Study on a Hybrid Approach for Improving Clinical Behavior of Cancer by Assorting Informative Genes

Golam Moktader Daiyan
Assistant Professor, Head,
Department of CSE
East Delta University (EDU)

Abrar Hussain
Lecturer, Department of EEE
Southern University
Bangladesh

Fahmida Akter
Faculty Member, Department of CSE
East Delta University (EDU)

Hrishi Rakshit
Lecturer, Department of ECE
Southern University
Bangladesh

ABSTRACT
In recent times in the classification and diagnosis of cancer nodules, gene expression profiling by micro array techniques are playing a fundamental role. A range of researchers have proposed a number of machine learning and data-mining approaches for identifying cancerous nodule using gene expression data. The process of gene selection for the cancer classification encounters with some major problems due to the properties of the data such as the small number of samples compared to the huge number of genes, irrelevant genes, and noisy data. Hence, this paper aims at selecting a near-optimal subset of informative genes that is most relevant for the cancer classification. This paper also proposes an efficient BFSS (Boost Feature Subset Selection) technique to improve the performance of single-gene based discriminative scores using bootstrapping techniques. The proposed hybrid approach (Filter-Wrapper) will be implemented on three publicly available microarray datasets. These microarray datasets are: Acute Lymphoblastic Leukemia Cancer (ALL), Lung Cancer and Colon Cancer.

General Terms
Genetic Algorithm, Cyclic Genetic Algorithm, Boost Feature Subset Selection, Filter-Wrapper Technique.

Keywords
Evolutionary Algorithms, Binary Coded Genetic Algorithm.

1. INTRODUCTION
People live in the information-age—accumulating data is easy and storing it is inexpensive. In 1991 it was assumed that the amount of stored information doubles every twenty months [1]. Unfortunately, as the amount of machine readable information increases, the ability to understand and make use of it does not keep pace with it's growth . This necessitates to transform enormous amounts of data into useful information and knowledge. This study focuses on feature selection. A feature selection technique is a pre-processing step to eliminate irrelevant and redundant data and in many cases, improves the performance of learning algorithms [1]. Microarray technology is a developing technology used to study the expression of many genes at once. [2]. Microarrays experiments are used to gather information from tissue and cell samples for finding gene expression differences that are useful in diagnosing diseases [3]. It produces gene expression

data as the final product. Therefore, it provides a new way for people to understand molecular behaviors in abnormal tissues and improve classification performances for accurate cancer diagnosis and treatment [4]. Recognition of patterns and other subsequent analysis from the thousands of gene expression values is particularly difficult and primary role of an effective feature selection is to simplify this task [5]. Feature selection attempts to identify and highlight the most informative genes in the microarray data sets which have dominant effects on the biological states of human cells and that this is the main objective of this study. Removal of less informative genes helps to alleviate the effects of noise and redundancy [6] and simplifies the task of disease classification and prediction of medical conditions such as cancer [7]. Feature selection techniques can be classified into three broad categories: Filter Technique, Wrapper Model and Embedded Technique. Filter technique performs individual gene selection process which is independent of the classification model [8]. Filter techniques are simple, fast and they tend to easily scale the data sets of different dimensions. However the filter technique does not take into consideration the performance of its selections and most importantly they ignore the biological relationships that exist among biological markers [9] and studies show that the medical states of individuals are a result of gene interactions [7]. Wrapper model generates and evaluates various possible subsets from the original dataset and seeks to identify the most informative subsets among them. As the search space grows exponentially, the heuristic measures are put to use to guide the search for the optimal subset. Although the filter method considers the gene dependencies, wrapper models have an over fitting problem [3] and as a wide range of possibilities needs to be addressed, the computational cost is high, which in turn results to a longer convergence time.

The ultimate goal of this study is to find the best way of feature selection that produces the ideal classification using an

evolutionary approach (Genetic Algorithm). This will pave the way for better machine perception and hence better data intelligibility.

In this paper, the focus is on the study of existing implementations of the different variants of Evolutionary Algorithms. Comparative performance analysis of the various methods will be done to identify strengths and weaknesses. Construction of a model representing the new goal will then be done and finally simulation of the model will be done to evaluate the effectiveness of the approaches developed through the study. In section 1 a brief introduction of the study has been provided. Section 2 highlights literature review which consists of shifting towards evolutionary approach along with a brief introduction on Genetic Algorithm. Section 3 elaborates on the proposed methodology. Section 4 shows the results from implementing our proposed approach on the datasets taken. A comparative analysis also creates a fruitfulness of our proposed method over other existing implementations based on performance analysis. Finally section 5 states conclusion and the possible future works on the proposed framework with some existing limitations.

2. LITERATURE REVIEW

2.1 Evolutionary Algorithms

PSO: Particle Swarm Optimization is a method that optimizes a problem iteratively. PSO treats each solution as a particle and starts with a pool of candidate solutions. The particles are moved in the search space and the movement of each particle is guided by the best known local position. The process is iterated as each local solution in found to guide the solution towards the global best position or intended optimal solution.

ACO: Ant Colony Optimization relies on a probabilistic model so solve problems. The original algorithm was used to find the best path in a graph. The algorithm was later modified to solve a wide class of problems across various applications.

GA: Genetic Algorithm is a type of Evolutionary Algorithm (EA) inspired by the biological method of evolution in which an environment is created in which potential solutions can evolve. From the population a fitness function selects some solutions based on its "goodness" which are subjected to genetic operators such Mutation and Crossover which generates new population. From this population the entire process is repeated until the optimal solution has been found.

In this study binary coded genetic algorithm is highlighted.

2.1.1 Binary coded Genetic Algorithm

The binary coded genetic algorithm is a probabilistic search algorithm that iteratively transforms a set (called a population) of mathematical objects (typically fixed-length binary character strings), each with an associated fitness value, into a new population of offspring objects using the Darwinian principle of natural selection and using operations that are patterned after naturally occurring genetic operations, such as crossover (sexual recombination) and mutation. Following the model of evolution, they establish a population of individuals, where each individual corresponds to a point in the search space. An objective function is applied to each individual to rate their fitness. Using well conceived operators, a next generation is formed based upon the survival of the fittest. Therefore, the evolution of individuals from generation to generation tends to result in fitter individuals, solutions, in the search space. Empirical studies have shown that genetic algorithms do converge on global optima for large class problems. In binary coded genetic algorithms, a population is nothing but a collection of "chromosomes" representing possible solutions. These chromosomes are altered or modified using genetic operators through which a new generation is created. This process is repeated a predetermined number of times or until no improvement in the solution.

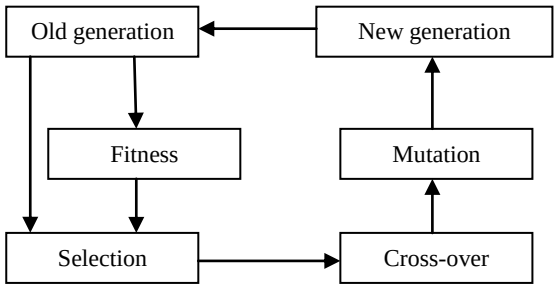

Figure 1: Genetic Loop

3. PROPOSED METHODOLOGY

In this paper, a cancer classification mode is proposed which has two phases: 1) Gene selection and 2) Classification. The first phase uses a gene selection method to select genes, while in the second phase a classifier is implemented to perform classification process which is shown below-

Here gene expression data is represented as micro-array dataset. For gene selection, BFSS is used (Boosted Feature Subset Selection) & Cyclic GA (Genetic Algorithm) method and classification purpose SVM (Support Vector Machine) classifier is used.

3.1 Boost Feature Subset Selection

The term boosting means producing a very accurate prediction rule by combining rough and moderately inaccurate "rules of thumb". In the context of single gene based feature selection boosting would improve the performance by identifying the weak performers in a particular iteration and in the successive iterations it would try to identify features that would perform well for those weak performers. The performance of the process is said to be "boosted" by giving more emphasis on the weak performers in a particular iteration as shown in Xian Xu and Aidong Zhang in [5].

Before moving to the details of the BFSS algorithm some terminologies need to be discussed.

A bootstrap sample set S^m is a multiset of samples randomly drawn with replacement from the original set of samples S. $s \in S$ can appear more than once or does not appear at all in

S^m. The sampling probability of each sample in S is determined by a probability table p(s) where s∈ S.

A bootstrap B of a training dataset using bootstrap sample set S^m is a dataset containing the samples of Bootstrap sample set

The worst set of samples S_{worst} with respect to bootstrap dataset B and a single-gene based scoring function F is defined as a multiset:

$$argmax\ (\ F\ (E(\ g,\ S^M - S\ )\ )\ )$$

Here S^m - S means a set by removing S from S^m. We also call $S^m - S_{worst}$, the best set of samples.

A microarray data set E can be considered as a collection S samples each containing G number of genes with each g, g ∈ G having a an expression value. Boost Feature Subset Selection (BFSS) algorithm (as shown in [1]) starts of by generating a set of samples called a bootstrap sample set S^m which is a multiset of samples obtained by random sampling from the pool of all samples S. The probability of a sample being selected is equal to p(s) where s ∈ S and initially all samples have a probability of 1/S. Using the S^m a bootstrap B is generated which is reduced version of the full microarray expression set containing all the genes and expression values, but containing only the samples present in S^m. After the creation of B a single gene based score for each of the score is calculated, in [5] F score was used. The gene with the best score g for the current B is identified and added to the list of selected genes G`. In the next step the samples for which g is not informative is identified i.e the worst set of samples. The probability of the remaining set of samples i.e good samples are reduced so that in subsequent iterations genes that would perform well on the current worst set of samples can be selected. g is marked as selected and is not evaluated again by the algorithm. BFSS would run until the number of selected genes has been found which depends on the dataset being evaluated.

Algorithm 2: BFSS: Boost Feature Subset Selection

n' is the number of genes to be selected; F is a single gene based discriminative score

1. Initialize $p(s)$ to be $1/m$ (m is the total number of samples in the dataset). Set
2. G' as an empty set
3. For $|G'|<n'$ **do**
4. Generate the bootstrap sample set S^M
5. Calculate score F on bootstrap; keep track of the best score so far
6. Add top ranked gene g based on F score to G'
7. Find worst δ samples S_{worst} based on gene g and S^M using algorithm 1
8. Reduce the probability for the *best set of samples* (those samples which are classified accurately by the gene g)
9. Remove g from dataset
   End for
10. Return G'

Algorithm 1: Worst Sample Set: Calculate the worst set of samples

1. S' , S_0 to be empty sets
2. For all s in S do
3. $S_1 \leftarrow S$ – {s }
4. Calculate F score for the gene g , add score to , S_0
5. End for
6. Sort, S_0, add samples s corresponding to top δ scores in , S_0 to S'
7. Return S'

The Boost feature subset selection algorithm (BFSS) is depicted by Algorithm 1 and Algorithm 2. After initialization, the Algorithm 2 generates a bootstrap B of training set E which also includes the creation of bootstrap sample set and then the bootstrap itself using random sampling with replacement from S using probability table p(s).

After bootstrap B is produced, the F score is calculated for each gene in B. The gene with best F score for the current B is selected and added to the selected gene set G'. Based on this best gene g, BFSS ascertains the worst set of samples with respect to g and the single-gene based scoring function F using Algorithm 1. The probability table p(s) which affects the generation of further bootstrap B is updated by reducing the probabilities of the non-worst or good samples. Thus the probability of selecting these good samples being in the later iterations is thus reduced, causing BFSS to shift the focus onto those samples that previously selected genes would not perform well on i.e the worst set of samples for the current *g*. The currently selected gene is then marked as selected and hence it is not considered further by the BFSS algorithm. BFSS repeats this process until *n`* genes are selected. Experimentally *d* was chosen to be 0.96 of the number of training samples in a dataset and e to be 0.96.

3.2 Cyclic Genetic Algorithm (CGA):

CGA starts executes cycles iteratively which is repeated until the required number of genes has been selected. Generating a potential subset of genes in the current cycle c, which is used for the next cycle (c+1) as its input set, thus selection of genes in (c+1) only uses genes given by cycle c to generate potential subset. A near-optimal subset is selected among the potential

subsets based on the highest fitness value which is the aggregate of the highest LOOCV (leave-One-Out Cross-Validation) accuracy given by the smallest number of selected genes. The cyclic process results a near-optimal subset of genes. In each iteration CGA chooses the highest possible number of genes in order to avoid the over fitting problem as the inclusion of the largest possible subset ensures that the combined power of a subset of genes are taken into consideration. The working procedure is shown in the flowchart below-

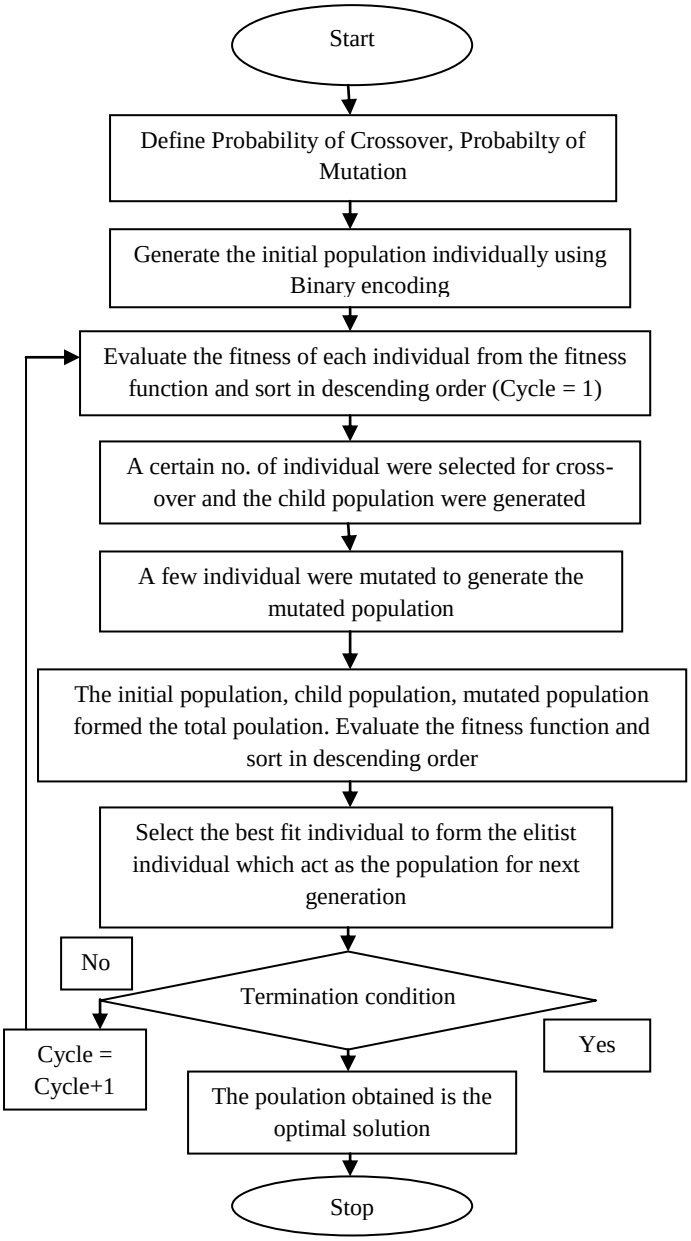

Figure 2: Proposed flowchart for Cyclic Genetic Algorithm

4. EXPERIMENTAL RESULTS:

4.1 Dataset Details

Experiments start with the implementation of Boost feature subset selection (BFSS). As mentioned earlier the main purpose of using BFSS is to provide a better initial population for cyclic genetic algorithm (CGA) rather than generating it completely on random basis thus avoiding early convergence and over-fitting problems. To implement BFSS we have used t-score to acquire the scoring of each gene within a specific sample. Process of t-score is amalgamated with the algorithm of BFSS successfully. This BFSS algorithm is then applied on the three publicly available microarray datasets. These microarray datasets are: Acute Lymphoblastic leukemia cancer (ALL), Lung cancer and colon cancer. Table 1 summarizes the data sets.

Table 1: Summary of Microarray datasets

Dataset	Number of Classes	Number of Samples in the Dataset	Number of Genes
ALL	2 (B-cell ALL and T-cell ALL)	128 (95 B-cell ALL and 33 T-cell ALL)	12625
Lung	2 (MPM and ADCA)	181 (31 MPM and 150 ADCA)	12533
Colon	2 (Normal and tumor)	62 (22 normal and 40 tumor)	2000

From table 1 it is seen that dataset for Colon cancer contains the lowest number of genes comparing to other two datasets, exposing higher possibilities of misclassifications and over fitting. It is because the more the number of samples the more we can train classifiers to classify test samples. The outputs of BFSS of 3 different datasets which are taken as the initial population by CGA are shown in Table 2.

Table 2: Reduced number of genes by Boost Feature Subset Selection

Datasets	Original Number Of Genes	BFSS output
Leukemia (ALL) Cancer	12625	4000
Lung Cancer	12533	4000
Colon	2000	1500

The number of genes to be given by BFSS as output for the initial population of Cyclic GA was set roughly as 30-32% of the total number of genes for leukemia and lung cancer datasets. Therefore Cyclic GA will take 4000 as the initial population for these two datasets. As the number of genes in colon cancer datasets is only 2000, output of BFSS for this dataset was set to 75% of total number of genes, which is 1500 genes.

4.2 Performance Analysis

The implementation of Boosted Cyclic GA requires the implementation of Boost Feature Subset Selection (BFSS) and then GA and Cyclic GA successively. Then BFSS and Cyclic GA are combined to produce our proposed approach Boosted Cyclic GA (BCGA). This sequence of implementation provided the opportunity to compare the proposed method Boosted Cyclic GA (BCGA) with the related previous works like GA and simple Cyclic GA. Different predefined parameters for GA, CGA and BCGA are given in Table 3.

Table 3: Parameters for Cyclic-GA

Dataset	ALL	Lung	Colon
Initial Population	4000	4000	1500
Number of Generations	200	200	200
Crossover Rate	0.9	0.9	0.9
Mutation Rate	0.03	0.03	0.03

To explore and observe the diversified results of the Boosted Cyclic GA, two classifiers are used namely KNN and SVM. On the other hand GA and Cyclic GA were implemented and tested only with SVM classifier. Cyclic GA creates possible potential subset of genes in each cycle and it reduces the number of genes in successive cycles. Now the termination condition could be set as the lowest number of genes produced by Cyclic GA beyond which it cannot be reduced further. But the motivation was to combine some reasonable number of genes that can hold the characteristics of diversity, providing higher accuracy and can classify maximum number samples avoiding the possibility of over fitting problem. For example in this experiment, 10 is used as the minimum number of genes which can be produced in a cycle of Cyclic GA. This condition holds for both Cyclic GA and our proposed approach Boosted cyclic GA throughout the experimental processes. The results after applying our proposed approach Boosted Cyclic GA on the leukemia (ALL) dataset are given in Table 4. This table contains the accuracies provided by the two classifiers along with the respective number of selected genes in 10 independent runs. At the end of the table the average values and also the standard deviations are given. The best accuracy with the number of selected genes and also the best average accuracy are shaded for noticing them easily.

Table 4: Classification accuracies and number of genes selected by Boosted Cyclic GA for leukemia (ALL) dataset

No. of Runs	Leukemia Cancer			
	BCGA-KNN	#Selected Genes(KNN)	BCGA-SVM	#Selected Genes(SVM)
1	94.63	16	95.48	17
2	96.19	15	94.79	15
3	93.89	10	96.24	18
4	95.34	13	94.49	11
5	96.29	13	97.87	16
6	96.78	15	96.21	14
7	98.19	18	97.64	18
8	97.92	18	95.84	22
9	97.16	18	97.98	16
10	95.49	16	97.14	15
Average ± S.D.	96.08 ±1.28	15.2±2.61	96.37 ±1.26	16.2±2.89

Table 4 shows that the highest accuracy was achieved by BCGA-KNN approach with accuracy of 98.19% and it was achieved only with 18 genes from a total of 12625 genes. On the other hand the BCGA-SVM provides the best average accuracy of 96.37% with a standard deviation of 1.26. So if we consider the overall performance and compare the average accuracies then definitely BCGA-SVM has shown better performance. Results after applying BCGA-KNN and BCGA-SVM on lung cancer dataset are given in Table 5.

Table 5: Classification accuracies and number of genes selected by Boosted Cyclic GA for lung cancer dataset

No. of Runs	Lung Cancer			
	BCGA-KNN	#Selected Genes(KNN)	BCGA-SVM	#Selected Genes(SVM)
1	87.93	23	89.71	26
2	86.19	27	90.56	23
3	89.63	23	87.13	21
4	89.56	23	89.53	23
5	89.13	26	89.92	29
6	87.89	34	87.31	21
7	88.23	23	89.59	23
8	88.73	29	87.94	29
9	86.27	21	88.43	22
10	89.76	27	87.29	23
Average± S.D	88.33±1.30	25.6±3.86	88.74±1.27	24±2.98

From table 5 it is seen that BCGA-SVM provides both highest accuracy which is 90.56% with only 23 genes from a total of 12533 genes and better average accuracy of 88.74% with 1.27 as standard deviation.

Results after applying the proposed approaches on Colon cancer dataset are given in Table 6.

Table 6: Classification accuracies and number of genes selected by Boosted Cyclic GA for colon cancer dataset

No. of Runs	Colon Cancer			
	BCGA-KNN	#Selected Genes(KNN)	BCGA-SVM	#Selected Genes(SVM)
1	87.34	24	88.23	23
2	86.47	19	86.65	26
3	86.32	19	86.74	19
4	89.43	18	87.43	22
5	88.54	22	85.76	23
6	88.75	21	88.84	19
7	88.83	19	86.64	18
8	87.85	23	87.82	24
9	85.23	19	89.73	20
10	88.27	19	91.43	22
Average ± S.D	87.70±1.34	20.3±2.06	87.93±1.70	21.6±2.55

Table 6 shows that the highest accuracy is achieved by BCGA-SVM which is 91.43% with only 22 genes and the best average accuracy is 87.93% with standard deviation of 1.70. Results for GA and CGA on leukemia cancer dataset, lung cancer dataset and colon cancer dataset are given in Table 7, Table 8 and Table 9 respectively.

Table 7: Classification accuracies and number of genes selected by GA and Cyclic GA for leukemia (ALL) cancer dataset

No. of Runs	Leukomia (ALL) Cancer			
	GASVM	#Selected Genes(GASVM)	CGASVM	#Selected Genes(CGASVM)
1	83.16	14	89.64	21
2	84.38	16	88.91	16
3	82.29	18	90.13	19
4	83.61	20	96.12	18
5	85.27	22	92.38	20
6	81.37	24	89.49	16
7	83.43	26	93.37	15
8	88.24	28	94.42	17
9	80.94	30	91.57	19
10	85.31	32	89.29	18

Average SD is 83.80±2.14 (GASVM), No. of Selected Genes are 23±6.06; Average SD for CGASVM (91.53±2.48), No. of Selected Genes 17.90±1.91**.** From Table 7 it is seen that best result for GASVM is 88.24% with 28 genes. On the other hand best result for CGASVM is 96.12% with only 18 genes and it has average 91.53% accuracy with standard deviation of 2.48, which is higher than basic GASVM method. Another noticing fact we have found is that CGASVM selects 17.90 genes, which can be rounded to 18 genes on average with standard deviation of 1.91; it is lesser than the GASVM method.

After applying the GASVM and CGASVM on lung cancer dataset it is found that CGASVM provides the best accuracy of 89.32% with only 32 genes among the two methods and it has also better average accuracy of 86.89% with 1.54 as standard deviation. Best accuracy for GASVM method is 87.92% with 32 genes and this method has an average of only 84.85% with standard deviation of 1.54.

Table 8: Classification accuracies and number of genes selected by GA and Cyclic GA for lung cancer dataset

No. of Runs	**Lung Cancer**			
	GASVM	**#Selected Genes(GASVM)**	**CGASVM**	**#Selected Genes(CGASVM)**
1	85.39	14	85.73	35
2	84.16	16	87.19	33
3	83.27	18	89.13	29
4	84.96	20	86.17	34
5	86.49	22	87.73	27
6	83.31	24	89.32	32
7	84.78	26	84.74	34
8	82.96	28	86.28	31
9	85.17	30	87.43	34
10	87.92	32	85.27	34
Average ± S.D	84.85±1.54	23±6.06	86.89±1.54	32.3±2.58

Table 9: Classification accuracies and number of genes selected by GA and Cyclic GA for colon cancer dataset

No. of Runs	**Colon Cancer**			
	GASVM	**#Selected Genes(GASVM)**	**CGASVM**	**#Selected Genes(CGASVM)**
1	84.76	14	87.63	27
2	82.43	16	84.83	26
3	84.89	18	84.72	28
4	83.94	20	85.96	25
5	81.54	22	86.13	25
6	85.32	24	86.43	24
7	86.27	26	85.39	26
8	82.64	28	86.13	25
9	81.76	30	89.74	23
10	81.29	32	86.64	26
Average ± S.D	83.48±1.77	23±6.06	86.36±1.46	25.5±1.43

From Table 9 it is see that the best accuracy is achieved by CGASVM which is 89.74% with only 23 genes and CGASVM yields 86.36% accuracy on average with 1.46 as standard deviation. On the other hand GASVM method provides 86.27% as highest accuracy with 26 genes and 83.48% accuracy on average with 1.77 as standard deviation. This indeed shows that CGASVM performs better than GASVM method.

4.3 Comparative Analysis

Table 10 and Table 11 show a comparison among BCGA-KNN, BCGA-SVM and related previous works namely GA and CGA by means of average accuracies, best accuracies and also the number of selected genes.

Table 10: Comparison of accuracies (%) obtained by Boosted Cyclic GA and other related previous methods.

Dataset	ACO	PSO	GA-SVM	CGA-SVM	BCGA-KNN	BCGA-SVM
Leukemia (Average ±SD; The Best)	83.89 ---- ----	84.22 ---- ----	83.80±2.14; 88.24	91.53±2.48; 96.12	96.08±1.28; 98.19	96.37±1.26; 97.98
Colon (Average ±SD; The Best)	76.87 ---- ----	79.69 ---- ----	83.48±1.77; 86.27	86.36±1.46; 89.74	87.70±1.34; 90.43	87.93±1.70; 91.43
Lung (Average ±SD; The Best)	79.76 ---- ----	80.09 ---- ----	84.85±1.54; 87.92	86.89±1.54; 89.32	88.33±1.30; 89.97	88.74±1.27; 90.56

From Table 10 it is seen that our proposed method Boosted Cyclic Genetic Algorithm has significantly outperformed previous related works namely Genetic Algorithm and Cyclic Genetic Algorithm. Both BCGA-KNN and BCGA-SVM provides better accuracy than the other two but BCGA-SVM shows better result than BCGA-KNN.

Table 11: Comparison of number of genes selected by Boosted Cyclic GA and other related previous methods

Dataset	Original Genes	Selected Genes			
		GA-SVM	CGA-SVM	BCGA-KNN	BCGA-SVM
Leukemia (Average ±S.D.)	12625	23±6.06	17.90±1.91	15.2±2.61	16.2±2.89
Colon (Average ±S.D.)	2000	23±6.06	25.5±1.43	20.3±2.06	21.6±2.55
Lung (Average ±S.D.)	12533	23±6.06	32.3±2.58	25.6±3.86	24±2.98

Table 11 shows that the proposed method has selected less number of genes than the other two methods. The average number of genes selected by BCGA-KNN is lesser than GA, CGA and BCGA-SVM except for the Lung cancer dataset where BCGA-SVM has selected lesser number of genes than other methods.

The comparison among different methods is graphically represented in Figure 1, Figure 2 and Figure 3 for leukemia, lung and colon cancer datasets respectively where X-axis represents number of selected genes and Y-axis represent accuracies (%) for each independent run.

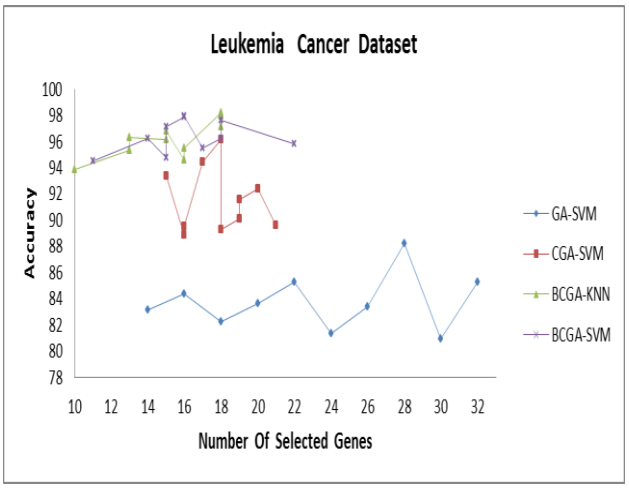

Figure 3: Graphical representation of comparison for leukemia (ALL) cancer dataset.

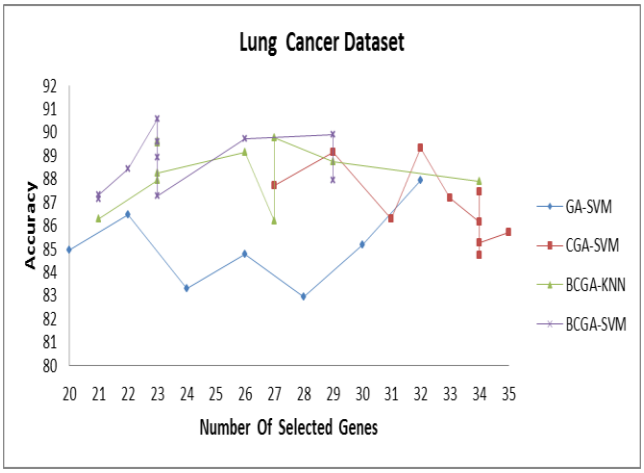

Figure 4: Graphical representation of comparison for lung cancer dataset.

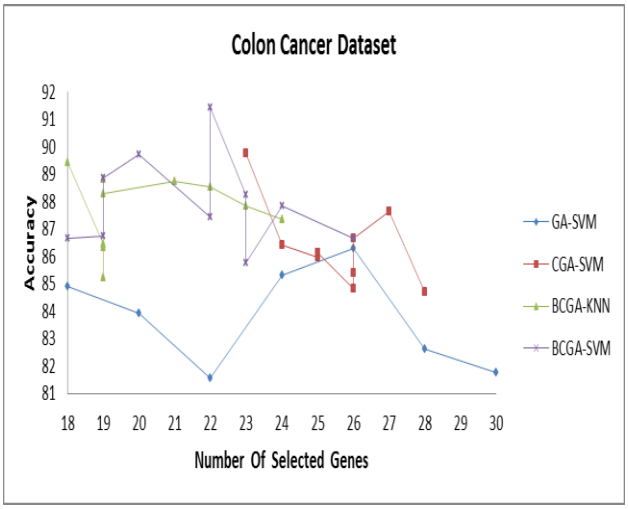

Figure 5: Graphical representation of comparison for colon cancer dataset.

From the graphs, a clear contrast is viewed between the performance of BCGA and other methods. For GA we predefined the number of genes for output. This number was set to be different for every independent run in order to observe the fluctuation of accuracies for different number of genes. From Figure 5 it is seen that CGA-SVM, BCGA-KNN and BCGA-SVM provide accuracies which fluctuate between the ranges 10-22 along the X-axis which represents the number of selected genes. With a closer inspection it can be observed that BCGA-KNN and BCGA-SVM provides better accuracies starting from 10 along the X-axis. Thus it is clear that BCGA starts exploration within the problem space from different points and yet it can provide better solution in most cases with lesser number of genes comparing with GA and CGA. The same phenomenon can be observed for other two datasets in Figure 6 and Figure 7.

5. CONCLUSION AND FUTURE WORK

In this study, the proposed framework is used only on microarray datasets. Traditional GA suffered from some problems namely random initial population, high convergence time, overfitting among others. Using the proposed framework the limitations of traditional GA were alleviated as shown earlier. Using a Boosted Filter Approach as the processing step, the random initial population and high convergence time problems problem were removed. Then using CGA overfitting problem was successfully reduced by incrementally reducing the steps in each iteration. For future development, this framework can also be used for other high dimensional data used in other fields such as archeology, geography, climate study, data mining, image processing and many others. The data analysis is expected to produce good results for these other datasets. Even in the field of specialization, the proposed framework was not used on all the microarray datasets such as brain cancer, bone cancer, stomach cancer etc. Also there are many classifiers available; in our study we have used SVM and KNN classifiers. Other classifiers such as C4.5, Naïve Bayes Classifier can also be integrated with the proposed framework for an enhanced comparative analysis.

6. ACKNOWLEDGEMENTS

We are grateful to almighty, the most merciful that by his boundless grace we are able to publish our research work. We am expressing our hearty complements and indebted to my honorable supervisors for their affectionate guidance, instructions & informative suggestions, helpful assistance, patience & encouragement throughout the whole term which made us to the do this research. Finally, we are grateful to all of our friends, seniors, juniors and colleagues who have always been part of our life. Continuous inspiration and understanding of these individuals enabled us to cross the hurdles of our life.

7. REFERENCES

[1] Correlation-based Feature Selection for Machine Learning, By Mark A. Hall, Department of Computer Science, The University of waikato, Hamilton, Newzealand.

[2] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittman, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown, "Expression Monitoring by Hybridization to High-density Oligonucleotide Arrays," *Nature Biotechnology*, Vol. 14, No. 13, pp. 1675–1680, 1996.

[3] T. S. Furey, N. Cristianini, N. Duffy, M. Schummer, D. W. Bednarski, and D.Haussler,"Support Vector Machine Classification and Validation of Cancer Tissue Sample Using Microarray Expression Data," *Bioinformatics*, Vol. 16, No. 10, pp. 906–914, 2000.

[4] Studies on Intelligent Approaches to Select Informative Genes from Gene Expression Data for Cancer Classification, Mohd Saberi Bin Mohamad, 大阪府立大学, 2009, 博士論文, 2009.

[5] Xian Xu and Aidong Zhang (2010), Boost Feature Subset Selection: A New Gene Selection Algorithm for Microarray Dataset, *State University of New York at Buffalo, Buffalo, NY 14260, USA.*

[6] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. GaasenBeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Blomfield, E.S. Lander **(**1999), Molecular classification of cancer: class discovery and class prediction by gene-expression monitoring, Science, 286, 531–537.

[7] Shital Shah, Andrew Kusiak (2007), Cancer gene search with data-mining and genetic algorithms, *Computers in Biology and Medicine 37 (2007) 251 – 261.*

[8] Mohd Saberi Mohamad, Sigeru Omatu, Safaai Deris, Michifumi Yoshioka (2010), A Three-Stage Method to Select Informative Genes from Gene Expression Data in Classifying Cancer Classes, *2010 International Conference on Intelligent Systems, Modelling and Simulation*.

[9] Yvan Saeys, I ˜naki Inza and Pedro Larra˜naga (2007), A review of feature selection techniques in bioinformatics, *Oxford University Press.*