A comprehensive evaluation of machine learning techniques for cancer class prediction based on microarray data

Khalid Raza* and Atif N. Hasan

Department of Computer Science,
Jamia Millia Islamia (Central University),
New Delhi 110025, India
Email: kraza@jmi.ac.in
Email: atifnhasan@gmail.com
*Corresponding author

Abstract: Prostate cancer is among the most common cancer in males and its heterogeneity is well known. The genomic level changes can be detected in gene expression data and those changes may serve as standard model for any random cancer data for class prediction. Various techniques were implied on prostate cancer data set in order to accurately predict cancer class including machine learning techniques. Large number of attributes but few numbers of samples in microarray data leads to poor training; therefore, the most challenging part is attribute reduction or non-significant gene reduction. In this work, a combination of interquartile range and *t*-test is used for attribute reduction. Further, a comprehensive evaluation of ten state-of-the-art machine learning techniques for their accuracy in class prediction of prostate cancer is done. Out of these techniques, Bayes Network outperformed with an accuracy of 94.11% followed by Naïve Bayes with an accuracy of 91.17%.

Keywords: cancer class prediction; machine learning; microarray analysis; prostate cancer.

Reference to this paper should be made as follows: Raza, K. and Hasan, A.N. (2015) 'A comprehensive evaluation of machine learning techniques for cancer class prediction based on microarray data', *Int. J. Bioinformatics Research and Applications*, Vol. 11, No. 5, pp.397–416.

Biographical notes: Khalid Raza is an Assistant Professor at the Department of Computer Science, Jamia Millia Islamia, New Delhi. He obtained his PhD in the area of Soft Computing and Computational Biology. He has contributed over 15 research articles in refereed international journals, conference proceedings and as book chapters. He has been PI of two Govt. funded research projects. He is reviewer of several international journals, member of several conference review committees, and member/life member of ACM, CSI, SCRS, AIRCC, and MIR-Lab.

Atif N. Hasan has completed his MSc (Bioinformatics) from Jamia Millia Islamia. Currently, he is working as a research volunteer and preparing for PhD.

1 Introduction

Tumour state of prostate cancer is difficult to detect, as prostate cancer is heterogenic in nature (Aihara et al., 1994). The conventional diagnostic techniques are not always effective as they rely on the physical and morphological appearance of the tumour. Early stage prediction and diagnosis is difficult with those conventional techniques. Moreover, these techniques are costly, time-consuming, and require large laboratory set-up and highly skilled persons. Cancers are involved in genome-level changes (Stratton et al., 2009). Thus, it implies that for a specific type of cancer there could be pattern of genomic change. If those patterns are known, then it can serve as a model for the detection of that cancer (Singh et al., 2002) and will help in making better therapeutic decisions.

Owing to recent advancements in high-throughput techniques for measuring gene expression, it is now possible to monitor the expression levels of tens of thousands of gene at once. Several researchers have done significant researches, using microarray gene expression data to classify cancers (Golub et al., 1999), but still predicting cancer class with high accuracy remains a challenge. An overview of the analysis and interpretation of various omics data can be found in (Dinasarapu et al., 2013). In this paper, we have done a comparative evaluation of several machine learning techniques for their accuracy in predicting the cancer sample class, i.e. tumour or normal.

The main difficulty with any machine learning technique is to get trained with large number of genes and comparatively very few samples (Saeys et al., 2007). This is known as *'curse of dimensionality'* problem. Machine learning is effectively good when the samples are more and attributes are less but this is rarely possible with gene expression data. Thousands of genes in gene expression data make the data huge and tough for any machine learning technique to get trained on it. Attribute reduction or gene filtering makes data more meaningful (Quackenbush, 2002). Most of the genes do not participate in tumour development which means that they are irrelevant for cancer prediction. Many researchers have used various techniques for attribute reduction or gene filtering. We have applied combination of statistical techniques such as quartile range and *t*-test, which has been effective in filtering significant genes and minimising noise and irrelevant attributes from data.

Ten different machine learning techniques were used, such as *Bayes Network (BN)*, *Naive Bayes (NB)*, *Logistic Model Tree (LMT)*, *C4.5*, *Decision Table (DT)*, *Sequential Minimal Optimisation Support Vector Machine* (*SMO-SVM)*, *LogitBoost (LB)*, *Random Forest (RF)*, *Neural Network (NN)* and *Genetic Algorithm* (GA).

2 Previous related works

Several machine learning techniques (Lu and Han, 2003), such as SVM (Guyon et al., 2002), *k*-Nearest Neighbours (*k*NN) (Varambally et al., 2005), Artificial Neural Networks (ANN) (Khan et al., 2001), Genetic Programming (Vanneschi et al., 2011), Genetic Algorithms (Jirapech-Umpai and Aitken, 2005), Bayesian Network (Friedman et al., 2000a), Naive Bayes (Wang et al., 2005), Decision Trees (Brown et al., 2000), Rough Sets (Wang and Gotoh, 2009), Emerging Patterns (Li and Wong, 2002), Self-Organising Maps (Hsu et al., 2003), have been used for feature selection, attribute reduction, class prediction and classification using gene expression data. SVM has been used for knowledge-based gene expression analysis, recognition of functional classes of genes

(Brown et al., 2000), and gene selection (Tang et al., 2007). A statistical-based method has been used in Raza and Mishra (2012) for identification and filtration of most significant genes that can act as a best drug target. *k*NN was successfully used for making a model which was capable of predicting the identity of unknown cancer samples (Singh et al., 2002). The problem of gene reduction from huge microarray data set was solved by neural network (O'Neill and Song, 2003); moreover, it was able to identify the genes responsible for particular type of cancer occurrence. Genetic Algorithm was used for building selectors where the state of allele denotes whether it (gene) is selected or not (Liu et al., 2005). Genetic Programming has been shown to work excellent in case for recognition of structures for large data sets (Moore et al., 2001). It was also applied to microarray data to generate programs that predict the malignant states of cancerous tissue, as well as to classify different types of tissues (Rosskopf et al., 2007). Bayesian Networks were well applied for identification of gene regulatory network from time course microarray data (Zou and Conzen, 2005). Self-Organising Maps show good result for gene clustering (Sturn et al., 2002). Naive Bayes has been used by several researchers for gene selection and classification (Li et al., 2004; Yeung et al., 2005). Emerging Pattern is markedly good for microarray data analysis. Moreover, it has an advantage of designing interaction among genes which enhances classification accuracy (Boulesteix et al., 2003). In our recent work, we constructed a prostate cancer-specific gene regulatory network from gene expression profiles and identified some highly connected hub genes using Pearson's correlation coefficient (Raza and Jaiswal, 2013). We also applied information theoretic approach for reconstruction and analysis of gene regulatory network of colon cancer in one of our recent works (Raza and Parveen, 2013). But still which method one should apply for the classification and prediction of a particular cancer with high accuracy remains a challenge.

3 Materials and methods

The prostate cancer data set was taken from Kent Ridge Biomedical Data Repository. Data were collected from the pioneer publication of Singh et al. (2002). In his experiment, 235 radical prostatectomy specimens were analysed from different patients. From that, 65 samples specimen were identified for having tumour on opposing side of the tissue. Gene expression profile was successfully carried for 52 prostate tumour and 50 non-tumour prostate samples containing expression profile of 12,600 genes. This makes the training data set for our experiment. Whereas the test data set was having nearly tenfold difference in overall microarray, intensity forms the training data set and was taken from independent experiment. It has 25 tumour samples and nine non-tumour/ normal samples. The methodologies used in this study are discussed as follows and pseudo-code of analysis pipeline used in this paper is given in Figure 1.

Data Normalisation and Attribute Reduction: As the available data set has large number of genes compared to samples, before training machine learning we have done data normalisation using Inter-Quartile Range (IQR) and attribute reduction using *t*-test. Normalisation is a data pre-processing technique used to rescale attribute values to fit in a specific range. Normalising data is important when dealing with attributes of different units and scales. The IQR is a measure of dispersion which is defined as the difference between the upper and lower quartiles, i.e. IQR = $Q3-Q1$, where the first quartile $Q1$

represents a quarter and the third quartile *Q3* represents three quarters of the list of all the data. The IQR is essentially the range of the middle 50% of the data and hence it is not affected by outliers or extreme values.

After normalising data we used a two-tailed *t*-test for extracting differentially expressed genes among two types of sample, i.e. normal and tumour, at a significant level $\alpha = 0.001$. General formula of *t*-test of unequal sample size is:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{S_{x_1 x_2}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{1}$$

where

$$S_{x_1 x_2} = \sqrt{\frac{(n_1 - 1)S_{x_1}^2 + (n_2 - 1)S_{x_2}^2}{n_1 + n_2 - 2}} \tag{2}$$

x_1 and x_2 are two unequal samples and $\bar{x}_1 - \bar{x}_2$ is the difference between the sample mean. S_{x1x2} is the standard error of the difference, n_1 and n_2 are the size of the samples. In this case, we have n_1 and n_2 as 52 and 50 for tumour and normal samples, respectively. After applying *t*-test, out of 12,600 genes, 856 genes were extracted out. These genes are the differentially expressed genes. From the test data set, we extracted out the same genes as remained in the training data set after normalisation and attribute reduction.

Figure 1 Pseudo-code of our complete analysis pipeline

Algorithm: Evaluation of machine learning techniques for cancer class prediction

1: Prostate cancer data taken from Kent Ridge Biomedical Data Repository
2: Data normalisation using IQR
   IQR = Q3 – Q1
3: Attribute reduction using *t*-test (significance $\alpha = 0.001$)
4: Training & testing machine learning techniques:
   Bayes Net, Naive Bayes, Logit Boost,
   Logistic Model Tree, Random Forest, Decision Table,
   SMO-SVM, Neural Network and Genetic Algorithm
5: Cross validation of results by modifying the original test datasets by four different way
   modified_dataset1 = (original_dataset)/2
   modified_dataset2 = (original_dataset)/10
   modified_dataset3 = (original_dataset)/20
   modified_dataset4 = (original_dataset)*2
   modified_dataset5 = (original_dataset)*10
   modified_dataset6 = (original_dataset)*20
6: Sensitivity, specificity, precision and accuracy analysis of prediction results
7: Comparison of results with the work of others

3.1 Classification and machine learning techniques

The classification is a term that covers any context where some decisions or forecasts are made on the basis of currently available information. The construction of classifiers from a set of previously available data for which the true classes are known is called supervised learning. Machine learning is an automatic computing procedure that learns a task from a series of examples. It aims to generate classifying expressions simple enough to be understood by the human and mimic human reasoning sufficiently to provide insight into the decision process. The main goal of a learning machine is to achieve generalisation from its learned experience, i.e. ability to perform accurately on new and unseen examples.

We have taken following state-of-the-art machine learning techniques for the cancer classification:

1. *Naive Bayes:* It is a simple probability-based techniques mainly based on Bayes theorem with high independence assumption (Friedman et al., 1997). The presence or absence of any attribute is not dependent on other. It requires small amount of training data in order to estimate the parameters required for classification (Zhang, 2004; Rish, 2001). The probability of posterior (p) which depends over a class variable $\boldsymbol{C}$ conditional on variable features F_1, …, F_n, where n is the number of features, is given by:

$$p\left(C\middle|F_1,\ldots,F_n\right)=\frac{p(C)p\left(F_1,\ldots,F_n\middle|C\right)}{p\left(F_1,\ldots,F_n\right)} \tag{3}$$

A Naive Bayes classifier is a function that assigns a class label to an example. According to Bayes Rule and from the probability perspective, the probability of an example $E = (x_1, x_2, \ldots, x_n)$ being class c is:

$$p(c\,|\,E)=\frac{p(E\,|\,c)P(c)}{P(E)} \tag{4}$$

When all attributes are independent and the value of class variable is given, that is

$$p(c\,|\,E)=p(x_1,x_2,\ldots,x_n\,|\,c)=\prod_{i=1}^{n}p\left(x_i\,|\,c\right) \tag{5}$$

Then the resulting classifier is

$$f_{nb}(E)=\frac{p(C=+)}{p(C=-)}\prod_{i=1}^{n}\frac{p\left(x_i\,|\,C=+\right)}{p\left(x_i\,|\,C=-\right)} \tag{6}$$

Where the function $f_{\mathrm{nb}}(E)$ is called Naive Bayesian classifier.

2. *Bayes Net:* Bayes Net was developed for improving the performance of Naive Bayes. These are the directed acyclic graphs (DAGs) which allow an effective representation of joint probability distribution for a set of random variables (Friedman et al., 1997). Consider a finite set U = {X_1, …, X_n}, of discrete random variables X_i, where each value of X_i may take values from a finite set. The vertices of the DAG corresponds to the random variables X_i, …, X_n. The edges of the graph show direct dependencies between the variables. The graph develops independent assumption; if the given graph is the parent graph of X_i then each variable X_i is

independent of its non-descendants. The 'θ' component of the graph represents the set of parameters that quantifies the network.

$$\theta_{xi|\Pi xi} = P_B(_{xi|\Pi xi}) \tag{7}$$

The joint probability distribution P_B over the variables of set U is given by:

$$P_B(X_1,\ldots,X_n) = \prod_{i=1}^{n} P_B\left(X_i \mid \prod X_i\right) = \prod_{i=1}^{n} \theta_{X_i|\prod X_i} \tag{8}$$

A training set D = $\{t_1, \ldots, t_n\}$ of instances of T finds a network B which best matches the training set D**.**

3. *LogitBoost:* Boosting was well describe by 'Freund and Schapire' that it is a classification which works by sequential implementation of a classification algorithm to reweighted training data and then taking the sequence classifiers produced by the weighted majority vote (Friedman et al., 2000b). For two classes problem boosting can be taken as an approximation to additive modelling on the logistic scale based on Bernoulli likelihood as a criterion. When cost-function of logistic regression is applied on generalised model of AdaBoost, LogitBoost is derived. It can be seen as an optimisation process, that is convex optimisation, where convex function/sets are minimised. Logitboost minimises the logistic loss or log likelihood loss given as:

$$\sum_i \log\left(1 + e^{-y_i f(x_i)}\right) \tag{9}$$

A linear change occurs between the function and the classification error which minimises noise. Quasi – Newton step – is used iteratively for fitting an additive symmetric logistic model which makes a strong classifier by introducing iteratively new weak classifiers. The probability of any instance 'Q' falling in class 1 (among classes 0 and 1) is represented by $p(x)$, where

$$p(x) = \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}} \tag{10}$$

The algorithm for LogitBoost is as follows:

i Weights $w_i = 1/N$ $i = 1, 2, \ldots, N$, $F(x) = 0$ and probability estimates $p(x_i) = ½$.

ii Repeat for $m = 1, 2, \ldots, M$.

iii Compute the working response and weights,

$$z_i = \frac{Q_i - p(x_i)}{p(x_i)\left(1 - p(x_i)\right)} \tag{11}$$

$$w_i = p(x_i)\left(1 - p(x_i)\right) \tag{12}$$

iv Fit the function $f_m(x)$ by a weighted least square regression of z_i to x_i using weights w_i.

v Update $F(x) – F(x) + ½ f_m(x)$ and $p(x)$ on equation (9)

vi The classifier output is given by the decision function,

Q = sign $[F(x)]$ = {1 if $F(x) > 0$ and 0 if $F(x) < 0$}

4. *C4.5:* It is an extension of ID3 algorithm proposed by Quinlan (1993) which is used to generate a decision tree for classification. C4.5 is also known as statistical classifier. The C4.5 constructs decision tree from a set of training data set using information entropy concepts. If we have training data set $S=\{s_1, s_2, s_3, \ldots\}$ which are already classified samples, each sample s_i consists of a p-dimensional vector $(x_{1i}, x_{2i}, \ldots, x_{pi})$, where x_j stands for attributes of the sample and the class in which sample s_i falls. At every node of the tree, C4.5 chooses the attribute that most effectively divides its set of samples into subsets. The division condition is based on normalised information gain, i.e. difference in entropy. The attribute having highest normalised information gain is selected to take decision. The C4.5 algorithm then recursively works on the smaller sub-lists. C4.5 avoids over-fitting of data, determines how deeply a decision tree would grow, reduces error pruning, rules post-pruning, handles continuous attributes, choosing a suitable attribute selection measure, handles training data with missing values and improves computational efficiency.

5. *Logistic Model Trees:* This is the combined version of linear logistic regression and tree induction. The former produces low variance high bias and the later produces high variance low bias. These two techniques were combined into learner which depends upon simple regression models if only little and/or noisy data are present. It adds more complex tree structures if enough data are available to such structures. Thus, LMTs are the decision trees having linear regression model at leaves (Landwehr et al., 2003). A LMT consists of a set of non-terminal nodes N and a set of terminal nodes T. Let $S = D_1, \ldots, D_m$ be the whole instance space, spanned by all attributes $V = \{v_1,\ldots, v_m\}$ that are present in the data. Then the tree structure divides S into regions S_t and every region is represented by a leaf in the tree,

$$S = \bigcup_{t\in T} S_t, S_t \cap S_{t'} = \varphi \quad \text{for } t \neq t' \tag{13}$$

The leaves $t \in T$ have an associated logistic regression Function f_t which takes into account an arbitrary subset $Vt \subset V$ of all attributes present in the data, and models the class probabilities as:

$$\Pr(G = j \mid X = x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{Fk(x)}} \tag{14}$$

where

$$F_j(x) = \alpha_0^j + \sum_{v\in V_t} \alpha_v^j . v \tag{15}$$

When $\alpha^j_{vk} = 0$ for $vk \notin Vt$. The model represented by the whole LMT is then given by

$$f(x) = \sum_{t\in T} f_t(x).I\left(x \in S_t\right) \tag{16}$$

6. *Random Forest*: It is type of ensemble learning classification method. During training it constructs many decision trees. Mode class is extracted which is the mode of the classes output by individual trees (Breiman, 2001). Random vectors are generated which leads to the growth of individual trees in the ensemble. As the definition given by Breiman (2001), *'Random forest is a classifier consisting of a*

collection of tree-structured classifiers {h(x, θk), k = 1, …} where the {θk} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x'. For a forest of *M* trees, the prediction that the *m*th tree makes for ***X*** can be written as

$$T_m(X) = \sum_{i=1}^{n} W_{im}(X) Y_i \tag{17}$$

Where $W_{im} = 1/k_m$ if X and X_i are in the same leaf in the mth tree and 0 otherwise, and k_m is the number of training data which fall in the same leaf as X in the mth tree. The whole forest is predicted as

$$= \sum_{i=1}^{n} \left(\frac{1}{M} \sum_{m=1}^{M} W_{im}(X) \right) Y_i \tag{18}$$

A random forest prediction is a weighted average of the Y_i's, with weights

$$W_i(X) = \frac{1}{M} \sum_{m=1}^{M} W_{im}(X) \tag{19}$$

7. *Decision Table:* It is an easy way to model complicated logics. These are flowcharts based on if, then, else, switch cases statements and associate conditions with actions to perform. Each decision is related to a variable, relation, condition alternatives dependencies. Operations to be performed are actions which correspond to specific entry. Each entry specifies whether or in what order the action is to be performed for the given set of condition alternatives the entry corresponds to (Cragun and Steudel, 1987).

   The four basic elements of decision tables are conditions, actions, condition alternatives and action alternatives. These basic elements make the decision table quadrant.

Decision Table Quadrant

Condition (If/Else)	Condition alternatives
Action (Then)	Action alternatives

   The upper left quadrant has all the condition for a problem. Each condition is associated with if/else condition. The lower left quadrant has the set of action, which is to be taken once the condition is fulfilled. All the possible states of condition are present in the upper right quadrant and the lower right quadrant has all the possible action or rule alternatives. In the case of a two-class classification, the instances are associated with conditions which are generally a specific gene expression level, which finally classifies any instance into either of the classes.

8. *SMO-SVM:* The original SVM algorithm was coined by Vladimir N. Vapnik in 1979. In its simplest and linear form, an SVM is a hyperplane that separates a set of positive examples from a set of negative examples with largest distance to the nearest positive and negative examples (Platt, 1998). The non-linear form of SVM was proposed by Bernhard E. Boser et al. in 1992 that uses kernel trick to maximum-

margin hyperplane. The SVMs have been shown to provide good generalisation performance on a wide range of classification problems but still its application is limited because SVM training algorithms are slow for large data sets.

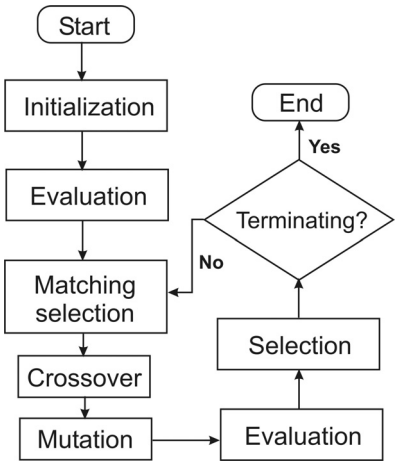
The SMO is a simple, fast and efficient SVM learning algorithm, designed by Platt at Microsoft Research (1998) having better scaling property for large problems. The original SVM learning algorithm uses numerical Quadratic Programming (QP) as inner loop, whereas SMO uses an analytic QP step. The SMO solves the optimisation problem of SVM by breaking it into several sub-problems, which are then solved analytically. The larger multiplier α_1 of the problem has linear equality constraint and therefore the smallest possible problem has two such multipliers. Then for any two multiplier (α_1 and α_2), the constraints are reduced to $0 \leq \alpha_1, \alpha_2 \leq C$, $y_1\ \alpha_1 + y_2\ \alpha_2$ and solved analytically. The comprehensive review on SMO-SVM can be found in Platt (1998).

9. *Neural network:* An ANN is a computational model inspired by the structural and functional facets of biological central nervous systems. It consists of simple artificial nodes, called neurons or processing elements, represented as systems of interconnected neurons that can compute values based on inputs by feeding information through the network. A typical ANN has input layer, one or more hidden layers and output layer. An ANN can be defined by three types of parameters: (i) the interconnection pattern between different layers of neurons, (ii) learning algorithm to update the synaptic weights of the interconnections and (iii) activation function that converts a neuron's weighted input to its output activation (Wikipedia, 2015). With each connect between neurons, synaptic weight is maintained that shows the strength of the connection which is activated during training and prediction. Backpropagation is most popular learning algorithm used to train ANN, which is a supervised learning that requires a data set of the desired output for many inputs. The errors are backpropagated and synaptic weights are updated by the learning algorithm. The capabilities of ANNs to learn from the data approximate any multivariate non-linear function, and its robustness to noisy data makes ANN a suitable candidate to solve classification problem using microarray data (Hopfield, 1982).

10. *Genetic algorithm:* Genetic Algorithms (GA) are basically search algorithms which are based on the mechanism of natural selection process and survival of the fittest, i.e. strong tend to adapt and survive, while the week tend to die out. The GA was proposed by John Holland in 1975. The GAs have the capability to generate an initial random population of possible solutions, and then recombine these solutions in a way to lead their search to only the most promising areas within the solution space. Every feasible solution is encoded in the form of chromosome (string), also known as a genotype. A fitness function is associated with each chromosome that decides its capability to survive and produce offspring. It uses probabilistic rules for evolving a population from one generation to another. The new solutions are generated by genetic recombination operators such as reproduction, crossover and mutation. The reproduction operator selects the fitness to reproduce, crossover combines parent chromosomes to produce children and passes superior genes to the next generation, and mutation alters few genes in a chromosome and confirms that the entire state-space is searched that lead the population out of local minima. Some of the key parameters of GAs are population size, evaluation function, crossover methods and mutation rate.

Selecting the size of population is non-trivial. Taking too small population size may have risk to converge to local minima prematurely, because population will not have sufficient genetic material to cover entire problem space. On the other hand, a larger population size has a better chance to find global optimum but at the cost of processing time. The size of the population remains unchanged throughout the generations (Mitchell, 1996). A schematic diagram of the working of GA is shown in Figure 2.

Figure 2 A general schema of genetic algorithm



4 Results and discussions

The experiment was carried on prostate cancer data set taken from Kent Ridge Biomedical Data Repository (http://datam.i2r.a-star.edu.sg/datasets/krbd/) as discussed in Section 3. We have used independent data set training and testing because of more reliability. Our work is unique from others because the same microarray data were not used for training as well as a percentage split of it for testing. The non-reduced or non-filtered data were full of noise and irrelevant data. The maximum and minimum values within the data prior to normalisation were 17,530 and –1807, respectively, therefore the range was 19,337. After normalisation maximum and minimum values were 18.067295 and –13.441490, respectively, and therefore the range reduced to 31.508785. Thus, the range after normalisation is far less prior to normalisation and shows that normalised data are less scattered. This satisfies the basic requirement for machine learning techniques. Table 1 shows a part of considered prostate cancer training set before normalisation, whereas Table 2 shows data set after normalisation.

The weka software tool (Hall et al., 2009) has been used for all the considered machine learning techniques except for neural network and genetic algorithm. Neuro Solutions 5.0 software tool (NeuroDimension, n.d.) has been applied for neural network and genetic algorithm. Following are the description of the results of various techniques used for training and testing. Table 3 shows a brief performance comparison of different techniques used with different statistical measures. It is clearly depicted that Bayes Network outperforms as compared to other techniques. Figure 3 shows the comparison of

different techniques used for their accuracy in classifying samples correctly and incorrectly, whereas Figure 4 shows accuracy level of different techniques used. Bayes net was found to be the best technique for classifying cancer class. Out of 34 samples Bayes net classified 32 samples correctly. Bayes Net outperforms the other techniques with an accuracy level of 94.11% followed by Naive Bayes with an accuracy level of 91.17%. Instance Based and LMT have same accuracy level of 85.29%, Random Forest and Genetic Algorithm have the same accuracy level of 82.35%, and Decision Tree and SMO-SVM have same accuracy level of 79.41%. C4.5 has the lowest accuracy level of 70.58%.

Table 1 Training data set before normalisation

	Gene 1	*Gene 2*	*Gene 3*	*Gene 4*	*Gene 5*	*Gene 6*
Sample 1	–9	1	1	15	–2	–3
Sample 2	–2	1	1	4	–2	–5
Sample 3	–6	17	6	29	4	–11
Sample 4	0	9	4	19	–10	–18
Sample 5	–1	0	1	5	0	–4
Sample 6	0	17	1	20	–20	–18
Sample 7	–5	5	–1	9	–10	–17
Sample 8	–3	1	1	5	–2	–6
Sample 9	–8	–2	–1	–32	–20	–41
Sample 10	–12	11	–3	21	–10	–9

Note: Table 1 shows the first six genes in columns and their corresponding gene expression values in first ten samples. The values of genes are largely scattered throughout the matrix. This expression matrix is raw, non-reduced and non-filtered.

Table 2 Training data set after normalisation

	Gene 1	*Gene 2*	*Gene 3*	*Gene 4*	*Gene 5*	*Gene 6*
Sample 1	–0.14	0.30	0.89	–1.11	0.19	0.56
Sample 2	–0.14	0.39	0.66	–1.79	–0.41	0.31
Sample 3	0.05	–0.36	–1.13	–0.59	–0.56	–0.99
Sample 4	–0.43	–1.04	–0.65	–0.20	1.54	–0.92
Sample 5	–0.34	0.04	–0.03	0.07	–0.41	–0.03
Sample 6	0.53	0.62	–0.99	–0.38	–0.56	–0.60
Sample 7	0.34	–0.62	–0.08	–0.64	0.12	–0.77
Sample 8	–0.92	0.87	–0.37	–1.20	–0.71	0.32
Sample 9	1.98	–1.51	–1.37	2.25	1.69	0.12
Sample 10	0.24	–0.97	–1.05	–0.51	0.19	0.06

Note: Table 2 shows the first six genes in columns after and their corresponding gene expression values in first ten samples after normalisation. Difference between the highest and the lowest values of genes is far less as compared to non-normalised values.

Figure 3 Correctly Classified samples (CCS) versus Incorrectly Classified Samples (ICS). During testing, total number of samples were 34

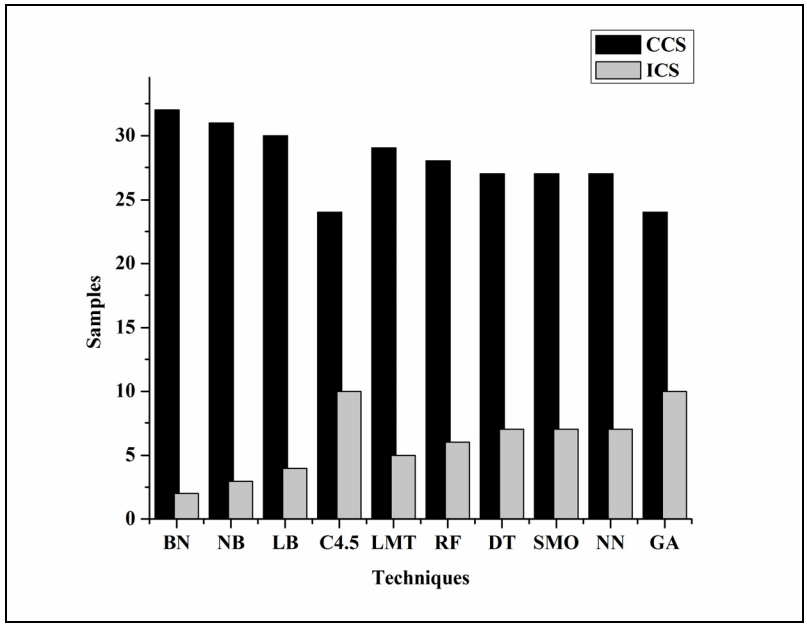
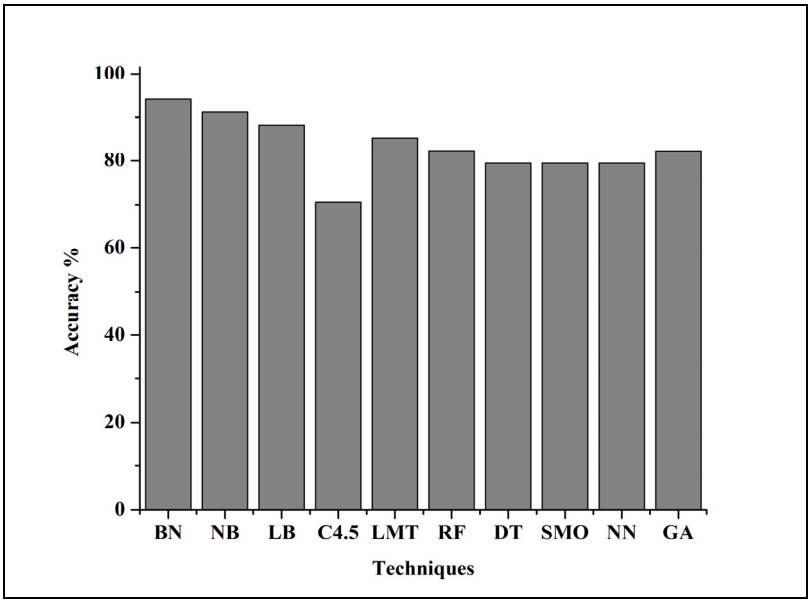


Figure 4 Accuracy level of various techniques on test data set



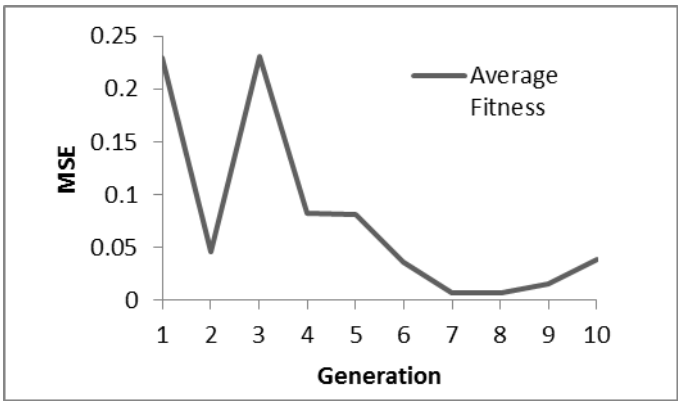
Note: BN (Bayes Net), NB (Naive Bayes), LB (Logit Boost), LMT (Logistic Model Tree), RF (Random Forest), DT (Decision Table), SMO-SVM (Sequential Minimal Optimization-Support Vector Machine), **NN** (Neural Network), **GA** (Genetic Algorithm).

4.1 Training and testing

The whole experiment is based on training of different techniques with training data first and then testing on independent testing data. During training the best accuracy was observed as 100% for LogitBoost, LMT and SMO-SVM techniques. All these techniques were successful in correctly classifying all the 102 training samples during training. Techniques such as C4.5 and Random Forest achieved an accuracy of 99.01% and were successful in classifying 101 samples correctly. Decision Table, Bayes Net and Naive Bayes achieved an accuracy of 97.05%, 91.17% and 87.25%, respectively. Neural network showed the minimum root mean squared error (RMSE) of 0.1277. This RMSE was the final RMSE for the training experiment. Whereas Genetic Algorithm while training showed the minimum Mean Squared Error (MSE) for Best Fitness and Average Fitness as 2.91173E-05 and 0.006661297, respectively, the final MSE for Best Fitness and Average Fitness was 2.91173E-05 and 0.038575474, respectively.

The outcome after testing these techniques on testing data is summarised in Table 3. It is clear from the outcomes that Bayes Net outperforms all the techniques with a classifying accuracy of 94.11%. Figure 5 shows the average fitness versus generation graph during testing in case of GA. The best fitness was for the seventh and eighth generation with MSE less than 0.05.

Figure 5 Average fitness versus generation graph. Among the ten generation the best fitness was for the fourth generation with mean square error (MSE) less than 0.05. The average MSE is 0.164830



4.2 Cross validation by modifying test data sets

As the test data set was taken from independent experiments and was having an overall tenfold difference from the training data set, we have modified the test data set in six different ways to cross validate our results. The test data set was divided by 2 (Div 2), 10 (Div 10), 20 (Div 20) and multiplied by 2 (Mul 2), 10 (Mul 10), 20 (Mul 20), making six different data sets. Table 4 shows the accuracy level of different techniques for all the six modified test data as well as the average accuracy level. It is clear from the table that the average accuracy level of Bayes Net is more than other techniques.

Table 3 Performance comparison of techniques on test data

Techniques	*CCS (%)*	*ICS (%)*	*RMSE*	*TPR (tumour)*	*TPR (normal)*	*FPR (tumour)*	*FPR (normal)*
Bayes Net	32 (94.11)	2 (8.82)	0.2189	0.92	1	0	0.8
Naive Bayes	31 (91.17)	3 (8.82)	0.297	0.88	1	0	0.12
LogitBoost	30 (88.23)	4 (11.76)	0.3736	0.84	1	0	0.16
C4.5	24 (70.58)	10 (29.41)	0.4918	0.76	0.55	0.44	0.24
Logistic Model Tree	29 (85.29)	5 (14.70)	0.3429	0.8	1	0	0.2
Random Forest	28 (82.35)	6 (17.64)	0.3523	0.76	1	0	0.24
Decision Table	27 (79.41)	7 (20.58)	0.3523	0.76	0.889	0.111	0.24
SMO-SVM	27 (79.41)	7 (20.58)	0.4537	0.72	1	0	0.28
Neural Network	27 (79.41)	7 (20.58)	0.253693	0.88	1	0	0.12
Genetic Algorithm	24 (76.44)	10 (29.41)	0.3747	0.94	0.47	0.52	0.058

Note: CCS (correctly classified samples), ICS (incorrectly classified samples), RMSE (root mean squared error), TPR (true positive rate), FPR (false positive rate). Bayes Net gives the most accurate prediction of prostate cancer class with an accuracy of 94.11%. Total number of sample was 34 out of which it predicted 32 samples correctly.

4.3 Sensitivity and specificity analysis

Sensitivity is one of the statistical methods for measuring the performance of binary classification (Provost and Fawcett, 1997). It measures the True Positive rate. True Positives (TP) are the positives which are correctly identified as positive. A high sensitivity corresponds to higher accuracy. Whereas, specificity describes the ability (of any technique used) to identify negatives as negative or true negatives (TN) as true negative. Thus, a high specificity indicates that any technique used has a high ability to identify true negatives. General formula for sensitivity and specificity are

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{20}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{21}$$

Table 4 shows sensitivity and specificity of all the techniques used for all the six modified data. Sensitivity and specificity of all the techniques were calculated for all the six modified data. Then the actual (mean) sensitivity and specificity of a technique were calculated by averaging the sensitivity and specificity obtained for individual modified data sets. Table 5 shows the actual (mean) sensitivity and specificity of all the techniques. The outcome shows Bayes Net has the sensitivity and specificity of 0.89 and 1, respectively, which is higher than the others.

4.4 Precision and accuracy analysis

Precision is also known as reproducibility or repeatability. It shows how a measurement under repeating condition remains unchanged. Precision is the degree of measurement of

true positive against true positive and false positive, whereas accuracy is the degree of closeness of obtained value to the actual value. A high accuracy and high precision signify that testing process is working well with a valid theory. General formula for precision and accuracy is

$$\text{Precision} = \frac{TP}{TP + FP} \tag{22}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{23}$$

Precision and accuracy of all the techniques were calculated for all the six modified data set. Then the actual precision and accuracy of a technique were calculated by taking the mean of precisions and accuracies obtained for all the modified data sets. Table 6 shows the actual (mean) precision and accuracy of all the techniques for modified data sets. It shows clearly that Bayes Net has the highest precision and accuracy of 1.0 and 0.92, respectively.

Table 4 Sensitivity and specificity

Techniques	*Sn/Sp*	*Div 2*	*Mul 2*	*Div 10*	*Mul 10*	*Div 20*	*Mul 20*
BN	Sn	0.96	0.92	0.88	0.8	1	0.8
	Sp	1	1	1	1	1	1
NB	Sn	1	0.52	1	0.2	1	0.12
	Sp	0.88	1	0	1	0	0.88
LB	Sn	0.92	0.76	0.92	0.68	1	0.68
	Sp	1	1	0.44	1	0.11	1
C4.5	Sn	0.64	0.76	0.68	0.84	0.68	0.84
	Sp	1	0.55	0.66	0.55	0.66	1
LMT	Sn	0.84	0.76	1	0.64	1	0.64
	Sp	1	1	0	1	0	1
RF	Sn	0.92	0.72	0.92	0.48	1	0.44
	Sp	1	1	0.66	0.77	0.44	0.77
DT	Sn	0.76	0.72	0.68	0.72	0.68	0.72
	Sp	0.88	0.88	0.66	0.88	0.55	0.88
SMO	Sn	0.8	0.64	1	0.64	1	0.64
	Sp	1	1	0	1	0	1
NN	Sn	0.8	0.76	1	0.72	0	0.84
	Sp	1	1	0	1	1	1
GA	Sn	0.68	0.72	0.68	0.76	1	0.72
	Sp	0.88	1	1	0	0.88	1

Note: Sn = sensitivity and Sp = specificity. Table 4 shows sensitivity and specificity of different techniques for all the six modified data sets.

Table 5 Mean sensitivity and specificity comparison

Techniques	*BN*	*NB*	*LB*	*C4.5*	*LMT*	*RF*	*DT*	*SMO*	*NN*	*GA*
Sensitivity	0.89	0.64	0.82	0.74	0.81	0.74	0.71	0.78	0.68	0.76
Specificity	1	0.62	0.75	0.73	0.66	0.77	0.78	0.66	0.83	0.79

Note: Table 5 shows the mean sensitivity and means specificity of all the techniques. The means were calculated by averaging the sensitivity and specificity of all the techniques for all the six modified data sets. Bayes Net has the highest sensitivity and specificity of 0.89 and 1, respectively.

Table 6 Precision and accuracy comparison

Techniques	*BN*	*NB*	*LB*	*C4.5*	*LMT*	*RF*	*DT*	*SMO*	*NN*	*GA*
Precision	1	0.86	0.92	0.77	0.71	0.71	0.71	0.76	0.7	0.7
Accuracy	0.92	0.63	0.8	0.74	0.77	0.75	0.73	0.75	0.7	0.7

Note: Table 6 shows the comparison of precision and accuracy. Bayes Net has the highest precision and accuracy.

In the following section, Table 7 shows a comparison of classification accuracy of our work with the works of other researchers on the same kind of data set (prostate cancer). Our Bayesian network and Naive Bayes-based techniques show the highest accuracy over the others, i.e. an accuracy of 94.11 and 92.17, respectively. The *k*NN-based method of Singh et al (2002) shows the next highest accuracy between 86 and 92.

Table 7 Comparison of accuracy with the work of others

Author(s)	*Techniques*	*Accuracy (%)*
Our method	Bayesian network	94.11
	Naive Bayes	92.17
Zupan et al. (2000)	Naive Bayes	70.8–78.4
	Decision Tree	68.8–77
	Cox	69.7–79
Wagner et al. (2004)	kNN	87.4–89.9
	Fisher Linear	87.9–89.1
	Linear SVM	89.5–91.9
Tan and Gilbert (2003)	Single C4.5	52.38
	Bagging C4.5	85.71
	AdaBoost	76.19
Singh et al. (2002)	kNN	86–92

Note: Table 7 shows the comparison of classification accuracy with the works of others on prostate cancer data set. Our Bayesian network and Naive Bayes based classification method outperforms over the others.

5 Conclusions and future challenges

In this paper, we have comparatively evaluated various machine learning techniques for their accuracy in class prediction of prostate cancer data set. As per our evaluation, Bayes Net gave the best accuracy for prostate cancer class prediction with an accuracy of 94.11% which is higher than any previously published work on the same data set. Bayes Net is followed by Naive Bayes with an accuracy of 91.17%. We tested our data set on different techniques and selected those techniques which gave best results. Our aim was to identify the best technique in terms of accuracy which can classify prostate cancer date set and to reveal a good procedure for meaningful attribute reduction, which we have acquired by using a combination of *t*-test and IQR. Similar process can be applied and checked for their accuracy in classification of other types of cancers. One of the biggest challenges is to develop a single classifier which is best suitable for classifying all types of cancer gene expression data into meaningful number of classes. Nature-inspired optimisation techniques such as Ant Colony Optimisation (ACO), Artificial Bee Colony (ABC) optimisation, Particle Swarm Optimisation (PSO) are successfully being used in many challenging problems. In the future work, we will try to hybridise these nature-inspired optimisation techniques with different classifiers for better classification accuracy.

Acknowledgements

The authors would like to thank the anonymous reviewers for providing fruitful suggestions and comments. Authors also acknowledge the entire scientist behind Kent Ridge Bio-medical Data Set Repository for making the data sets publicly available. Author K. Raza acknowledges the funding from University Grants Commission, Government of India, through research grant 42-1019/2013(SR).

References

Aihara, M., Wheeler, T.M., Ohori, M. and Scardino, P.T. (1994, January) 'Heterogeneity of prostate cancer in radical prostatectomy specimens', *Urology*, Vol. 43, No. 1, pp.60–66.

Boulesteix, A-L., Tutz, G. and Strimmer, K. (2003) 'A CART-based approach to discover emerging patterns in microarray data', *Bioinformatics*, Vol. 19, No. 18, pp.2465–2472.

Breiman, L. (2001) 'Random forests', *Machine Learning*, Vol. 45, No. 1, pp.5–32.

Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares, M. and Haussler, D. (2000) 'Knowledge-based analysis of microarray gene expression data by using support vector machines', *Proceedings of the National Academy of Sciences*, Vol. 97, No. 1, pp.262–267.

Cragun, B.J. and Steudel, H.J. (1987, May) 'A decision-table-based processor for checking completeness and consistency in rule-based expert systems', *International Journal of Man–Machine Studies*, Vol. 26, No. 5, pp.633–648.

Dinasarapu, A.R., Gupta, S., Maurya, M.R., Fahy, E., Min, J., Sud, M., Gersten, M.J., Glass, C.K. and Subramaniam, S. (2013) 'A combined omics study on activated macrophages – enhanced role of STATs in apoptosis, immunity and lipid metabolism', *Bioinformatics*, Vol. 29, No. 21, pp.2735–2743.

Friedman, N., Geiger, D. and Goldszmidt, M. (1997) 'Bayesian Network classifiers*', *Machine Learning*, Vol. 29, pp.131–163.

Friedman, J., Hastie, T. and Tibshirani, R. (2000a) 'Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)', *The Annals of Statistics*, Vol. 28, No. 2, pp.337–407.

Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000b, January) 'Using Bayesian networks to analyze expression data', *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology*, Vol. 7, Nos. 3–4, pp.601–620.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. and Lander, E.S. (1999, October) 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *Science (New York, N.Y.)*, Vol. 286, No. 5439, pp.531–537.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002) 'Gene selection for cancer classification using support sector machines', *Machine Learning*, Vol. 46, Nos. 1–3, pp.389–422.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009) 'The WEKA data mining software: an update', *SIGKDD Explorations*, Vol. 11, No. 1, pp10–18.

Hopfield, J.J. (1982) 'Neural networks and physical systems with emergent collective computational abilities', *Proceedings of the National Academy of Sciences*, Vol. 79, No. 8, pp.2554–2558.

Hsu, A.L., Tang, S-L. and Halgamuge, S.K. (2003) 'An unsupervised hierarchical dynamic self-organizing approach to cancer class discovery and marker gene identification in microarray data', *Bioinformatics*, Vol. 19, No. 16, pp.2131–2140.

Jirapech-Umpai, T. and Aitken, S. (2005, January) 'Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes', *BMC Bioinformatics*, Vol. 6, p.148.

Khan, J., Wei, J.S., Ringnér, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C. and Meltzer, P.S. (2001, June) 'Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks', *Nature Medicine*, Vol. 7, No. 6, pp.673–679.

Landwehr, N., Hall, M. and Frank, E. (2003) 'Logistic model trees', in Lavrac, N., Gamberger, D., Todorovski, L. and Blockeel, H. (Eds): *Machine Learning: ECML 2003*, Springer, Berlin, pp.241–252.

Li, J. and Wong, L. (2002) 'Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns', *Bioinformatics*, Vol. 18, No. 5, pp.725–734.

Li, T., Zhang, C. and Ogihara, M. (2004) 'A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression', *Bioinformatics*, Vol. 20, No. 15, pp.2429–2437.

Liu, J.J., Cutler, G., Li, W., Pan, Z., Peng, S., Hoey, T., Chen, L. and Ling, X.B. (2005) 'Multiclass cancer classification and biomarker discovery using GA-based algorithms', *Bioinformatics*, Vol. 21, No. 11, pp.2691–2697.

Lu, Y. and Han, J. (2003, June) 'Cancer classification using gene expression data', *Information Systems*, Vol. 28, No. 4, pp.243–268.

Mitchell, M. (1996) *An Introduction to Genetic Algorithms, 1996*, PHI Pvt. Ltd., New Delhi.

Moore, J.H., Parker, J.S. and Hahn, L.W. (2001) 'Symbolic discriminant analysis for mining gene expression patterns', *Machine Learning: ECML 2001*, Vol. 2167, pp.372–381.

NeuroDimension (n.d.) *Neuro Solutions 5.0*. Available online at: http://www.neurosolutions.com/

O'Neill, M. and Song, L. (2003) 'Neural network analysis of lymphoma microarray data: prognosis and diagnosis near-perfect', *BMC Bioinformatics*, Vol. 4, No. 1, p.13.

Platt, J. (1998) 'Sequential minimal optimization: a fast algorithm for training support vector machines', in Scholkopf, B., Burges, C. and Smola, A. (Eds): *Advances in Kernel Methods – Support Vector Learning*, MIT Press, Cambridge, MA.

Provost, F. and Fawcett, T. (1997) 'Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions', *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Newport Beach, CA, pp.43–48.

Quackenbush, J. (2002, December) 'Microarray data normalization and transformation', *Nature Genetics*, Vol. 32, pp.496–501.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Francisco, CA.

Raza, K. and Jaiswal, R. (2013, June) 'Reconstruction and analysis of cancer-specific gene regulatory networks from gene expression profiles', *International Journal on Bioinformatics & Biosciences*, Vol. 3, No. 2, pp.25–34 [Preprint arXiv:1305.5750, 2013].

Raza, K. and Mishra, A. (2012) 'A novel anticlustering filtering algorithm for the prediction of genes as a drug target', *American Journal of Biomedical Engineering*, Vol. 2, No. 5, pp.206–211.

Raza, K. and Parveen, R. (2013) 'Reconstruction of gene regulatory network of colon cancer using information theoretic approach', *4th International Conference (CONFLUENCE-2013): The Next Generation Information Technology Summit*, 26–27 September, pp.461–466.

Rish, I. (2001) 'An empirical study of the Naive Bayes classifier', *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, Vol. 3, No. 22, pp.41–46.

Rosskopf, M., Schmidt, H.A., Feldkamp, U. and Banzhaf, W. (2007) *Genetic Programming Based DNA Microarray Analysis for Classification of Cancer*, Memorial University of Newfoundland, Newfoundland.

Saeys, Y., Inza, I. and Larrañaga, P. (2007, October) 'A review of feature selection techniques in bioinformatics', *Bioinformatics (Oxford, England)*, Vol. 23, No. 19, pp.2507–2517.

Singh, D., Febbo, P.G., Ross, K., Jackson, D.G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A.A., D'Amico, A.V., Richie, J.P., Lander, E.S., Loda, M., Kantoff, P.W., Golub, T.R. and Sellers, W.R. (2002, March) 'Gene expression correlates of clinical prostate cancer behavior', *Cancer Cell*, Vol. 1, No. 2, pp.203–209.

Stratton, M.R., Campbell, P.J. and Futreal, P.A. (2009, April) 'The cancer genome', *Nature*, Vol. 458, No. 7239, pp.719–724.

Sturn, A., Quackenbush, J. and Trajanoski, Z. (2002) 'Genesis: cluster analysis of microarray data', *Bioinformatics*, Vol. 18, No. 1, pp.207–208.

Tan, A.C. and Gilbert, D. (2003) 'Ensemble machine learning on gene expression data for cancer classification', *Applied Bioinformatics*, Vol. 2, pp.75–83.

Tang, Y., Zhang, Y-Q. and Huang, Z. (2007) 'Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 4, No. 3, pp.365–381.

Vanneschi, L., Farinaccio, A., Mauri, G., Antoniotti, M., Provero, P. and Giacobini, M. (2011, January) 'A comparison of machine learning techniques for survival prediction in breast cancer', *BioData Mining*, Vol. 4, No. 1, p.12.

Varambally, S., Yu, J., Laxman, B., Rhodes, D.R., Mehra, R., Tomlins, S.A., Shah, R.B., Chandran, U., Monzon, F.A., Becich, M.J., Wei, J.T., Pienta, K.J., Ghosh, D., Rubin, M.A. and Chinnaiyan, A.M. (2005, November) 'Integrative genomic and proteomic analysis of prostate cancer reveals signatures of metastatic progression', *Cancer Cell*, Vol. 8, No. 5, pp.393–406.

Wagner, M., Naik, D.N., Pothen, A., Kasukurti, S., Devineni, R.R., Adam, B., Semmes, O.J. and Wright, G. L., Jr. (2004) 'Computational protein biomarker prediction : a case study for prostate cancer', *BMC Bioinformatics*, Vol. 9, pp.1–9.

Wang, X. and Gotoh, O. (2009, January) 'Microarray-based cancer prediction using soft computing approach', *Cancer Informatics*, Vol. 7, pp.123–139.

Wang, Y., Tetko, I.V., Hall, M.A., Frank, E., Facius, A., Mayer, K.F.X. and Mewes, H.W. (2005, February) 'Gene selection from microarray data for cancer classification – a machine learning approach', *Computational Biology and Chemistry*, Vol. 29, No. 1, pp.37–46.

Wikipedia (2015) *Artificial Neural Networks*. Available online at: https://en.wikipedia.org/wiki/Artificial_neural_network (accessed on 3 March).

Yeung, K.Y., Bumgarner, R.E. and Raftery, A.E. (2005) 'Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data', *Bioinformatics*, Vol. 21, No. 10, pp.2394–2402.

Zhang, H. (2004) 'The optimality of naive Bayes', *Proceedings of the 17th Florida Artificial Intelligence Research Society Conference*, The AAAI Press, pp.562–567.

Zou, M. and Conzen, S.D. (2005) 'A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data', *Bioinformatics*, Vol. 21, No. 1, pp.71–79.

Zupan, B., Demsar, J., Kattan, M.W., Beck, J.R. and Bratko, I. (2000) 'Machine learning for survival analysis: a case study on recurrence of prostate cancer', *Artificial Intelligence in Medicine*, Vol. 20, No. 1, pp.59–75.