LAUTECH Journal of Computing and Informatics (LAUJCI) – ISSN : 2714-4194
Volume 1– No.1, Sept. 2020 – www.laujci.lautech.edu.ng

A Modified Genetic Algorithm Used for Dimensionality Reduction in Record Classification

Kamal Bakari Jilahi
Dept. of Mathematical Sciences
Taraba State University,
Jalingo, Nigeria
kamalbakari@gmail.com

Ahmet Ünveren
Department of Computer Engr.
Eastern Mediterranean University, Famagusta TRNC via Mersin 10, Turkey
ahmet.unveren@emu.edu.tr

ABSTRACT
This work proposes a modified Genetic Algorithm and compares its performance with the conventional Genetic Algorithms (GA) used for Dimensionality Reduction in record classification. A specialized elite voting crossover and mutation was introduced to the conventional GA and the population composition of every generation was compartmented into elite and non-elite individuals, and a proportion of offspring generated in each generation are derived from the elite individuals using the introduced voting crossover and mutation. The performance of the two algorithms was tested with 3 datasets from the UCI ML repository using different levels of elitism, crossover and mutation with the Extreme Learning Machine classifier. At higher rate of elitism, the results were highly in favor of the modified GA in both convergence time and classifier accuracy. While, at lower levels of elitism the two algorithms seen to be comparable in convergence time but the modified algorithm had better classifier accuracy. Furthermore, at higher rate of crossover, the modified algorithm tends to be slower in convergence than the conventional algorithm but better classifier accuracy. On the other hand, at higher mutation rate the modified algorithm tends to be faster in convergence than the conventional algorithm. In conclusion, except for the added computational cost due to the specialized voting crossover and mutation in the modified algorithm the results are in favor of the modified algorithm.

Keyword: Genetic Algorithm, Dimensionality Reduction, Record Classification, Crossover, Mutation, Elitist

1. INTRODUCTION

The importance of Dimensionality Reduction cannot be over emphasized in any Machine Learning process. This is because each algorithm is best implemented with a more appropriate set of features and the presence of redundant and irrelevant attribute causes overfitting (Dy and Brodley, 2004). Therefore, there is an inherent need to select the most relevant, appropriate and optimal set of features to be used with the intended algorithm in other to optimize the performance of the learning algorithm. This problem can be expressed as an optimization problem as thus:

$$\min z = \left(\frac{n!}{(m-n)!\,m!}, -f(w)\right)$$

Subject to

$$x_i \le n$$
$$x_i \in X$$
$$x_i \in \{0,1\}$$

Where *m* is the total number of available features, *n* is the number of features selected, *f(w)* is measure of goodness of the Machine Learning algorithm (e.g. accuracy, convergence and efficiency).

Record classification is the task of categorizing records into classes based on some known data that is the training set. The classification process is broken down into training and prediction stages (Ma and Huang, 2008). During the training stage the classifier algorithm may be supplied with too much, redundant, irrelevant attributes or a small number of observations which leads to overfitting or over generalization (Liu and Yu, 2005). As such, while minimizing the number of features used in a classification process the accuracy of the ensuring classifier is expected to be maximized. This is why all irrelevant and redundant features must be eliminated from the training dataset before it is fed to the classifier algorithm.

Over time, a number of Dimensionality Reduction algorithms have been used in the ML community. These can be broadly categorized into statistical, heuristic and meta-heuristic algorithms. Of all these categories the meta-heuristics are the most promising (Haleh and Kenneth, 2012). Although they do not guarantee to always arrive at the optimal solution but the speed and low computational cost they assure compensate for their less precision (Wettschereck, *et al*, 1997). One of the most popular meta-heuristic algorithms is the GA which mimics the evolutionary process of optimizing individual fitness through transfer of traits from one generation to another by genetic operation (crossover and mutation).

2.0 REVIEW OF LITERATURE

Genetic Algorithm is an optimization algorithm that imitates the Darwin's process of natural selection. This algorithm is applied to solving combinatorial and other types of optimization problems where the objective equations are complex to compute. GA form a part of the widely known Evolutionary Algorithm (EA) that use natural selection techniques such as crossover, mutation, inheritance etc. to generate solutions to optimization and search problems (Bhanu and Yingqiang, 2003).

Jihon and Vasant (1998) used Genetic Algorithm for Dimensionality Reduction in pattern classification and knowledge discovery. It was noted that, although meta-heuristics generally and Genetic Algorithm specifically do not always guarantee the optimal solution, the low computational cost and time compensate for the precision of optimality and this makes the algorithms feasible alternatives for the most computationally prohibitive methods of Dimensionality Reduction.

Frohlich *et al* (2003) applied Genetic Algorithm to the problem of classification of protein-protein interactions. The algorithm was modified to take into account the existing bounds of the generalization error for Support Vector Machines SVM. The performance of this algorithm was compared with the conventional Genetic Algorithm and Cross-Validation methods of Dimensionality Reduction. The obtained were in favor of the proposed algorithm.

A two stage Dimensionality Reduction method was used by Harun (2011), where Information Gain method was first applied then Principal Component Analysis (PCA) and Genetic Algorithm (GA) where Applied in the second stage to the Rueters-21578 and Classic3 datasets with *k-nearest neighbor (KNN)* and *C4.5* classifier algorithms for document classification. Precision, recall and F-measure were used as measure of goodness of the learning algorithm. The research remarked the drop in classifier time as compared with application of single algorithm. It further noted the effectiveness of this method especially in document classification where the attributes are numerous and a large part of them are less important in the classification process.

Riccardo and Amparo (1998) reviewed the positive and negative aspects of applying Genetic Algorithm in Dimensionality Reduction with Partial Least Squares PLS models. It is noted on several datasets that if correctly applied the algorithm always produces very good result and concluded that to some extend the optimality of the selected feature set depends on the initial population of solutions. Furthermore, it is noted that in situations where the original feature set does not contain much redundant features the algorithm may not perform well.

Chaikla and Yulu (1999) applied the Genetic Algorithm to the problem of Dimensionality Reduction in situations where there is a dependency between feature set. The method explores the search space for possible subsets to obtain the set of features that maximizes the prediction accuracy and minimizes the irrelevant and redundant attributes. Furthermore, a correlation was introduced into the fitness function of the Genetic Algorithm. A comparison between the conventional fitness function used for this problem and the proposed method was reported. The results show the superiority of the proposed method over the conventional methods in situations where the search space is multi-modal, combinatorial and large enough.

Younes *et al* (1997) applied the Genetic Algorithm to the problem of Dimensionality Reduction in Seed classification using *k-nearest neighbor (KNN)* classifier. The work noted that "the number of selected features was directly related to the probability of initialization of the population at the first generation of the GA and when the probability was fixed at 0.01 the GA selected about five features less than other values which increased the classification performance with the number of generations." It further acknowledged the great potential of the Genetic Algorithm for Dimensionality Reduction.

3.0 THE PROPOSED MODIFIED GENETIC ALGORITHM

As in the conventional GA, the modified algorithm begins by creating a population of randomly generated individuals *N*. Next the classifier is executed using the encoded alleles then the fitness of each individual is evaluated as the accuracy of the classifier using the selected attributes as encoded by each individual (other fitness evaluation criteria may be used) . Next, N or N*rate of elitism =N’ individuals with fitness above the average fitness in the generation are selected and designated as elite individuals from the population. As in the conventional GA these $N^{'}$ individuals are taken to the next generation without any alteration. Furthermore, in this modified algorithm these individuals are sent to a special mating pool in other to be used with the special crossover and mutation for generating $N^{''}$ individuals. Then any of the conventional crossover and mutation for binary encoded GA is used to generate the remaining *N-(N’+N”) = N’’’* individuals. Therefore, every generation other than the initial generation is a combination of N’ elites individuals, N’’ offspring of the elite individuals using the special crossover and mutation and *N’’’* offspring using the conventional crossover and mutation expressed as thus *N = N’+N’’+N’’’*. The selected attributes as encoded by the individuals are sent to the classifier and the accuracy of the classifier is used as the fitness of each individual. This is repeated until a predefined number of iterations or some required level of accuracy from the classifier or any other stopping criteria is met.

To retain the randomness of the conventional GA the special crossover and mutation creates less number of individuals. While to encourage greediness of the algorithm, the minimum requirement to serve as elite is an average of individual fatnesses of all individuals in the generation. This procedure can be translated as only alleles agreed upon by the elites are considered as good traits because only alleles with high correlation with the target class are important in Dimensionality Reduction. This can be diagrammatically represented as in flowchart 1:

The Special Crossover and Mutation

In a GA with a population of N individuals where each individual is composed of M alleles, then this population can be represented as a matrix $A_{N,M}$. The fitness of each individual is denoted by $f(A_i)$, the fitness required to be considered as an elite is denoted as f_g, the average fitness in the population is denoted by f_{avg}. N’ = A_n:$f(A_n) \geq f_g$ are selected to undergo the special crossover and mutation to create a new individual. That is to say out of N individuals, N’ good individuals (those with fitness $\geq f_g$) are selected for the proposed reproduction. The value of f_g is obtained using

$$f_g = g \times F_{avg} \tag{3.1}$$

Where g is a constant [0, 1] which signifies the relevance of the average fitness in the process and f_{avg} is the average fitness in the population and is given by

$$f_{avg} = \frac{\sum_{i=1}^{N} f(A_i)}{N} \tag{3.2}$$

If $g=1$ then, f_g will be equal to f_{avg}. The reason for selecting g [0, 1] is to ensure that the whole search space is been explored. After obtaining the minimum requirement to be selected as a parent (i.e. f_{avg} and f_g), the sum of 1’s alleles across both horizontal and vertical directions of the matrix $A_{N,M}$ is obtained. The sum of alleles in the horizontal direction serves as indicator of the number of alleles which should be present in the new individual and is obtained as

$$L = h \times L_{avg} \tag{3.3}$$

Where *L* is the number of 1’s in the parent h is a constant [0, 1] and L_{avg} is the average 1’s alleles in the horizontal direction given as

$$L_{avg} = \frac{\sum_{i=1}^{N'} L_i}{N} \tag{3.4}$$

Where L_n is the sum of occurrences of 1’s alleles in the horizontal direction which represents the number of attributes selected by an individual and is given by

$$L_n = \sum_{m=1}^{M} a_{nm} \tag{3.5}$$

And the sum of 1’s in the vertical direction is the voting weight of a selected feature that determines which allele should be a 1 in the generated offspring and defined by

$$V_m = \sum_{m=1}^{M} a_{nm} \tag{3.6}$$

The created offspring will be composed of 1 alleles selected from the highest constant V_m $m=1$ to M. A single individual is considered for mutation using bit flip mutation where a single allele with a bi value of one standard deviation below the mean (i.e. 1 value below L_{avg} is flipped from a zero to a 1 to generate another individual. More individuals are generated by repeating this procedure for all other alleles with one value below L_{avg} until the required number of N''' is obtained

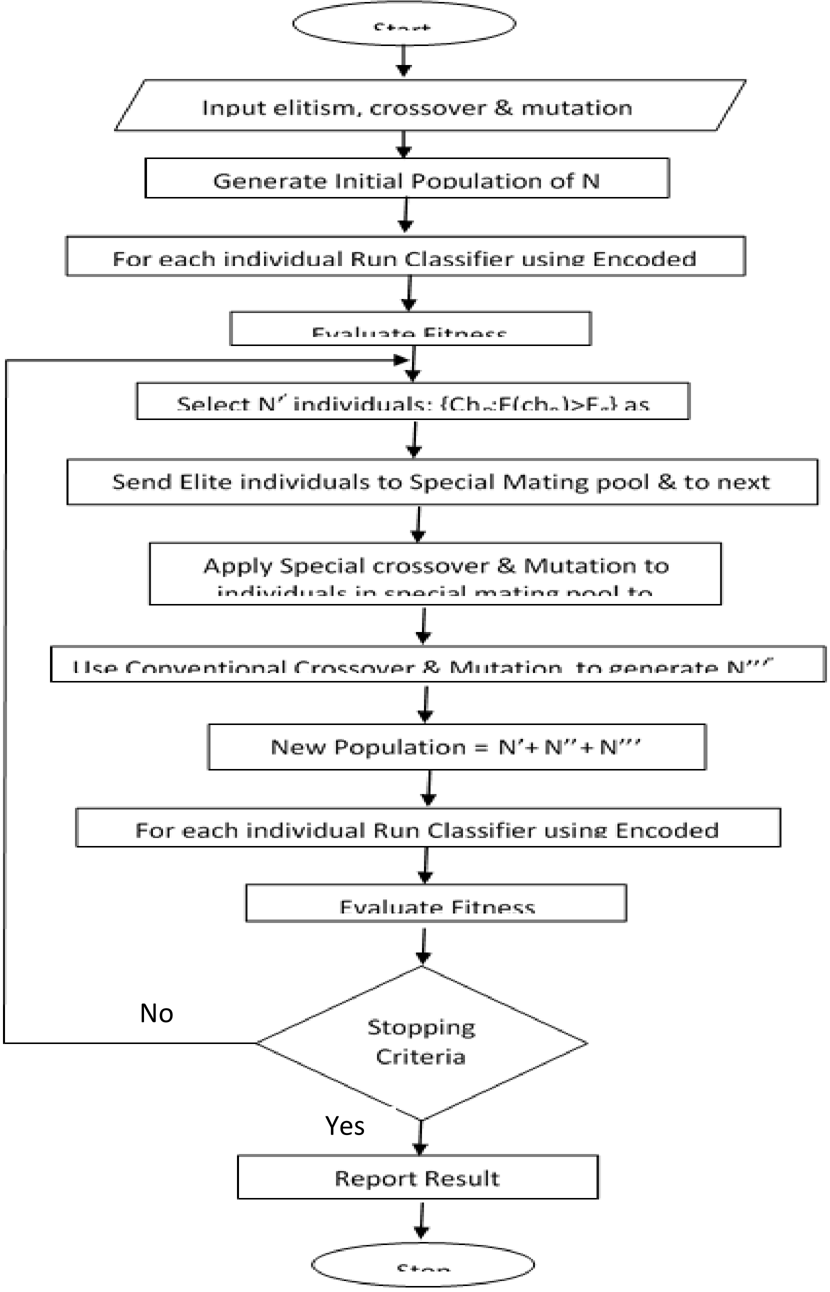
Start
Input elitism, crossover & mutation
Generate Initial Population of N
For each individual Run Classifier using Encoded
Evaluate Fitness
Select N′ individuals: {Ch:F(ch)>F} as
Send Elite individuals to Special Mating pool & to next
Apply Special crossover & Mutation to individuals in special mating pool to
Use Conventional Crossover & Mutation to generate N'''
New Population = N'+ N''+ N'''
For each individual Run Classifier using Encoded
Evaluate Fitness
No
Stopping Criteria
Yes
Report Result
Stop

Figure 1: Flowchart for the Proposed Modified Genetic Algorithm

Algorithm for the Modified Genetic Algorithm

```
Step1: Input genetic operators: rate of crossover $R_{xo}$, rate of mutation $R_{mt}$, and rate of elitism $R_{el}$
Step 2: Input Number of attributes in dataset M and Number of Individuals in a population N
Step 3: Initialize a Counter: C to 1
Step 3: REPEAT Step 3.1 to Step 3.3 UNTIL C=N
    Step 3.1: Generate an individual with M alleles of 1's and 0's
    Step 3.2: If there is an individual with the same alleles in the population then discard the individual else add individual to population and increment C by 1
Step 4: For each individual in population run classifier using encoded alleles
Step 5: Evaluate the fitness of each individual as the goodness of the classifier using the encoded alleles.
Step 6: Select $R_{el}*N = N'$ Individual with Fitness> Average Fitness from the population as elites
Step 7: Move the selected individuals in Step 6 to a Special Mating Pool $MP_{sp}$ and to the next generation $G_{n+1}$.
Step 8: Apply the special crossover and mutation to the individuals in the $MP_{sp}$ to generate N" offspring
Step 9: Apply any conventional binary crossover and mutation using $R_{xo}$ , $R_{mt}$ on the population to generate N''' offspring
Step 10: New population N = N'+N"+N'''
Step 11: For each individual in population run classifier using encoded alleles
Step 12: Evaluate the fitness of each individual as the goodness of the classifier using the encoded alleles.
Step 13: IF Stopping criteria is met THEN Report Result ELSE GOTO Step 6.
Step 15: Stop
```

4.0 RESULTS AND DISCUSSION

This work compared the performance of the proposed algorithm with the conventional algorithm using Pima Indians, Cleveland and Arrhythmia datasets all from UCI Machine Learning repository. The choice of these three datasets is to see the performance of the algorithms on datasets with small (Pima), medium (Cleveland) and large (Arrhythmia) number of features. This work also investigated the effects of population size, mating pool size, rate of cross over and mutation on both the algorithms using a stopping criterion of 100 iterations. The proposed algorithm was implemented using R programming language while galgo plug-in of RStudio was used for the conventional algorithm. Figure 2 shows the performance of the two algorithms using a population size of 50, maximum iteration of 100, elitism of 25%, and crossover rate of 20% and mutation rate of 2% as suggested by (Yang and Honavar, 1998).

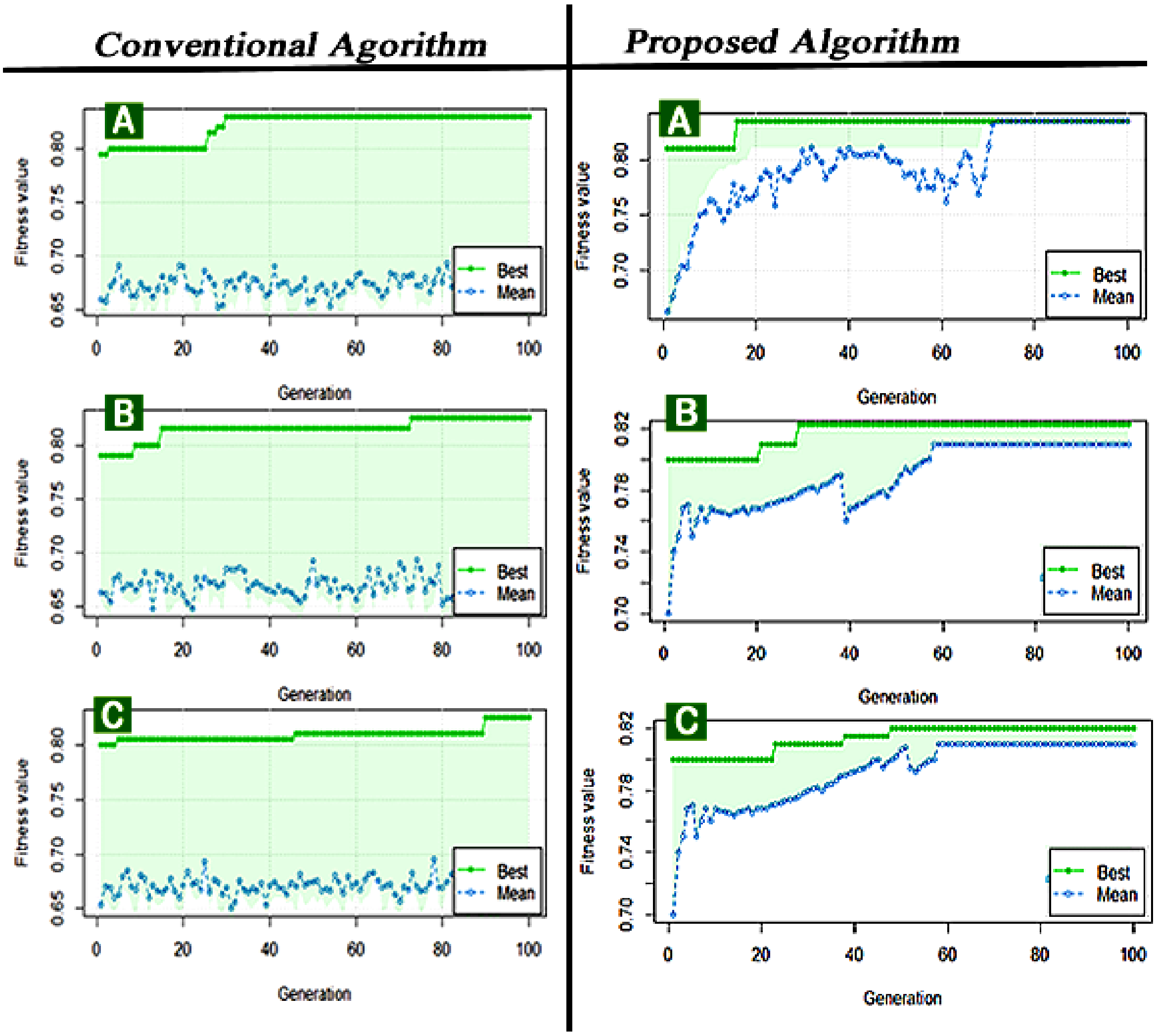

Figure 2: Comparison of performance between Conventional and Proposed Algorithms with (A) Pima Indians, (B) Cleveland and (C) Arrhythmia datasets

From figure 2, it can be seen that the proposed algorithm always has better classifier accuracy (0.80 against 0.81 for Pima Indians and 0.80 against 0.82 for both Cleveland and Arrhythmia respectively), better mean fitness (the mass below the best fitness line) and converges faster than the conventional algorithm. Next, the research used the Original Cleveland dataset (with 76 features) to evaluate the performance of the two algorithms on Population size, Mating pool size, Rate of Elitism, Mutation and Crossover the results are presented in Tables 1 to 4

Table 1: Performance Comparison Based on Population Size. Columns: (1) Number of Features Selected (2) Classifier Accuracy (3) Mean Fitness (4) Number of Iterations before Convergence (5) Worst Fitness

	Conventional Algorithm					Proposed Algorithm				
Population Size	1	2	3	4	5	1	2	3	4	5
30	30	0.780	0.46	58	0.381	23	0.795	0.66	38	0.651
40	28	0.788	0.45	61	0.391	23	0.798	0.68	39	0.660

50	28	0.791	0.43	62	0.321	22	0.799	0.61	40	0.581
60	27	0.793	0.38	64	0.311	20	0.799	0.71	42	0.692
70	27	0.796	0.42	67	0.309	16	0.801	0.73	46	0.702
80	24	0.798	0.41	68	0.288	16	0.804	0.76	49	0.741
90	22	0.799	0.40	74	0.265	15	0.808	0.75	50	0.732
100	20	0.799	0.40	78	0.231	14	0.815	0.75	56	0.721

In table 1, the population size of the two algorithms was incremented from 20 to 100 individual using a step size of 10. It can be seen that the classifier accuracy increased with every increase in the population size in both algorithms except for the increase from 90 to 100 in the case of the conventional algorithm and 5 to 60 in the case of the modified algorithm. Furthermore, the modified algorithm always had a better classifier accuracy, mean fitness and worst fitness. To further compare, the conventional algorithm had most number of selected features

Table 2: Performance Comparison Based on Mating Pool Size. Columns: (1) Number of Features Selected (2) Classifier Accuracy (3) Mean Fitness (4) Number of Iterations before Convergence (5) Worst Fitness

	Conventional Algorithm					**Proposed Algorithm**				
Special Mating Pool Size	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
0.1	24	0.780	0.38	50	0.243	23	0.782	0.67	37	0.521
0.2	23	0.779	0.38	53	0.246	22	0.787	0.68	38	0.523
0.3	21	0.776	0.43	56	0.244	22	0.802	0.70	40	0.522
0.4	21	0.778	0.44	58	0.311	20	0.809	0.71	42	0.578
0.5	20	0.796	0.46	61	0.369	16	0.813	0.73	46	0.601
0.6	24	0.780	0.47	64	0.368	16	0.808	0.76	49	0.598
0.7	26	0.777	0.49	69	0.379	15	0.808	0.75	50	0.622
0.8	26	0.767	0.51	74	0.408	14	0.803	0.75	50	0.629
0.9	27	0.769	0.54	76	0.420	14	0.799	0.74	53	0.631
1.0	30	0.761	0.55	79	0.420	14	0.799	0.74	55	0.633

In table 2, the performance of the two algorithms under varying mating pool sizes was compared. In the case of the conventional algorithm, the main mating pool was varied while the special mating pool was used. Just as in the first test scenario, the proposed algorithm had better classifier accuracy, mean fitness and number of selected features. It can be noted that the proposed algorithm produced its best classifier accuracy when the mating pool size set at 50% of the population.

Table 3: Performance Comparison Based on Crossover rate. Columns: (1) Number of Features Selected (2) Classifier Accuracy (3) Mean Fitness (4) Number of Iterations before Convergence (5) Worst Fitness

	Conventional Algorithm					**Proposed Algorithm**				
Crossover rate	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
0.1	34	0.684	0.43	77	0.381	24	0.785	0.58	55	0.481
0.2	28	0.782	0.45	74	0.392	23	0.788	0.59	50	0.490
0.3	24	0.781	0.43	67	0.331	21	0.790	0.57	49	0.501
0.4	25	0.783	0.41	65	0.341	21	0.791	0.60	49	0.576
0.5	22	0.782	0.38	67	0.311	22	0.793	0.61	46	0.559
0.6	23	0.784	0.42	64	0.307	18	0.788	0.68	43	0.582
0.7	21	0.786	0.37	60	0.300	16	0.785	0.64	40	0.601
0.8	22	0.780	0.36	61	0.298	17	0.808	0.59	41	0.611
0.9	19	0.790	0.37	58	0.299	14	0.810	0.61	44	0.612

In table 3, the two algorithms were compared under varying crossover rate. It can be seen that just as in the first two scenarios the classifier accuracy of both algorithms increased with every increase in the cross over rate. This may be due to the increase in the exploratory capability the algorithms as signified by the crossover rate. In comparison to the first two scenarios, both algorithms produced worst classifier accuracy (0.810 against 0.815 and 0.813 respectively) and mean fitness (0.58 against 0.66 and 0.67 respectively) but better convergence time in terms of iterations before best classifier accuracy (42 against 46 and 44 respectively)

Table 4: Performance Comparison Based on Mutation rate. Columns: (1) Number of Features Selected (2) Classifier Accuracy (3) Mean Fitness (4) Number of Iterations before Convergence (5) Worst Fitness

	Conventional Algorithm					**Proposed Algorithm**				
Mutation rate	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
0.01	31	0.780	0.40	64	0.301	24	0.795	0.66	37	0.549
0.02	28	0.787	0.42	65	0.321	23	0.798	0.66	38	0.550
0.03	27	0.790	0.43	68	0.303	22	0.799	0.68	39	0.601
0.04	26	0.793	0.46	67	0.306	20	0.800	0.71	41	0.621
0.05	25	0.797	0.48	68	0.309	18	0.801	0.73	42	0.632
0.06	24	0.798	0.51	71	0.323	16	0.804	0.75	40	0.642
0.07	21	0.800	0.54	70	0.333	15	0.807	0.76	43	0.651
0.08	20	0.801	0.58	74	0.321	14	0.809	0.78	46	0.652

0.09	19	0.807	0.60	78	0.342	15	0.816	0.79	48	0.661

In table 4, the two algorithms were compared under varying mutation rate and it can be seen that both algorithms performed better than the first three scenarios in terms classifier accuracy, mean fitness and number of features selected as the mutation rate increases. This may be due the exploitive capability of the algorithms been enhanced with every increase in the mutation rate there by making both algorithms been able to find better solution faster. In comparison to the first three scenarios, we have the worst convergence time of 37 iterations against 44, 46 and 48 respectively.

5.0 SUMMARY AND CONCLUSION

This work reviewed the problem of data dimensionality reduction. Subsequently, a new crossover and mutation technique was introduced to the conventional GA which is used in the task of dimensionality reduction. Pima Indians, Cleveland and Arrhythmia datasets of UCI ML repository were used to test the performance of the proposed method. The result obtained suggest the superiority of the proposed method over the conventional algorithm in convergence, classifier accuracy and population diversity. Additionally, issues that affect the performance of the proposed method such as probability of crossover and mutation, elitism, mating pool size and population size were also investigated. The result obtained showed that the proposed algorithm performed better when elitism and crossover rate were high and population size and mutation rate are low. Finally, it is worth noting that in a situation where all individuals in the special mating pool give the same importance to all features in the dataset (i.e. equal vote for all alleles) this algorithm may not work well. This research suggest that further research can be carried out to study the algorithms premature convergence.

REFERENCES

1 Bhanu, B. and Yingqiang L. (2003). *Genetic Algorithm Based Feature Selection for Target Detection in SAR Images*, Image and Vision Computing Journal 21 (2003) 591–608

2 Chaikla, N & Qi, Y. (1999). *Feature Selection Using the Domain Relationship with Genetic Algorithms*. Knowl. Inf. Syst.. 1. 257-268. 10.1007/BF03325105.

3 Cheng-Lung H., and Chieh-Jen W. (2006). *A GA-based Feature Selection and Parameters Optimization for Support Vector Machines* Journal of Expert Systems with Applications 31 (2006) 231–240

4 Dy, J.G. and Brodley, C.E. (2004). *Feature Selection for Unsupervised Learning.* The Journal of Machine Learning Research, 5:845–889, 2004.

5 Fröhlich, H. and Chapelle, O. (2003) *“Feature Selection for Support Vector Machines by Means of Genetic Algorithms,”* Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento, 3-5 November 2003, pp. 142-148.

6 Haleh V, Kenneth De J. (2012). *Genetic Algorithms as a Tool for Feature Selection in Machine Learning*, Springer.

7 Harun U (2011) *A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm* Knowledge-Based Systems, Volume 24, Issue 7, October 2011, Pages 1024-1032

8 Inza,I., Larranaga, P., Blanco, R. and Cerrolaza, A. J. (2004). *Filter Versus Wrapper Gene Selection Approaches in DNA Microarray Domains.* Artificial Intelligence in Medicine, 31(2):91–103, 2004.

9 Kim, S and Xing, E. (2009). *Statistical Estimation of Correlated Genome Associations To A Quantitative Trait Network*. PLoS genetics, 5(8):e1000587.

10 Leardi, R and Lupiáñez G, A. (1998). *Genetic Algorithms Applied to Feature Selection in PLS Regression: How and When to Use Them.* Chemometrics and Intelligent Laboratory Systems. 41. 195-207. 10.1016/S0169-7439(98)00051-3.

11 Liu, H. and Yu, L. (2005). *Toward Integrating Feature Selection Algorithms for Classification and Clustering.* IEEE Transactions on Knowledge and Data Engineering, 17(4):491, 2005.

12 Ma, S and Huang, J (2008). *Penalized Feature Selection and Classification in Bioinformatics*. Briefings in Bioinformatics, 9(5):392–403.

13 Mitra, P., Murthy, C. A. and Pal, S. (2002). *Unsupervised Feature Selection Using Feature Similarity*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24:301–312.

14 Priyanka K., and Kavita. D., (2016). *Burse Feature Selection Using Genetic Algorithm and Classification using Weka for Ovarian Cancer* International Journal of Computer Science and Information Technologies, Vol. 7 (1), 194-196

15 Riccardo L. (2000). *Application of Genetic Algorithm–PLS for Feature Selection in Spectral Datasets* Journal Of Chemometrics 2000; Vol 14: 643–655

16 Shahamat, H., and Pouyan, A. A. (2012). *Feature Selection Using Genetic Algorithm for Classification of Schizophrenia Using fMRI Data* Journal of AI and Data Mining Vol 3, No 1, 2015, 30-37.

17 Wettschereck, D., Aha, D. and Mohri, T. (1997). *A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms.* Artificial Intelligence Review, 11:273–314, 1997.

18 Yang J., Honavar V. (1998) *Feature Subset Selection Using a Genetic Algorithm. In: Liu H., Motoda H. (eds) Feature Extraction, Construction and Selection.* The Springer International Series in Engineering and Computer Science, vol 453. Springer, Boston, MA

19 Zou, H. and Hastie, T. (2005). *Regularization and Variable Selection via The Elastic net.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2):301–320.