AN ENHANCED GAGS BASED MTSVSL LEARNING TECHNIQUE FOR CANCER MOLECULAR PATTERN PREDICTION OF CANCER CLASSIFICATION

I. Julie[1] & E. Kirubakaran[2]
[1]Department of Computer Science, Arignar Anna Government Arts College, Musiri– 621 201, India
[2]Senior Deputy General Manager, BHEL, Trichy – 620 014, India

ABSTRACT

Cancer Classification is becoming the critical basis in patient therapy. Researchers are made continuously in developing and applying the most accurate classification algorithms based on the gene expression profiles of patients. Microarray technologies have made an enormous encroachment on cancer genome research. To predict the Cancer Classification, there are two methods namely Signal-to-Noise Ratio (SNR) based Genetic Algorithm on Gene Selection (GAGS) and Multi-Task Support Vector Sample Learning Technique (MTSVSL) had proposed. The GAGS is a Filter, which is used to select target genes in the diagnosis of cancer. The MTSVSL Learning Technique is a Wrapper, which is based on Back Propagation Neural Network and Linear Support Vector Machine. This work yield good classification accuracy for Leukaemia cancer genes. From the literature survey, this research work revealed that the classification performance interms of Accuracy and Error Rate could be improved if Counter Propagation Neural Network (CPNN) is combined with MTSVSL instead of BPNN. This is called as Enhanced MTSVSL (EMTSVSL) Learning Technique. From the experimental result, it is established that this proposed Technique achieves higher classification performance interms Accuracy and Error Rate as compared with existing technique.

Keywords: *Gene Prediction, Genetic Algorithm Gene Selection, Cancer Classification, Multi Task Learning, Support Vectors, Back Propagation Neural Network, Counter Propagation Neural Network.*

1. INTRODUCTION

Micro array technologies, which measure the expression level for thousands of gene expression simultaneously, have had a great impact on cancer genome research over the past few years. The Microarray Gene Selection[1,2,4,7] procedure is shown in the Figure 1. Currently, microarray-based gene expression profiling has been viewed as a promising approach in predicting cancer classes and prognosis outcomes. In most cases, cancer diagnosis depends on the use of a complex combination of clinical and histopathological data. However, it is often difficult or impossible to recognize a tumor type in some atypical instances. Large scale profiling of genetic expression and genomic alternations using DNA microarrays can reveal the differences between normal and malignant cells, genetic and cellular changes at each stage of tumor progression and metastasis, and the difference among cancers of different origins. Cancer classification is becoming the critical basis in patient therapy. Researchers are continuously developing and applying the most accurate classification algorithms based on the gene expression profiles of patients.

Several Data Mining techniques[1,2,4,5,7,8,9] such as Support Vector Machines (SVM), K-Nearest Neighbors, Ensemble Rough Hypercuboid Approach, Multiple-Filter-Multiple-Wrapper Approach, Principal Component Analysis (PCA), Nonnegative Principal Component Analysis (NPCA), Nonparallel Plane Proximal Classifier (NPPC), Back Propagation Neural Network and Multiple Filter with Multiple Wrapper (MFMW) had been proposed and applied in cancer diagnosis and classification.

In a microarray chip, the number of genes available is far greater than that of samples, a well-known problem called the curse of dimensionality [8]. However, most genes in a microarray give little benefits to the sample classification problem. Therefore, prior to sample classification, it is important to perform gene selection whereby more interpretable genes are identified as biomarkers, so that a more efficient, accurate, and reliable performance in classification can be expected. These biomarkers may also be useful for assessing disease risk [6] and understanding the basic biology of a disorder [8]. There are, in general, two approaches to gene selection, namely filters and wrappers [8]. The filter approach selects genes according to their discriminative powers with regard to the class labels of samples. Methods such as Signal-to-Noise Ratio (SNR), t-statistics (TS), threshold number of misclassifications (TNoM) score, and F-test have been shown to be effective scores for measuring the discriminative power of genes. In all cases, genes are ranked according to their statistical scores, and a certain number of the highest ranking genes are selected for the purpose of classification. However, these Filters have failed to select more interpretable genes. To overcome this identified problem, this paper planned to focus Signal-to-Noise Ratio (SNR) based Genetic Algorithm on Gene Selection (GAGS), which will improve the performance of Gene Selection Technique whereby more interpretable genes can be identified, so that a more efficient, accurate, and reliable performance in classification can be achieved.

In the wrapper approach, genes are selected sequentially one by one so as to optimize the training accuracy of a particular classifier [8]. That is, the classifier is first trained using one single gene, and this training is performed for the entire original gene set. The gene that gives the highest training accuracy is selected. Then, a second gene is added to the selected gene and the gene that gives the highest training accuracy for the two-gene classifier is chosen. This process is continued until a sufficiently high accuracy is achieved with a certain gene subset. From the literature survey, it is observed the existing classifiers such as Support Vector Machine (SVM), k-Nearest Neighbor have its own limitations such as False Positive and False Negative classification.

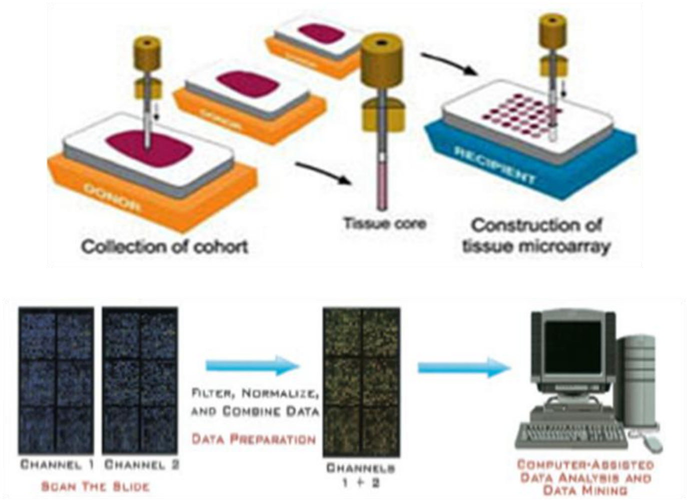


Figure. 1. Microarray Gene Selection Mechanism

To overcome this, Austin H and et.al., have proposed the MTSVSL Learning Technique, which is based on Back Propagation Neural Network and Linear Support Vector Machine. This work yield good classification performance in terms of Accuracy and Error Rate.

1.1 Objective of this Work

However, from our literature survey[1,2,8,9], it is identified that the performance of Multi-Task Support Vector Sample Learning (MTSVSL) technique could be improved if Counter Propagation Neural Network is introduced with Genetic Algorithm based Gene Selection (GAGS) rather Back Propagation Neural Network (BPNN), which can be named as Extended MTSVSL Learning Technique. This will achieve to find an optimal information gene subset, thereby avoiding the over-fitting problem caused by attempting to apply a large number of genes to a small number of samples.

2. BACKGROUND

In this Section, the features of Signal-to-Noise (SNR) Gene Selection Method, Genetic Algorithm based Gene Selection (GAGS) method, Support Vector Sampling Technique (SVS) and Multi-Task Learning (MTL) method are discussed.

2.1 Signal-to-Noise (SNR) based Gene Selection Method

Gene Selection is widely used to select target genes in the diagnosis of cancer. One of the prime goals of gene selection is to avoid the over-fitting problems caused by the high dimensions and relatively small number of samples of microarray data. Theoretically, in cancer classification, only informative genes which are highly related to particular classes should be selected. In the study of Austin H and et.al., it had used Signal-to-Noise Ratio (SNR) as the Gene Selection method [1]. For each gene, this work has normalized the gene expression data by subtracting the mean and then dividing by the standard deviation of the expression value. Every sample is labeled with {+1,-1} as either a normal or a cancer sample. The following formula is used to calculate each gene's F score.

$$F(g_i)=\frac{|\mu^{+1}(g_i)-\mu^{-1}(g_i)|}{\sigma^{+1}(g_i)+\sigma^{-1}(g_i)}$$ ……………………………..(1)

The µand σ are the mean and standard deviation of the samples in each class (either +1or -1) individually. This work rank these genes with an F score

2.2 Genetic Algorithm based Gene Selection (GAGS) Technique

The genetic algorithm[1] is an effective algorithm in searching complex high-dimensional space and in finding the optimal solution. Austin H and et.al., proposed this Genetic Algorithm based Gene Selection method that can find the most informative gene set. The genetic algorithm is a type of evolutionary computing method widely used in simulating the process of natural selection. The basic concept behind the genetic algorithm is consisted of four steps. They are

- Population
- Reproduction
- Crossover And
- Mutation

Before beginning the genetic algorithm, this work has randomly separated the gene expression data into three parts. They are Testing Dataset, Training Dataset And Validation Dataset. The Testing Dataset is an independent dataset used purely for measuring the classification performance.

2.3 Population

Here, all the genes are randomly separated into m chromosomes and each chromosome contains n genes. Each chromosome represents a possible gene subset. The system is designed to set the value of m and n depends upon the requirement.

2.4 Reproduction

In the biological evolutionary process, only the organisms that adapt to the environment survive. Only chromosomes with high fitness scores replicate and are passed onto the next stage. The fitness function is defined as

$$Fitness=\frac{1}{3}ATR+\frac{2}{3}ATV$$ ……………………………..(2)

where ATR is the predictive accuracy of the training dataset using the support vector machine and ATV is the predictive accuracy of the validation dataset. The reproduction rate may influence the variety of chromosomes. If the variety of chromosomes is low, the genetic algorithm may catch a local optimum solution instead of a global optimum solution.

2.5 Crossover

After the reproduction phase, offsprings are created by crossing over the parent chromosomes at the cross point. The single-point crossover approach was used. The crossover point is randomly generated and two chromosomes are randomly selected to do so at this point

2.6 Mutation

To increase the possibility of finding the optimal solution, a mutation phase is applied. We will set P and p as the mutation possibility of each chromosome and each gene respectively. Here, every chromosome may generate a random number R, and if R > P then this chromosome will be added to the mutation pool. Every gene in these chromosomes may also generate a random number r, where if r > p then the gene will be replaced with another randomly selected gene from the F-gene pool.

2.7 Multi-Task Support Vector Sample Learning (MTSVSL)

This Multi-Task Support Vector Sample Learning (MTSVSL) has two methodologies[1]. These technologies combined together to improve the classification accuracy from the gene expression data. The technologies are

Support Vector Sample (SVS) method and Multi-Task Learning (MTL) Method. By using this approach, a classifier can learn two tasks. They are i. the main task is "which kind of sample is this?" and the second task is "is this sample a support vector sample?". This work categorize the samples into four classes, namely

1. The sample which belongs to class 1 and is a support vector sample
2. The sample which belongs to class 2 and is a support vector sample
3. The sample which belongs to class 1 but is not a support vector sample
4. The sample which belongs to class 2 but is not a support vector sample

2.8 Support Vector Sampling Technique (SVS)

A binary SVM[1,8,9] attempts to find a hyperplane which maximizes the "margin" between two classes (+1/-1).
Let

$$\{X^i, Y^i\} i = 1,2...j, Y^i \in \{-1,1\}, X \in R \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

be the gene expression data with positive and negative class labels. The SVM learning algorithm should find a maximized separating hyperplane W * X +b = 0, where W is the n-dimensional vector, which is called the normal vector that is perpendicular to the hyperplane, and b is the bias. The SVM decision function is showed in formula(4), where α_i is a positive real numbers and φ is mapping function

$$W^T \phi(X) + b = \sum_{i=1}^{j} \alpha_i Y_i \phi(X_i)^T \phi(X) + b \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

Only $\phi(X_i)$ of $\alpha_i > 0$ would be used, and these points are support vectors. The support vectors lay close to the separating hyperplane. Here $0 < \alpha_i < C$, where C is the penalty parameter of Error Term. If α_i becomes zero, there is no influence to the hyperplane.

2.9 Multi-Task Learning (MTL) method

The principle goal of multi-task learning[1] is to improve the performance of a classifier. The multi-task learning technique can be considered as an inductive transfer mechanism where the inductive transfer leverages additional sources of information to improve learning performance within a current task. Variables which were not used as the initial inputs may contain some useful information. Instead of discarding these variables, MTL get the inductive transfer benefit from discarded variables by using them as an extra output. The Back Propagation Neural Network (BPNN) is modeled as MTL and learn tasks.

2.10 Identified Problems

From our literature survey, it is identified that the performance of Multi-Task Support Vector Sample Learning (MTSVSL) technique is improved as compared with Back Propagation Neural Networks. However, the learning technique of MTSVSL has failed to select more interpretable genes and hence unable to improve the classification accuracy. That is the Wrapper of this system leads to poor Gene Classification. This is the major drawback. To overcome this identified problem, this paper planned to improve the performance of Wrapper.

3. ENHANCED MTSVSL

As stated in the previous section, the Multi-Task Support Vector Sample Learning (MTSVSL) technique has two methodologies namely Support Vector Sample (SVS) Technique and Multi-Task Learning (MTL) Technique. These technologies combined together to improve the classification accuracy of the gene expression data. The main objective of this work is to improve the performance of MTSVSL. That is this work is improved the performance of MTL with Counter Propagation Neural Networks.

3.1 The Principle of Counter Propagation Neural Networks

The Counter-Propagation Network is a combination of a portion of the Kohonen Self-Organizing Map [10] and Grossberg Outstar Structure [10]. During learning, pairs of the input vector X and output vector Y are presented to the input and interpolation layers, respectively. These vectors propagate through the network in a counterflow manner to yield the competition weight vectors and interpolation weight vectors. Once these weight vectors become stable, the learning process is completed.

The output vector Y^1 of the network corresponding to the input vector X is then computed. The vector Y^1 is intended to be an approximation of the output vector Y, i.e. $Y^1 \approx Y = f(X)$. The equations of the network are described briefly as follows.

Let Uj = [u_{ji}] be the arbitrary initial competition weight vector for the j-th neuron in the competition layer where u_{ji} is the weight connecting the j-th neuron in the competition layer to the i-th neuron in the input layer. The Euclidean distance between the input vector X and the competition weight vector U_j of the j-th neuron is calculated, That is

$$d_j = \| X - U_j \| = \sqrt{\sum_{i=1}^{m} (x_i - u_{ji})^2} \qquad \text{………………………(5)}$$

Once the distance d_j for each neuron has been calculated, the neuron with the shortest Euclidean distance to X is selected to represent the winning neuron. As a result of the competition, the output of the winning neuron is set to unity and the outputs of the other neurons are set to zero. Thus, the output of the j-th neuron in the competition layer can be expressed as

$$Z_j = \begin{cases} 1.0 & \text{if } \mathrm{d_j} < d_i \text{ for all i} \\ 0.0 & \text{otherwise} \end{cases} \qquad \text{………………………(6)}$$

The weight u_{ji} connecting the j-th neuron in the competition layer to the i-th neuron in the input layer is adjusted based on the Kohonen learning rule, that is

$$u_{ji}(p+1) = u_{ji} + \beta(x_i - u_{ji}(p))Z_j \qquad \text{………………………(7)}$$

where **β** is the learning coefficient and ***p*** is the iteration number. After the competition weight vector $\mathbf{U_j}$ stabilizes, the interpolation layer starts to learn the desired output vector **Y** by adjusting the interpolation weight vector. Let **Vj = [$\mathbf{v_{ji}}$]** be the arbitrary initial interpolation weight vector for the *j*-th neuron in the interpolation layer where $\mathbf{v_{ji}}$ is the weight connecting the *j*-th neuron in the interpolation layer to the *i*-th neuron in the competition layer. The weight $\mathbf{v_{ji}}$ is adjusted based on the Grossberg learning rule, that is

$$v_{ji}(p+1) = v_{ji} + \gamma(y_i - v_{ji}(p))Z_i \qquad \text{………………………(8)}$$

where **γ** is the learning coefficient.

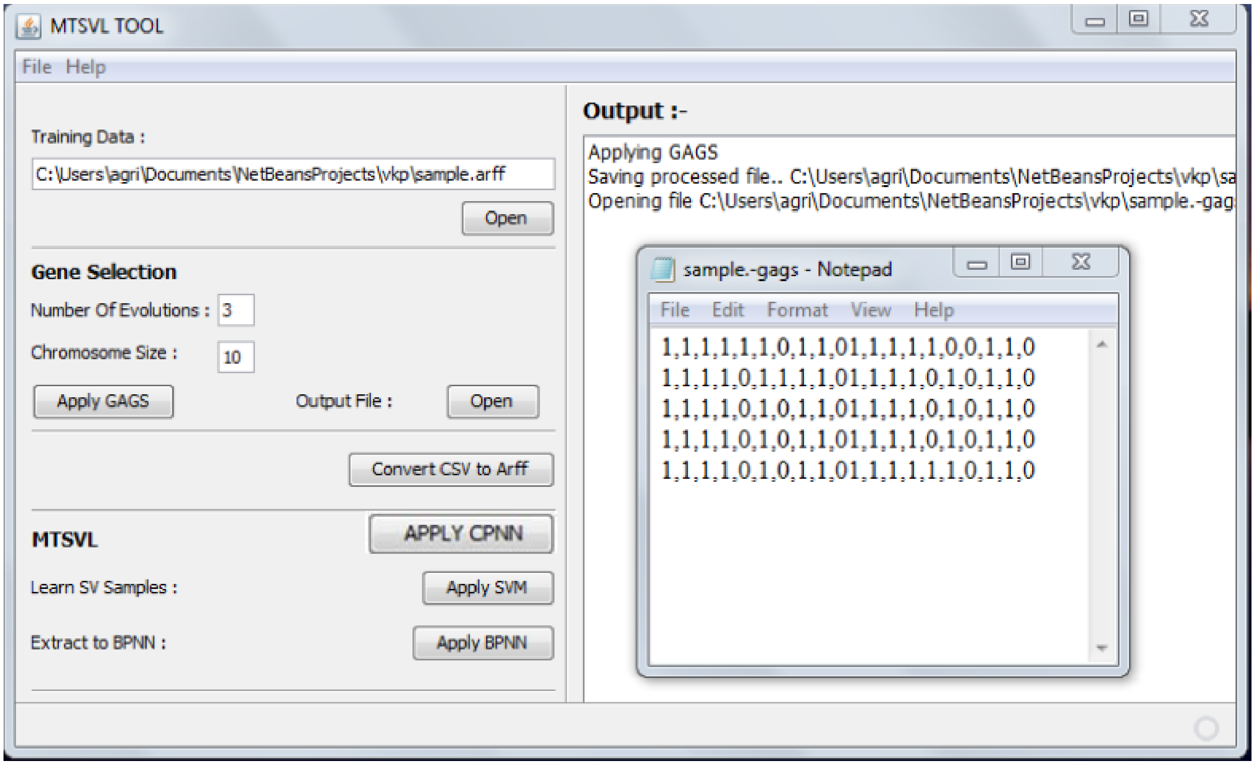
This is repeated until the interpolation weight vector $\mathbf{V_j}$ converges to a preset value. The output vector Y^1 of the network corresponding to the input vector ***X*** can be calculated using a weighted summation function. The *j*-th component $\mathbf{y^1_j}$ of the output vector $\mathbf{Y^1}$can be expressed as

$$y_j^1 = \sum v_{ji} Z_i \qquad \text{………………………(9)}$$

In the foregoing discussion, the counter-propagation network functions as a look-up table. The learning process associates the input vector with the corresponding output vector based on two well-known algorithms, namely the Kohonen self-organizing map for finding the most similar training vector and the Grossberg outstar map for projecting the corresponding output vector. Once the network is trained, the application of an input vector can quickly produce the corresponding output vector. This is the enhanced MTL

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

We have been developed MTSVSL Tool with NetBeans and it is configured with BioWeka0.6.1. As shown in the



Training Data : C:\Users\agri\Documents\NetBeansProjects\vkp\sample.arff

Number Of Evolutions : 3

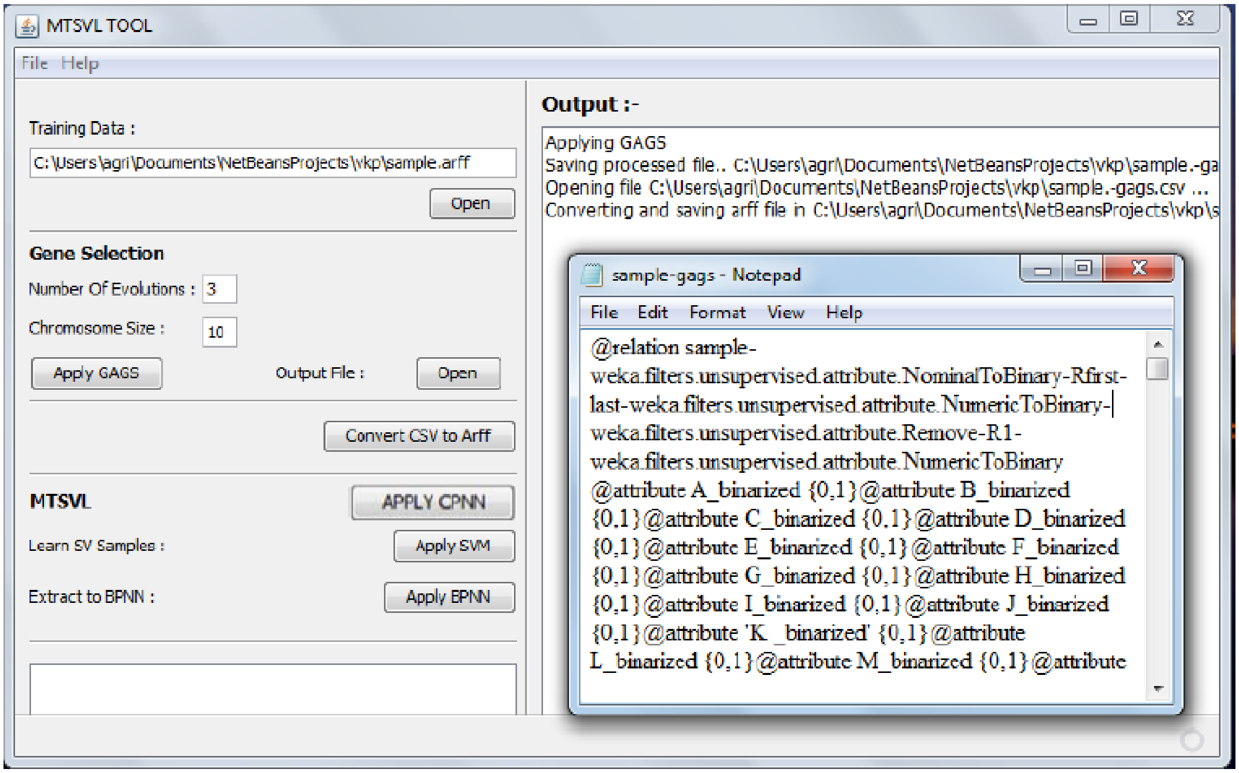
Chromosome Size : 10

Output :-

```
Applying GAGS
Saving processed file.. C:\Users\agri\Documents\NetBeansProjects\vkp\sa
Opening file C:\Users\agri\Documents\NetBeansProjects\vkp\sample.-gag
```

```
1,1,1,1,1,1,0,1,1,01,1,1,1,1,0,0,1,1,0
1,1,1,1,0,1,1,1,1,01,1,1,1,0,1,0,1,1,0
1,1,1,1,0,1,0,1,1,01,1,1,1,0,1,0,1,1,0
1,1,1,1,0,1,0,1,1,01,1,1,1,0,1,0,1,1,0
1,1,1,1,0,1,0,1,1,01,1,1,1,1,1,0,1,1,0
```

Figure. 2. GAGS based MTSVSL Tool with BioWeka0.6.1



Training Data : C:\Users\agri\Documents\NetBeansProjects\vkp\sample.arff

Number Of Evolutions : 3

Chromosome Size : 10

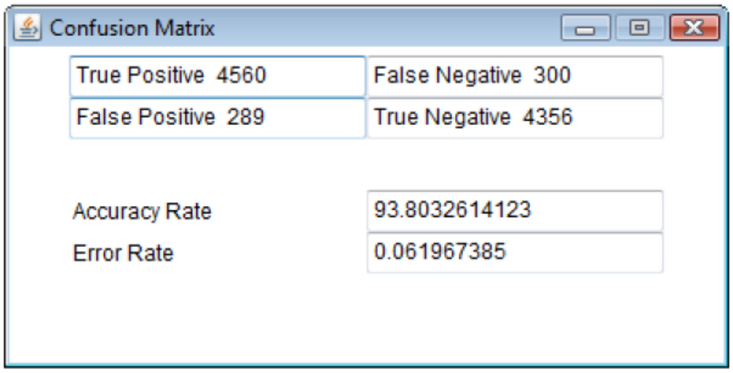
Output :-

```
Applying GAGS
Saving processed file.. C:\Users\agri\Documents\NetBeansProjects\vkp\sample.-ga
Opening file C:\Users\agri\Documents\NetBeansProjects\vkp\sample.-gags.csv ...
Converting and saving arff file in C:\Users\agri\Documents\NetBeansProjects\vkp\s
```

```
@relation sample-
weka.filters.unsupervised.attribute.NominalToBinary-Rfirst-
last-weka.filters.unsupervised.attribute.NumericToBinary-
weka.filters.unsupervised.attribute.Remove-R1-
weka.filters.unsupervised.attribute.NumericToBinary
@attribute A_binarized {0,1}@attribute B_binarized
{0,1}@attribute C_binarized {0,1}@attribute D_binarized
{0,1}@attribute E_binarized {0,1}@attribute F_binarized
{0,1}@attribute G_binarized {0,1}@attribute H_binarized
{0,1}@attribute I_binarized {0,1}@attribute J_binarized
{0,1}@attribute 'K _binarized' {0,1}@attribute
L_binarized {0,1}@attribute M_binarized {0,1}@attribute
```

Figure. 3. MTSVSL SVM Classification

Figure 2, it consists of two modules. The first module is a Filtering Module, where GAGS is implemented. In this module, the Chromosome Size can be fixed. The second module is the Wrapper Module, where MTSVSL and EMTSVSL have been implemented. As shown in the Figure 3, SVM with BPNN is classifying the Cancer Pattern from the Dataset. For experimental study, the work is considered Leukaemia Cancer Pattern Datasets and number of Top Genes are taken as 100 and 150. The Confusion Matrices and their Accuracy and Error Rate are shown in the Figure from Figure. 4. to Figure. 7.

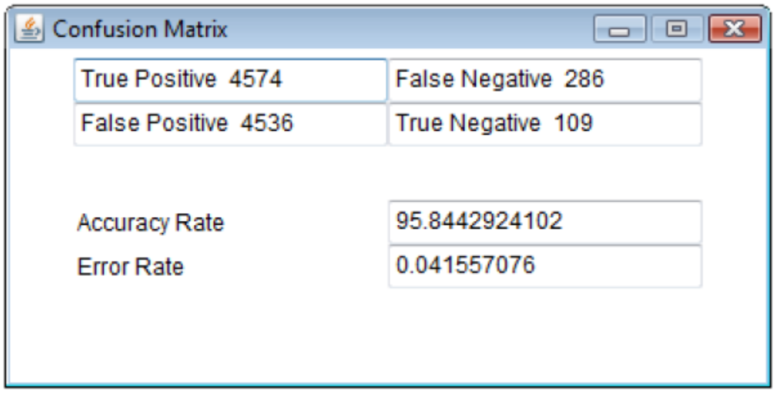


True Positive 4560	False Negative 300
False Positive 289	True Negative 4356

Accuracy Rate	93.8032614123
Error Rate	0.061967385

Figure. 4. Confusion Matrix of MTSVSL for Leukaemia Cancer pattern (Top Genes : 100)

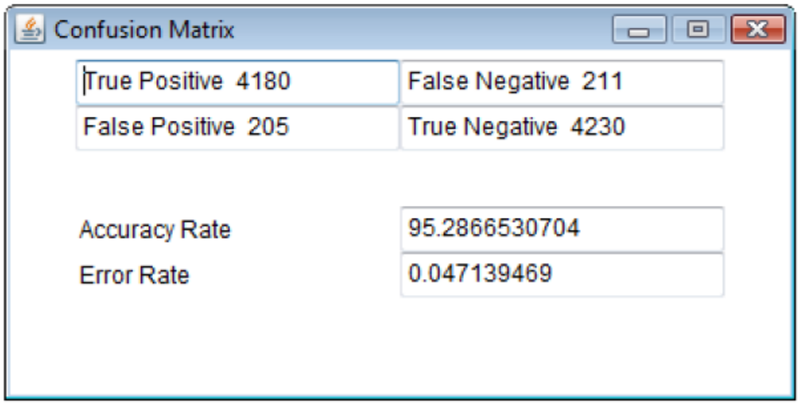
From the Figure 4, it is noted that the existing GAGS based MTSVSL obtained 93.8033 is the Classification Accuracy and 0.06197 is the Error Rate for Leukaemia Cancer pattern with number of Top Genes are 100. And also observed that this proposed GAGS based EMTSVSL Technique achieves 95.8443 and 0.4156 as its Classification Accuracy and Error Rate respectively, which is shown in the Figure 5. It is revealed that our proposed work performs well as compared with existing system. With Top Genes as 150, the same experiment is repeated, which is shown in the Figure 6 and Figure 7 and also realized that this proposed work outperform GAGS based MTSVSL.



True Positive 4574	False Negative 286
False Positive 4536	True Negative 109

Accuracy Rate	95.8442924102
Error Rate	0.041557076

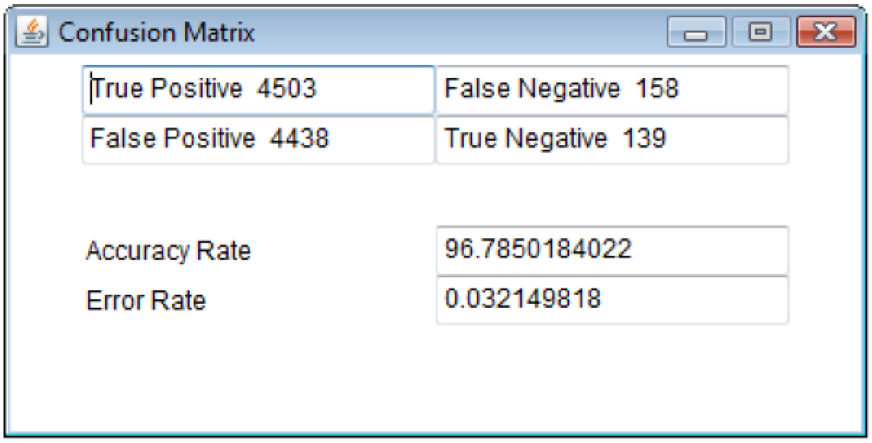
Figure. 5. Confusion Matrix of EMTSVSL for Leukaemia Cancer pattern (Top Genes : 100)



True Positive 4180	False Negative 211
False Positive 205	True Negative 4230

Accuracy Rate	95.2866530704
Error Rate	0.047139469

Figure. 6. Confusion Matrix of MTSVSL for Leukaemia Cancer pattern (Top Genes : 150)



Confusion Matrix

True Positive 4503	False Negative 158
False Positive 4438	True Negative 139

Accuracy Rate	96.7850184022
Error Rate	0.032149818

Figure. 7. Confusion Matrix of EMTSVSL for Leukaemia Cancer pattern (Top Genes : 150)

5. CONCLUSION

Microarray technologies have made an enormous encroachment on cancer genome research. To predict the Cancer Classification, the GAGS is used to select target genes in the diagnosis of cancer and the MTSVSL Learning Technique based on Back Propagation Neural Network and Linear Support Vector Machine were implemented for classification. To improve its classification accuracy, this paper proposed an efficient enhanced MTSVSL (EMTSVSL) is proposed. From the experimental result, it is established that this proposed Technique achieves higher classification accuracy with less error rate as compared with existing MTSVSL Technique. For experimental study, the Leukaemia Cancer Pattern is used.

REFERENCES

[1]. Austin H, Chen and Jen-Chieh Hsu, "Exploring novel algorithms for the prediction of cancer classification," International Conference on Software Engineering and Data Mining (SEDM), ISBN: 978-1-4244-7324-3 pp. 378 – 383, 2010.

[2]. Statnikov A, Aliferis CF, Tsamardinos I, Hardin D, Levy S, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," Bioinformatics , 2005, vol. 21, pp. 631–643

[3]. Ramaswamy S. et al., "Multiclass cancer diagnosis using tumour gene expression signatures," Proc. Natl Acad. Sci. USA 98, 2001 ,_ pp. 15149–15154.

[4]. Greer BT, Khan J, "Diagnostic classification of cancer using DNA microarrays and artificial intelligence," Ann N Y Acad Sci , 2004, vol. 1020, pp. 49-66.

[5]. Ramirez L, Durdle NG, Raso VJ, Hill DL, "A support vector machines classifier to assess the severity of idiopathic scoliosis from surface topology," IEEE Trans. Inf. Technol. Biomed., 2006, 10, no. 1, pp. 84-91, Jan. 2005.

[6]. Y. Wang, I.V. Tetko, M.A. Hall, E. Frank, A. Facius, K.F.X. Mayer, and H.W. Mewes, "Gene Selection from Microarray Data for Cancer Classification—A Machine Learning Approach," Computational Biology and Chemistry, vol. 29, no. 1, pp. 37-46, 2005.

[7]. Rhodes, and et.al., "Oncomine 3.0: Genes, Pathways, and Networks in a Collection of 18,000 Cancer Gene Expression Profiles," Neoplasia, vol. 9, no. 2, pp. 166-180, 2007.

[8]. Yukyee Leung and Yeungsam Hung, "A Multiple Filter Multiple Wrapper to gene selection and microarray data classification," IEEE/ACM Transcations computational Biology and Bioinformatics, VOL. 7, NO. 1, JANUARY-MARCH 2010.

[9]. Minghao Piao, Jong Bum Lee, Khalid E.K. Saeed, and Keun Ho Ryu, "Discovery of significant classification rules from Incrementally inducted decision tree ensemble for diagnosis of disease". 2009.

[10]. S.C. Juang, Y.S. Tarng, and H.R. L, "A comparison between the back-propagation and counter-propagation networks in the modeling of the TIG welding process," Journal of Materials Processing Technology, pp. 54 – 63, 1998.