7.1. Introduction:

Feature selection is one of the major concerns when it comes to irrelevant and noisy features or attributes. Researchers work on feature selection methods so as to improve the overall performance of the prediction system. Strong features are filtered out that affect the prediction. As far as Fuzzy Logic is concerned the major issue lies in building up of membership functions and making decision on its appropriate parameters (MFs, their distribution and composition of fuzzy rules) and that in Neural Network, the architecture is an important issue. So taking such issues into consideration, the hybrid system of the two is used wherein GA optimization technique is also used that optimizes various parameters like number of membership functions for each input. These softcomputing techniques when combined prove to be faster than the individual ones.

Further, the antecedent and consequent parameters are the important parameters in training process. Premise parameters are the membership functions parameters and Consequent parameters are the fuzzy rule parameters. So, parameter optimisation is also the work focus in the chapter. The output is the actual classified data in FIS created with optimised parameters. Further we calculate the belongingness of the output to the actual target value. A new proposed approach using hybrid of NN, FIS and EA has been given to increase the efficiency of optimisation, thus decreasing the mean square error. This error is further used to set up the parameters for the next iteration thus improving the predictability. Evaluation for improvements and comparisons have been done through Matlab simulation. The modifications have been made in an attempt to improve the overall performance. Eco and GNFS have been proposed inorder to improve the overall performance of the system.

7.2. Use of Softcomputing Techniques:

Numerical data used in the construction of neuro fuzzy system consists of Input Output Space. Such construction requires two important steps [Fakhreddin, 2009] i.e.:

7.2.1. Structure learning or Identification Phase determining the structure of Fuzzy Rules.

7.2.2. Parameter Learning Phase optimizing and tuning up the parameters
7.2.1. Structure learning Phase:

Under this phase the focus is on the Input-Output space partitioning wherein each rule is represented by a partition. So the number of rules created depends on the number of partitions created. Overlapping of boundaries in case of crisp data is impossible but when the data is fuzzy there can be occurrence of overlapping boundaries. So to form the antecedent of the fuzzy rules in the fuzzy system we go for Input space partitioning, thus explaining the structure and relation between the input space and fuzzy rules. The various clustering techniques are grid based, partitional based, density based and hierarchy based.

7.2.2. Parameter Learning Phase:

This Phase focuses on modifying and adjusting the parameters (weights and membership functions). The learning algorithm used for optimizing weights and biases of network can be classified under three categories i.e. supervised, unsupervised and reinforcement learning. The Backpropagation learning algorithm is based on the gradient descent optimization, but it may get trap in local minima. However, increasing number of hidden layer neurons in the network decreases the number of local minima thus increasing the chance of convergence in the algorithm [Fekhreddin,2009] [Baldi, 1995] In hybrid models like ANFIS, a combination of the least square method and the gradient descent technique is used for tuning the parameters [Jang, 1992] i.e. gradient decent used for learning and tuning antecedent parameters and least squares method is used to learn the consequent parameters [Fakhreddin, 2009]

Obviously, the rules will be easier to interpret if they are defined by the most influential variables and the system behaviour will be easier to understand as the number of rules is getting smaller [Serge, 2001].

Due to some complex relationships in the data, nonlinear artificial neural network estimator has been used for higher accuracy, however the design and weight assignment issue in ANN makes it lag behind. The network architecture and the parameters used in the network such as the No. of layers and nodes used, inputs, learning rates etc. control the weights of the network [Kampouropoulos et al., 2014]. As far as weight assignment is concerned, it has a direct influence over the performance of network. A comprehensive study in which a global
methodology for ANN has been presented wherein sensitivity analysis on their parameters has been performed [Samanta, 2004].

So, the next step was to use an optimization method in the network so as to improve the performance. Researchers combined ANN with GA and simulated annealing Lavenberg – Marquardt to improve the overall performance [Tahmasebi et al., 2012]. GA was used to optimize the parameters and design of ANN, by which researchers proved a better performance in ANN. Apart from this, researchers have used Fuzzy logic with ANN [Samanta, 2009] [Chatterjee, 2010] [Bandopadhayay, 2009].

[Cheng and Agterberg 1999] gave the concept of fuzzy weights so as to focus on conceptual and empirical aspects of data. Further to obtain the fuzzy posterior probability, the data-based conditional probability was combined with Knowledge based membership values in hybrid fuzzy weights of evidence model. However the issue in Fuzzy logic and the lack of the knowledge lead the researchers to combine FL with ANN.

Researchers like Jang in 1993 tried to combine the three softcomputing intelligent techniques FL, GA and NN to have a more accurate predictor. Neural Network having the learning and high computational ability and being the dynamic estimator is hybridized with the explanatory nature of rules in FIS (membership functions) along with GA that optimally selects the right parameters so as to solve the uncertainties in the data and reach best performance.

The aim is to develop a powerful tool for diagnosis based on Genetic-neuro–fuzzy system (GNFS) wherein the appropriate parameters are selected optimally and has less error and better accuracy.

7.3. Feature or Attribute Selection

Feature selection process is used in machine learning algorithms to increase the predictability or classification and clustering accuracy [Peng et al., 2004]. It is considered to be a fundamental tool for processing high dimensional data. This NP-Complete problem of selecting strong features needs an exhaustive search technique to filter out relevant feature subset. Two Important approaches for feature selection are [Kumari et al., 2011]:
1. **Forward selection** wherein the technique starts with an empty feature set and then keeps adding up features to the set, one by one, focusing on the minimal error features.

2. **Backward selection** that starts with taking up all the features or attributes in the feature set and removing the features, one by one, focusing on maximal error features.

Feature selection process is broadly classified as filter and wrapper method. In filter method the features are ranked on some statistical criteria and the procedure for subset selection does not depend on any learning or preprocessing step whereas in wrapper method a certain subset is selection out of possible feature subset based on some learning algorithm and evaluated accordingly. However, Wrapper method has been recommended by researchers for better validation of the results [Kumari and Swarnkar, 2011]. Heuristic approaches are more efficient and accurate for feature selection in high dimensional microarray dataset.

Fig 7.1 represents the feature selection process in Filter and wrapper method. One of the important filter methods for attribute selection is the Best Fit. This technique starts with no attribute followed by forward search and uses the best first strategy to navigate attribute subset [Svetlana, 2004]. In classification, feature selection reduces the curse of dimensionality and helps in model learning, minimizes computation cost and helps in better prediction i.e. better accuracy.
Figure 7.2 below shows Genetic Algorithm as a randomized wrapper technique in the feature selection Taxonomy.
7.4. Genetic Algorithm as an Adaptive Feature selection process

Genetic algorithm is an optimization strategy that searches for the best optimal solution to a problem. In order to reduce number of features to a manageable size, specific strategies have been developed and applied on the knowledge base [Dom, 1989]. This algorithm is a valuable tool that has been used for feature subset selection in an ANN designed for classification and knowledge discovery [Yang and Vasant, 1998]. Such filtered subsets of features or attributes or variables obtained from genetic algorithms prove more efficient and strong in predicting the class of data instance than those obtained from other feature selectors (classical ones) [Leardi et al, 1992]. Mathematical experiments have been performed for comparing the traditional models of feature selection and genetic algorithm to prove that the later approach is better for feature selection process [Babatunde et al., 2014]

Generic heuristics techniques like greedy algorithm can be applied for selecting efficient feature subset in case of unavailable or expensive extraction of knowledge [Kitler, 1978]. However, experiments have been performed to compare greedy search and genetic algorithm. Although greedy search algorithm proved to be very fast, but that may produce poor results. As far as GA-based method is concerned, it can provide a more robust solution, although the computational complexity may increase [Vafaie and Iman, 1994].

In medical field also, genetic algorithm has been used in combination with ANN classifier (LM-NN) designed to identify Abnormal Cardiac beat so as to detect heart ailments (Bundle Branch Block). Genetic algorithm helped in finding the best optimized features using the ECG data taking into consideration the changes in the ECG data [Kora and Krishna, 2016]. So, Genetic Algorithm provides a powerful means of finding near-optimal subsets of features from large sets.

Fig 7.3 below represents the parameter optimization process that starts with an input parameter set and applies genetic operators over it to select a subset of parameter set that actually predict the output class.
Fig 7.3. Parameter optimization process

7.5. Hybrid model of Genetic Algorithm as a feature selector and ANN as Evaluator:

Various univariate techniques like filter, wrapper and embedded have some drawbacks which can be resolved by focusing on population based randomized technique of Genetic Algorithm along with classifiers to provide accurate solution [Yvan,2007][Li,T.,2004][Petricoin,2002].

Genetic Algorithm provides the different combinatorial set of features among which the best combination to achieve optimal accuracy is achieved. It has proved to be a very adaptive and efficient method of feature selection wherein the number of features is being reduced to an optimal number.

The major issue in genetic algorithm is the fitness function. This function is used to check how efficiently the selected subset of features (model generated solution) still represents the whole dataset. A classifier is applied every time a new combination of attributes is prepared and check the extent to how well that combination or feature subset is performing. This is done by comparing the error calculated by classifier for every feature subset i.e. the less the error, the better shall be the selected subset.

The algorithm ranks the function values in an order and passes those solution to next generation that has the best scores to produce those solution that’ll prove to be better. Feature selection is
one of the important techniques that can be used in many application areas especially in the bioinformatics domain. The technique for feature selection using Genetic algorithm provides a filtered set for further analysis of useful information. Here, we shall focus on the supervised learning technique of Classification, wherein the class labels are already known.

As discussed in chapter 3, the learning mechanism of ANN classifier makes it an adaptive nonlinear processing model that has the characteristic of self-organization, real-time learning and self-adaptation [McCulloch and Pitts, 1943] [Ding et al. 2013]. The learning mechanism is supervised when learning is based on gradient descent method wherein direct comparison between expected output and observed or calculated output or unsupervised when the learning is based on correlation of the input data.

The stochastic process of genetic algorithm generates population randomly and searches the solution by applying consecutive operations of crossover and mutation. So, evolutionary process of Genetics is based on the basic principles of natural selection wherein basic operations i.e. selection, crossover and mutation are applied on the individuals in the population that go through continuous competition and exchange of information with each other [Ding et al. 2013] [Shabia et al, 2016]. Till now genetic algorithm has proved the best tool for feature selection due to its optimization power and heuristic feature.

Here, in our case study, we shall use genetic algorithm as feature selection technique and ANN based classification error as a fitness function so as to obtain optimal accuracy. As the high dimensional dataset is supplied we need to select strong features that would best predict the class.

The hybrid model of GA and Classifier or an evaluator creates a wrapper model for feature subset selection. Since the evaluations in chapter 3 came up with accurate results in case of neural network, so we shall use neural network as an evaluator to select the best possible feature set. Figure 7.4 below presents the hybrid model that helps in finding the optimal solution wherein ANN is being used as an evaluator that assures the accuracy in the system. The figure shows general flow of feature selection process using Genetic Algorithm and Neural Network.
The general technique of GA in combination with neural network flows the steps listed out in Algorithm 7.1. below:
Algorithm: 7.1. **Algorithm behind feature selection using GA-NN model**

The Algorithm begins with the random creation of population and followed by the fitness evaluation that calculates the rank of each chromosome. The chromosome is the set of binary values wherein a ‘1’ denotes presence of the gene or feature i.e. the particular feature has been selected and a ‘0’ denotes the absence of gene or feature i.e. feature not to be considered in the chromosome or feature-set evaluation.

**Start**

**Step-1:** Create initial random population consisting of individuals represented by x.

**Step-2:** Divide dataset into train and test data.

**Step-3:** Select the set of features by setting ‘1’ for presence of feature and ‘0’ for absence.

**Step-4:** Evaluate feature set using ANN classifier.

   Fitness evaluation by calculating cost value using error calculated by classifier.

**Step-5:** Selection of parent according to fitness value // discussed in chapter GA

   a) Search best cost value, position of which shall participate in next generation

   b) Search worst cost value, position of which shall not participate in next generation.

   **Step-6:** Acceptable Solution found i.e. Termination/Stopping Criteria satisfied?

   If Yes then Exit

   Else Search qualifying feature sets for further processing

   a. The Recombination or crossover to generate offspring

   b. Mutation of the offspring

   c. Replace population by offspring to form the next generation.

   d. Goto 3

Thus, creation of new feature sets to participate in next generation
7.5.1. Selection of Classifier or Evaluator

Mean Squared Error (MSE), a statistical evaluation measure or estimator, is the mean or average of the squares of the errors or deviations wherein the error is the difference between the expected and observed value.

\[
\text{MSE} = \frac{1}{n} \sum_{i}^{n} (T_i - O_i)
\]

This measure can help in estimating more accurate models. Another measure is the Root Mean Square which is calculated by square rooting MSE as follows:

\[
\text{RMSE} = \sqrt{(MSE)}
\]

A classifier identifies the category or class to which an observation or instance belongs. This depends on the training, validation and testing set provided to the classifier. The best classifier chosen for the dataset is Multilayer Perceptron wherein the calculated RMSE is lowest than other classifiers applied. The lower the error, the better the classifier.

7.5.2. Fitness or Cost Function:

The default evaluation method is Mean Squared Error (MSE), a statistical evaluation measure or estimator, is the mean or average of the squares of the errors or deviations wherein the error is the difference between the expected and observed value.

\[
E = \frac{1}{n} \sum_{i}^{n} (T_i - O_i)
\]

Further the cost or fitness value is calculated using the equation below:

Fitness or Cost = E* (1 + P + SFR)

Where,

E is the mean error, Parameter ‘P’ is the probability of the best chromosome or individuals selected compared to the average probability of selection of all chromosomes or individuals and SFR is the selected feature ratio.
7.6. New Approach Crossover operation for Feature Selection

One of the issues in Genetic algorithm is the careful selection and development of genetic operators [Yadav et al., 2011]. Focusing on the genetic operators, there are some important methods that have been used so far in order to perform the crossover between the feature set or parameter set or chromosome. The process of crossover involves combining two chromosomes to produce in order to produce an offspring solution chromosome. The basic reason behind performing this operation is to search a new chromosome that would have the better cost values than the parent chromosomes taking the best characteristic from each parent i.e. the performance might be better than the previous chromosomes.

As discussed in chapter 4, crossover involved both parental chromosomes usually split at a randomly determined crossover point(s). Some important and basic crossover techniques are single or one point crossover, double point crossover and uniform arithmetic crossover. Single Point Crossover technique divides the first chromosome in two parts depending upon single point selected and appends its 1st part with 2nd part of other divided chromosome [Reeves and Rome, 2003] [Kellegoz et al. 2008]. Apart from this, techniques have been provided that use multiple cut points to perform the crossover operation so that the problem space can be searched more in depth. In double or two point crossover, the chromosome is divided into three sections using two crossover cut points [Booker, 1987] [Kaya et al., 2011] [Shabia et al., 2016].

Uniform Arithmetic Crossover (UAC) technique implements the concept of Intermediate recombination that involves copying of samples from the two participating parents depending on alpha value. Uniform Arithmetic point crossover takes the sample from 1st parent chromosome if corresponding alpha value (randomly generated) is 1 and or it takes sample from 2nd parent if the corresponding alpha value is ‘0’ [de garis, 1990] i.e. new alpha value (0 / 1) is chosen for each variable.

Uniform Arithmetic Crossover (UAC) where two offspring feature sets or parameter sets are arithmetically produced are calculated as:

New Chromosome_1= alpha* 1st parent + (1-alpha) * 2nd parent

New Chromosome_2 = alpha* 2nd parent + (1-alpha) * 1st parent
The effects of the crossover function on the performance of genetic algorithm have been analyzed [Spears, 1992] and the results concluded uniform arithmetic crossover function as an outperformer among all. Further one or single point crossover has also proved better than double point crossover operation. Experimental study of crossover functions, i.e. Single-crossover, two point crossover, uniform arithmetic crossover concluded that single point crossover function with best performance. Further, uniform function was found to be the next best [Jeorge, 2013] [Vekaria and Clack, 1998]. Apart from this two point operator has been considered to be least disruptive and uniform crossover as most disruptive [De Jong and Spears, 1992] [de garis, 1990 (a,b)] [Hong and Mrinal,2006] [Sivanandam, 2007(b)] [Hartono and Erianto, 2015] [Michalewicz, 1998] [Kaelo and Ali, 2007]. Apart from this, experiments performed over dataset of 50 instances using some of the important crossover functions have proved single point to be better crossover function followed by the uniform crossover [Mendes, 2013]

Genetic Algorithm is a randomised wrapper method that avoids the issue of getting stuck in local minima and interacts with an evaluator or classifier. It filters out the strong features that best predict the class of an instance.

When feature selection is applied over the gene expression data, we call that as gene selection. [Ang et al., 2016] lists out various issues in the gene expression data like irrelevant and noisy data that leads to the problematic analysis of data.

It is the process of obtaining a subset of optimal features for use in algorithm. Sometimes the high dimensional feature data set may affect the performance of the system due to some redundant or non-informative or irrelevant features or factors (often called as noise) [Kim, 2000]. To avoid such inefficiency and poor performance, we try to find out the best and smallest that represent the whole database even after removing some of the useless attributes or features. So, Feature Selector (Fs) acts as an operator to map m dimensional feature set to n dimensional feature set. The process is being used in order to get the filtered dataset with reduced dimension that improves the efficiency of the algorithm. The main aim of feature selection is to improve the performance of model by providing optimal subset of relevant features possible thus proving better prediction through any supervised classifier specified. As far as unsupervised technique is concerned, it can search better clusters using clustering technique. It helps in avoiding problem of overfitting and provide faster and cost efficient models [Yvan, 2007] [Daelemans, 2003]
Feature subset generation method has been classified under three main categories [Nianyi Chen, 2004] i.e. Exhaustive Search method which is suitable for the small datasets as it is time consuming, however searches the best feature subset whereas Heuristic Search method and Non-deterministic Search method usually applied over larger dataset e.g. evolutionary algorithm etc. (esp. genetic algorithm) to determine the feature subset.

To resolve the issue of operator selection, as discussed above, we came up with novel idea of ensemble function which is a hybrid of Single-Point Crossover (SPC), Double-Point Crossover (DPC) and Uniform Arithmetic Crossover (UAC) operation between Parent (I) and Parent (J) wherein each feature set (Fs) is parent input to crossover function and different Crossover Points (Cp) are set for the three different functions. The technique performs the three operations and calculates the respective OFVs (Objective function values). Each operation results in two feature solution sets, thus the whole process producing six different solution sets. Their OFVs are compared and the minimum two of the six along with their respective feature set solutions (FSSs) are selected for the further processing so as to participate in the next generation.

Principle of work: Choosing all 3 techniques to participate in crossover emphasizes the effectiveness of crossover
7.6.1. Novel Idea of Ensemble Crossover operation- (ECO)

A series of experiments have been conducted to compare the important traditional feature selection algorithms and the new proposed one with modified crossover function i.e. SPC, DPC, UAC and our novel ECO technique, thus verifying the novel method.

The algorithm views the whole dataset in two parts input set and target set. The various factors used for experimental design are shown in the table 7.1 below:

<table>
<thead>
<tr>
<th>Factors in experimental design of GA</th>
<th>Experimental Value Setting in Feature Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>10</td>
</tr>
<tr>
<td>Generation Size or Maximum number of iterations.</td>
<td>10</td>
</tr>
<tr>
<td>Selection Function</td>
<td>Roulette wheel function</td>
</tr>
<tr>
<td>Selection Pressure (β)</td>
<td>8</td>
</tr>
<tr>
<td>Crossover Percentage</td>
<td>0.7</td>
</tr>
<tr>
<td>Mutation Percentage</td>
<td>0.3</td>
</tr>
<tr>
<td>Mutation Rate</td>
<td>0.1</td>
</tr>
<tr>
<td>Classifier Evaluation Runs (r)</td>
<td>3</td>
</tr>
</tbody>
</table>

To create a feature set, each feature is associated ‘S’ value. This value is set randomly with ‘0’ or ‘1’ indicating absence or presence of particular feature respectively.

S = 1 indicates presence of feature/attribute/variable

S = 0; indicates absence of feature/attribute/variable

After feature set is obtained, features with value S ≠ 0 are used to create solution feature set that is further evaluated for fitness.

Calculate Objective Fitness Function Value or Cost Value (OFV):

\[ FV = (1 + \beta \times FSR) \times \sum_{i} Error \]

Where, Feature Selection Ratio: \[ FSR = \frac{\text{No.of features selected}}{\text{Total No.of features in Input Set}} \]
The classifier used is Artificial Neural Network and its parameter setting is shown in the table 7.2 below:

Table 7.2. Parameter Setting of NN Classifier

<table>
<thead>
<tr>
<th>ANN Classifier Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training data percentage</td>
<td>70</td>
</tr>
<tr>
<td>Validation data percentage</td>
<td>15</td>
</tr>
<tr>
<td>Testing data percentage</td>
<td>15</td>
</tr>
<tr>
<td>Training Epochs</td>
<td>50</td>
</tr>
<tr>
<td>Performance function</td>
<td>Mean Square Error (MSE)</td>
</tr>
</tbody>
</table>
Algorithm 7.2: Best Cost Genetic Algorithm using ECO

Traditionally, selecting a crossover function among various options available was done randomly or design specified. The basic principle of survival of the fittest should be applied on internal processing as well so that we can get the offspring (Solution set) that best fits to the environment. Algorithm above provides the novel idea for that.

**Input:**

- Input $F_{S_I}[1:end]$ and $F_{S_J}[1:end]$
- Parameters: $C_p$, $C_{p_{min}}$, $C_{p_{max}}$, alpha \hspace{1cm} // using randi ( ) function in Matlab

**Process:**

//SPC operation

FSS_1: $F_{S_I}[1 \text{ to } C_p]$; $F_{S_J}[C_p +1 \text{ to end}]$

FSS_2: $F_{S_J}[1 \text{ to } C_p]$; $F_{S_I}[C_p +1 \text{ to end}]$

// DPC operation:

FSS_3: $F_{S_I}[1 \text{ to } C_{p_{min}}]$; $F_{S_J}[C_{p_{min}} +1 \text{ to } C_{p_{max}}]$; $F_{S_I}[C_{p_{max}} +1 \text{ to end}]$

FSS_4: $F_{S_J}[1 \text{ to } C_{p_{min}}]$; $F_{S_I}[C_{p_{min}} +1 \text{ to } C_{p_{max}}]$; $F_{S_J}[C_{p_{max}} +1 \text{ to end}]$

// UAC operation

FSS_5: alpha * $F_{S_I} + (1\text{-alpha}) * F_{S_J}$

FSS_6: alpha * $F_{S_J} + (1\text{-alpha}) * F_{S_I}$

// Cost Evaluation

Calculate and Record the Arrangements of FSS and respective Cost Values.

Compare the Costs of 1$^{st}$, 3$^{rd}$ & 5$^{th}$ FSS and select FSS with min cost as Arrng_1

Compare the Costs of 2$^{nd}$, 4$^{th}$ & 6$^{th}$ FSS and select FSS with min cost as Arrng_2

**Output:**

Two Features Arrangements: \hspace{1cm} // applying vector min function over the cost array

Arrng _1 and Arrng _2 as the qualified best feature set solutions (FSS) for participation in Next generation
Search for the Best and Worst Solutions depending on the OFV and on the basis of these values determine the individuals for the further processing for the participation in the next generation.

Select the parents for Crossover using Roulette Wheel function. An important factor that we shall focus on is the selection of Crossover function which is considered to be one of the major issues in genetic algorithm. Some of the important crossover functions are Single point function, Double point function and Uniform arithmetic crossover. Usually the desired crossover function is specified in the design only or roulette wheel selection function is used to select randomly of the functions. Researchers have worked on the comparisons of the functions, however they did not conclude to a particular best crossover function. Fig 7.5 shows the process flow in ECO operation.

Fig 7.5. Ensemble Crossover operation in Genetic Algorithm
7.6.2. Crossover followed by Mutation

The next step is the process of mutation wherein the major concern is the value after mutation of samples. For that we need to find out the number of mutations to be performed in a feature set and the location of the mutation in the set. This is calculated on the basis of the specified mutation rate.

\[
\text{No. of Mutations in a Sample Set} = \text{Mutation rate} \times \text{No. of Samples}
\]

The location or position of mutation is determined randomly (Using randsample( ) function) in Matlab.

For feature Selection, wherein the ‘S’ value is ‘0’ or ‘1’, the mutation value is calculated as:

\[
\text{New Mut}_{\text{Value}} = 1 - \text{Old}_{\text{Value}}
\]

Sort the cost values (OFCs) of the population created by fittest population, mutated population and Crossover population and select Best Cost (min of the cost values) and its respective arrangement. The features with associated ‘1’ are the features that best predict the target value.

7.7. Fusion of Neuro Fuzzy and Best Cost Genetic Algorithm (BCGA) With ECO.

The complexity of training these above parameters is the main issue in the research field. To resolve such a problem, we started with the basic training algorithm that is based on gradient descent, however, chain rule might lead to trap in local minimum and result in unacceptable prediction error during training and testing of the network.

So, to reduce the training error and increase the training accuracy many efforts have been made such as improving accuracy of fuzzy space partition in premise parameter identification [Li et al., 2009]. Based on this fuzzy space partition and subsequent structure (premise) parameters, the least square method identifies the consequent (conclusion) parameter. Actually in the premise parameter identification, partitioning the fuzzy space is performed so as to get the center and width of fuzzy set. This is done by chaos optimization strategy which is able to avoid trapping into local minimum during the optimization of the clustering objective function. One of the optimization tools i.e. Genetic Algorithm which is used with Anfis for training can increase the performance of training accuracy and reduce the prediction error [Ho et al, 2009]
Further, it has been also used [Kim, 2000] with Anfis for stock market forecasting.

Recently, Genetic Algorithm has been used to tune up the parameters of FIS taking into consideration the ANFIS architecture. This adaptive network is known by ANFIS-Genetic Algorithm (AGA). The difference between AGA and ANFIS lies in the process of normalization and adaptation. As far as adaptation is concerned, ANFIS adapts or changes the weights, but in case of AGA the weights are kept constant and membership values act as the weights in AGA. The parameters are adapted using the principle of the evolution theory of natural selection. [Varnamkhasti, 2011][Sarkheyli et al., 2015]. The algorithm of Best Cost Genetic Algorithm (BCGA) acts in the similar way as AGA and uses genetic algorithm with ECO for parameter optimization.

Experiments have been performed to prove that the proposed novel method of Neural Network with BCGA-ECO works much better than the traditional one such as Genetic Neural Network (GANN).

7.8. Calculation of Feature Count Value (FCV) – A Novel Idea

In the evolutionary process of genetic algorithm, the features or attributes that survived in all or maximum generations are the strongest features and thus play an important role in the accurate output prediction of output. Using genetic algorithm for feature selection provides the best solution featureset that is evaluated through fitness function (fitness value or cost value). The less the cost value the more fitness lies in the feature set.

A Novel Idea of Feature Count Value (FCV) has been introduced that can search the strongest features that greatly affect the output of the classifier. The number of times a particular feature is being selected is an evidence that the particular feature is that much effective and has that much of impact over the target value, i.e.

\[
\text{No. of Times of Occurrence (Count) } \propto \text{ Strength in Prediction of Target Value}
\]

Actually, the process uses a specified factor i.e. ‘maximum iterations’ to perform the genetic operations, each iteration or generation ending up with the best solution. We need to keep track of the count that represents the frequency of a feature being selected (filtered out) in the best
solution featuresets. The more the feature frequency (FCV), the stronger is the associated feature known to be.

The best solution featuresets are recorded in a structure along with the respective cost values. These feature sets contain the samples or S values as ‘0’ indicating the absence of a particular feature and ‘1’ indicating the presence of a feature set. So, to search the features that are stronger we need to check their respective S values.

If the S value is ‘1’, then it adds up to the FCV for the associated feature else the FCV remains unchanged. The Features with maximum FCV will represent the strong features or attributes in determining or predicting the output of the System.

Each generation keeps track of the count value for every attribute that participated in the best cost feature set. e.g. the one which is selected in all the generations is the most efficient one and should be focused on as it can really affect the output prediction.

FCV can help the medical field to search the attributes that need to be focussed on that can turn help in deciding the proper medication of appropriate intensity and type. This can help in increasing the survival rate of patients suffering from severe diseases and in predicting the future chance of any occurrence of unforeseen or abnormal condition.

Case Studies below calculates the FCV and accordingly finds out the strong attributes for further consideration.
Algorithm 7.3. : Algorithm for calculating Feature Count Value (FCV)

The number of times a particular feature is being selected is an evidence that the particular feature is that much effective and has that much of impact over the target value, i.e.

\[ \text{No. of Times of Occurrence (Count) } \propto \text{ Strength in Prediction of Target Value} \]

Input:
- Input Variable Set FS\(_1\)
- Specify Parameters
- Specify Max No. of Generations

Process:
1. Perform Selection operation
2. Set value '1' for presence of variable/attribute
   value '0' for absence of variable/attribute
3. Create new Feature Set FS\(_2\) with all position values as ‘1’
4. Perform Crossover Operation using the new set FS\(_2\)
5. Perform Mutation Operation
6. Compute the best cost value for each respective combination
7. Record the FCV for each variable in the combination
8. If max generation reached exit
   else
     Goto step 1
9. Compute Total FCV for each variable of FS\(_1\)

Output:
- Strong Feature Set FS\(_3\) // The features with highest FCV;
- The strong feature are the ones that can really affect the class prediction of instance.
7.9. Optimization algorithm in Search of efficient biomarkers

Experiments: Results and Discussion

In case of statistics, once a conceptual model is built (hypothesis), we go for the validation of that hypothesis which leads us to the final acceptance or rejection of the null hypothesis.

In contrast, machine learning works almost in an opposite way wherein the first step does not start with the hypothesis. Rather we just have a data set and we don’t really know what we are looking for. So, here we start by applying the algorithm over the dataset in an attempt to get some interesting knowledge forming the basis of the hypothesis thus the name “Hypothesis discovery”.

Following Case studies in the below sections represent the applications of computational Intelligence that involves the Medical Implementation of Algorithm for Diagnosis.

7.10. Case Study 1: Tennesse dataset

Determining Survival After Surgery of Patients with Pancreatic Cancer

Traditionally, diagnosis depends on identifying some patterns from data that are obtained from human experiences. However this kind of diagnosis is prone to human error and is time consuming [Ahmad, 2015]. So, we need a better solution for overcoming the drawbacks and provide an efficient model. Genetic algorithm is used for attribute selection for better prediction and reduce the dimensionality of selected dataset.

The traditional method for diagnosing the disease relies on human experiences to identify the presence of certain pattern from the database. It is prone to human error, time consuming and labor intensive. Therefore, an evolutionary algorithm shall be used for filtering a feature subset and evaluated using a classifier to find out the best prediction system for diagnostics.

7.10.1. Pancreatic Tumor Dataset Description:

Pancreatic cancer, a disease with poor prognosis and one of the major causes of death in the world, needs to be detected as early as possible [Sheema et al., 2014] [Moschopoulos, 2013]. So our main aim is to find out the factors or features that actually affect the survival of the tumor patient after the surgery. This will help in early detection and treatment so as to increase the
survival rate of the patients [Ahmad, 2015]. The tumor tissues have been stained with muc13 (transmembrane mucin) in the similar way as CA-19 test is done [Sheema et al., 2014]. Fig 7.6 below shows the cancerous cell and its effects in different parts of cell i.e. cytoplasm, membrane and nucleus, using protein expression of MUC13 in the tissue.

Fig 7.6. Protein expression of MUC13 in human pancreatic cancerous samples through Western blotting

The Pancreatic tissue array in Fig 7.7 below, represents 90 cases including the cancerous and adjacent normal tissue and then based on this a dataset is produced containing various attributes related to each sample. Before introducing the dataset to the feature selector, it is preprocessed to avoid any kind of strong relationships between the different attributes.

Fig 7.7. Microarray Panel display containing pancreatic cancer tissue samples.
Dataset Description: ("tennessee.mat")

Source: Health Science Center Laboratory

Creator: Dr. S.Khan & Dr. S. Chauhan,
The University of Tennessee Health Science Center,
Tennessee, USA

Date of Creation: August 2015

Related Information: Study on the survival of patients who had undergone surgery for Pancreatic Cancer. On the basis of survival months after, with threshold set up, the dataset has been preprocessed with instances divided into two classes—Survived and Deceased.

Number of Instances: 74 (i.e. 74 different patients, deceased or survived)

Number of Attributes: 20 + class = 21 attributes

Class Distribution: (2 classes)

Table 7.3. Class table for ‘tennessee’ dataset

<table>
<thead>
<tr>
<th>Class</th>
<th>No. of Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survived</td>
<td>24</td>
</tr>
<tr>
<td>Deceased</td>
<td>50</td>
</tr>
</tbody>
</table>

The experiment uses dataset containing tumor tissue samples taken from 74 patients. The dataset has been obtained from Health Science Centre, Tennessee, and USA. This dataset (74*21) consists of 20 features (factors or attributes), 1 class attribute and 74 instances. The features or attributes have been defined in the table 7.4 below.
Table 7.4. Attribute Description in ‘tennessee’ dataset:

<table>
<thead>
<tr>
<th>S.No</th>
<th>Attribute</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sex :</td>
<td>Whether the patient is female or male.</td>
<td>Nominal</td>
</tr>
<tr>
<td>2</td>
<td>Age:</td>
<td>The age of the tumor patient.</td>
<td>Continuous</td>
</tr>
<tr>
<td>3</td>
<td>Pathology</td>
<td>Type of pancreatic tumor</td>
<td>Nominal</td>
</tr>
<tr>
<td>4</td>
<td>Grade</td>
<td>Degree of aggressiveness of tumor</td>
<td>Nominal</td>
</tr>
<tr>
<td>5</td>
<td>Stage</td>
<td>Extent of tumor spread</td>
<td>Nominal</td>
</tr>
<tr>
<td>6</td>
<td>T_Tumor invading</td>
<td>Whether tumor invades submucosa or muscularis propria, muscularis propria into subserosa or into non-peritonealized pericolic or perirectal tissues or other organs or structures and/or perforate visceral peritoneum.</td>
<td>Nominal</td>
</tr>
<tr>
<td>7</td>
<td>N_lymph nodes</td>
<td>Whether Regional lymph node metastasis or Metastasis in 1 to 3 regional lymph nodes or in 4 or more regional lymph nodes.</td>
<td>Nominal</td>
</tr>
<tr>
<td>8</td>
<td>M_metastasis</td>
<td>Whether Distant Metastasis or not.</td>
<td>Nominal</td>
</tr>
<tr>
<td>9</td>
<td>Memb_S_Int</td>
<td>Intensity value of staining in membrane</td>
<td>Continuous</td>
</tr>
<tr>
<td>10</td>
<td>Memb_Percent</td>
<td>Percentage of tissue stained</td>
<td>Continuous</td>
</tr>
<tr>
<td>11</td>
<td>MembMCS</td>
<td>Mean Composite Score. Memb Intensity* Memb Percentage</td>
<td>Continuous</td>
</tr>
<tr>
<td>12</td>
<td>Cyto__Int</td>
<td>Intensity value of staining in cytoplasm</td>
<td>Continuous</td>
</tr>
<tr>
<td>13</td>
<td>Cyto_Perc</td>
<td>Percentage of tissue stained</td>
<td>Continuous</td>
</tr>
<tr>
<td>14</td>
<td>CytoMCS</td>
<td>Mean Composite Score. Cyto Intensity* Cyto Percentage</td>
<td>Continuous</td>
</tr>
<tr>
<td>15</td>
<td>Nucl_Int</td>
<td>Intensity value of staining in Nucleus</td>
<td>Continuous</td>
</tr>
<tr>
<td>16</td>
<td>Nucl_Perc</td>
<td>Percentage of tissue stained</td>
<td>Continuous</td>
</tr>
<tr>
<td>17</td>
<td>NucMCS</td>
<td>Mean Composite Score. Nucleo Intensity* Nucleo Percentage</td>
<td>Continuous</td>
</tr>
<tr>
<td>18</td>
<td>OI</td>
<td>Overall Intensity</td>
<td>Continuous</td>
</tr>
<tr>
<td>19</td>
<td>OP</td>
<td>Overall Percentage</td>
<td>Continuous</td>
</tr>
<tr>
<td>20</td>
<td>OIxOP</td>
<td>Overall Intensity * Overall Percentage</td>
<td>Continuous</td>
</tr>
<tr>
<td>21</td>
<td>Survival_status</td>
<td>Survived or Deceased after surgery &lt;Class attribute&gt;</td>
<td>Nominal</td>
</tr>
</tbody>
</table>
7.10.2. Feature Selection in ‘tennessee’ dataset using BestFit+NN:

Feature Selection has been performed using one of the traditional methods of filter approach. The filter method of Best Fit algorithm has been applied over the Tennessee dataset for feature extraction. For Evaluation, a Neural Network classifier or a learner has been used to identify the category or class to which an observation or instance belongs. This depends on the training validation and testing set provided to the classifier.

Using feature selector “best fit attribute selector and classifier Neural Network, the dataset has been reduced to just 5 strong features i.e. (Sex, T_Tumor invading, N_Lymph nodes, Nuclear Intensity, Nuclear Percentage) thus making prediction of survival or decease much clearer and accurate. Figure 7.8 below shows the filtered strong attributes for prediction by Best Fit – Six attributes including class attribute.

![Image of filtered attributes](image)

Fig 7.8. Filtered Attributes for prediction by Best Fit

After applying feature selection using ‘Best First’ (BF) Algorithm, further the filtered attributes are evaluated using certain classifiers.

Figure below shows the results of the classifiers used among which Multiperceptron proved the best taking into consideration certain important evaluation measures. The RMSE value obtained
from Best fit +NN is 0.6336. Figure 7.9 below represents the screenshot showing the error calculations from Multiperceptron evaluation with RMSE as 0.6336.

![Screenshot showing the error calculations from Multiperceptron classifier](image.png)

Fig 7.9. Screenshot showing the error calculations from Multiperceptron classifier

7.10.3. Feature Selection in ‘tennessee’ dataset using GA+NN:

The statistical measure focused on is Mean Squared Error (MSE) which is an estimator and is calculated as the mean or average of the squares of the errors or deviations wherein the error is the difference between the expected and observed value. After selecting the classifier we first need to filter out the attribute set depending upon fitness value considered to be heart of Genetic Algorithm. This shall keep the balance between the applied classifier and the solution feature set. Here the fitness value to which we refer is RMSE.

The Neural Network uses the performance function as: `net.performFcn = 'mse'`

The calculated MSE is used to calculate the RMSE to compare it with best fit RMSE value as $RMSE = \sqrt{MSE}$

Further on applying GANN i.e. Genetic algorithm along with the same classifier (NN), the results show that after reducing the features and removing useless and inefficient features the prediction of survival or decease seems to be much clearer and accurate. Figure 7.10 below sows the output values after the feature selection using GANN.
The algorithm extracted eight attributes with the feature ratio of 0.4 and the solution set resulted in mean square error value as 0.0867 with RMSE as 0.2944. The eight attributes selected included the cytoplasm and nucleus MCS values, thus indicating these among the attributes that really can affect the output. Further the iteration resulted in best cost value reduced to 0.1041 (see fig 7.11. below)
The graph above shows the best cost level for 10 iterations. As the iterations increase, the algorithm filters out stronger attributes due to which the prediction cost lessens. At the 10th iteration we got smallest efficient set of attributes with lowest prediction error and cost value.

7.10.4. Comparative Analysis between Best Fit+NN and GANN over Tennessee dataset:

The correct prediction of survival or decease of patient before surgery depends on some of the factors that needs to be filtered out for further analysis. Fig 7.12 below sows the flow of the two techniques that have been applied over the dataset.

Fig 7.12. Filter and Wrapper method used over Tennessee dataset

Table below shows the comparison between the two- best first attribute selector and the genetic algorithm. In both cases, MLP is used for evaluation. It is clear that the wrapper method of Genetic Algorithm (GA) has better performance than the bestfit (BF) filter algorithm. Table 7.5. below shows the comparison between

<table>
<thead>
<tr>
<th>Attribute Selector</th>
<th>Classifier</th>
<th>Root Mean Square Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best First Algorithm</td>
<td>Multilayer Perceptron</td>
<td>0.6336</td>
</tr>
<tr>
<td>Genetic Algorithm</td>
<td>Multilayer Perceptron</td>
<td>0.2944</td>
</tr>
</tbody>
</table>
Experimental results show that GA approach to the feature discretization model outperforms the conventional model of best first attribute selection using the same classifier of NN.

Here, the RMSE calculated using the best first attribute selection and classifier NN is larger than the calculated RMSE using Genetic algorithm and the same classifier. This proves that the correct prediction of survival or disease depends on some of the factors (esp. the membrane and cytoplasm staining) that GA filtered out and should be worked upon for early diagnosis.

Figure 7.13 represents a graph showing the features selected by the two respective algorithms. As we can see nuclear intensity and T_inv being the strongest attributes in prediction of survival of the patient as both selected them and the values related to membrane being the poorest and least predicts the survival. By strong attribute, we mean even the less intensity of staining can also help in future prediction. The membrane and cytoplasm staining least predicts the survival as best fit technique selected neither of the two attributes for the prediction. Figure 7.13 below shows the feature based comparison.

![Diagram showing features selected by Genetic Algorithm and Best-Fit](image)

Fig 7.13. Comparison of features selected by GANN and BestFit+NN
7.10.5. Feature Cost Value (FCV) using GANN over Tennessee dataset

Further we have seen that Nuclear_ Mcs outperforms other attributes in prediction of class-Deceased or survived. The machine learning technique used has filtered out some of the strong features among which Nucl_MCS is the strongest one and should be worked upon for early diagnosis. This is evident by the calculated FCV during the process of GANN.

Figure below shows that Nucl_MCS is much stronger in prediction of the class. This means that even the less staining can depict the class of the patient before surgery. Referring to table 7.4 above, Fig 7.14 shows the occurrence of attribute 17 (Nucl_MCS) in all the 10 iterations of Genetic algorithm.

![Image showing the occurrence of attribute 17 (Nucl_MCS) in all the 10 iterations of Genetic algorithm.]

Fig 7.14. Attribute 17 (Nucl_MCS) as strongest survival attribute

7.10.6 Feature Selection using Best Cost Genetic Algorithm (BCGA) with ECO over Tennessee dataset:

Applying GANN i.e. Genetic algorithm along with the same classifier (NN), the results show that after reducing the features and removing useless and inefficient features the prediction of survival or decease seems to be much clearer and accurate. Figure 7.15 below sows the output values after the feature selection using GANN.
The number of features selected by BCGA-ECO are 12 with a feature ratio of 0.6. The efficiency of 12 element feature set is being verified by the Error value (MSE) as 0.07923 as shown in fig above. The RMSE value is 0.2816 which is less than the RMSE value obtained from GANN system above. Further the iteration resulted in best cost value reduced to 0.1031 (see Fig 7.16 below)
Figure above graphs out the best costs obtained in each iteration. The graph has same cost value in ending iterations. The twelve attributes selected included the nucleus and Cytoplasm intensity values, thus indicating these among the attributes that really can affect the output. Figure below lists out the output feature set solution (see Fig 7.17 below)

Fig 7.17: Features extracted by BCGA-ECO represented as numbers

Fig 7.18. below shows the view of binary way of selecting the features in case of BCGA-ECO. A binary solution set is obtained wherein Binary ‘1’ indicates the presence of feature (considering the feature among the feature set solution) and Binary ‘0’ indicating the absence of feature.

Fig 7.18. Binary Feature Selection in BCBA-ECO
7.10.7. Comparative Analysis between GANN and BCGA with ECO.

The attributes selected by Genetic Algorithm best predict the class of the tumor patient. So the major focus for early diagnostics should be on some of the have been filtered out by the genetic algorithm. Fig 7.19 below shows the comparative graph for features extracted from GANN and BCGA-ECO.

![Comparative Graph](image)

Fig 7.19. Comparison between GANN and BCGA over Tennessee dataset

Table 7.6 below compares the evaluation measures resulted from GANN and BCGA-ECO algorithms. The proposed model BCGA-ECO proved to be the better algorithm.

<table>
<thead>
<tr>
<th>Evaluation measure</th>
<th>GA NN</th>
<th>BCGA with ECO (proposed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error (MSE)</td>
<td>0.0876</td>
<td>0.0793</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.2944</td>
<td>0.2816</td>
</tr>
<tr>
<td>Objective/Cost function (z)</td>
<td>0.1041</td>
<td>0.1031</td>
</tr>
<tr>
<td>Features Selected</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>Feature Ratio</td>
<td>0.4000</td>
<td>0.6000</td>
</tr>
</tbody>
</table>
7.10.8. Count value (FCV) using BCGA with ECO

Further, using the technique for more knowledge extracting we check the FCV that were recorded during the whole process of selection, crossover and mutation. The algorithm found two strongest features that really affect the output of an instance. They were having the FCV same as the number of evolutionary iterations specified. Screenshot Fig 7.20 below shows highest value of FCV for attributes ‘Age’ represented by Feature No. 1 and attribute ‘Nucl_perc’ represented by Feature No. 16.

Fig 7.20. Attribute 1(Age) and Attribute 16 (Nucl_perc) as survival attribute

7.10.9. Conclusion of applying BCGA-ECO Algorithm over Tennessee dataset:

The algorithm proved to be better algorithm in terms of error rate and accuracy. The algorithm selected more features than traditional one to increase the accuracy and decrease the error rate.

Apart from evaluation of the algorithm, we could find out some important nugget of knowledge i.e. focus on nuclear intensity value. This was determined by checking the FCV of each feature in the whole evolution process (GA). The feature(s) with highest FCV (the number can be equal to the maximum generations specified) are the strongest one(s) that need to be focused on and utilized for decision making in case of finding the proper medication to the cancer patients.
7.11. Case Study 2: Lung Cancer dataset

**Determining the pathological class of lung cancer patient.**

The data consists of instances each belonging to one of the three types of pathological lung cancers. The attribute or variable Information has been kept hidden for privacy purposes. However, the data is original and the features have been serially numbered for identification.

### 7.11.1. Data Description for Lung Cancer (LC) – lungC.mat

- **Source:** UCI Machine Learning Repository
- **Creator:** Hong, Z.Q., Yang, J.Y and S.Aeberhard
- **Date Created:** May 1992
- **Related Information:** 3 types of pathological lung cancers
- **Number of Instances:** 32
- **Number of Attributes:** 56 +1 class attribute
- **Class Distribution:** (3 classes)

<table>
<thead>
<tr>
<th>Class</th>
<th>No. of Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 7.7 Class Distribution for LungC dataset
7.11.2. Traditional use of Best Fit and Neural Network over lungC data.

Applying the filter method of Best Fit algorithm over the lung cancer dataset for feature extraction results in the reduction of variable or attribute no. from 32 to 10. Further, for evaluation, a Neural Network classifier or a learner has been applied to identify the category or class to which an observation or instance belongs and evaluate the feature set that is filtered out by best fit algorithm. This depends on the training validation and testing set provided to the classifier (see Fig 7.21)

![Fig 7.21. Best Fit attribute selector on lung cancer dataset](image)

Using feature selector “best fit attribute selector and classifier Neural Network, the dataset has been reduced to just 10 features that can help in making the prediction of pathology class much clearer and accurate. The filtered feature set is evaluated by NN classifier that resulted in RMSE value as 0.5174 (see Fig 7.22).
7.11.3. Feature Selection in ‘LungC’ dataset using GANN:

Further on applying GANN i.e. Genetic algorithm along with the same classifier (NN), the results show that after reducing the features and removing useless and inefficient features the prediction of survival or decease seems to be much clearer and accurate. The feature selection resulted in the reduction of no. of features to 28. The best cost value is reduced to value of 0.0865 as shown in the graph below (Fig 7.23)
The above graph is drawn on the basis of the recorded ten best cost values for each iteration (top one being the ‘best cost’) as shown in the screenshot below (Fig 7.24)

![Graph showing ten best cost values](image)

Fig 7.24. Ten ‘best cost values’ by applying GANN over lungC

The evaluation value of MSE after applying NN is 0.0692 and the calculated RMSE value to be focused on is 0.26305. As we can see, GANN proved better feature selector for lungC dataset as compared to the traditional best fit filter algorithm that resulted in RMSE value of 0.5174.

**Search for Strong Attribute(s) using FCV:**

No attribute definitions have been given in the UCI depository for lung cancer dataset. Serializing the attribute names, excluding first attribute taken as class attribute, some of the strong features with FCV highest are listed below:

| 10 | 11 | 12 | 23 | 34 | 41 | 46 | 49 |
7.11.4. Best Cost Genetic Algorithm BCGA with ECO using LungC data:

Applying BCGA-ECO i.e. Best Cost Genetic Algorithm with ECO algorithm over LungC dataset, the results show the reduction of feature number to 32. The efficiency of 32 element feature set is being verified by the Error value (MSE) as 0.0437. The RMSE value is 0.2090 which is less than the RMSE value obtained from GANN system above. The iteration resulted in best cost value reduced to 0.0561 (see Fig 7.25).

Fig 7.25. Best Cost value vs iteration No. in BCGA with ECO over LungC dataset

Graph in Fig 7.25 above shows the best cost values against the iteration number. The above graph is drawn on the basis of the recorded ten best cost values for each iteration as shown in the screenshot below (Fig 7.26).

Fig 7.26. Ten ‘best cost values’ by applying GANN over lungC
Search for Strong Attribute(s) using FCV:

No attribute definition have been given in the UCI depository. Serializing the attribute names, excluding first attribute taken as class attribute, some of the strong features with FCV highest are listed below:

| 9 | 10 | 11 | 12 | 18 | 19 | 20 | 23 | 41 | 46 | 48 | 49 | 53 |

7.11.5. Comparative Analysis of BF+NN, GANN and BCGA with ECO

The basic principle of feature selection is that the No. of features selected should not be small enough or large enough to increase the error and decrease the accuracy. The BCGA with ECO helps in maintaining the efficient number of features that would increase accuracy and decrease the error. Table 7.8. Below shows the Comparative analysis between the proposed model and the traditional algorithms.

<table>
<thead>
<tr>
<th>Evaluation Measures</th>
<th>Best Fit+NN</th>
<th>GANN</th>
<th>BCGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error (E)</td>
<td>0.2659</td>
<td>0.0692</td>
<td>0.0437</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.5174</td>
<td>0.2630</td>
<td>0.2090</td>
</tr>
<tr>
<td>Features Selected</td>
<td>10</td>
<td>28</td>
<td>32</td>
</tr>
<tr>
<td>Cost Value</td>
<td>-</td>
<td>0.0865</td>
<td>0.0561</td>
</tr>
</tbody>
</table>

After applying various algorithms over lung cancer dataset for feature selection, the algorithm of BCGA with ECO proved to be better algorithm in terms of error rate and accuracy. The features should not be small enough or large enough to disturb the prediction accuracy. The proposed algorithm of BCGA selected more features than traditional one to increase the accuracy and decrease the error rate.
7.12 Case Study 3: Saliva Pancreatic Cancer

**Recognizing patients as cancerous or healthy: Saliva Pancreatic Cancer**

Pancreatic cancer, a disease with poor prognosis and one of the major causes of death in the world, needs an early and efficient diagnostic system [Khan et al., 2014][Moschopoulos, 2013][Zhang et al., 2010]. So our main aim is to find out the factors or features that actually affect or predict the survival of the tumor patient after the surgery. This will help in early detection and treatment so as to increase the survival rate of the patients [Ahmad et al., 2015].

**7.12.1. Dataset Description:**

The dataset is based on analysis of saliva supernatant from pancreatic cancer patients and healthy subjects. Oral fluid (saliva) meets the demand for non-invasive, accessible, and highly efficient diagnostic medium. Results provide insight into salivary biomarkers for detection of pancreatic cancer.

The experiment uses a salivary analytical dataset that consists of evaluated performance of gene features identified by gene symbol and reference transcript ID. It uses Affymetrix Human Genome U133 Plus 2.0 whole genome array so as to discover altered gene expression in saliva supernatant. The detection of salivary biomarkers can help in early diagnosis of pancreatic cancer specifically without the complication of chronic pancreatitis. The freely available dataset has been taken from Gene Expression Omnibus (GEO) is a database repository and can be refereed for more information- GDS4100.

The total gene samples taken from pancreatic cancer patients and from healthy subjects are 24 with 12 healthy tissues and 12 tumor tissues. The dataset (77x24) has been provided in two separate matrixes:
Salivary Analytical Dataset (SAD)- salpan.mat

Source: Gene Expression Omnibus (GEO) is a database repository
Date of publish: December, 2008
Related Information: Affymetrix Human Genome U133 Plus 2.0 0 whole genome Array
Number of Instances: 24 (Sample count)
Number of Attributes: 76 + class
Class Distribution: (2 classes)

Table 7.9. Class distribution in Salpan

<table>
<thead>
<tr>
<th>Class</th>
<th>Class Code</th>
<th>No. of Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tumor</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>Healthy</td>
<td>0</td>
<td>12</td>
</tr>
</tbody>
</table>

7.12.2. Feature Selection using GANN (Traditional Genetic Algorithm with NN Classifier) over ‘Salpan’ dataset:

We first need to filter out the attribute set depending upon fitness value which is considered to be heart of Genetic Algorithm. This shall keep the balance between the applied ANN classifier and the solution feature set. The experiment goes through an iterative process wherein each iteration results in a biomarker set that is being evaluated by the classifier. Each iteration provides the best cost value for the respective gene set selected. Fig 7.27 below shows the best cost value reduction after applying GANN over the Salivary Analytical Dataset.

Fig 7.27. Best Cost Value against each iteration using GANN over Salpan dataset
Fig 7.24. above shows the screenshot of simulator calculating the best cost at 10th iteration as 0.0249 with 76 attributes or gene set reduced to 40. It shows the graph for best cost value for each iteration. The evaluation of feature set resulted in MSE as 0.0197 and RMSE as 0.1403. The above graph is drawn on the basis of the recorded ten best cost values for each iteration as shown in the screenshot below (Fig 7.28).

<table>
<thead>
<tr>
<th>Fields</th>
<th>Position</th>
<th>Cost</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1x76 double</td>
<td>0.0249</td>
<td>1x1 struct</td>
</tr>
<tr>
<td>2</td>
<td>1x76 double</td>
<td>0.0449</td>
<td>1x1 struct</td>
</tr>
<tr>
<td>3</td>
<td>1x76 double</td>
<td>0.0482</td>
<td>1x1 struct</td>
</tr>
<tr>
<td>4</td>
<td>1x76 double</td>
<td>0.0505</td>
<td>1x1 struct</td>
</tr>
<tr>
<td>5</td>
<td>1x76 double</td>
<td>0.0510</td>
<td>1x1 struct</td>
</tr>
<tr>
<td>6</td>
<td>1x76 double</td>
<td>0.0520</td>
<td>1x1 struct</td>
</tr>
<tr>
<td>7</td>
<td>1x76 double</td>
<td>0.0530</td>
<td>1x1 struct</td>
</tr>
<tr>
<td>8</td>
<td>1x76 double</td>
<td>0.0537</td>
<td>1x1 struct</td>
</tr>
<tr>
<td>9</td>
<td>1x76 double</td>
<td>0.0564</td>
<td>1x1 struct</td>
</tr>
<tr>
<td>10</td>
<td>1x76 double</td>
<td>0.0587</td>
<td>1x1 struct</td>
</tr>
</tbody>
</table>

Fig 7.28. Ten ‘best cost values’ by applying GANN over Salpan dataset

Further, the Experimental results show that GA makes some of the gene features to outperform other genes or in other words we say the attributes or genes selected by Genetic Algorithm best predict or define the class of the patient i.e. either healthy or tumorous.

So the major focus for early diagnostics should be on some of the have been filtered out by the genetic algorithm. Figure below shows the graph fir the best costs obtained in each iteration. The graph has same cost value in ending iterations.

**Search for Strong Attribute(s) using FCV:**

Genetic algorithm is a wrapper method of feature selection process that works on the principle of “survival of the fittest”. This concept can help in binary feature selection wherein the selected subset of features satisfy the fitness value specified and further participate in the next generation.. The implementation of GA along with classifier ANN filtered out some gene features that act as biomarkers and can strongly predict the class of the subject (patient) i.e. whether the subject is healthy or tumorous. This listed out gene set could be worked upon for early diagnosis of pancreatic tumor. In table below, the experiment lists out set of gene symbols
on the basis of highest FCV that strongly predict the class of the subject (patient) i.e. whether the subject is healthy or cancerous.

Table 7.10 below lists out gene features that strongly predict the class of subject (cancerous or healthy patient).

Table 7.10. List of gene features using GANN on Salpan dataset

<table>
<thead>
<tr>
<th>Gene Symbol</th>
<th>Gene Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR1</td>
<td>discoidin domain receptor tyrosine kinase 1</td>
</tr>
<tr>
<td>HSPA6</td>
<td>heat shock 70kDa protein 6 (HSP70B’)</td>
</tr>
<tr>
<td>PAX8</td>
<td>paired box 8</td>
</tr>
<tr>
<td>CCL5</td>
<td>chemokine (C-C motif) ligand 5</td>
</tr>
<tr>
<td>SCARB1</td>
<td>scavenger receptor class B, member 1</td>
</tr>
<tr>
<td>TTLL12</td>
<td>tubulin tyrosine ligase-like family, member 12</td>
</tr>
<tr>
<td>WFDC2</td>
<td>WAP four-disulfide core domain 2</td>
</tr>
<tr>
<td>WFDC2</td>
<td>WAP four-disulfide core domain 2</td>
</tr>
<tr>
<td>MAPK1</td>
<td>mitogen-activated protein kinase 1</td>
</tr>
<tr>
<td>DPM1</td>
<td>dolichyl-phosphate mannosyltransferase polypeptide 1, catalytic subunit</td>
</tr>
</tbody>
</table>
7.12.3. Feature Selection using Best Cost Genetic Algorithm BCGA- ECO over ‘Salpan’ dataset

Applying BCGA-ECO i.e. Best Cost Genetic Algorithm with ECO algorithm over ‘Salpan’ gene expression dataset, the results show the reduction of feature number to 33. The efficiency of 33 element feature set is being verified by the Error value (MSE) as 0.0182. The RMSE value is 0.1349 which is less than the RMSE value obtained from GANN system above. The graph for best cost values against the iteration number is shown in figure below.

![Graph](image)

Fig 7.29. Best Cost Value graph using BCGA over ‘Salpan’ dataset

The best cost value is reduced to 0.0222 (Fig 7.29 above). The above graph is drawn on the basis of the recorded ten best cost values for each iteration as shown in the screenshot below (Fig 7.30)

![Table](image)

Fig 7.30. Ten ‘best cost values’ by applying BCGA over Salpan dataset
Search for Strong Attribute(s) using FCV:

In table below, the experiment lists out set of gene symbols on the basis of highest FCV that strongly predict the class of the subject (patient) i.e. whether the subject is healthy or cancerous. Features with Count Value equal to the maximum iterations represent the strong features as listed below (Table 7.11):

Table 7.11. List of strong gene features using BCGA on Salpan dataset

<table>
<thead>
<tr>
<th>Gene Symbol</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR1</td>
<td>discoidin domain receptor tyrosine kinase 1</td>
</tr>
<tr>
<td>PAX8</td>
<td>paired box 8</td>
</tr>
<tr>
<td>THRA</td>
<td>thyroid hormone receptor, alpha (erythroblastic leukemia viral (v-erb-a) oncogene homolog, avian)</td>
</tr>
<tr>
<td>PTPN21</td>
<td>protein tyrosine phosphatase, non-receptor type 21</td>
</tr>
<tr>
<td>CYP2E1</td>
<td>cytochrome P450, family 2, subfamily E, polypeptide 1</td>
</tr>
<tr>
<td>EPHB3</td>
<td>EPH receptor B3</td>
</tr>
<tr>
<td>ESRRA</td>
<td>estrogen-related receptor alpha</td>
</tr>
<tr>
<td>CYP2A6</td>
<td>cytochrome P450, family 2, subfamily A, polypeptide 6</td>
</tr>
<tr>
<td>SCARB1</td>
<td>scavenger receptor class B, member 1</td>
</tr>
<tr>
<td>TTLL12</td>
<td>tubulin tyrosine ligase-like family, member 12</td>
</tr>
<tr>
<td>NCRNA00152</td>
<td>non-protein coding RNA 152</td>
</tr>
<tr>
<td>WFDC2</td>
<td>WAP four-disulfide core domain 2</td>
</tr>
<tr>
<td>MAPK1</td>
<td>mitogen-activated protein kinase 1</td>
</tr>
<tr>
<td>PRR22</td>
<td>proline rich 22</td>
</tr>
<tr>
<td>ACRV1</td>
<td>acrosomal vesicle protein 1</td>
</tr>
<tr>
<td>DMXL2</td>
<td>Dmx-like 2</td>
</tr>
<tr>
<td>DPM1</td>
<td>catalytic subunit</td>
</tr>
</tbody>
</table>
7.12.4. Comparison between GANN and BCGA with ECO:

Table below shows the Comparative analysis between the proposed model and the traditional algorithms. As we can see that the novel approach of BCGA with ECO outperforms the traditional algorithm of GANN. Table below shows the comparative analysis of BCGA and GANN.

Table 7.12: Comparative analysis between GANN and BCGA using Salpan dataset

<table>
<thead>
<tr>
<th>Evaluation measure</th>
<th>GA NN</th>
<th>BCGA with ECO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error (E)</td>
<td>0.0197</td>
<td>0.0182</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.1403</td>
<td>0.1349</td>
</tr>
<tr>
<td>Objective/Cost function (z)</td>
<td>0.0249</td>
<td>0.0222</td>
</tr>
<tr>
<td>(Best Cost)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features Selected</td>
<td>40</td>
<td>33</td>
</tr>
</tbody>
</table>

7.13. RULE BASED CLASSIFICATION

Neural Network classifier is basically meant for its learning capability. In above section it has been used for evaluation of feature set solution. Apart from using neural network as an evaluator, it can be used for providing structure for Fuzzy Inference System. The basic reason behind using fuzzy controllers is its ability of human like data representation and quality of rule generation. This fuzzy inference system converts the complex data into human understandable knowledge in the form of rules that are being evaluated in terms of their accuracy and error rate. The data partitioning technique that shall be used further is Fuzzy C means as it has proved to be better than previous techniques (Chapter 6).

The new proposed technique of GNFS shall be an adaptive network that implements neuro fuzzy structure of ANFIS [Tharwat, 2010] wherein the premier and consequent parameters are being updated by the modified genetic algorithm (GA) in an aim to perform rule based classification.

The basic difference between a simple neural network structure and ANFIS is that the latter shall consider membership functions as weights. Further the technique will use evolutionary algorithm (GA) with necessary modifications to update the parameter set of FIS, that are being evaluated in
each iteration using MSE as basic part of fitness function. The hybrid technique of Genetic Neuro Fuzzy System (GNFS) can use genetic algorithm with ECO for optimization of parameter set.

The Algorithm starts with the initial creation of Fuzzy Inference system (FIS) using the filtered feature set consisting of the input set and target set. Several options need to be specified in case of FIS design (See Table 7.13)

Table 7.13. Value setting for FIS

<table>
<thead>
<tr>
<th>FIS design options</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Partitioning method</td>
<td>FCM (fuzzy C Means)</td>
</tr>
<tr>
<td>No. of clusters</td>
<td>10</td>
</tr>
<tr>
<td>FIS generator</td>
<td>Genfis3( )</td>
</tr>
<tr>
<td>FIS type</td>
<td>Takagi Sugeno</td>
</tr>
<tr>
<td>FIS evaluation function</td>
<td>MSE (mean square error)</td>
</tr>
<tr>
<td>Objective function</td>
<td>RMSE (root mean square error)</td>
</tr>
</tbody>
</table>

This Partitioning method of FCM algorithm will generate the specified number of clusters which actually determines the number of rules and membership functions for antecedent and consequent. If the No. of clusters is not specified as an integer then ‘auto’ option applies sub clustering algorithm to find out the No. of clusters using radii value as 0.5. If the number of output is one, then genfis3 is used to generate initial FIS for ANFIS training. The Genfis 3 function generates Anfis structure by specifying the input and target data sets, type of FIS to be used, no. of clusters and the fcm options.

The FCM method is used as a partitioning technique that needs the specification of following options along with their default values in Matlab tool:

i. Membership function or fuzzy partition matrix ‘U’ wherein value U (i,j) is the membership value of data object ‘i’ in cluster ‘j’,

ii. Min amount of Improvements (default is <= -5).

iii. Maximum iterations (default is 100),
iv. Fuzzifier exponent for the partition matrix (>1; default in Matlab is 2.0), Fuzzifier exponent represents the fuzziness of the classification. It is called as Fuzzifier or fizziness index or weighing exponent and its value can influence the performance of fuzzy c-means (FCM. The higher the value of Fuzzifier exponent, the softer are the boundaries between the clusters and vice versa). The value of Fuzzifier exponent should be large enough to increase robustness to noise and outliers in FCM and small enough to avoid any unique optimizer [Kuo-Lung Wu, 2012]

The NeuroFuzzy Structure created is further trained using Genetic Algorithm. For that we need to get the Input (premise) & Output (consequent) Parameter Sets from the FIS structure. Parameter values in the set are optimized for the best solution parameter set. This is done in an attempt to decrease the objective function (probably RMSE or MSE).

Next we need to set the some genetic algorithm parameters for GNFS design as shown in table below:

<table>
<thead>
<tr>
<th>GNFS design options for GA</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range for parameter value</td>
<td>[-25 25]</td>
</tr>
<tr>
<td>Maximum training iteration</td>
<td>1000</td>
</tr>
<tr>
<td>Population size</td>
<td>25</td>
</tr>
<tr>
<td>Selection Pressure (β)</td>
<td>8</td>
</tr>
<tr>
<td>Crossover Percentage</td>
<td>0.4</td>
</tr>
<tr>
<td>Mutation Percentage</td>
<td>0.7</td>
</tr>
<tr>
<td>Mutation Rate</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Genetic operation is applied over the Parameter Set that consists of premise and output parameters for optimization. The initial FIS is evaluated for cost values that results in the best solution with lowest cost value and select worst cost with highest cost value. Further, population is selected for crossover genetic operation.

Algorithm 7.4 below lists out the steps for performing the Genetic Neuro Fuzzy System for optimization of parameters in FIS structure.
Algorithm: 7.4. Steps behind GNFS Algorithm.

The Algorithm begins with the creation of ANFIS raw structure that sets the parameter list (premise and consequent). This is followed by training of the parameter set by evolutionary algorithm. Fitness evaluation calculates the rank of each chromosome / parameter set on the basis of which the chromosome is selected for the participation in the next generation.

Start

Step-1: Create initial random population consisting of individuals represented by x.

Step-2: Divide dataset into train and test data.

Step-3: Select the set of features by setting ‘1’ for presence of feature and ‘0’ for absence.

Step-4: Evaluate feature set using ANN classifier.

   Fitness evaluation by calculating cost value using error calculated by classifier.

Step-5: Selection of parent according to fitness value // discussed in chapter GA

   a) Search best cost value, position of which shall participate in next generation
   b) Search worst cost value, position of which shall not participate in next generation.

Step-6: Acceptable Solution found i.e. Termination/Stopping Criteria satisfied?

   If Yes then Exit

   Else Search qualifying feature sets for further processing

   a) The Recombination or crossover to generate offspring
   b) Mutation of the offspring
   c) Replace population by offspring to form the next generation.
   d) Goto Step-3

Thus, creation of new feature sets to participate in next generation
7.13.1. Crossover in GNFS:

The Uniform Arithmetic Crossover (UAC) Operator is used for performing the crossover over the parameter set arithmetically. Here, the alpha value is the uniform distribution function in contrast to rand function that randomly generates the alpha values as 0s and 1s and is suitable for feature selection. However, the alpha values generated by uniform distribution is suitable for parameter value updation and a range is specified in this case to make sure that the values generated fall within this range.

Uniform Arithmetic Crossover (UAC) where two offspring feature sets or parameter sets are arithmetically produced are calculated as:

New Chromosome_1 = alpha* 1st parent + (1-alpha) * 2nd parent

New Chromosome_2 = alpha* 2nd parent + (1-alpha) * 1st parent

7.13.2. Mutation in GNFS- A Novel Idea

Till now mutation has been given a formula that can be used to mutate the value in each generation that was based on some certain parameters. The modified mutation takes into consideration the ratio of No. of mutations in a solution to the total no. of variables in the solution. This proposed GNFS algorithm changes that formula each time new mutation occurs i.e. it will depend on number of mutations to be performed over the solution set/ chromosome.

Algorithm 7.5 lists out the steps that are being performed in the proposed modified mutation operation.

Start Genetic Algorithm for optimization of parameters and check whether the output produced by ANFIS algorithm is appropriate- Go through Back Propagation process to change the premise parameters
Algorithm 7.5: The Modified Mutation operation in GNFS

**Input:**

Input the arrangement or parameter set for mutation ‘X’.

**Set Parameters:**

No.of Positions to be Mutated = (Mutation Rate) * (Length of Parameter Set)

Sigma (σ) = 10% of Rng

Rng = VarMax – VarMin // specify the range

**Process:**

‘O’ represents the Old Value in ‘X’ at position (J) to be Mutated,

‘Sz’ = randn (size of ‘J’)

// returns an array of random entries that is the same size as ‘J’.

Output Mutated Array ‘Y’ at position ‘J’

\[
Y = \frac{O_{old \text{ at position } J} + \sigma \times \text{randn (size of ‘J’)} - \text{No.of Positions to be mutated}}{\text{Total No.of Variables}}
\]

**Output:**

If (VarMin < Y < VarMax) then return Y

Else if (Y < VarMin) then return VarMin

Else if (Y > VarMax) then return VarMax

Evaluate Cost and Use fit mutated parent for the next generation. The randn(x) function returns matrix of random entries or numbers and whose elements are normally distributed with mean 0, standard deviation (σ=1) and variance (σ²=1).

**Repeat:**

Maximum Iteration reached or

Threshold value reached
7.14. Experiment for implementation of GNFS with modified mutation:

GNFS with modified mutation procedure is applied over the real datasets i.e. ‘Heart disease’ dataset and ‘darmatlogy’ dataset. Apart from this, other important traditional algorithms have been also applied over the dataset for comparative analysis. The description and experiment results for each of them are listed below:

7.14.1. Cleveland Heart disease directory HSD- Heart.mat

- **Source:** UCI Machine Learning Repository
- **Creator:** Andras Janosi, William Steinbrunn, Matthias Pfisterer, Robert Detrano, And David W. Aha
- **Date Created:** July, 1988
- **Related Information:** All attributes are numeric-valued. The data was collected from Cleveland Clinic Foundation (cleveland.data). Class attribute is the angiographic disease status.
- **Number of Instances:** 303
- **Number of Attributes:** 14 (including class attribute)
- **Class Distribution:** (5 classes)

<table>
<thead>
<tr>
<th>Class Code</th>
<th>No. of Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>164</td>
</tr>
<tr>
<td>1</td>
<td>55</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>303</strong></td>
</tr>
</tbody>
</table>

Table 7.15. Class distribution of HSD dataset
The various attributes are age, sex, chest pain type, resting blood pressure, cholesterol, sugar level cardio-graphic results, max heart rate, angina, ST depression ST segment, No. of major vessels, Thalassemia defect and the Class attribute that represents the diagnosis of heart disease.

7.14.2. Comparative analysis of GNFS, ANFIS and ANFISGA:

Data retrieved from UCI repository Cleveland dataset is used to compare the two traditional models with the proposed one. Table 7.16 below shows the performance of GFNS compared with traditional models of ANFIS and ANFISGA using 4-test operation.

Table 7.16. Four test for HSD-Heart dataset After 500 iterations

<table>
<thead>
<tr>
<th>Test No. (heart)</th>
<th>Evaluation Measure</th>
<th>ANFIS (FCM genfis3)</th>
<th>ANFIS GA</th>
<th>GNFS (modified mutation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>MSE</td>
<td>24.3579</td>
<td>0.13715</td>
<td>0.10916</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>4.9354</td>
<td>0.37034</td>
<td>0.3304</td>
</tr>
<tr>
<td></td>
<td>Error Mean</td>
<td>-0.22376</td>
<td>-0.059266</td>
<td>-0.010342</td>
</tr>
<tr>
<td></td>
<td>Err St.D.</td>
<td>4.9385</td>
<td>0.36759</td>
<td>0.33207</td>
</tr>
<tr>
<td>Test 2</td>
<td>MSE</td>
<td>1.8995</td>
<td>0.15031</td>
<td>0.1332</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>1.3782</td>
<td>0.3877</td>
<td>0.36496</td>
</tr>
<tr>
<td></td>
<td>Error Mean</td>
<td>0.033722</td>
<td>0.011992</td>
<td>-0.015348</td>
</tr>
<tr>
<td></td>
<td>Err St.D.</td>
<td>1.3801</td>
<td>0.38966</td>
<td>0.36666</td>
</tr>
<tr>
<td>Test 3</td>
<td>MSE</td>
<td>4.603</td>
<td>0.18676</td>
<td>0.13592</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>2.1455</td>
<td>0.43216</td>
<td>0.36867</td>
</tr>
<tr>
<td></td>
<td>Error Mean</td>
<td>-0.019405</td>
<td>0.045206</td>
<td>-0.024013</td>
</tr>
<tr>
<td></td>
<td>Err St.D.</td>
<td>2.1489</td>
<td>0.43217</td>
<td>0.36993</td>
</tr>
<tr>
<td>Test 4</td>
<td>MSE</td>
<td>3.9545</td>
<td>0.12085</td>
<td>0.10398</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>1.9886</td>
<td>0.34764</td>
<td>0.32246</td>
</tr>
<tr>
<td></td>
<td>Error Mean</td>
<td>-0.12282</td>
<td>-0.029238</td>
<td>0.02249</td>
</tr>
<tr>
<td></td>
<td>Err St.D.</td>
<td>1.9881</td>
<td>0.34832</td>
<td>0.32346</td>
</tr>
</tbody>
</table>
The prediction by proposed method of modified mutation process in GNFS proved efficient for the heart disease diagnosis. This is evident by comparing the values of various evaluation measures as shown in Table 7.16.

The results in the above table 7.16 clearly shows that GNFS with modified mutation outperforms among the algorithms used. Further we concluded that the new proposed proportional based mutation are effective for these problem. Figure 7.31 below shows the screenshots captured during Test 1 ad Test 2.

![Graph a. Test 1 (ANFIS GA)](image)

![Graph b. Test 1 (GNFS)](image)

![Graph c. Test 2 (ANFIS GA)](image)

![Graph d. Test 2 (GNFS)](image)

Fig 7.31. Two-test operation over ‘heart disease’ dataset.
The screenshots in Fig 7.31 above represent the 2-test operation displaying the MSE and RMSE values for comparative study of GFS ad AFISGA. As we can see GNFS provides better results i.e. more accuracy and less error.

7.14.3. Dermatology Dataset (DD) – dermat.mat

Dermatology is the study of various types of skin diseases. This real dataset has been downloaded from UCI machine learning repository. It consists of six different classes, each class representing a different skin disease sharing the same features or attributes.

Source: UCI Machine Learning Repository
Creator: Dr.N. Ilter & Dr. A.Guvenir, Gazi University, School of Medicine
Date of Creation: January, 1998
Number of Instances: 358
Number of Attributes: 34 + 1 class attribute
Class Distribution: (6 classes)

Table 7.17. Class distribution of Dermat dataset

<table>
<thead>
<tr>
<th>Class Code</th>
<th>Class:</th>
<th>Number of instances:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>psoriasis</td>
<td>112</td>
</tr>
<tr>
<td>2</td>
<td>seboreic dermatitis</td>
<td>61</td>
</tr>
<tr>
<td>3</td>
<td>lichen planus</td>
<td>72</td>
</tr>
<tr>
<td>4</td>
<td>pityriasis rosea</td>
<td>41</td>
</tr>
<tr>
<td>5</td>
<td>cronic dermatitis</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>Pityriasis rubra pilaris</td>
<td>20</td>
</tr>
</tbody>
</table>

Data retrieved from UCI repository dermatology dataset is used to compare the two traditional models with the proposed one. The prediction by proposed method proved efficient for the heart disease diagnosis. This is evident by comparing the values of various evaluation measures. Table below shows the performance of GFNS compared with traditional models of ANFIS and ANFISGA.

Table 7.18 Four-test experiment over Dermat dataset after 500 iterations

<table>
<thead>
<tr>
<th>Test No.(dermat)</th>
<th>Evaluation Measure</th>
<th>ANFIS (FCM genfis3)</th>
<th>ANFIS GA</th>
<th>GNFS (modified mutation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>MSE</td>
<td>0.4308</td>
<td>0.39231</td>
<td>0.32735</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>0.65635</td>
<td>0.62635</td>
<td>0.57215</td>
</tr>
<tr>
<td></td>
<td>Error Mean</td>
<td>-0.11734</td>
<td>-0.037818</td>
<td>-0.024888</td>
</tr>
<tr>
<td></td>
<td>Err St.D.</td>
<td>0.65184</td>
<td>0.62815</td>
<td>0.5743</td>
</tr>
<tr>
<td>Test 2</td>
<td>MSE</td>
<td>0.99592</td>
<td>0.37699</td>
<td>0.37099</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>0.99796</td>
<td>0.61399</td>
<td>0.060909</td>
</tr>
<tr>
<td></td>
<td>Error Mean</td>
<td>0.1585</td>
<td>-0.07982</td>
<td>-0.038102</td>
</tr>
<tr>
<td></td>
<td>Err St.D.</td>
<td>0.99454</td>
<td>0.61165</td>
<td>0.61076</td>
</tr>
<tr>
<td>Test 3</td>
<td>MSE</td>
<td>0.79669</td>
<td>0.48371</td>
<td>0.46238</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>0.89258</td>
<td>0.69549</td>
<td>0.67999</td>
</tr>
<tr>
<td></td>
<td>Error Mean</td>
<td>-0.025146</td>
<td>0.097041</td>
<td>-0.077626</td>
</tr>
<tr>
<td></td>
<td>Err St.D.</td>
<td>0.9006</td>
<td>0.69193</td>
<td>0.67872</td>
</tr>
<tr>
<td>Test 4</td>
<td>MSE</td>
<td>1556.9255</td>
<td>0.42737</td>
<td>0.29227</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>39.4579</td>
<td>0.65374</td>
<td>0.54062</td>
</tr>
<tr>
<td></td>
<td>Error Mean</td>
<td>-1.1408</td>
<td>-0.014761</td>
<td>0.08941</td>
</tr>
<tr>
<td></td>
<td>Err St.D.</td>
<td>39.4966</td>
<td>0.65664</td>
<td>0.53569</td>
</tr>
</tbody>
</table>

Focusing on MSE and RMSE values, the results in the table above clearly shows that GNFS with modified mutation outperforms among the algorithms applied over the datasets.

Further we concluded that the new proposed proportional based mutation are effective for these problem. The proposed algorithm results in less MSE and RMSE values and thus providing better accuracy in the prediction of disorder class in dermatology dataset (Table 7.18 above)
7.15. Prediction of Belongingness to the class using ANFIS:

**Belongingness Value**

When we say that an instance belongs to a particular class, we need to check whether it perfectly belongs to the said class or not and if not perfectly, then how much it belongs to the next higher or lower class. To make it more clear we have introduced the concept of belongingness to the class.

After adding learning capability to FIS using Neural Network and optimizing the parameters for appropriate prediction, further we added up another layer for checking the belongingness of an instance or a feature set to the particular class. This approach introduces the deep learning concept in GNFS.

The basic working principle of deep learning is to add a layer that would give us proper knowledge about the exact class of the data. If it does not belong to class A, then what is the probability of its belongingness to the next higher or lower class [Michael Nielsen, 2016] [Nicola Jones, 2014].

Determination of membership to the particular class can be performed using Gaussian Probability distribution function is represented as

\[
G_{pdf} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]

Where the two parameters are:

Sigma (σ) indicating the Variance from the mean

Mu (μ) indicating the expected value to be taken

x is the calculated output

The neuro fuzzy controller applied to the dataset uses Gaussian distribution function to split the data variable into membership values and accordingly converts them into crisp outputs (Takagi Sugeno) that we compare with the actual outputs / target outputs to calculate the accuracy.

However, apart from this the crisp values can be used further for more knowledge extraction. Once the calculated outputs are obtained, further belongingness percent is calculated by applying
the Algorithm blow to know exactly to what particular class it belongs or what is the probability of a particular instance to belong to next higher or lower class or category.

Algorithm 7.5 below lists out the steps performed to obtain the belongingness of the instance to a particular class.

Matlab provides the signum function ie. \( \text{sign}(d) \) that helps in checking the sign of a output value. It the value is positive i.e. greater than zero then it gives 1 as result and if the value is negative then it gives -1 as output and if the value is zero then its output is also zero. This function is used in the algorithm to know the diversion ‘d’ of an instance class.

---

Algorithm 7.5: Algorithm for calculating belongingness

1. Start

2. Calculate output on the basis of Gaussian distribution function (membership Values) by applying NeuroFuzzy Controller.

3. Compare the calculated output to the expected output to calculate the Diversion

4. Set \( S = \text{sign} \) (Diversion)

5. for each instance ‘i’ in the dataset

   If \( s==1 \)

   \[
   \text{Belongingness (i)} = (1-\text{abs} (\text{diversion (i)}))*100;
   \]

   Else if \( s==0 \)

   \[
   \text{Belongingness (i)} = 0;
   \]

   Else

   \[
   \text{Belongingness (i)} = \text{abs} (\text{diversion (i)}) *100;
   \]

   Repeat (i)

6. End
• If the diversion value is positive, it means that the instance does not exactly fit in the class specified as its label. Rather, it mostly belongs to the lower class with belongingness percent as follows:

Belongingness for lower class = 100 - Belongingness (i)

E.g. If Actual Class is 1 but the Predicted Class is 0.178 then Diversion is positive i.e. 0.82. So,

Belongingness for Lower class is (100-17.8) = 82.2

• If the diversion value is negative then, it means that the instance does not fit in the class specified as it label. Rather, it has some extensions towards the upper class.

Belongingness for higher class = Belongingness (i)

E.g. If Actual Class is 3 but the Predicted Class is 3.198 then Diversion is negative i.e. -0.198. So,

Belongingness for higher class is \( \text{abs}(-0.198) \times 100 = 19.8 \)

Experiment in section 7.11.1 shows these examples as applied on ‘dermat’ dataset.

7.16. Experiment and Discussion for determining Belongingness

The belongingness concept can be clearly understood by applying the algorithm over the dermatology data set (Refer Section 7.14.3). The diversion and belongingness is calculated using the crisp output and actual output vectors. Applying the belongingness algorithm over the ‘dermat’ datasets, we get the belongingness values as recorded in the structure below (Fig 7.32).
Fig 7.32. Calculation of diversion and belongingness for each instance.

It calculates data output and compares it for checking the diversion of the calculated value from the actual one. This calculated output value is the crisp value determined by the neuro fuzzy structure of GNFS. The less the diversion, the more is the belongingness to the class.

Some of the belongingness values, after applying GNFS algorithm over ‘dermat’ dataset, are listed below (Table 7.19).

Table 7.19. Experimental Results of Class Belongingness Values for each Instance in Dermat Dataset

<table>
<thead>
<tr>
<th>ACTUAL OUTPUT</th>
<th>CALCULATED OUTPUT</th>
<th>DIVERSION</th>
<th>BELONGINGNESS VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.178156768090552</td>
<td>0.821843231909448</td>
<td>17.8156768090552</td>
</tr>
<tr>
<td>5</td>
<td>5.59274179941323</td>
<td>-0.592741799413234</td>
<td>59.2741799413234</td>
</tr>
<tr>
<td>3</td>
<td>3.30517454296828</td>
<td>-0.305174542968276</td>
<td>30.5174542968276</td>
</tr>
<tr>
<td>5</td>
<td>4.84505580762879</td>
<td>0.154944192371215</td>
<td>84.5055807628785</td>
</tr>
<tr>
<td>5</td>
<td>5.50889047038326</td>
<td>-0.508890470383264</td>
<td>50.8890470383264</td>
</tr>
<tr>
<td>1</td>
<td>1.72413574819803</td>
<td>-0.724135748198035</td>
<td>72.4135748198035</td>
</tr>
<tr>
<td>2</td>
<td>1.57656624145817</td>
<td>0.423433758541826</td>
<td>57.6566241458174</td>
</tr>
<tr>
<td>3</td>
<td>2.72416951413712</td>
<td>0.275830485862884</td>
<td>72.4169514137116</td>
</tr>
<tr>
<td>3</td>
<td>2.86689612737683</td>
<td>0.13310387263169</td>
<td>86.6896127376832</td>
</tr>
<tr>
<td>3</td>
<td>3.04306153146798</td>
<td>-0.0430615314679801</td>
<td>4.30615314679801</td>
</tr>
<tr>
<td>4</td>
<td>3.64401960801068</td>
<td>0.355980391989318</td>
<td>64.4019608010682</td>
</tr>
<tr>
<td>2</td>
<td>1.82443099248213</td>
<td>0.175569007517866</td>
<td>82.4430992482134</td>
</tr>
<tr>
<td>1</td>
<td>1.27993655452989</td>
<td>-0.279936554529888</td>
<td>27.9936554529888</td>
</tr>
<tr>
<td>2</td>
<td>2.36350326079160</td>
<td>-0.363503260791606</td>
<td>36.3503260791606</td>
</tr>
<tr>
<td>3</td>
<td>3.14363651367595</td>
<td>-0.143636513675951</td>
<td>14.3636513675951</td>
</tr>
<tr>
<td>2</td>
<td>1.81592985414901</td>
<td>0.184070145850993</td>
<td>81.5929854149007</td>
</tr>
<tr>
<td>2</td>
<td>2.48313852520485</td>
<td>-0.483138525204852</td>
<td>48.3138525204852</td>
</tr>
<tr>
<td>2</td>
<td>1.57448098516092</td>
<td>0.425519014839079</td>
<td>57.4480985160921</td>
</tr>
<tr>
<td>2</td>
<td>2.09679001971905</td>
<td>-0.0967900197190454</td>
<td>9.67900197190454</td>
</tr>
<tr>
<td>4</td>
<td>3.78488231338195</td>
<td>0.215117686618048</td>
<td>78.4882313381952</td>
</tr>
<tr>
<td>1</td>
<td>1.07068283583009</td>
<td>-0.00768283580300937</td>
<td>7.06828358300937</td>
</tr>
<tr>
<td>4</td>
<td>3.71519442179645</td>
<td>0.284055578203547</td>
<td>71.5194421796453</td>
</tr>
<tr>
<td>5</td>
<td>4.97608920150697</td>
<td>0.0239107984930342</td>
<td>97.6089201506966</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>3</td>
<td>3.22157633354788</td>
<td>-0.221576333547876</td>
<td>22.1576333547876</td>
</tr>
<tr>
<td>3</td>
<td>2.91065334637692</td>
<td>0.0893466536230809</td>
<td>91.0653346376919</td>
</tr>
<tr>
<td>4</td>
<td>3.70670829773808</td>
<td>0.293291702261919</td>
<td>70.6708297738081</td>
</tr>
<tr>
<td>2</td>
<td>1.60664030481728</td>
<td>0.393359695182721</td>
<td>60.6640304817279</td>
</tr>
<tr>
<td>5</td>
<td>5.29325918140746</td>
<td>-0.293259181407456</td>
<td>29.3259181407456</td>
</tr>
<tr>
<td>1</td>
<td>0.347528944507086</td>
<td>0.652471055492914</td>
<td>34.7528944507086</td>
</tr>
<tr>
<td>1</td>
<td>1.03438161156227</td>
<td>-0.034381611562265</td>
<td>3.43816115622650</td>
</tr>
<tr>
<td>5</td>
<td>5.56956622705259</td>
<td>-0.569566227052594</td>
<td>56.9566227052594</td>
</tr>
<tr>
<td>6</td>
<td>5.96913381352774</td>
<td>0.0308661864722604</td>
<td>96.9133813527740</td>
</tr>
<tr>
<td>5</td>
<td>4.32461963281595</td>
<td>0.675380367184051</td>
<td>32.4619632815949</td>
</tr>
<tr>
<td>3</td>
<td>3.29570275297818</td>
<td>-0.295702752978182</td>
<td>29.5702752978182</td>
</tr>
<tr>
<td>3</td>
<td>3.11370088443769</td>
<td>-0.113700884437685</td>
<td>11.3700884437685</td>
</tr>
</tbody>
</table>

### 7.17. Overall Structure for Realistic Dataset:

An Algorithm has been proposed that uses GA to filter out features and then uses FIS over the filtered data for rule base classification. Further, the parameters of nonlinear equation in FIS are modified by GA using neural network learning mechanism.

This problem can be eliminated if non-revisiting GA is used in place of simple GA. Moreover, GA uses fitness of various test cases to judge when to finish searching. At present this fitness evaluation is done by a human being hence it is a costly affair. This work can be automated by replacing a human being with a neural network. A combination of neural network and genetic algorithm has been presented to overcome the existing problems of automatic test data generation for white box testing. The algorithm uses all the techniques in each iteration of crossover and then finds out the two arrangements with lowest cost value. These two low cost arrangements shall be used as crossover parents for the next generation. Further, Mutated value
is calculated each time new iteration occurs for mutation. This calculated value will depend on number of times it has been previously mutated.

Apart from this, novel idea has been presented to find out the belongingness to the class. Once the crisp values are obtained in GNFS, it tries to find out the value of belongingness to the class. This is done using Gaussian distribution function. The number of times a particular feature is being selected is an evidence that the particular feature is that much effective and has that much of impact over the target value i.e. No. of Times of Occurrence(Count) ∝ Strength in Prediction of Target Value

Fig 7.33 below shows the overall performance improvement mechanism from feature selection towards prediction. It can be applied to realistic medical data for filtering out the strong features and predicting the exact class.

Fig 7.33. Overall Performance improvement mechanism from feature selection towards prediction
7.18. Conclusion:

The aim is to develop a powerful tool for diagnosis based on softcomputing techniques, wherein the appropriate parameters are selected optimally and has less error and better accuracy. The paper presented the algorithm that blends the advantages of three softcomputing techniques - ANN, FIS and GA so as to produce an improved technique for real-time classification method. The ANFIS is the result of ANN and FIS combination wherein the qualitative as well as quantitative aspects are reached and provides an optimal solution for classification or prediction problems.

Using experimental data, an optimized adaptive neuro-fuzzy inference system model has been developed on MATLAB platform for future prediction. The method of optimized feature selection using ECO operation has been proposed and compared with previous known best feature selection algorithms. Further, the new proposed optimized method of GNFS with mutation operation based on the number mutations in the parameter set or chromosome has been presented. Both the proposed techniques can be applied in diagnosis or classification. The new Optimized ANFIS model of GNFS provides better results than previous ANFISGA model and ANFIS.

Till now either GA has been used for feature selection or in modifying the various parameters in a nonlinear equation for classification or clustering.