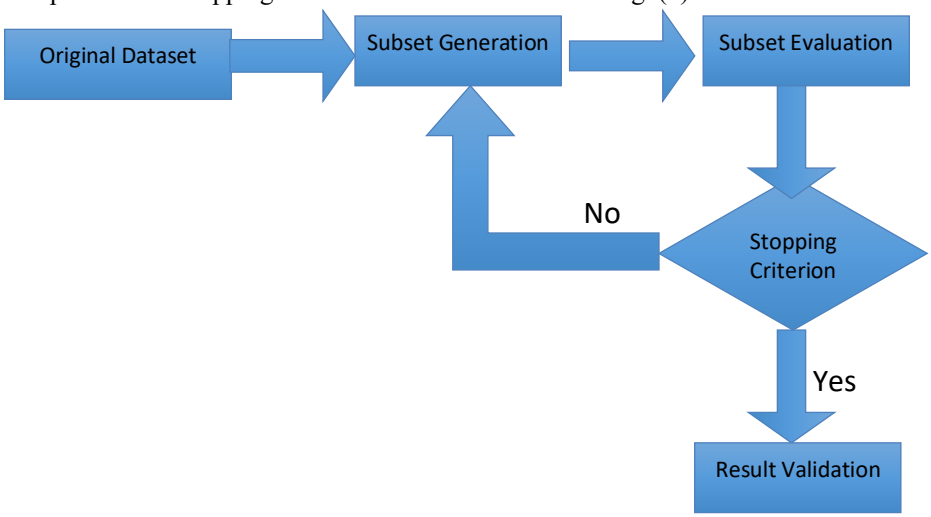

Fig. 1 Four steps for feature selection process [3]

There are three approaches to feature selection: filter, wrapper or embedded approach [1], [6], [7], and [8]. Filter feature selection methods apply a statistical measure to assign a weight to each feature according to its degree of relevance. Filters independently measure the relevance of feature subsets to classifier outcomes where each feature is evaluated with a measure such as the distance to outcome classes, correlation or Euclidean distance. All the features in the dataset are then ranked according to these measures. The advantages of filter methods are that they are fast, scalable and independent of a learning algorithm. The most distinguishing characteristic of the filters is that the relevance index is calculated solely on a single feature without considering the values of other features [9]. Such implementation implies that the filter assumes orthogonality of features, which is often not true in practice. Therefore, filters omit any conditional dependences (or independence) that might exist, which is known to be one of the weaknesses of filters. Wrapper methods use the predictor as a black box and the predictor performance as the objective function to evaluate the feature subset [1]. The expression wrapper approach covers the category of variable subset selection algorithms that apply a learning algorithm in order to conduct the search for the optimal or a near-optimal subset [10]. The number of the created subset is equal to 2n becomes an NP-heard problem, a suboptimal subset is selected by applying the search algorithm that finds the subset heuristically. The embedded approach is with specific learning algorithms that perform feature selection

in the process of training. An important aspect of using feature selection algorithms is that they can improve inductive learning, either in terms of general capabilities, learning speed or reducing the complexity of the induced model and classification accuracy [2]. Often a compromise is reached in achieving these various objectives in a feature selection approach.

This work focuses on applying the Genetic Algorithms (GAs) as a feature selection technique for Heart Failure data sets in order to improve the classification accuracy and reduce the number of features. The GAs was tested as a 'wrapper' features selection method. GAs makes up one of the global methods for optimization, for searching in complex, large and multidimensional datasets ([1], [9], [11], [12], [13], [14], [15]). First, the GAs was built using different populations, generations, and neighborhoods (k). Secondly, selected features from the best performing GAs were tested again, using different populations and k values. Finally, the GAs investigation was carried out by setting a population of up to 800. In terms of classification accuracy, two different classifiers were used namely; Bayes Nets (BN) and Random Forest (RF).

2 Genetic Algorithms (GAs) as a Feature Selection Tool

GAs is optimizing and search technique based on natural biological evolution theory (survival for the fittest) ([1], [6], [7]). Over successive generations, the population "evolves" toward an optimal solution. The advantage of GAs over others is that allows the best solution to emerge from the best of the prior solution. The idea of GAs is to combine different solutions generation after generation to extract the best genes from each one. GAs can manage data set with a large number of features and it does not need any extra knowledge about the problem under study. The subsets of features selected by genetic algorithms are generally more efficient than those obtained by classical methods of feature selection since they can produce a better result by using a lower number of features [16].

The individuals in the genetic space are called chromosomes. The chromosome is a collection of genes where the real value or binary encoding can generally represent genes. The number of genes is the total number of features in the data set. If genes are binary values that mean each chromosome in the GAs population has value of 1 or 0. A value of (1) in a chromosome representation means that the corresponding feature is included in the specified subset. A value of (0) indicates that the corresponding feature is not included in the specified dataset. Each solution in a genetic algorithm is represented through chromosomes. The collection of all chromosomes is called 'population' as shown in Fig. (2). As a first step of GAs, an initial population of individuals is generated at random or heuristically. In each generation, the population is evaluated using fitness functions.

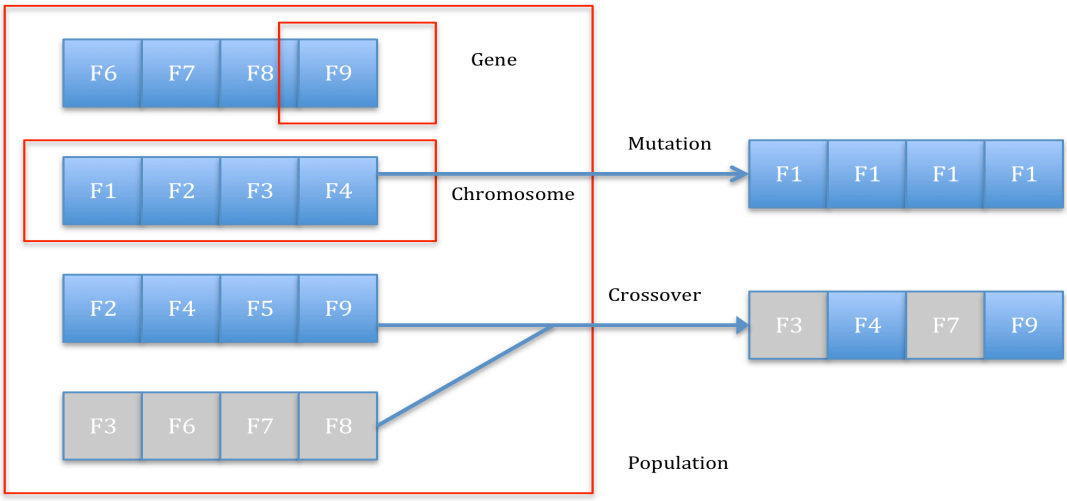

Fig. 2 Genetic Algorithms

The next step is the selection process, where in the high fitness chromosomes are used to eliminate low fitness chromosomes. Better feature subsets have a greater chance of being selected to form a new subset through crossover or mutation. In this manner, good subsets are "evolved" over time [17]. The commonly used methods for reproduction or selection are Roulette-wheel selection, Boltzmann selection, Tournament selection, Rank selection, and Steady-state selection. The selected subsets are ready for reproduction using crossover and mutation. The crossover combines different features from a pair of subsets into a new subset as shown in Fig. (2). Cross over tends to create a better string. The mutation changes some of the values (thus adding or deleting features) in a subset randomly as shown in Fig. (2). The new population generated undergoes the further selection, crossover, and mutation until the termination criterion is satisfied or maximum numbers of generation were reached as shown in Fig. (3).

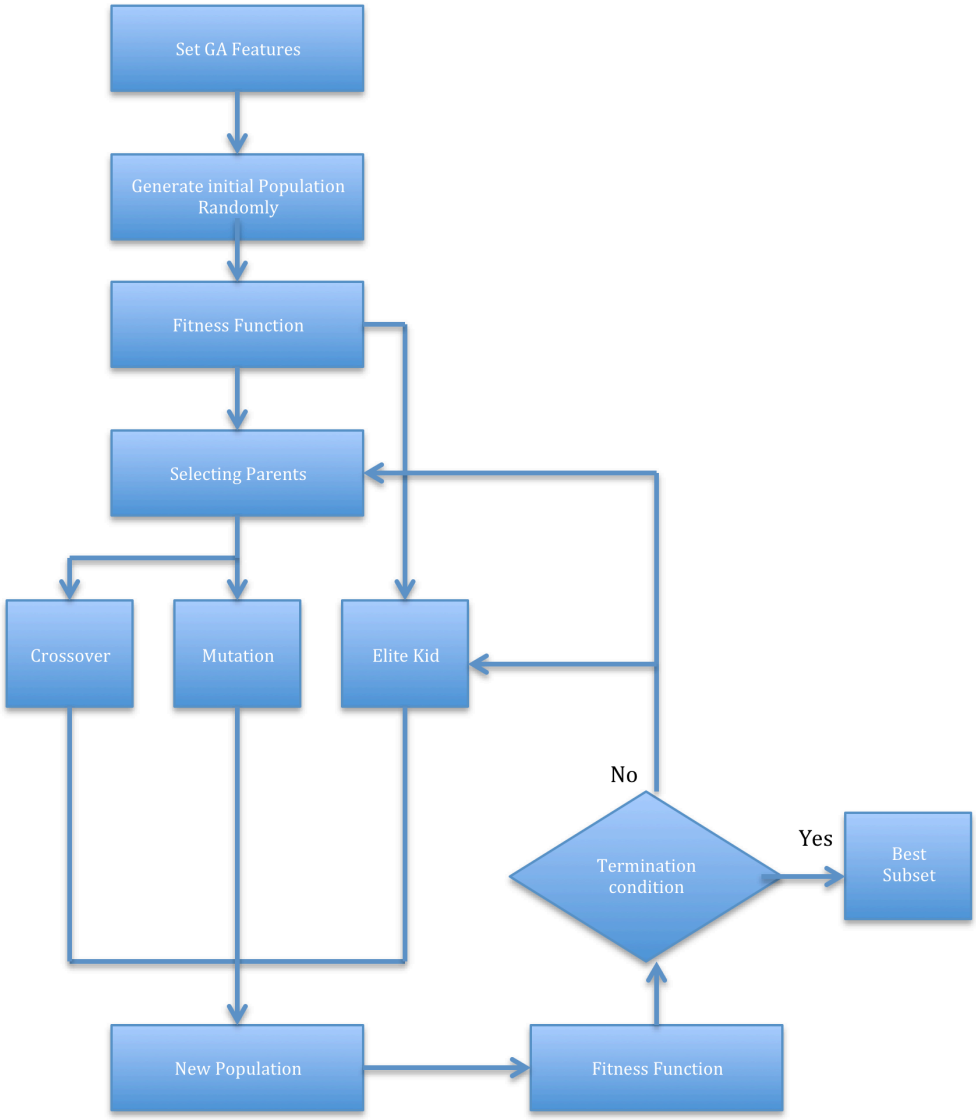

Fig.3 GA as a feature selection [9]

3 Genetic Algorithms (GAs) Experiments

In this experiment, the Matlab GAs toolbox is used. GAs started by initially creating a random population then it will be evaluated by using a fitness function. The elite kids have then pushed automatically to the next generation and the remaining kids in the

current population are allowed to genetically pass through the function of cross over and mutilation to form a new generation [13]. The dataset is a real-life heart failure dataset.

In this dataset, there are 60 features for 1944 patient records. The class is “dead” or “alive”. The data sets were imputed by different methods such as Concept Most Common Imputation (CMCI) and Support Victor Machine (SVM). Different classification methods have been applied to these datasets to select which dataset will be trained [18]. The performance of these datasets was measured using accuracy, sensitivity, and specificity. SVM dataset was chosen since its accuracy, sensitivity and specificity were the best. The experiments were designed using Weka (version 3.8.1-199-2016). The accuracy was the best using Bayes net, random forest, decision tree, REP tree, J48. In this work, BN and RF were selected as classifiers since the accuracy was the highest value as shown in Table 1. The feature’s name is displayed in Appendix A.

Table 1 Imputed dataset

	Classification Algorithms	Accuracy	Sensitivity	Specificity
SVM	J48	77.8%	86.09%	52.99%
	Random Forest	84.72%	96.78%	48.45%
	Decision Tree	83.6%	95.27%	48.87%
	REP tree	81.2%	92.66%	46.8%
	Bayes.Net	87.34%	89.1%	82.06%

GAs parameters are shown in Table 2.

Table 2 GAs parameters

GAs Parameter	Value
Number of Features	60
Population size	50,75,100
Genomelength	60
Population type	Bite Strings
Fitness function	kNN-based classification error
Number of generation	100,130

Crossover	Arithmetic crossover
Mutation	Uniform mutation
Selection Scheme	Roulette wheel
Elite Count	2

As discussed above, the number of chromosomes used in a particular implementation is of particular interest, in evolutionary computation ([19], [20], [21]). Various results about the appropriate population size can be found in the literature [22], [23]. Researchers usually argue that a "small" population size could guide the algorithm to poor solutions ([24], [25], [26]) and that a "large" population size could make the algorithm expend more computation time in finding a solution ([24], [26],[27]).

For GAs to select a subset feature, a fitness function must be defined to evaluate the fitness of each subset feature. In this work, the fitness function was based on Oluleye's fitness function [14] that is based on error minimization and reducing the number of features. The fitness of each chromosome in the population is evaluated using kNN-based fitness function as defined in FSP1. The kNN algorithm computes Euclidean distance between test data and the training sets then finds the nearest point from the training set to the test set. The individuals are evaluated and their fitness is ranked based on the kNN based classification error. Individuals with minimum fitness have a better chance of surviving into the next generation. GA ensures that the GA reduces the error rate and picks the individual with the best fitness error rate that will reduce the number of features as well.

The model representation for KNN is the entire training dataset. Predictions are made for a new data point by searching through the entire training set for the K most similar instances (the neighbours) and summarizing the output variable for those K instances. For classification problems, this might be the mode (or most common) class value.

Roulette wheel selection was used as the selection method for these experiments as it was discussed in the earlier section. With roulette wheel selection, each individual is assigned as a 'slice' of the wheel in proportion to the fitness value of the individual. Therefore, the fitter an individual is, the larger the slice of the wheel. The wheel is simulated by normalization of fitness values of the population of individuals.

4 Results and Discussions

In this work, different population size was tested to find the optimal size. The optimal accuracy was achieved using GAs where the population is 100 and k =5 as shown in Table 3. The number of features was dropped from 60 to 27 features. As K is increased,

the accuracy changes as well as shown in Table 3. The researcher should try different values for k to reach the optimal solution. BN accuracy was 87.8% that can be interpreted as predicting 12.2% as being a false classified.

Table 3 the performance of classification algorithms using various GA variables

Population	**RF Accuracy**		
	K=3	K=5	K=9
100	84.82%	85.03	85.4%
75	83.75%	80.76%	84.51%
50	86.7%	85%	83.69%
Population	BN Accuracy		
	K=3	K=5	K=9
100	83.79%	**87.8%**	86.21%
75	84.92%	82.25%	83.84%
50	86.7%	83.07%	85.18%

The number of features was 60, for kNN, the trick is in how to determine the similarity between the data instances. The simplest technique, if the attributes are all of the same scale (all in inches for example), is to use the Euclidean distance. A number it can be calculated directly based on the differences between each input variable. In this case, it is impossible because the features are recorded on different scales.

The idea of distance or closeness can break down in very high dimensions (lots of input variables) which can negatively affect the performance of the algorithm on this problem. This is called the curse of dimensionality.

In order to improve the GAs performance, it's suggested only use those input variables that are most relevant to predicting the output variable [2829]. In the next experiments, the selected features from GAs, where accuracy was the highest (population 100, generation 130, k=5), were tested and the results are shown in Table 4. BN accuracy was 86.77% that can be interpreted as predicting 13.233% as being a false classified. Sensitivity of 91.0% can be interpreted as the algorithm predicting 8.91% dead when they should have been predicted as alive. Specificity shows a performance of 74.02% which can be interpreted as the algorithm predicting 25.98% FP (alive).

Table 4 the results of GAs for different generations and k using 27 features

Features Selection	**Classification Algorithms**	**Accuracy**	**Sensitivity**	**Specificity**
GAs 100,130,k=3 2,4,6,16,21,23,31,32,34,39, 41,46	Random Forest	83.02%	93.35%	51.95%
	Bayes.Net	86.36%	91.09%	72.16%
GAs 100,130,k=5 4,6,16,21,23,24,25,31,32, 34,39,45,48,55	Random Forest	84.92%	94.65%	55.67%
	Bayes.Net	**86.77%**	91.02%	74.02%
GAs 100,130,k=9 6,16,21,23,24,25,32,34,39, 41,55	Random Forest	83.84%	94.03%	53.19%
	Bayes.Net	84.49%	90.88%	67.21%
GAs 75,130,k=3 4,21,23,31,32,34,39,40 41,55	Random Forest	83.02%	93.35%	49.89%
	Bayes.Net	86.36%	92.39%	60.61%
GAs 75,130,k=5 4,6,16,21,23,25,31,32, 42,46,55	Random Forest	82.71%	94.24%	50.72%
	Bayes.Net	84.46%	91.02%	69.27%
GAs 75,130,k=9 6,16,23,24,25,32,39,41,55	Random Forest	84%	94.04%	53.81%
	Bayes.Net	85.39%	91.91%	65.77%
GAs 50,130,k=3 2,4,6,21,23,31,32,39,41	Random Forest	82.20%	93.42%	48.45%
	Bayes.Net	84.00%	91.43%	61.64%
GAs 50,130,k=5 4,6,24,28,31,32,34,39,40 41,46,48	Random Forest	83.12%	93.35%	52.57%
	Bayes.Net	85.85%	90.95%	70.51%
GAs 50,130,k=9 4,7,16,21,23,23,28,32,34 40,41,45,46,48	Random Forest	84.1%	96.02%	48.24%
	Bayes.Net	85.03%	90.95%	67.21%

The performance of GAs has not improved significantly regarding the accuracy; however, the number of selected features was reduced from 27 to 14 features as shown in Table 4.

The number of populations was increased to 400,600, and 800 in order to investigate if there will be any improvement on the GAs performance. Table 5 shows the accuracy for different generations, the optimal accuracy is 86.3% which is less than 87.7% that was achieved using 100 populations. The results showed that it took a long time and almost the same number of selected features.

Table 5 GAs results for 400,600, & 800 populations where k=3

Feature Selection	**Classification Algorithms**	**Accuracy**	**Sensitivity**	**Specificity**
Genetic algorithms (400,100) K=3 1,5,7,13,15,16,17,18,24,28,39,31,33,34,38,39,40,42,45,46,49,55,59,60	Random Forest	85.03%	95.54%	53.40%
	Bayesian Networks	86.3 %	88.8%	78.96%
Genetic algorithms (600,100) K=3 1,7,10,11,14,15,16,17,19,23,24,25,28,29,30,32,33,39,40,43,46,48,50,52,53,55,56,58,59,60	Random Forest	84.77%	96.23%	50.3%
	Bayesian Networks	85.75%	88.8%	76.9%
Genetic algorithms (800,100) K=3 1,2,3,5,6,7,8,14,19,20,28,30,35,37,38,42,45,47,48,50,54,59,60	Random Forest	82.76%	94.37%	47.83%
	Bayesian Networks	82.30%	87.11%	67.83%

Al Khaldy [29] investigated several feature selection methods including wrapper and filter methods and used a representative set of classification methods for evaluating the features selected. These methods enabled the identification of a core set of features, from the same dataset. As shown in Table 11, there are many common features between his findings and this work.

Table 11 Common Factors

	GA	Al Khaldy
Urea(mmol/L)	1	4
UricAcid(mmol/L)	1	4
MCV(fL)	1	5
Iron(umol/L)	1	6
Ferritin(ug/L)	1	4
CRP(mg/L)	1	3
White Cell Count	1	2
CT-proET1	1	7
LVEDD(HgtIndexed)	1	6
E	1	3
Height(Exam)(m)	1	2
PCT	1	1
MR-proADM	1	5
FVC(L)	1	6

5 Conclusions

The experiments in this paper demonstrate the feasibility of using GA as a feature selection tool for large data sets. While the number of features was reduced from 60

to 27 features using GA, the accuracy - being 87.8% - was almost the same. In order to improve the GA performance, the input variables were the most relevant to predicting the output variable (27 features). Whilst the performance of GA has not improved significantly regarding the accuracy, the number of selected features was reduced from 27 to 14 features thus identifying the most important features. GA picked up the three variables that are used by clinicians in diagnosing heart failure [30], namely Urea, Uric acid and Creatinine. In order to validate the performance of GA, different feature selection experiments were carried out using WEKA tool to show this is a viable technique for such problems.

References

1. Chandrashekar, G., Sahin, F.: A survey on feature selection methods, Computer and Electrical Engineeing 40(1), 16-28 (2014).
2. Panthong ,R., Srivihok, A.: Wrapper feature subset selection for dimension reduction based on ensemble learning algorithm, Procedia Computer Science 72, 162-169 (2015).
3. Dash, M ., Liu, H. :Feature selection methods for classifications, Intelligent Data Analysis 1(3), 131-156 (1977).
4. Kumar V., Minz S., Feature Selection: A Literature review, Smart Computing Review 4(3), 2014.
5. Liu, H.,Yu, L.: Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17(4), 491 - 502 (2005).
6. Jain, a. K., Duin, R. P. W., Mao, J., Statistical pattern recognition: A review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 4-37(2000).
7. Cai, J., Luo, J., Wang,S., Yang,S.: Feature selection in machine learning :A new perspective, Neurocomputing 300,70-79(2018).
8. Shikhpour, R., Sarram, M. A., Gharaghani, S., Ali, M., Chahooki, Z.: A Survey on semi-supervised feature selection methods, Pattern Recognition 64, 141-158 (2017).
9. Masilamani, A. Anupriya, Iyenger, N.: Enhanced prediction of heart disease with feature subset selection using genetic algorithm, International Journal of Engineering Sciences and Technology 2(10), 5370-5376 (2010).
10. Kohavi, R., John, G.H. : The wrapper approach. In H. Liu and H. Motoda, editors, Feature extraction, construction and selection, page 33. Kluwer Academic Publisher, 1998.
11. Tiwari, R., Singh, M. P.: Correlation-based attribute selection using genetic algorithm, International Journal of Computer Applications 4(8), 28-34 (2010).
12. Karthikeyan, T., Thangaraju, P.: Genetic algorithm based CFS and Naïve Bayes algorithm to enhance the predictive accuracy, Indian Journal of Science and Technology 8(27), (2015).
13. Oluleye, B., Armstrong, L. J., Leng, J., Diepeveen, D.: Zernike Moments and Genetic Algorithm: Tutorial and Application, British Journal of Mathematics & Computer Science 4(15), 2217-2236 (2014).
14. Alander, J. T.: On optimal population size of genetic algorithms, In Proceedings of the IEEE Computer Systems and Software Engineering, 65–69 (1992).
15. Jabbar M., Deekshatulu B. L., Chandra P.: Classification of heart disease using K-Nearest neighbor and genetic algorithm, International conference on Computational Intelligence: Modeling Techniques and Applications, vol. 10, pp. 85-94, (2013).

16. Boggia R., Riccardo L., Marco T.: Genetic Algorithms as a strategy for feature selection, Journal of Chemometrics, 6(5), 267 – 281 (1992).
17. Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection, Pattern Recognition Letters, 10, 335-347 (1989).
18. Khaldy M., Kambhampati C.: Performance analysis of various missing value imputation methods on heart failure dataset, SAI Intelligent systems Conference (2016), Sep 20-22. London UK.
19. Alander, J. T.: On optimal population size of genetic algorithms, In Proceedings of the IEEE Computer Systems and Software Engineering, pp. 65–69, (1992).
20. Diaz-Gomez, P. A, Hougen, D. F. : Initial population for genetic algorithms: A metric approachs, Proceedings of the 2007 International Conference on Genetic and Evolutionary Methods, Las Vegas, GEM 2007, June.
21. Piszcz, A., Soule, T.: Genetic programming: Optimal population sizes for varying complexity problems, In Proceedings of the Genetic and Evolutionary Computation Conference, pp. 953–954 (2006).
22. Reeves, C. R.: Using genetic algorithms with small populations, In Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest (ed.)(Morgan Kaufmmann, San Mateo), pp. 92-99, (1993).
23. Roeva, O.: Improvement of genetic algorithm performance for identification of cultivation process models, Advanced Topics on Evolutionary Computing, Book Series: Artificial Intelligence Series-WSEAS, pp. 34-39 (2008).
24. Koumousis, V. K., Katsaras, C. P.: A sawtooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance, IEEE Transactions on Evolutionary Computation, 10(1), 19–28 (2006).
25. Pelikan, M., Goldberg, D. E., Cantu-Paz, E.: Bayesian optimization algorithm, population sizing, and time to convergence, Illinois Genetic Algorithms Laboratory, University of Illinois, Tech. Rep., (2000).
26. Lobo, F. G., Goldberg, D. E.: The parameterless genetic algorithm in practice, Information Sciences Informatics and Computer Science, 167(1-4), 217–232 (2004).
27. Lobo, F. G., Lima, C. F. : A review of adaptive population sizing schemes in genetic algorithms, In Proceedings of the Genetic and Evolutionary Computation Conference, pp. 228–234 (2005).
28. Raymer, M. L., Punch, W. F., Goodman, E. D., Khun, L. A., Jain, A. K.: Dimensionality reduction using genetic algorithm, IEEE Transactions of Evolutionary Computation, 4(2), , 164-171 (2000).
29. Mohammad Al Khaldy, Clinical data issues and autoencoder framework to compress datamining methodology, PhD thesis, University of Hull, June 2017.
30. Lisa Kirke, Datamining for heart failure: An Investigation into the Challenges in Real Life Clinical Datasets, PhD thesis, The University of Hull, June 2015.

Appendix A

1	Age	31	MR-proADM
2	Sodium(mmol/L)	32	CT-proET1

3	Potassium(mmol/L)	33	CT-proAVP
4	Chloride(mmol/L)	34	PCT
5	Bicarbonate(mmol/L)	35	Rate(ECG)(bpm)
6	Urea(mmol/L)	36	QRSWidth(msec)
7	Creatinine(umol/L)	37	QT
8	Calcium(mmol/L)	38	LVEDD(cm)
9	AdjCalcium(mmol/L)	39	LVEDD(HgtIndexed)
10	Phosphate(mmol/L)	40	BSA(m^2)
11	Bilirubin(umol/L)	41	LeftAtrium(cm)
12	AlkalinePhophatase(iu/L)	42	LeftAtrium(BSAIndexed)
13	ALT(iu/L)	43	LeftAtrium(HgtIndexed)
14	TotalProtein(g/L)	44	AorticVelocity(m/s)
15	Albumin(g/L)	45	E
16	UricAcid(mmol/L)	46	Height(Exam)(m)
17	Glucose(mmol/L)	47	Weight(Exam)(kg)
18	Cholesterol(mmol/L)	48	BMI
19	Triglycerides(mmol/L)	49	Pulse(Exam)(bpm)
20	Haemoglobin(g/dL)	50	SystolicBP(mmHg)
21	WhiteCellCount(10^9/L)	51	DiastolicBP(mmHg)
22	Platelets(10^9/L)	52	PulseBP(mmHg)
23	MCV(fL)	53	PulseBP(mmHg)

24	Hct(fraction)	54	FEV1(L)
25	Iron(umol/L)	55	FEV1Predicted(L)
26	VitaminB12(ng/L)	56	FEV1
27	Ferritin(ug/L)	57	FVC(L)
28	CRP(mg/L)	58	FVCPredicted(L)
29	TSH(mU/L)	59	FVC
30	MR-proANP	60	PEFR(L)