A Novel Gene Selection Technique Using Wrapper Approach on Microarray Gene Expression Data

Project Report submitted in partial fulfillment of
The requirements for the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Of

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

By

ANKAN BANERJEE, 10, 10900114010
MAHASHETA KUNDU, 43, 10900114043
MD NASIRUL HAQUE, 46, 10900114046

Under the guidance of

Mr. Shilpi Bose

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NETAJI SUBHASH ENGINEERING COLLEGE
TECHNO CITY, GARIA, KOLKATA – 700 152

2017-18

CERTIFICATE

This is to certify that this project report titled "**A Novel Gene Selection Technique Using Wrapper Approach on Microarray Gene Expression Data**" submitted in partial fulfillment of requirements for award of the degree Bachelor of Technology (B. Tech) in Computer Science & Engineering of Maulana Abul Kalam Azad University of Technology is a faithful record of the original work carried out by,

ANKAN BANERJEE, **Roll no**. 10900114010, **Regd. No.** 141090110010 of 2014 - 2015

MAHASHETA KUNDU, **Roll no**. 10900114043, **Regd. No.** 141090110043 of 2014 - 2015

MD NASIRUL HAQUE, **Roll no.** 10900114046, **Regd. No.** 141090110046 of 2014 - 2015

under my guidance and supervision.

It is further certified that it contains no material, which to a substantial extent has been submitted for the award of any degree/diploma in any institute or has been published in any form, except the assistances drawn from other sources, for which due acknowledgement has been made.

Date: ……...

Dr. Atanu Das Head of the Department Department of CSE Netaji Subhash Engineering College Technocity, Garia, Kolkata - 700152	Mr. Sourav Dutta Project Coordinator Department of CSE Netaji Subhash Engineering College Technocity, Garia, Kolkata - 700152	Mr. Shilpi Bose Assistant Professor Department of CSE Netaji Subhash Engineering College Technocity, Garia, Kolkata – 700152

DECLARATION

We hereby declare that this project report titled

A Novel Gene Selection Technique Using Unsupervised Wrapper Approach on Microarray Gene Expression Data

is our own original work carried out as a under graduate student in Netaji Subhash Engineering College except to the extent that assistances from other sources are duly acknowledged.

All sources used for this project report have been fully and properly cited. It contains no material which to a substantial extent has been submitted for the award of any degree/diploma in any institute or has been published in any form, except where due acknowledgement is made.

Student's Name:	Signature:	Dates:
……..	……..	……
……..	……..	……
……..	……..	……..…

CERTIFICATE OF APPROVAL

We hereby approve this dissertation titled

A Novel Gene Selection Technique Using Unsupervised Wrapper Approach on Microarray Gene Expression Data

carried out by:

ANKAN BANERJEE, **Roll no**. 10900114010, **Regd. No.** 141090110010 of 2014-2015

MAHASHETA KUNDU, **Roll no**. 10900114043, **Regd. No.** 141090110043 of 2014-2015

MD NASIRUL HAQUE, **Roll no**. 10900114046, **Regd. No.** 141090110046 of 2014-2015

Under the guidance of

Mr. Shilpi Bose

of Netaji Subhash Engineering College, Kolkata in partial fulfillment of requirements for award of the degree Bachelor of Technology (B. Tech) in Computer Science & Engineering of Maulana Abul Kalam Azad University of Technology.

Date: ……..……..

Examiner's Signature:

1. …….

2. …….

ACKNOWLEDGEMENT

We would take this opportunity to express my profound gratitude and deep regards to my guide and co-guide for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The help and guidance given by them time to time have been a constant source of inspiration. We would also like to thank all our lecturers who have directly or indirectly helped us for our project. We would like to our respect and love to our friends and family for their cooperation and support throughout the project.

Dated: ……..

…..……
ANKAN BANERJEE

...……..
MAHASHETA KUNDU

...……..……..……..
MD NASIRUL HAQUE

Abstract

As of late, feature selection and dimensionality reduction have turned out to be major way out for data extraction and processing, particularly to process high-dimensional information, for example, gene expression data. Microarray Gene Expression data consists of thousands of features in a relatively small sample space. Moreover, these datasets consist of redundant and non-relevant information which affects the learning algorithms used in sample classification. The large dimension increases the complexity of these learning algorithms too. So, the test to diminish the information dimensionality emerges. Countless number of dimensional reduction and gene selection algorithms are applied to choose a subset of pertinent features to better demonstrate development and to look for better malignancy classification performance. This project work seeks to introduce a new unsupervised wrapper approach for selection of optimal subset of genes from the microarray gene expression data. In this research work, the clustering of the data has been performed and the representatives of the clusters are then chosen using a novel manner to apply the wrapper technique for finding the final optimal feature set. Multiple classification algorithms like SVM, KNN and NBC have been employed in the wrapper for five different microarray datasets to show the effectiveness of our proposed method. The efficiency of our method is demonstrated by comparing it with some of the well-known existing techniques.

Table of Contents:

1. INTRODUCTION

The rise of DNA microarray innovation has empowered synchronous measurements of excitation levels of thousands of genes. However, because of the high cost of experimentation to measure the said excitation levels, the sample size of these gene expressions are of a relatively small size, usually in hundreds, as compared to the thousands of genes present in a human DNA. As the ratio of sample to features(genes) are very unbalanced, selection of relevant genes for computation and information extraction becomes tough. Also, with the imbalance in input dimensionality, it becomes increasingly difficult to get accurate results with the present irrelevant data within the useful information. In this way, choice of applicable and relevant genes remains a challenge in gene expression data analysis. In our paper, we address the problem of selecting relevant genes from a microarray gene expression data and aim to reduce the dimensionality of the data in an unsupervised manner without losing accuracy of its information conveyed.

The features or excitation levels measured for every sample of the DNA can be treated as relevant, non-relevant or redundant features. The relevancy of a gene is typically measured by the class level of the sample which indicates its importance in the data. However, when approaching the problem in an unsupervised manner, annotations or class labels are not mentioned with the data that is provided. The algorithm needs to separate the relevant features from the redundant without any reference source point to guide it which makes the already difficult task much harder. Unsupervised feature selection, where there is no information about the actual functionality or relevancy of a class calculates the relevancy of a feature through the data structures, such as the distribution of data, the correlation of data with each other, data variance, etc.

The approach that has been incorporated in the paper is an unsupervised method of reducing the dimensionality of a matrix of gene expression data. The given data is first segregated into clusters. The measure for data independence or dependence has been calculated by a correlation measure instead of the commonly used Euclidean distance. Upon forming the required number of clusters, the data points closest to the mean of each cluster has selected into the new gene microarray as the relevant feature representatives from a pool of dozens of features in a cluster. Using these representatives, a wrapper method is implemented which checks for the effect on the accuracy of data due to each representative feature. The accuracy of the data is measured using multiple classification procedures such as SVM, NBC, etc (discussed later in the paper) and the relative division of the given dataset into a smaller dataset which shows maximum accuracy on various classifiers will be selected for the final result.

2. EXISTING WORK

Discussed below are certain known existing methodologies for feature selection and dimension reduction that have been used and researched upon. In our work, we have drawn inspiration from these and designed a framework which shall be discussed in the next section.

2.1 Outline of Feature Selection

Feature selection expects to choose a component subset from the existing set of components based on component's relevance and redundancy. Yu and Liu [1] group the component subsets into four classes: (a) completely irrelevant and noisy features, (b) weakly relevant and redundant features, (c) weakly relevant and non-redundant features, and (d) strongly relevant features. An ideal subset primarily contains all the features in the group (c) and (d). Strongly relevant features are essential for improvement of discriminative power and prediction accuracy. Occasionally, weakly relevant features can be beneficial in enhancing prediction accuracy if the feature is non-redundant and well matched with evaluation measures. An irrelevant feature implies that the feature does not have any contribution to prediction accuracy. As a result, in order to construct a decent model prediction, preferably all strongly relevant features and some weakly relevant features should be chosen, and insignificant, redundant or noisy features should be excluded. The primary reason behind excluding redundant features because they may have significant statistical relations with other features, not on the account of they contain useless data.

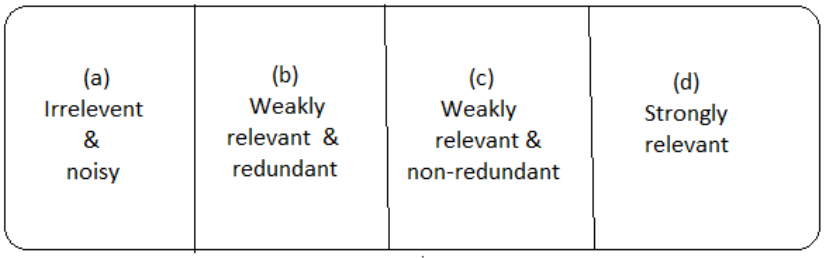


Fig. 1. Features classification based on relevancy and redundancy.

Feature selection gives a considerable measure benefits as it enhances prediction accuracy of a model if the right subset is chosen, understandability, adaptability, and speculation capacity of the classifier. It additionally diminishes the complexity of a model and makes it easier to interpret, gives faster and more practical demonstrate [2], and enhanced generalization by reducing overfitting. There are three kinds of feature selection strategy: supervised, unsupervised and semi-supervised.

2.2 Supervised Feature Selection

Supervised feature selection is mainly used for classification purpose. Supervised feature selection is the most familiar rehearse approach [3]. The accessibility of the class labels permits supervised feature selection algorithm to successfully choose discriminative features to differentiate samples from various classes. Features are first produced from training data. Rather than utilizing each and every data to train the supervised learning model, supervised feature selection will initially choose a subset features and after that process the data with the chosen features to the learning model. The feature selection stage will utilize the label data and the properties of the data, for example, information gain or Gini index, to choose relevant features. The last chosen features, and with the label data, are utilized to train a classifier, which can be utilized for prediction [4].

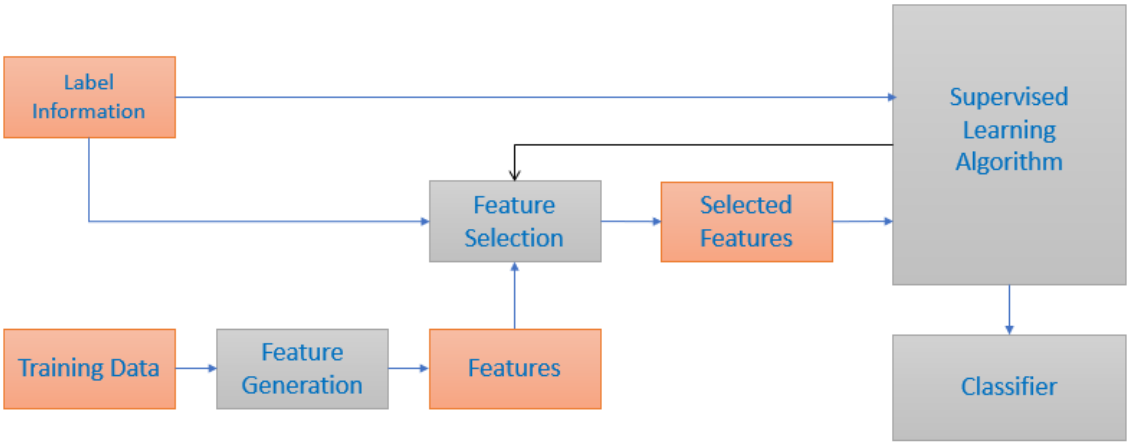


Fig. 2. General Framework of Supervised Feature Selection.

Filter: It is used to evaluate how much the features are relative to the problem by checking properties of the data. In most cases a feature relevance score is calculated, and the low scoring features are removed. Afterwards, high ranked features are presented as input to the classification algorithm.

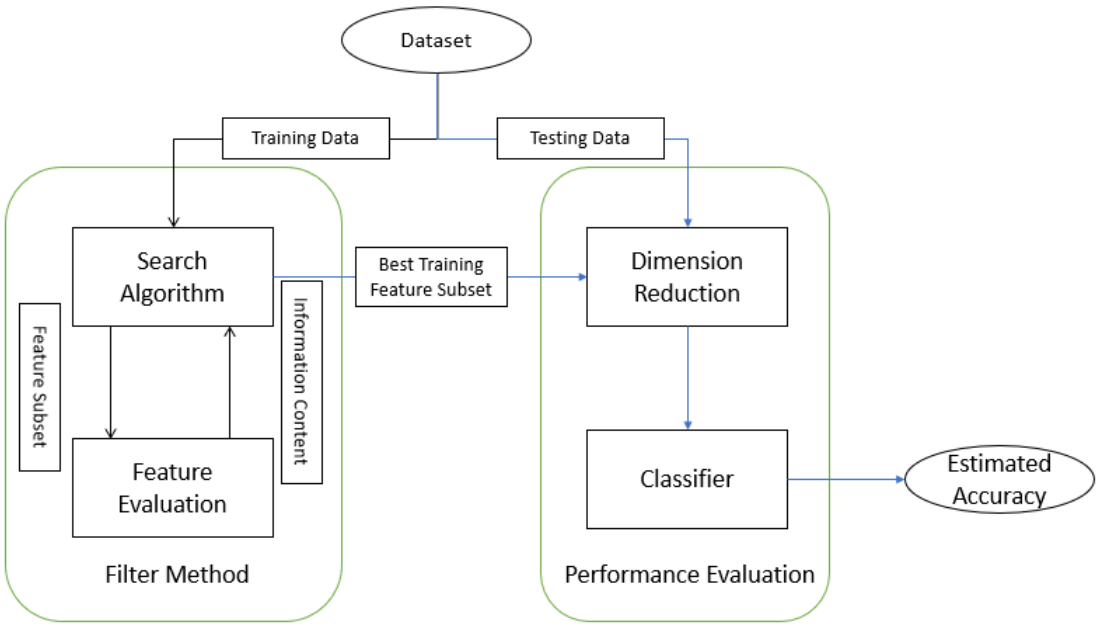


Fig. 3. General Framework of Filter Method

Wrapper: This technique evaluates subsets of features according to how informative they are to a given predictor. The method conducts a search for a good subset using the learning algorithm itself as a part of the evaluation function. The problem boils down to a problem of stochastic state space search.

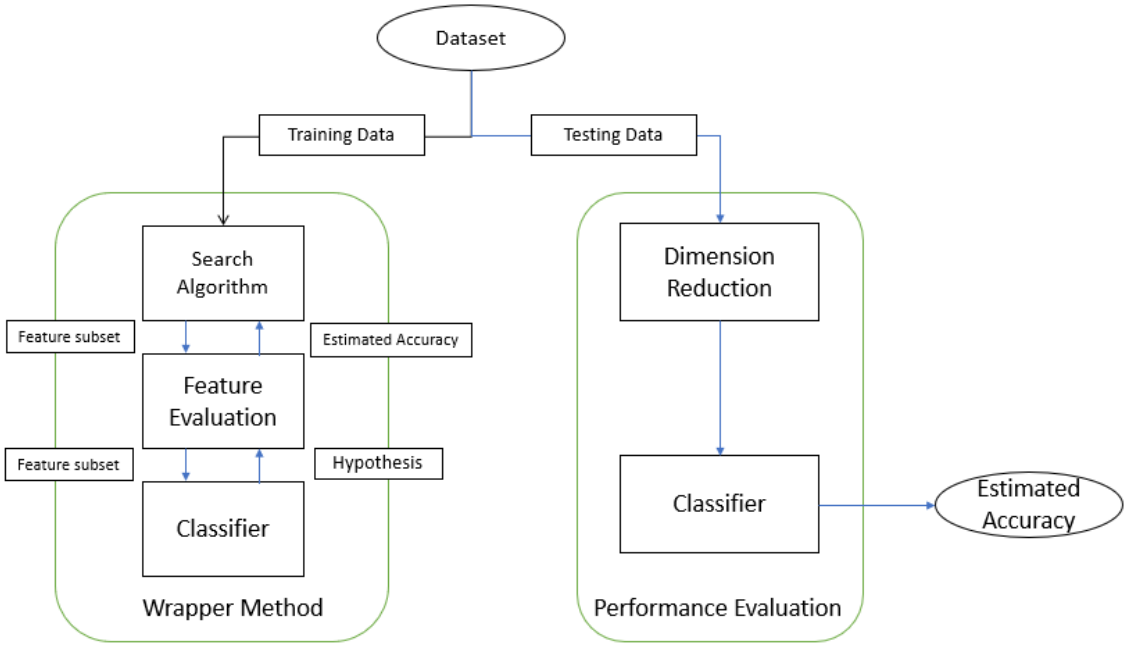


Fig. 4. General Framework of Wrapper Method.

Embedded: Embedded methods combine the qualities of filter and wrapper methods. It's implemented by algorithms that have their own built-in feature selection methods. Embedded method is more efficient and computationally more manageable than wrapper method while maintaining similar performance. This is because the embedded method ignores the repetitive execution of classifier and examination of every feature subset. Additionally, this method has lower risk to over-fitting compared to wrapper method [5].

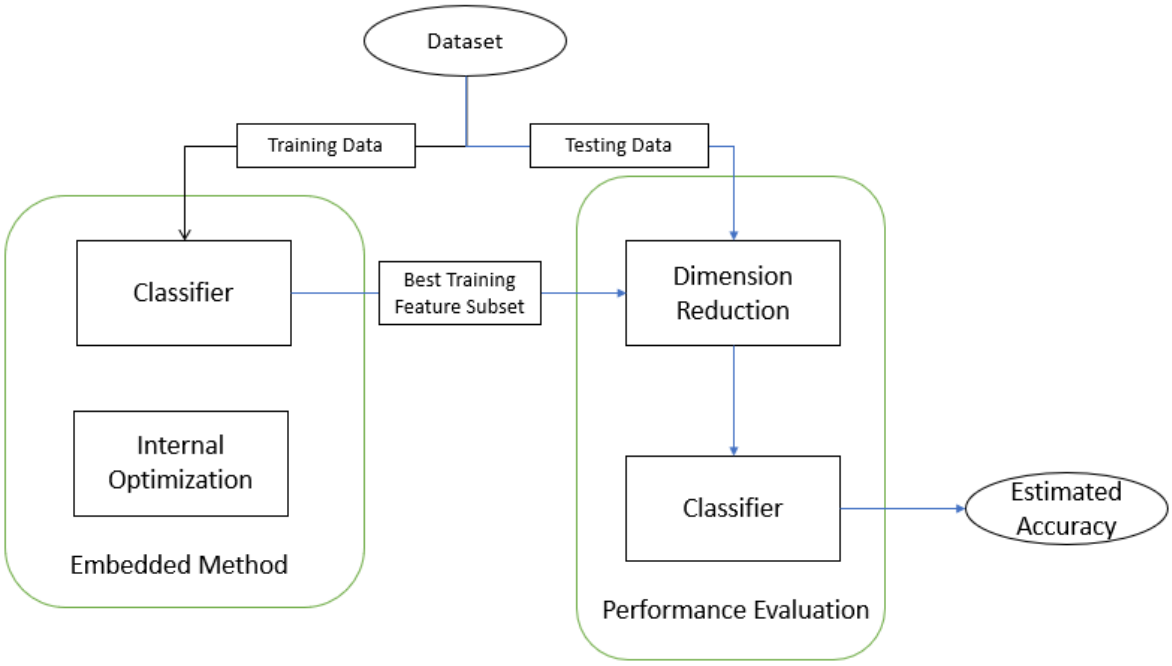


Fig. 5. General Framework of Embedded Method

Hybrid: Hybrid method can be either formed by merging two distinct methods (e.g. Filter and wrapper), two methods of the similar criteria, or two feature selection approaches. Hybrid method tries to acquire the advantages of both methods by combining their corresponding strengths [6]. To enhance the efficiency and prediction performance with better computational performance, it applies distinct evaluation criteria in different search stages.

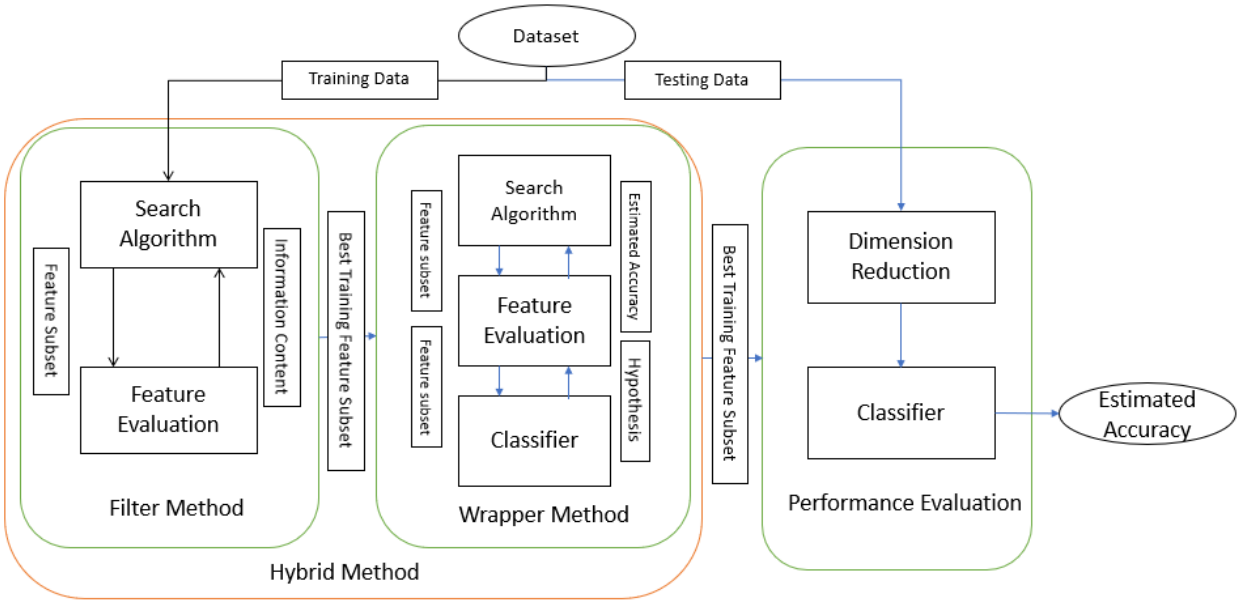


Fig. 6. General Framework of Hybrid Method.

Ensemble: Ensemble method is a method that plans to build a class of feature subsets and then deliver a total result out of the class [16]. It is intentionally intended to handle the uncertainty and agitation issues in numerous feature selection algorithm.

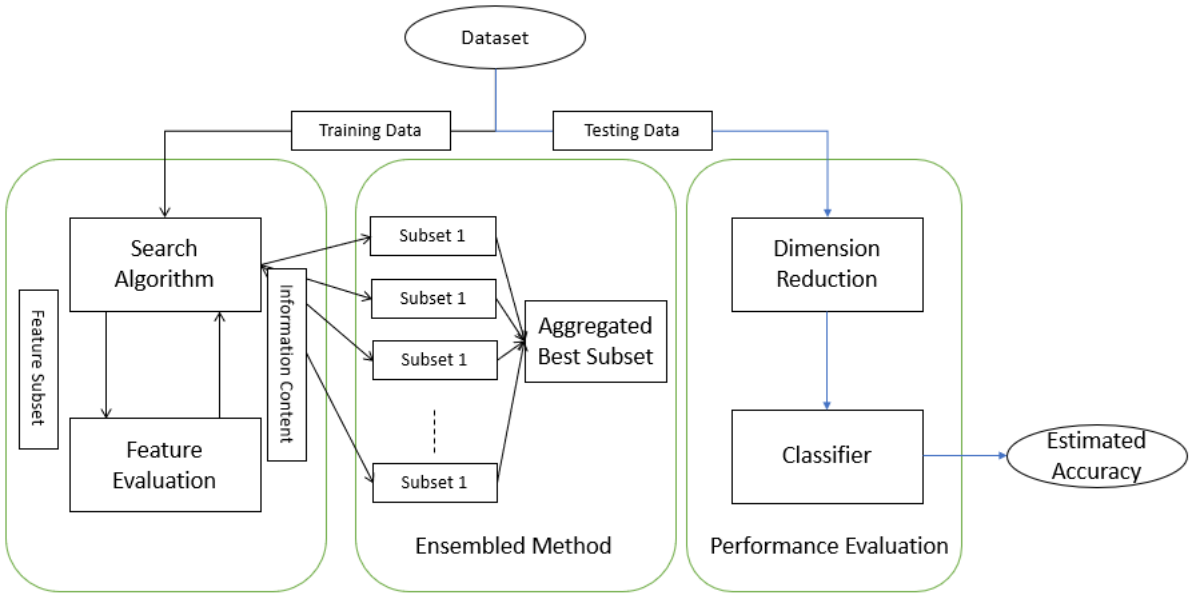


Fig. 7. General Framework of Ensembled Method.

2.3 Unsupervised Feature Selection

Unsupervised Feature Selection is generally utilized for clustering tasks. Unsupervised feature selection is more demanding than supervised and semi-supervised methodologies since label data may not be available, to be involved in feature selection phase and model learning phase. So, unsupervised feature selection depends on other alternative measure in the time of feature selection phase to describe feature relevance. One ordinarily utilized rule picks features that can best conserve the assorted structure of the original data. Another commonly used technique is to search for cluster pointers by the help of clustering algorithms and then change the unsupervised feature selection into a supervised framework. There are two distinct approaches to utilize this technique. One way is to look for cluster pointers and at the same instant perform the supervised feature selection within one combined framework. The other approach is to first look for cluster pointers, then execute feature selection to discard or choose certain features, and finally to repeat these two steps iteratively until certain criteria is met [4].

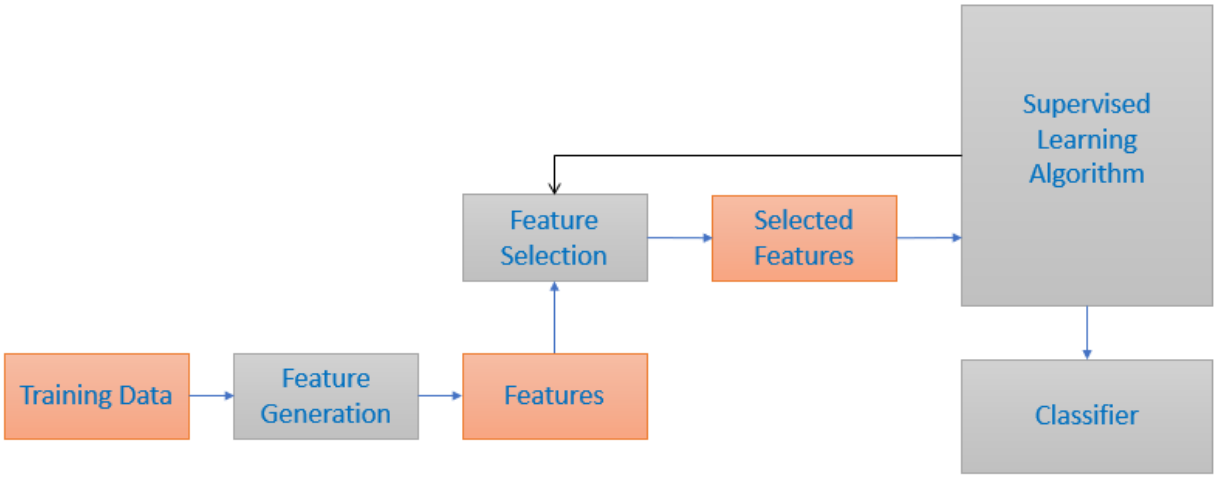


Fig. 7. General Framework of Unsupervised Feature Learning

2.4 Clustering

Clustering is a division of data into groups of similar objects. Each group (= a cluster) consists of objects that are similar between themselves and dissimilar to objects of other groups. From the machine learning perspective, Clustering can be viewed as unsupervised learning of concepts.

Types of clustering

Partitioning Clustering

They are iterative relocation calculation. This method separates the data objects into non-covering clusters such that each data object is in exactly one subset. A few strategies which are utilized, for example, (a) K-medoids, (b) K-means, (c) Probabilistic.

Prototype based Clustering

A cluster is an arrangement of items in which each question is nearer (more comparative) to the model that defines the group than to the model of some other bunch. For information with consistent traits, the model of a bunch is frequently a centroid, i.e., the normal (mean) of the considerable number of focuses in the cluster. At the point when a centroid isn't significant, for example, when the information has clear cut characteristics, the model is regularly a medoid, i.e., the most illustrative purpose of a bunch. For some sorts of information, the model can be viewed as the most essential issue, and in such occurrences, we regularly allude to model-based bunches as focus-based groups. As anyone might expect, such groups tend to be globular [8].

Graph Based Clustering

One of the families of clustering algorithm is Graph-Based Clustering. These algorithms ascertain the problem in terms of an undirected graph. Each node on the graph is connected to a sample in the feature being represented, while each edge defines the distance between each data point node based on parameters set which define the relationship between data points.

Graph based clustering depends on searching for the minimum cut or cliques in a proximity graph. Some clustering algorithms of this family also depend on eigen vectors and eigen values.

Graph Based Clustering algorithms majorly consist of a few crucial steps:

- Pre-processing of data: This step essentially converts the given data into a graph by analysing the data points and establishing a relationship to connect nodes in the graph.
- Partitioning of the Graph: This step essentially finds out the similarity of nodes and its dissimilarity to the other nodes and partitions the graph.

- Clustering: The partitioned graphs are now subjected to clustering procedures until the required number of clusters are not obtained.

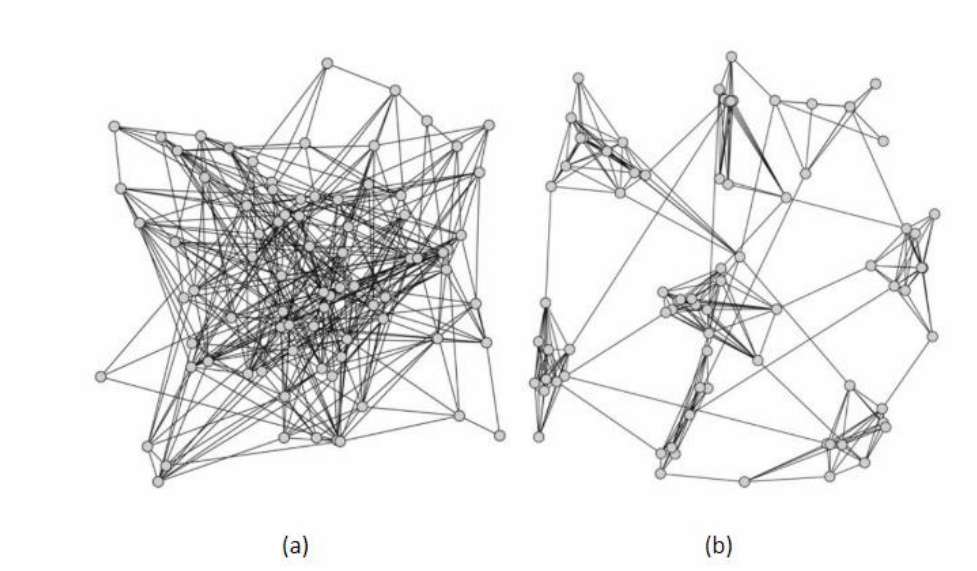


Fig. 8. Two graphs (a) and (b) both of which have 84 vertices and 358 edges. The graph on the left however is an un-clustered graph, whereas on the right is a clustered with similar nodes near a cluster

3. PROPOSED FRAMEWORK

In this section we shall discuss in detail the framework proposed in our work. The research we have done to develop a working model involves the accumulated knowledge of some algorithms and measures which need to be discussed separately (as has been done) to provide a full picture of what we are striving to achieve. This section will be divided into multiple sub-sections, each highlighting a method that has been used in our framework and the cumulative approach combining all these methods into one at the very end of this section.

3.1 Pearson Correlation

The Pearson correlation coefficient is a measure of the quality of a linear relationship between two features/samples/factors/variables and is denoted by r. Fundamentally, a Pearson correlation is an endeavour to draw a line of best fit through the data of two variables, and the Pearson correlation coefficient, r, shows how far away every one of these data points are from the line of best fit (i.e., how well the data fits this new model/line of best fit) [7].

The Pearson Correlation Coefficient can take values ranging from -1 to +1. A value 0 indicates that there is no correlation between the data points. Any negative value of the coefficient indicates that there is an inverse relation between the data points; i.e. when the value of a variable decreases, the value of the other variable correlated to it increases. A positive value of the coefficient indicates positive correlation where the increase in value of a variable translates to an increase in value of the other variable correlated to it.

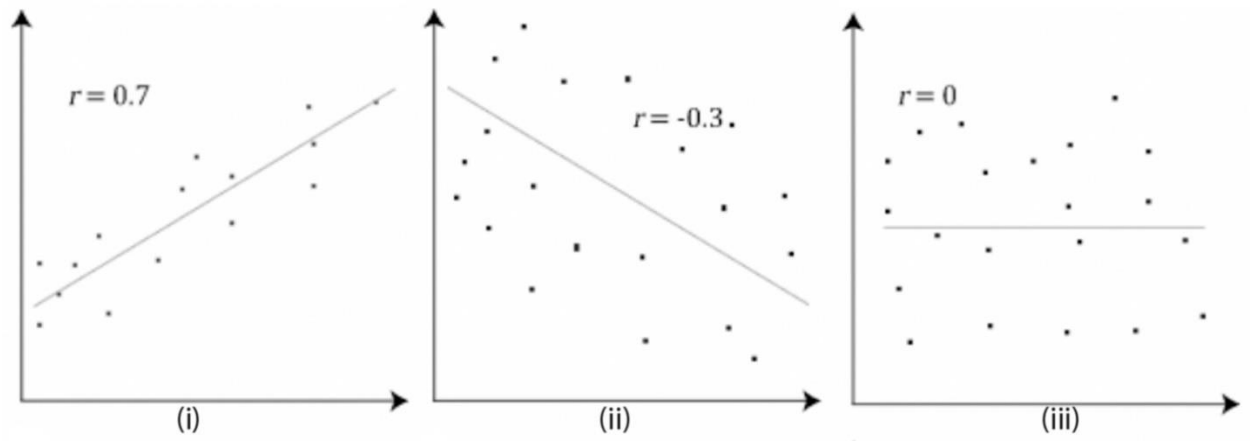


Fig. 9. (i)- Positive correlation (ii)-negative correlation (iii)-no correlation

Stronger association of the data points indicate that the Pearson correlation coefficient(r) is closer to either +1 or -1. When r is -1/+1, data points are perfectly aligned on the line of best fit.

If we want to inspect correlations, we'll have a computer calculate them for us. You'll probably never need the actual formula. However, for the sake of completeness, a Pearson correlation between variables X and Y is calculated by

$$r_{XY} = \frac{\Sigma_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\Sigma_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\Sigma_{i=1}^{n}(Y_i - \bar{Y})^2}} \qquad \ldots\ldots\ldots\ldots(1)$$

The formula basically comes down to dividing the covariance by the product of the standard deviations.

An outlier in this analysis is a data point that does not correspond the general outlay of the data given and appears as an extreme anomaly among the plotted data points. Outliers must be tested as they create a large impact on the line of best fit in a correlation analysis. This is illustrated in the example below which clearly shows how the scatter points with the outlier present and removed affect the line of best fit as well as the correlation coefficient.

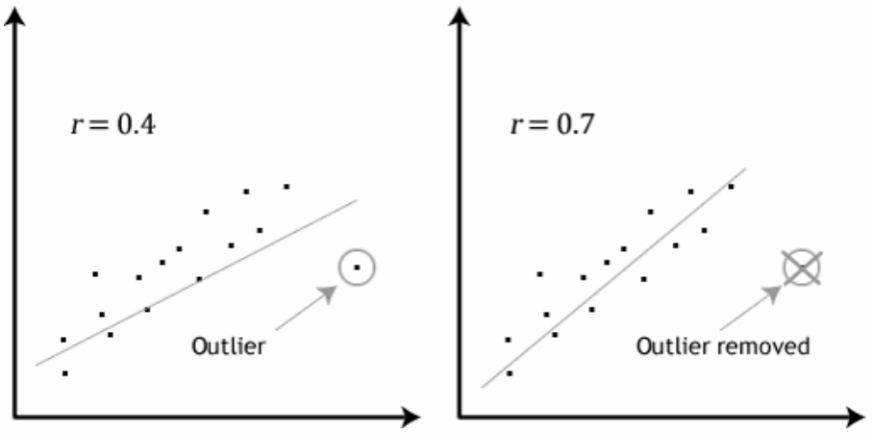


Fig. 10. Effect of Outlier on the value of r

3.2 K-means Algorithm

The K-means clustering technique is simple [8]. We first choose K initial centroids, where K is the number of clusters desired. Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated based on the points assigned to the cluster. We repeat the assignment and update steps until no point changes clusters, or equivalently, until the centroids remain the same.

Step 1: Select *K* points as initial centroids.

Step 2: repeat

Step 3: Form *K* clusters by assigning each point to its closest centroid.

Step 4: Recompute the centroid of each cluster.

Step 5. Until centroids do not change.

3.3 Wrapper Method

This technique evaluates subsets of features according to how informative they are to a given predictor. The method conducts a search for a good subset using the learning algorithm itself as a part of the evaluation function. The problem boils down to a problem of stochastic state space search. In Wrapper-based method, subset selection takes place based on the learning algorithm. That means wrapper-based method requires one predetermined learning algorithm in feature selection. It uses to evaluate performance and determine which features are selected. It gives superior performance as it finds features better suited to the predetermined learning algorithm, but computationally it is more expensive. This method based on successive elimination of features and leaving those features that lead to highest accuracy. Main objective of wrapper-based method is to search for the best subset of features. To achieve the best possible performance with a particular learning algorithm on a particular training set, a feature subset selection method should consider how the algorithm and the training set interact.

3.4 Naive Bayes Classifier

Naive Bayes Classifier or NBC is a classification technique based on Bayes Theorem of probability [8]. NBC operates with an assumption that the presence of a particular feature in a class or sample is independent of the presence of any other feature. NBC is easy to build and useful for very large data sets, much like the datasets used in our project. It usually outshines most other classification techniques which work with large datasets.

NBC predicts membership probabilities for every class such as the likelihood that given record or information point has a place with a specific class. The most likely class is the one with the highest probability and is called MAP or Maximum A Posteriori.

MAP(H)

= max(P(H|E))

$$= \max \left(\frac{P(E|H) * P(H)}{P(E)} \right) \quad \text{.....(2)}$$

= max (P(E|H) * P(H))

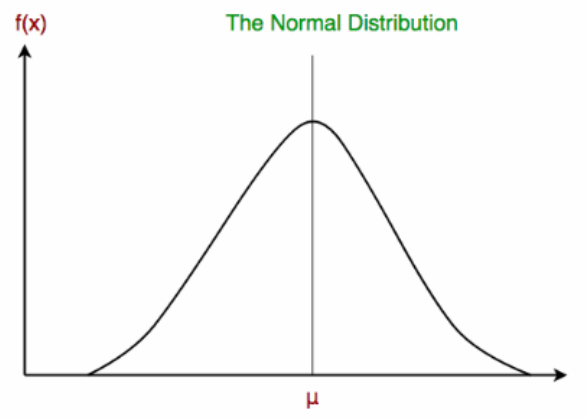
where

- P(H) is the probability of hypothesis H being true. This is known as the prior probability.
- P(E) is the probability of the evidence (regardless of the hypothesis).
- P(E|H) is the probability of the evidence given that hypothesis is true.
- P(H|E) is the probability of the hypothesis given that the evidence is there.

There are several types of NBC algorithms. The ones used in our work for classification of the data are Gaussian Naive Bayes algorithm and Multinomial Naive Bayes algorithm, which have been discussed.

Gaussian Naive Bayes

It is a special kind of NBC algorithm which assumes that all features in the dataset have continuous values. Also, it follows the assumption that the features are following a Gaussian distribution or Normal distribution. Whenever plotted, it gives a bell-shaped bend/curve which is symmetric about the mean of the element or in this case, feature as demonstrated:



The conditional probability is given by:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad \text{.....(3)}$$

Multinomial Naive Bayes

It predicts the conditional probability of a particular feature given a class as the relative frequency of term(t) or feature in documents belonging to the class(c).

$$P(t|c) = \frac{T_{ct}}{\sum_{t' eV} T_{ct'}} \quad \text{.....(4)}$$

Thus, this variation takes into consideration the number of repetitions of feature(t) in the training dataset from class c.

3.5 K-Nearest Neighbour Classifier

K-Nearest neighbour Classifier or KNN employs a simple algorithm for classification of data by storing all available features and classifies any new test data based on a set similarity measure [9]. It is a non-parametric technique which is used for classification of pattern recognition datasets.

KNN classifier works on a simple level. Data points across the plane have already been assigned to separate classes and when a test data point is plotted, based on the value of K, its nearest neighbours are traced. Based on the similarity of a data point to the class (c) of data which it is closest to, it can be ascertained with a good confidence that the test data point belongs to the class c. Choosing the value of K should thus be done with proper care. When a graph of the validation error to the value of K is plotted, depending on the dataset, it is seen that error decreases with the increase of K until it reaches a minima. This is the value of K that should be chosen as the optimal value where there is minimal underfitting or overfitting of data to provide a fairly accurate observation of classification.

Pseudo code for KNN:

(i) Load the data

(iii) k value is Initialised

(iii) For getting the predicted class, iterate from 1 to total number of training data points:

1.Calculate the distance between test data and each row of training data. Any distance metric can be used for this measurement.

2.Sort the calculated distances in ascending order based on distance metric.

3.Extract the top k rows from the array(sorted).

4.Get the most frequent class of these rows.

5.Return the predicted class.

3.6 Support Vector Machine

A Support Vector Machine (SVM) is a discriminative classifier formally characterized by an isolating hyperplane. At the end of the day, given labelled training dataset (supervised learning), the calculation yields an ideal hyperplane which orders training dataset. In two-dimensional space this hyperplane is a line separating a plane in two sections where in each class lay in either side. [10]

Given a plot of two label classes on graph as shown in the image

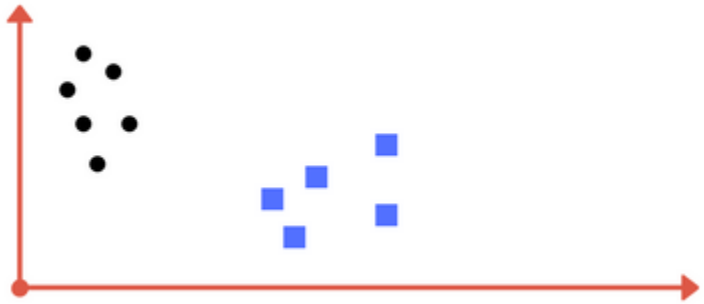


Fig. 11. Sample data to be classified

something similar to the image (image B) can be considered to demarcate between the two classes. It fairly separates the two classes. Any point that is left of line falls into black circle class and on right falls into blue square class.

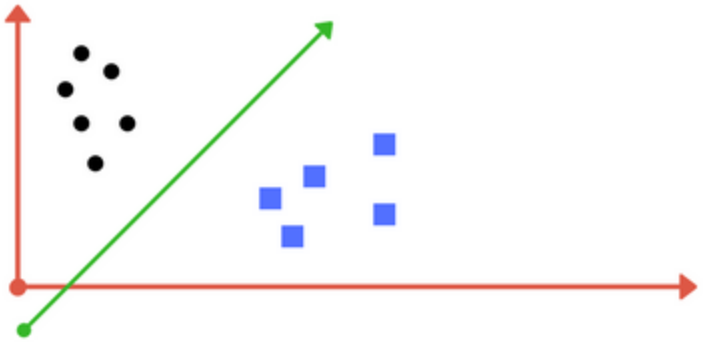


Fig. 12. Sample data classified by SVM

If there is no line that can separate the two classes, like in this x-y plane.

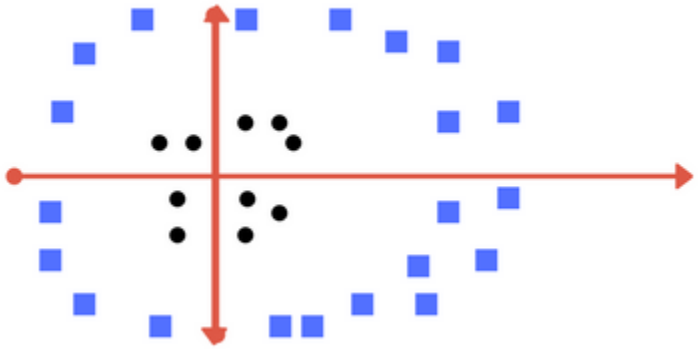


Fig. 13. Another sample data to be classified

Transformation is applied and one more dimension is added called z-axis. Let the value of points on z plane be such as, $w = x^2 + y^2$. In this case we can manipulate it as distance of point from z-origin. Now if we plot in z-axis, a clear separation is visible, and a line can be drawn. When it is transformed back to original plane this line maps to circular boundary as shown in image E. These transformations are called kernels.

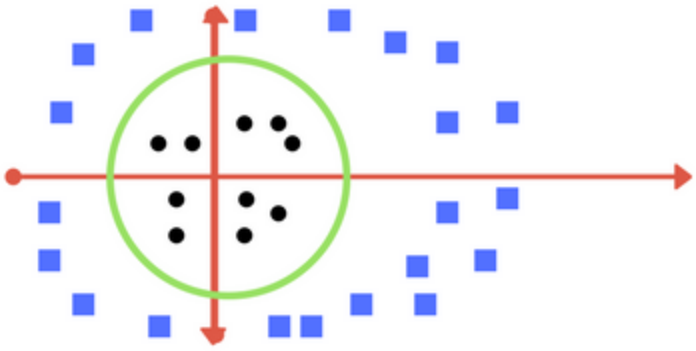


Fig. 14. Probable classification of the sample data

If a situation occurs something like this

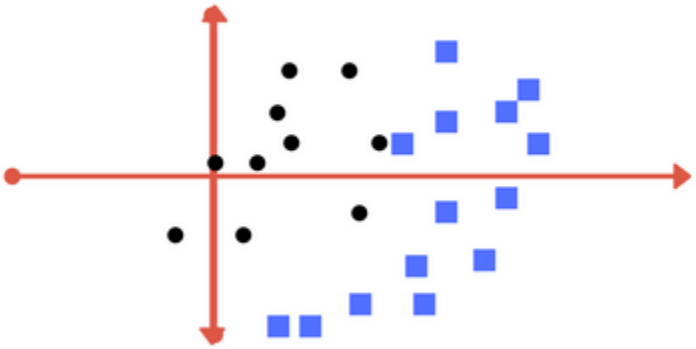


Fig. 15. One more sample data to be classified

There are two possible outcomes of this situation, namely

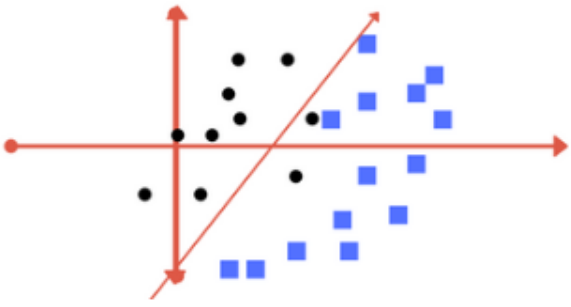


Fig. 16. First possibility of classification of the data

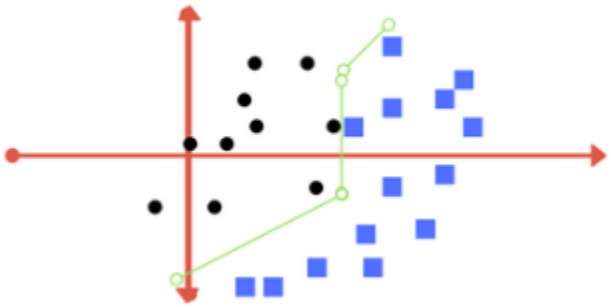


Fig. 17. Second possibility of classification of the data

But in real world application, finding perfect class for millions of training data set takes lot of time. There is something called regularization parameter, two terms are defined regularization parameter and gamma. These are the tuning parameters in SVM classifier. Tuning these considerable non-linear classification line with more accuracy can be achieved in reasonable amount of time.

One more parameter is kernel. It is defined as whether a linear of linear separation is required or not.

For linear kernel the equation for prediction for a new input using the dot product between the input (x) and each support vector (x_i) is calculated as follows:

$$f(x) = B(0) + sum(a_i * (x, x_i)) \quad \text{.....(5)}$$

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

The polynomial kernel can be written as:

$$K(x, x_i) = 1 + sum(x * x_i)^d \quad \text{.....(6)}$$

and exponential as:

$$K(x, x_i) = exp(-gamma * sum((x - x_i^2)) \quad \text{.....(7)}$$

3.7 Proposed Methodology

The whole framework of the algorithm is implemented in python 2.7. Various modules used are numpy (for making suitable numpy arrays for the various functions), scipy (for the Pearson Correlation function), biopython (for the K-means using different distance functions) and sklearn (for the classifiers and K-fold algorithm), these modules contain various algorithms which is used to implement the combination of algorithms we used.

Step 1: Loading the Matrix

The input data of m x n (m number of samples and n number of features n>>m) is first imported and divided into two lists. The first is a two-dimensional list consisting the genetic data (m x (n-1)) **X_features** and the second contains the class levels of each data (m x 1) **Y_classes**.

Step 2: Clustering

The list **X_features** is clustered in an unsupervised manner using the K-means algorithm. The function returns a list **cluster_id** (m x 1) with the cluster id of the subsequent columns, i.e. which feature belongs to which cluster. The cluster id is calculated according to the Pearson Correlation measure as the distance function. The number of clusters **d** is varied from 1 to 50 as given in the result section. Then a list is made which takes the **X_features** and clusters it according to the cluster id as **corr_data**.

Step 3: Centroid Selection

Then the cluster centroid is calculated using the function in biopython module and the cluster point's distance is calculated using Pearson Correlation. The point closest to the centroid is selected as the main centroid and representative of the cluster as **centroid_data** (m x d). The number of representative **r** is also varied from 1 to 30. So, a combination of variable clusters and variable cluster representative is used to form a table to check which combination gives the best result of the reduced dimension.

Step 4: Wrapper Method

A wrapper method is then implemented that combines a classification algorithm within the feature selection process. Out of the **d** clusters formed, the centroid (**centroid_data**) is taken first along with a feature **f** which is chosen such that its closest to the centroid $\mathbf{r_f \rightarrow 1}$. Classifiers are run on these chosen data to check if the accuracy of the classification using this particular feature increases or decreases or is unaffected.

Wrapper method normally uses forward and backward selection process based on ranking or class levels provided in an annotated dataset. However, the procedure undertaken here seeks to perform wrapper method in an unsupervised manner by using clustering on basis of Pearson Correlation of the data points to achieve the result. The Pearson Correlation employed takes only the positively correlated genes, as they exhibit better behaviour in gene expression.

It is seen that pattern-based genes exhibit similar behaviour, and so do the positively correlated genes. Negatively correlated genes do not exhibit such a promising behaviour compared to positively correlated ones. Taking the positively correlated genes does not hamper the chance of getting genes from negative correlation as positively correlated genes falls under separate cluster and negatively correlated under a different cluster as the clustering performed used Pearson Correlation as its distance metric. Moreover, two negatively correlated genes to another gene are positively correlated with each other and thus do not contribute to any significant data loss during the classification process.

A classifier is a tool used to test the data and to check how much accurately the algorithm has generated the reduced dataset. The classifiers used here are **KNN**, **SVM** and **NBC** (**G-NBC** and **M-NBC**).

The training is done on basis of leave but one method where if 10 samples are present, 9 are used to train and 1 is used to test. This is achieved by the K-fold algorithm where number of folds is equal to number of samples.

K-fold algorithm uses **(n-1)** folds out of **n** folds to train and **1** fold to test, and it is repeated until all the folds are tested. So, if the number of folds is equal to number of samples, all the samples will be tested at least once.

After this in every loop of the K-fold algorithm the accuracy is maintained as to how many times the tested data was correct, compared to the class level **Y_classes**, sorted out before, this is stored in the list **accuracy** (m x 1) where **m** is number of samples. An average is calculated after the completion of the K-fold algorithm as the result of the classification.

Using classifiers to guide the selection procedure, it is checked it is found that:

- If the feature is REDUNDANT, the accuracy remains SAME
- If the feature is IRRELEVANT, the accuracy DECREASES
- If the feature is RELEVANT, the accuracy INCREASES.

Using this observation, a particular feature is accepted if and only if it Increases the accuracy of the classification. This is done by comparing the previous accuracy without that feature and then with that feature. By this wrapper method it can be determined which features are relevant and which are redundant, and from those only the relevant features are selected, and the redundant features are discarded. So, at the end, a list **dataset_train** is generated of dimensions **(m x r x d)** where **m** is the number of samples, **d** is number of clusters and **r** is the number of representatives from each cluster. Finally, the accuracy generated is printed and tabulated and shown in the result section.

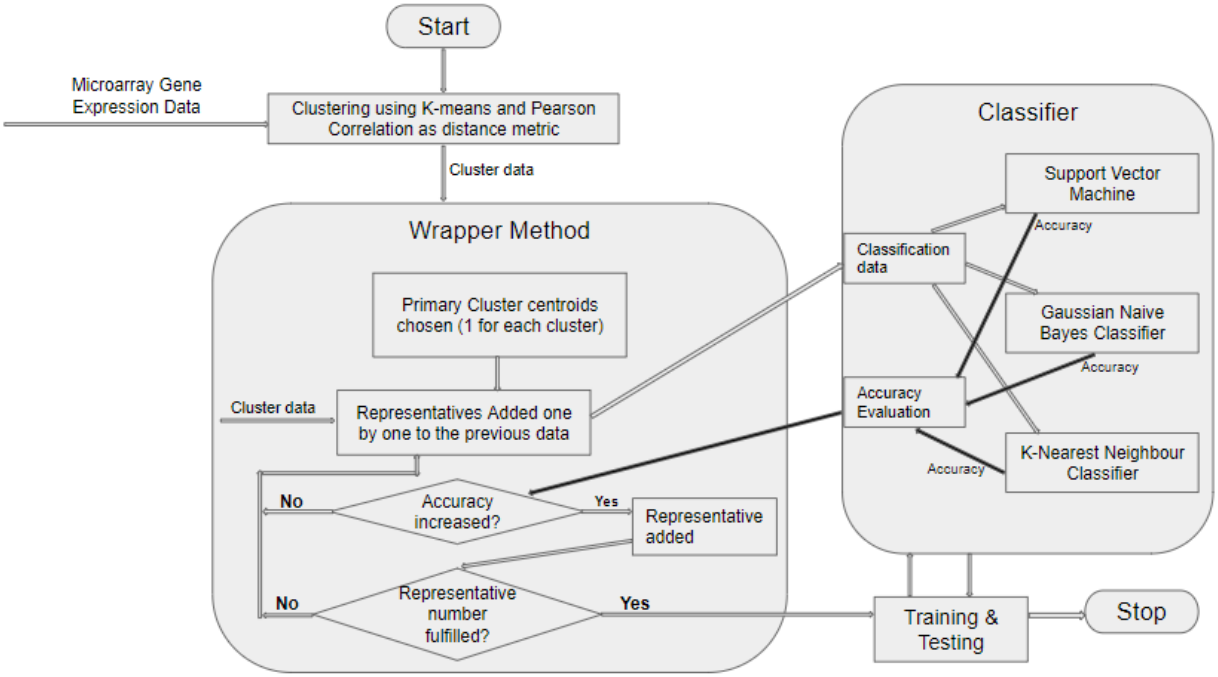


Fig. 18. Framework of implemented algorithm

4. RESULT

This section discusses the accumulation of all the techniques discussed above and combines them into the framework that has been proposed for an accurate dimensional reduction. The framework contains the research work done using the mentioned algorithms namely K-Means, in which the distance metric used is Pearson Correlation. As the dataset is mainly genetic dataset, the distance formula is used as such to provide a better estimation of the data points that need to be worked with. The correlation of the genes is used as a relevancy factor in determining the redundant genes and significant genes.

The training of the data for classification is done by leave but one method where one sample is used for testing the data for its accuracy whereas the other samples are used to train the data. This process is iterated over the entirety of the sample space where every individual sample is used to test the data in an iteration and the others are used to train. For example, if there are 10 samples, numbered 1 to 10, in the 1st iteration, sample 2-10 are trained and sample 1 is tested. The next iteration sample 2 is tested while 1 and 3-10 are trained, and so on. This procedure ensures that each and every sample is tested at least once in the process of classification.

The result section which comes after this contains a list of tables based on the research we did. The original dataset and the dimensionally reduced dataset both are trained on various classifiers or machine learning algorithms and checked for percentage accuracy.

For a given number of representatives, the best accuracy of the number of cluster is highlighted. For example, if the number of representatives is 5 and if the number of clusters is 10 which gives the best accuracy compared to other number of clusters, then that part is highlighted.

The datasets which are used to test the algorithm proposed are tabulated in the next page, which is namely, colon [11], lung [12], leukaemia [13], breast [14] and prostate [15]. The classifiers or the machine learning algorithms used here are K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Gaussian Naïve Bayes (G-NBC), and Multinomial Naïve Bayes (M-NBC). As Multinomial Naïve Bayes doesn't take up negative values in some of the percentage accuracy result table Multinomial Naïve Bayes (M-NBC) classifier might not be present.

Name	Dimension	Source
Breast cancer	7129 x 49	[11]
Colon cancer	2000 x 62	[12]
Lung cancer	12533 x 181	[13]
Leukaemia	7070 x 72	[14]
Prostate cancer	12600 x 136	[15]

Dataset description and details

The percentage accuracy result table is divided based on different dataset and classifiers with the significantly good results highlighted, and consolidated result table is provided at the end of the results tables of all the datasets.

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	58%	79%	72%	**85%**	81%	79%	75%	75%
2	58%	60%	79%	75%	**83%**	72%	79%	75%
3	56%	75%	79%	75%	**79%**	77%	77%	75%
4	50%	43%	72%	**81%**	79%	77%	70%	62%
5	47%	52%	**75%**	64%	75%	75%	72%	55%
10	50%	72%	**75%**	75%	68%	72%	72%	55%
15	56%	54%	**75%**	75%	70%	75%	75%	65%
20	43%	56%	**75%**	75%	72%	72%	75%	57%
25	43%	60%	68%	**75%**	75%	75%	72%	59%
30	58%	**75%**	64%	68%	75%	77%	75%	60%

KNN: K-Means and Pearson Correlation on Breast Cancer Dataset (7129 x 49)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	**52%**	52%	52%	52%	52%	52%	52%	52%
2	**52%**	52%	52%	52%	52%	52%	52%	52%
3	**52%**	52%	52%	52%	52%	52%	52%	52%
4	**52%**	52%	52%	52%	52%	52%	52%	52%
5	**52%**	52%	52%	52%	52%	52%	52%	52%
10	**52%**	52%	52%	52%	52%	52%	52%	52%
15	**52%**	52%	52%	52%	52%	52%	52%	52%
20	**52%**	52%	52%	52%	52%	52%	52%	52%
25	**52%**	52%	52%	52%	52%	52%	52%	52%
30	**52%**	52%	52%	52%	52%	52%	52%	52%

SVM: K-Means and Pearson Correlation on Breast Cancer Dataset (7129 x 49)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	50%	**79%**	72%	79%	77%	81%	77%	50%
2	52%	**81%**	66%	89%	79%	83%	79%	52%
3	35%	54%	58%	79%	81%	83%	**85%**	35%
4	45%	62%	75%	**87%**	87%	72%	75%	45%
5	50%	52%	**87%**	83%	87%	77%	77%	50%
10	58%	58%	**85%**	81%	83%	81%	72%	58%
15	45%	70%	**89%**	79%	87%	79%	83%	45%
20	47%	72%	85%	**87%**	87%	77%	79%	47%
25	47%	62%	66%	**83%**	83%	70%	79%	47%
30	54%	54%	**79%**	79%	75%	77%	77%	54%

G-NBC: K-Means and Pearson Correlation on Breast Cancer Dataset (7129 x 49)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	44%	68%	70%	**80%**	80%	72%	73%	44%
2	45%	73%	67%	77%	78%	**81%**	81%	45%
3	55%	65%	78%	77%	78%	80%	**81%**	55%
4	62%	67%	75%	73%	78%	**80%**	78%	62%
5	55%	77%	72%	77%	80%	**83%**	80%	55%
10	55%	65%	77%	80%	80%	82%	**85%**	55%
15	65%	72%	80%	80%	81%	**83%**	80%	65%
20	57%	65%	**83%**	80%	78%	83%	83%	57%
25	59%	60%	**83%**	81%	83%	82%	78%	59%
30	60%	63%	**85%**	83%	77%	82%	83%	60%

KNN: K-Means and Pearson Correlation on Colon Cancer Dataset (2000 x 62)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	63%	**65%**	65%	65%	65%	65%	65%	63%
2	**68%**	65%	65%	65%	65%	65%	65%	68%
3	**65%**	65%	65%	65%	65%	65%	65%	65%
4	**65%**	65%	65%	65%	65%	65%	65%	65%
5	**65%**	65%	65%	65%	65%	65%	65%	65%
10	**65%**	65%	65%	65%	65%	65%	65%	65%
15	**65%**	65%	65%	65%	65%	65%	65%	65%
20	**65%**	65%	65%	65%	65%	65%	65%	65%
25	**65%**	65%	65%	65%	65%	65%	65%	65%
30	**65%**	65%	65%	65%	65%	65%	65%	65%

SVM: K-Means and Pearson Correlation on Colon Cancer Dataset (2000 x 62)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	**65%**	49%	63%	65%	60%	59%	65%	65%
2	65%	54%	62%	63%	62%	**67%**	60%	65%
3	27%	62%	**63%**	60%	62%	63%	60%	27%
4	32%	50%	60%	55%	55%	58%	**62%**	32%
5	34%	50%	59%	**62%**	55%	62%	60%	34%
10	42%	55%	57%	55%	**60%**	57%	57%	42%
15	44%	52%	55%	55%	57%	**60%**	55%	44%
20	42%	52%	57%	55%	57%	**58%**	55%	42%
25	44%	57%	**63%**	59%	57%	57%	57%	44%
30	44%	55%	57%	55%	**62%**	52%	54%	44%

G-NBC: K-Means and Pearson Correlation on Colon Cancer Dataset (2000 x 62)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	65%	77%	83%	83%	78%	**85%**	83%	65%
2	60%	70%	70%	80%	85%	**88%**	83%	60%
3	65%	59%	78%	73%	83%	86%	**90%**	65%
4	62%	67%	82%	80%	85%	**90%**	88%	62%
5	62%	72%	78%	85%	**88%**	85%	86%	62%
10	65%	73%	**86%**	81%	83%	82%	85%	65%
15	59%	75%	81%	**86%**	85%	85%	86%	59%
20	54%	78%	80%	**88%**	86%	86%	86%	54%
25	62%	80%	82%	**86%**	86%	86%	86%	62%
30	65%	78%	80%	85%	**86%**	85%	86%	65%

M-NBC: K-Means and Pearson Correlation on Colon Cancer Dataset (2000 x 62)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	65%	**94%**	75%	80%	65%	79%	83%	65%
2	62%	**93%**	86%	86%	76%	81%	88%	62%
3	62%	**88%**	80%	70%	87%	81%	77%	62%
4	62%	**94%**	90%	75%	88%	90%	90%	62%
5	66%	87%	**93%**	77%	87%	87%	91%	66%
10	69%	90%	86%	80%	81%	87%	**93%**	69%
15	58%	88%	77%	79%	87%	87%	**91%**	58%
20	76%	86%	81%	88%	88%	**91%**	90%	76%
25	80%	83%	90%	87%	88%	**94%**	90%	80%
30	76%	87%	87%	90%	**91%**	91%	91%	76%

KNN: K-Means and Pearson Correlation on Leukaemia Dataset (7070 x 72)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	63%	**65%**	65%	65%	65%	65%	65%	63%
2	**65%**	65%	65%	65%	65%	65%	65%	65%
3	**65%**	65%	65%	65%	65%	65%	65%	65%
4	**65%**	65%	65%	65%	65%	65%	65%	65%
5	**65%**	65%	65%	65%	65%	65%	65%	65%
10	**65%**	65%	65%	65%	65%	65%	65%	65%
15	**65%**	65%	65%	65%	65%	65%	65%	65%
20	**65%**	65%	65%	65%	65%	65%	65%	65%
25	**65%**	65%	65%	65%	65%	65%	65%	65%
30	**65%**	65%	65%	65%	65%	65%	65%	65%

SVM: K-Means and Pearson Correlation on Leukaemia Dataset (7070 x 72)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	65%	90%	91%	94%	**97%**	94%	91%	65%
2	65%	**97%**	94%	81%	94%	94%	97%	65%
3	65%	93%	90%	95%	98%	**100%**	97%	65%
4	61%	91%	94%	90%	97%	93%	**98%**	61%
5	58%	95%	91%	97%	88%	**98%**	98%	58%
10	54%	95%	94%	98%	94%	98%	**100%**	54%
15	55%	94%	**98%**	95%	98%	98%	97%	55%
20	59%	95%	**98%**	97%	98%	98%	97%	59%
25	52%	97%	94%	95%	98%	98%	**100%**	52%
30	48%	95%	**100%**	98%	97%	100%	98%	48%

G-NBC: K-Means and Pearson Correlation on Leukaemia Dataset (7070 x 72)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	80%	93%	91%	94%	95%	**97%**	97%	80%
2	85%	97%	93%	97%	96%	**98%**	98%	85%
3	86%	92%	95%	97%	97%	97%	**98%**	86%
4	86%	96%	97%	**98%**	98%	97%	97%	86%
5	88%	97%	97%	97%	**98%**	98%	98%	88%
10	91%	96%	**98%**	98%	98%	98%	98%	91%
15	88%	97%	**98%**	98%	98%	98%	98%	88%
20	88%	97%	97%	97%	**98%**	98%	98%	88%
25	87%	**99%**	97%	98%	98%	98%	98%	87%
30	87%	**98%**	97%	98%	98%	98%	98%	87%

KNN: K-Means and Pearson Correlation on Lung Cancer Dataset (12533 x 181)

Number of Representatives	Number of Clusters							Full Dataset
	1	**5**	**10**	**20**	**30**	**40**	**50**	**Full**
1	**82%**	82%	82%	82%	82%	82%	82%	82%
2	**82%**	82%	82%	82%	82%	82%	82%	82%
3	**82%**	82%	82%	82%	82%	82%	82%	82%
4	**82%**	82%	82%	82%	82%	82%	82%	82%
5	**82%**	82%	82%	82%	82%	82%	82%	82%
10	**82%**	82%	82%	82%	82%	82%	82%	82%
15	**82%**	82%	82%	82%	82%	82%	82%	82%
20	**82%**	82%	82%	82%	82%	82%	82%	82%
25	**82%**	82%	82%	82%	82%	82%	82%	82%
30	**82%**	82%	82%	82%	82%	82%	82%	82%

SVM: K-Means and Pearson Correlation on Lung Cancer Dataset (12533 x 181)

Number of Representatives	Number of Clusters							Full Dataset
	1	5	10	20	30	40	50	Full
1	82%	96%	**97%**	97%	97%	97%	82%	82%
2	86%	96%	96%	**98%**	98%	98%	82%	86%
3	83%	98%	97%	**99%**	98%	98%	82%	83%
4	84%	98%	**99%**	99%	99%	99%	82%	84%
5	85%	94%	97%	**100%**	99%	99%	82%	85%
10	85%	**99%**	98%	99%	98%	97%	82%	85%
15	85%	98%	98%	**100%**	99%	99%	82%	85%
20	83%	97%	98%	**99%**	98%	98%	82%	83%
25	83%	98%	98%	**100%**	99%	98%	82%	83%
30	82%	97%	98%	**99%**	99%	98%	82%	82%

G-NBC: K-Means and Pearson Correlation on Lung Cancer Dataset (12533 x 181)

Number of Representatives	Number of Clusters							Full Dataset
	1	5	10	20	30	40	50	Full
1	63%	72%	61%	80%	75%	75%	**81%**	63%
2	63%	67%	78%	75%	74%	76%	**80%**	63%
3	62%	77%	73%	76%	75%	**79%**	78%	62%
4	61%	76%	73%	**80%**	80%	73%	77%	61%
5	60%	77%	72%	81%	81%	**85%**	78%	60%
10	72%	79%	78%	77%	80%	**83%**	77%	72%
15	65%	77%	82%	80%	75%	**83%**	83%	65%
20	68%	78%	80%	77%	78%	81%	**84%**	68%
25	71%	82%	75%	80%	75%	79%	**83%**	71%
30	64%	**83%**	76%	78%	81%	83%	82%	64%

KNN: K-Means and Pearson Correlation on Prostrate Cancer Dataset (12600 x 136)

Number of Representatives	Number of Clusters							Full Dataset
	1	5	10	20	30	40	50	Full
1	**57%**	56%	56%	56%	56%	56%	56%	57%
2	**56%**	56%	56%	56%	56%	56%	56%	56%
3	**57%**	56%	56%	56%	56%	56%	56%	57%
4	**56%**	56%	56%	56%	56%	56%	56%	56%
5	**56%**	56%	56%	56%	56%	56%	56%	56%
10	**56%**	56%	56%	56%	56%	56%	56%	56%
15	**56%**	56%	56%	56%	56%	56%	56%	56%
20	**56%**	56%	56%	56%	56%	56%	56%	56%
25	**56%**	56%	56%	56%	56%	56%	56%	56%
30	**56%**	56%	56%	56%	56%	56%	56%	56%

SVM: K-Means and Pearson Correlation on Prostrate Cancer Dataset (12600 x 136)

Number of Representatives	Number of Clusters							Full Dataset
	1	5	10	20	30	40	50	Full
1	55%	**58%**	55%	54%	55%	56%	55%	55%
2	55%	**56%**	55%	55%	55%	55%	55%	55%
3	55%	**57%**	55%	54%	55%	55%	55%	55%
4	**55%**	52%	55%	55%	55%	55%	55%	55%
5	55%	**56%**	55%	54%	55%	55%	55%	55%
10	**55%**	55%	55%	55%	55%	54%	53%	55%
15	55%	**56%**	54%	54%	55%	55%	55%	55%
20	**55%**	55%	55%	55%	55%	55%	55%	55%
25	55%	**56%**	55%	55%	55%	55%	55%	55%
30	55%	**56%**	55%	55%	56%	54%	55%	55%

KNN: K-Means and Pearson Correlation on Prostrate Cancer Dataset (12600 x 136)

1. **Breast Cancer Dataset:**

Classifier	No. of Clusters	No. of Cluster Representatives	Accuracy	Full Dataset Accuracy
KNN	20	1	85%	75%
SVM	1	1	82%	82%
G-NBC	10	15	89%	45%

2. **Colon Cancer Dataset:**

Classifier	No. of Clusters	No. of Cluster Representatives	Accuracy	Full Dataset Accuracy
KNN	50	10	85%	55%
SVM	5	1	65%	63%
G-NBC	40	2	65%	67%
M-NBC	50	3	90%	65%

3. **Leukaemia Dataset:**

Classifier	No. of Clusters	No. of Cluster Representatives	Accuracy	Full Dataset Accuracy
KNN	40	25	94%	80%
SVM	5	1	65%	63%
G-NBC	40	3	100%	65%

4. **Lung Cancer Dataset:**

Classifier	No. of Clusters	No. of Cluster Representatives	Accuracy	Full Dataset Accuracy
KNN	10	10	98%	91%
SVM	1	1	82%	82%
G-NBC	20	5	100%	85%

5. **Prostate Cancer Dataset:**

Classifier	No. of Clusters	No. of Cluster Representatives	Accuracy	Full Dataset Accuracy
KNN	40	5	85%	60%
SVM	1	1	57%	57%
G-NBC	5	1	58%	55%

On comparison with the existing algorithms, for example from the paper [17], based on clustering of relevant and significant genes from cancer data using supervised method, the information we gathered is tabulated below.

Datasets	Classifiers	Algorithm Accuracy (ACA)	Algorithm Accuracy (mRMR)	Full Dataset Accuracy	Our Algorithm Accuracy	Our Full Dataset Accuracy
Breast Cancer	SVM	81.6 %	85.7%	91.8 %	82 %	82 %
	KNN	81.6 %	89.8 %	73.5 %	85 %	75 %
	NB	81.6 %	89.8%	51.0 %	89 %	45 %
Colon Cancer	SVM	72.6 %	83.9 %	82.3 %	65 %	63 %
	KNN	77.4 %	83.9 %	74.2 %	85 %	55%
	NB	64.5 %	83.9 %	64.5 %	**90 %**	**65%**
Leukaemia	SVM	82.4 %	90.3 %	98.6 %	65 %	63 %
	KNN	82.4 %	93.1 %	76.4 %	**94 %**	**80 %**
	NB	88.2 %	94.4 %	65.3 %	**100 %**	**65 %**
Lung Cancer	SVM	82.9 %	97.8 %	98.9 %	82 %	82 %
	KNN	82.9 %	97.8 %	87.9 %	**98 %**	**91 %**
	NB	65.2 %	97.8 %	57.1 %	**100 %**	**85 %**
Prostate Cancer	SVM	71.3 %	71.3 %	91.9 %	57 %	57 %
	KNN	71.3 %	80.9 %	74.2 %	85 %	60 %
	NB	50.7 %	84.6 %	56.6 %	58 %	55 %

Comparison with existing algorithms

5. DISCUSSION

As the main objective is to dimensionally reduce the given dataset without loss in accuracy during the sample classification, i.e. removing only the irrelevant and redundant features, we implemented the above methodology as stated in proposed framework. As seen in the result section few of the dataset responded very adequately to the algorithm and in some sections, there is no loss of accuracy at all (the lung cancer dataset with both KNN and G-NBC classifier), so we can conclude that the algorithm we proposed is somewhat data specific and varies from dataset to dataset.

As, it can be seen that the whole data is not actually required to train and test the unknown data, so we can furthermore improve our algorithm to make it dataset independent, this can be done by wrapper method which is also stated in the proposed work with a diagram, as to how it is implemented, this reduces the dataset dependency of the algorithm making it more robust and general. We started with the wrapper method framework and testing is still in progress. The response is better than the filter method of feature selection.

In future, we are planning to combine multiple classifiers in the wrapper method as ensembled to produce more accurate results as each one of them, thus contributing more to the feature selection domain.

6. CONCLUSION

With growing technologies in the field of medicine and with the advent of modern technology, treatment of cancer and other diseases are spreading to the far corners of the world. Accurate and quick testing of genetic data thus has become increasingly important for the correct diagnosis of such diseases. Our work in this field seeks to further that cause in a manner that genetic data relevant to the diagnosis is weeded out from the clutter so that when the time comes for a quick and fast treatment process, the reduced genetic data can be put to good use to develop remedies for the curses of diseases that plague the modern world.

The rigorous testing of our proposed method upon several microarray gene expression datasets of cancer classification shows the utility of the proposed algorithm. The results are also compared with that of some well-known methods which actually justifies the necessity of newer techniques in this area of research. Though the proposed method is not capable of giving best results in all cases, but the method tries to provide a new dimension of idea in this area. The aim of our further research is to help further the merger of technology and medicine in healthcare of the people.

7. REFERENCE

1. L. Yu and H. Liu, “Efficient Feature Selection via Analysis of Relevance and Redundancy,” J Mach Learn Res, vol. 5, pp. 1205– 1224, Dec. 2004.
2. M. Gutkin, R. Shamir, and G. Dror, “SlimPLS: A Method for Feature Selection in Gene Expression-Based Disease Classification,” PLoS ONE, vol. 4, no. 7, p. e6416, Jul. 2009.
3. T. Bo and I. Jonassen, “New feature subset selection procedures for classification of expression profiles,” Genome Biol., vol. 3, no. 4, pp. 1–0017, 2002.
4. J. Tang, S. Wang, and H. Liu, “Feature selection for classification: A review,” 2013.
5. Jun Chin Ang, Andri Mirzal, Habibollah Haron, and Hazra Nuzly Abdulla Hamed, “Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection,” Transaction on computational biology and bioinformatics, vol. 13, no. 5, Sept/Oct. 2016.
6. M. Monirul Kabir, M. Monirul Islam, and K. Murase, “A new wrapper feature selection approach using neural network,” Neurocomputing, vol. 73, nos. 16–18, pp. 3273–3283, Oct. 2010.
7. J. L. Rodgers, W. A. Nicewander, "Thirteen ways to look at the correlation coefficient", *Amer. Statistician*, vol. 42, pp. 59-66, Feb. 1988.
8. R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification (Second Edition). Wiley & Sons, NY, 2001.
9. Tryosanka, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Bostein, D. and Altman, R.B. (2001) ‘Missing value estimation methods for DNA microarrays’, Bioinformatics, Vol. 17, pp.520–525.
10. Support vector machine classification and validation of cancer tissue samples using microarray expression data Terrence S. Furey Nello Cristianini Nigel Duffy David W. Bednarski Michèl Schummer David Haussler *Bioinformatics*, Volume 16, Issue 10, 1 October 2000
11. U. Alon, N. Barkai, D. A. Notterman, K. Gish, *et al.*, “Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays,” in Proc. National Academy of Science, USA, vol. 96, no. 12, 1999, pp. 6745–6750.
12. P. Maji and S. K. Pal, “Fuzzy-rough sets for information measures and selection of relevant genes from microarray data,” IEEE Trans. Syst. Man. Cybern. B. Cybern, vol. 40, no. 3, pp. 741–752, 2010.
13. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, *et al.*, “Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring,” *Science*, vol. 286, pp. 531–537, 1999.
14. G. J. Gordon, R. V. Jensen, L.-L. Hsiao, S. R. Gullans, J. E. Blumen stock, S. Ramaswamy, W. G. Richards, D. J. Sugarbaker, and R. Bueno, “Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma,” Cancer Res., vol. 62, pp. 4963–4967, 2002.
15. D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D’Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers, “Gene expression correlates of clinical prostate cancer behaviour,” Cancer Res., vol. 1, pp. 203–209, 2002.

16. Q. Shen, R. Diao, and P. Su, “Feature selection ensemble,” in Proc. Turing-100, 2012, vol. 10, pp. 289–306.
17. Pradipta Maji and Chandra Das “Relevant and Significant Supervised Gene Clusters for Microarray Cancer Classification”, IEEE TRANSACTIONS ON NANOBIOSCIENCE, VOL. 11, NO. 2, JUNE 2012