Chapter 2

Elitist GA based Feature Selection

2.1 Introduction

In the literature, classification accuracy has been used as fitness function. Muhammad *et al.* [53], Rodrigues *et al.* [54], Nakamura *et al.* [55], Xing *et al.* [56], Suresh *et al.* [57] and Cheng *et al.* [41] have performed feature selection using classification accuracy as fitness function.

To obtain reducts (set of selected features), Muhammad *et al.* [53] have used hybrid of tabu search and kNN classifier, Rodrigues *et al.* [54] have used Binary Cuckoo Search (BCS) optimization method, Nakamura *et al.* [55] have used Binary Bat optimization method, Xing *et al.* [56] and Suresh *et al.* [57] have have utilized Binary PSO optimization method to get optimal reducts and Cheng *et al.* [41] have used Genetic Algorithm (GA) method.

It has been observed that elitist GA (eGA) based approaches have not been attempted for feature selection using classification accuracy as fitness function. This chapter attempts feature selection using enhanced version of GA i.e. Elitist GA (eGA) to obtain reduct which is optimal in the sense of classification accuracy.

Elitist GA (eGA) encodes features as sequences of Boolean values and performs exploration of feature space. String of ones and zeros (chromosomes) is used where features selected are represented as ones. Length of chromosome is same as the number of features in given dataset. In this work, program for eGA has been written in MATLAB and it was interfaced with data mining workbench WEKA for computing the classification accuracy using J48 classifier [58]. J48 classifier is a Java version of decision tree based

C4.5 classifier [51].

2.2 Elitist Genetic Algorithm (eGA)

In order to use GAs to solve any problem, variables are first coded in some string structures. Binary coded strings having 1's and 0's are mostly used. The length of the string is usually determined according to the desired solution accuracy. GAs mimic the survival-of-the-fittest principle of nature to make a search. Therefore, GAs are naturally suitable for solving maximization problems. Minimization problems are usually transformed into maximization problems by some suitable transformation. In general, a fitness function is first derived from the objective function and used in successive genetic operations. The operators of GAs begins with a population of random strings representing design or decision variables. Thereafter, each string is evaluated to find the fitness value. The population is then operated by three main operators- reproduction(selection), crossover and mutation- to create a new population of points. The new population is further evaluated and tested for termination [43]. For the selection of the Mating pool, Roulette wheel is used, then single point crossover is performed and consequently mutation is performed with mutation probability.

The concept of elitism has been used while generating new population for the next generation. Elitism allows few best chromosomes from the current generation to carry over to the next, without any change. This strategy is known as elitist selection and guarantees that the solution quality obtained by the GA will not deteriorate from one generation to the next.

After completing the GA operations the final offspring chromosomes provide the optimal string of ones and zeros, i.e. features selected will be represented as ones and not selected features are shown as zeros in the string.

2.2.1 Steps of eGA

The steps of eGA [45] implemented in this work are as follows. The computational procedure involved in maximizing the fitness function $F(x_1, x_2, x_3, ..., x_n)$ in the genetic algorithm can be described by the following steps.

1. Choose a suitable string length N. Assume suitable values for the following parameters: population size p, crossover probability p_c, mutation probability p_m and maximum number of generations to be used as a convergence criterion.

2. Generate a random population of size p, Evaluate the fitness values $F_i, i = 1, 2, ..., p$.

3. Carry out the reproduction process.

4. Carry out the crossover operation using the crossover probability p_c.

5. Carry out the mutation operation using the mutation probability p_m to find the new generation of p strings.

6. Include elite chromosomes using elite percentage (e_p) for the next generation.

7. Evaluate the fitness values $F_i, i = 1, 2, ..., p$, of the p strings of the new population.

8. If fitness function does not improve or, does not change during last G (specified) number of generations, then stop and return the solution achieved. Otherwise, go to step 2.

2.3 Experimental Setup

String of ones and zeros (chromosomes) is being used where features selected are represented as ones and features which are not selected are represented as zeros. Length of chromosomes (string of ones and zeros) is the same as the number of features in the dataset.

While computing the fitness function for eGA, MATLAB has been interfaced with WEKA. MATLAB program is used for loading the input dataset into WEKA using the interface and computing the S-10-FCV accuracy [59,60] using WEKA. The value returned to MATLAB program is used as fitness function for performing the feature selection. eGA involves operators such as selection, crossover and mutation. Parameters used in GA are shown in Table 2.1. The proposed algorithm terminates when there is no change in the fitness function in last 30 generations. Number of generations for stopping criteria of the eGA depends upon the number of features in the given dataset. i.e., when less number of features are there in the given dataset, less number of generations will be required.

Table 2.1: Parameters used in Elitist Genetic Algorithm (eGA)

Population size	20
Crossover Probability(rate)	0.5 to 0.7
Mutation Probability(rate)	0.005
Elite Percentage	20
Generations	Varying*
*Stopping criteria for termination is taken as: fitness value does not change for specified number (30) of generations. Therefore, for all the different runs for all the dataset, total number of generations are different for different cases.	

2.4 Fitness Function

The stratified ten fold cross validation classification accuracy (S-10FCV) is used as a fitness function in this work. In S-10FCV the available input patterns are divided into ten parts, out of these 10 parts, at a time, 9 parts are assigned as training set (for training a classifier) and one part is assigned as testing set (for evaluating classification accuracy). This process is performed ten times (ten folds) to ensure that all the data are considered for testing. The stratification of the data prior to its division into folds ensures that each class label has equal representation in all folds, thus helping to reduce bias/variance problems. The advantage of S-10FCV is that all objects are used for both training and testing, and each object is used for testing only once. The stratified ten fold cross validation classification accuracy (S-10FCV) or fitness function is returned to the GA as a measure of the quality of a given input pattern.

Proposed algorithm stops when there is no change in the fitness function in last few generations. In this study we used MATLAB / WEKA interface to compute the fitness function. Four benchmark datasets are used in this work [61]. The eGA was run ten time for each of the four benchmark datasets Ecoli, Glass, Wisconsin and Cleveland to avoid initialization bias.

Table 2.2: Description of used benchmark datasets

Dataset	No. of Features	No. of objects
Ecoli	7	336
Glass	9	214
Wisconsin	9	683
Cleveland	13	297

2.5 Results and Discussions

As discussed earlier proposed method, eGA was run on four benchmark datasets [61]. The number of features, number of object and classification accuracy reported for unreduced dataset are illustrated in Table 2.2.

In cases where the feature set contains noisy features, the removal of the same improves the classification accuracy. Hence, it is expected that eGA would reduce the number of features in such cases where accuracy is more but the features are less than the original dataset.

Since eGA is based on evolutionary techniques and their results can be different in different runs, therefore for all the benchmark datasets the experiments are conducted for ten times to avoid initialisation bias. In the Table no. 2.3 to 2.6, the results shown are for ten independent runs. However, results are sorted according to decreasing order of classification accuracy for ease of reading. The results obtained for each dataset are discussed dataset wise.

2.5.1 Ecoli

Results for Ecoli dataset are shown in Table 2.3. In the Table 2.3, ten runs represent ten independent runs. It can be observed from Table 2.3 that maximum accuracy achieved for the dataset is 84.52 %. It is observed that for the same best accuracies, the feature-sets (reducts) were different, for example, at s.no. 6, the accuracy is 84.52 and reduct size is 6, whereas, the same accuracy is there for reduct size 5 also as given at s.no. 7.

The 6th feature from left is irrelevant or noisy and for the improved accuracy, it

Table 2.3: Result for Ecoli dataset for 10 independent runs

S. No.	Optimal substring (Chromosomes)	No. of features in optimal substring	Stratified tenfold cross validation Accuracy	Generations
1	1111101	6	84.52	35
2	1111101	6	84.52	31
3	1111101	6	84.52	31
4	1111101	6	84.52	50
5	1111101	6	84.52	60
6	1111101	6	84.52	31
7	1110101	5	84.52	55
8	1110101	5	84.52	31
9	1110101	5	84.52	37
10	1110101	5	84.52	31

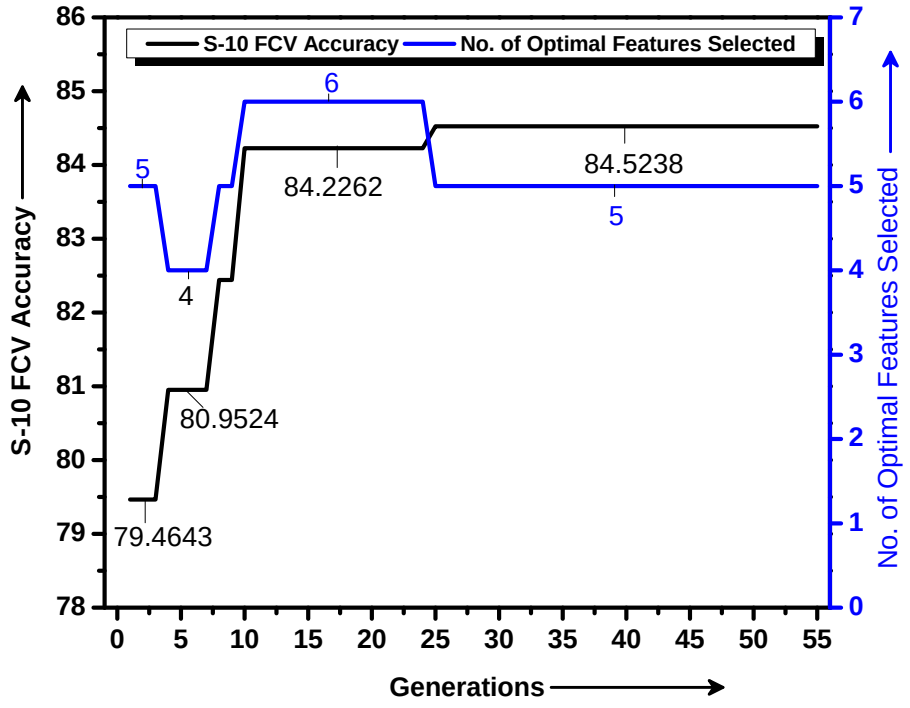

Figure 2.1: Accuracy vs generation and number of features selected vs generation for Ecoli dataset

must be removed and 4th feature is redundant because it is clear from Table 2.3 that presence/absence of this 4th feature does not affect the accuracy. It is observed that for best accuracy all the features except 4th and 6th are indispensable features; 6th feature is irrelevant and 4th feature is redundant. Figure 2.1 shows the plot of accuracy and reduct size with respect to generations.

In cases where the feature set contains noisy features, the removal of the same improves the classification accuracy. Hence, it is expected that eGA would reduce the number of features in such cases where accuracy is more but the features are less than the original dataset.

2.5.2 Glass

Results for Glass dataset are shown in Table 2.4. It can be observed from Table 2.4 that maximum accuracy achieved for the dataset is 77.57 %.

It is observed that 4th feature from the left is irrelevant, which means that taking 4th feature in the dataset deteriorates the classification accuracy. For 5 selected features accuracy is more than that of selecting 6 features. It is observed that for best accuracy all the features except 4th, 5th, 6th and 9th are indispensable features; 4th feature is irrelevant. Figure 2.2 shows the plot of accuracy and reduct size with respect to generations.

2.5.3 Wisconsin

Results for Wisconsin dataset are shown in Table 2.5. It can be observed from Table 2.5 that maximum accuracy achieved for the dataset is 95.99 %. It is observed that for the same best accuracies, the feature-sets were different, for example at s. no. 2, the accuracy is 95.99 % and reduct size is 3, whereas, the same accuracy is there for the reduct size 4 also as given at s. no. 3.

5th feature is redundant because it is clear from Table 2.5 that presence/absence of this 5th feature does not affect the accuracy. It is observed that for best accuracy 1st, 2nd and 6th are indispensable features.

Table 2.4: Result for Glass dataset for 10 independent runs

S. No.	Optimal substring (Chromosomes)	No. of features in optimal substring	Stratified tenfold cross validation Accuracy	Generations
1	111000110	5	77.57	49
2	111000110	5	77.57	33
3	111000110	5	77.57	31
4	111000110	5	77.57	53
5	111000110	5	77.57	36
6	111011001	6	75.23	36
7	111011001	6	75.23	33
8	111011001	6	75.23	36
9	111100010	5	73.83	36
10	010111110	6	72.89	42

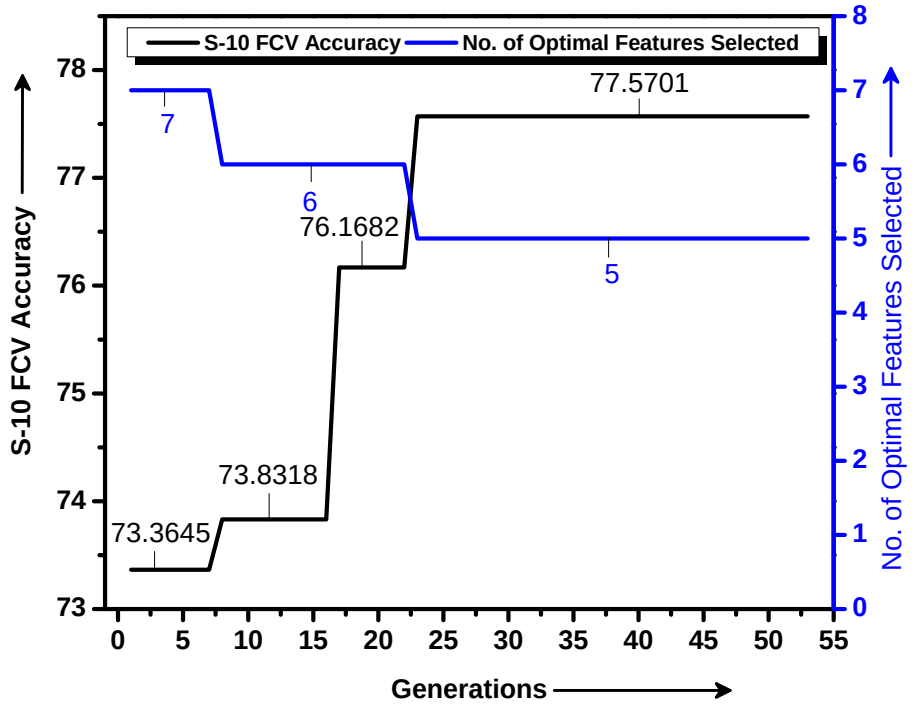

Figure 2.2: Accuracy vs generation and number of features selected vs generation for Glass dataset

Table 2.5: Result for Wisconsin dataset for 10 independent runs

S. No.	Optimal substring (Chromosomes)	No. of features in optimal substring	Stratified tenfold cross validation Accuracy	Generations
1	110001000	3	95.99	47
2	110001000	3	95.99	39
3	110011000	4	95.99	45
4	110011000	4	95.99	38
5	110011000	4	95.99	34
6	110011000	4	95.99	38
7	110011000	4	95.99	35
8	101010010	4	95.99	33
9	110011111	7	95.99	38
10	101001011	5	95.85	35

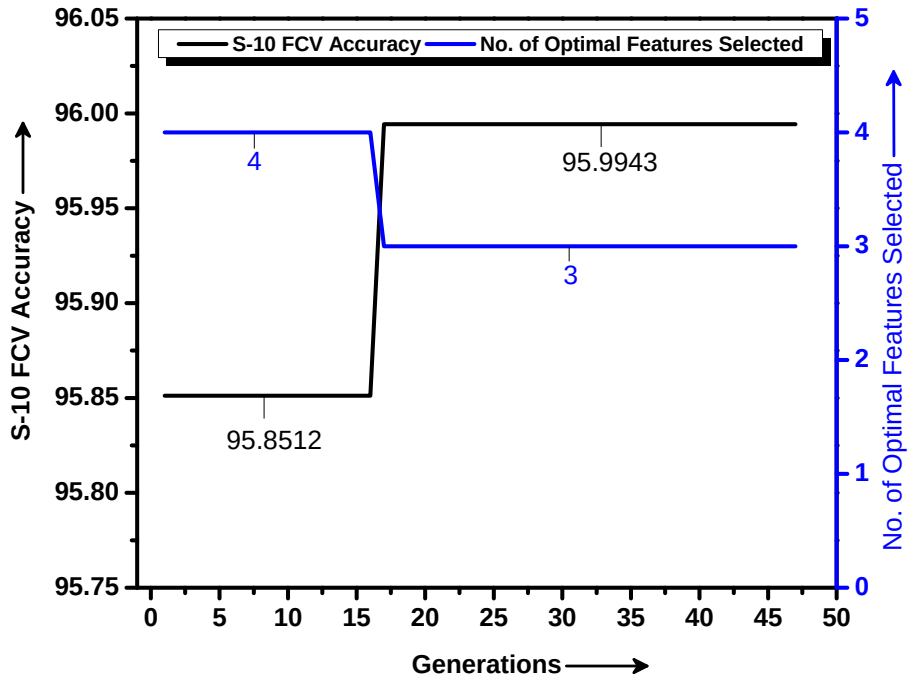

Figure 2.3: Accuracy vs generation and number of features selected vs generation for Wisconsin dataset

Table 2.6: Result for Cleveland dataset for 10 independent runs

S. No.	Optimal substring (Chromosomes)	No. of features in optimal substring	Stratified tenfold cross validation Accuracy	Generations
1	0010100010010	4	62.37	333
2	0010100010010	4	62.37	157
3	0010100010010	4	62.37	157
4	0010100010010	4	62.37	157
5	0010000000010	2	62.04	163
6	0001110110101	7	61.05	155
7	0001110110101	7	61.05	155
8	0001110110101	7	61.05	155
9	0111110000010	6	60.72	151
10	0011100100010	5	59.73	151

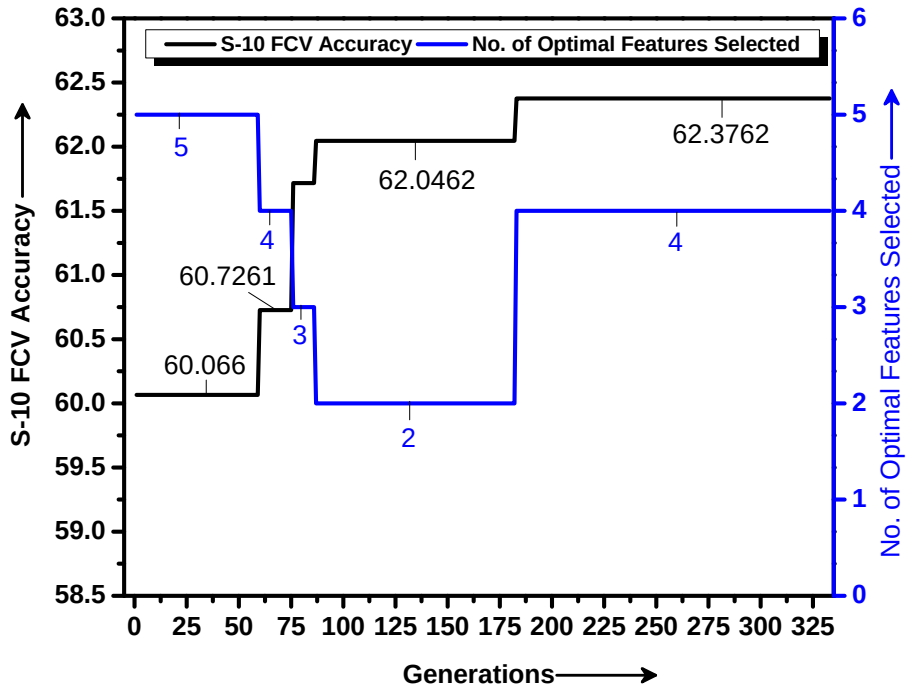

Figure 2.4: Accuracy vs generation and number of features selected vs generation for Cleveland dataset

2.5.4 Cleveland

Cleveland consists of 13 features, and it converges in 150 generations. Results for Cleveland dataset are shown in Table 2.6. It can be observed from Table 2.6 that maximum accuracy achieved for the dataset is 62.37%.

It is evident from Table 1.1, that the reported best accuracy in literature is 52.6%. Low classification accuracy is the characteristic of this dataset.

1st, 2nd, and 7th features are irrelevant (for the improved accuracy, it must be deleted). It is observed that for best accuracy 3rd, 5th, 9th and 12th are indispensable features.

2.5.5 Comparison of results

The Table 2.7 shows the comparison of the sizes of selected subsets of features, using different methods reported in [9] with proposed method, eGA. The methods reported in [9] are Unsupervised-Fuzzy Rough Feature Selection(UFRFS), Boundary region based Unsupervised Fuzzy Rough Feature Selection(B-UFRFS), Fuzzy Rough Feature Selection(FRFS) and Boundary region based Fuzzy Rough Feature Selection(B-UFRFS). In Table 2.7 at certain instances fractional number of features are reported corresponding to average number of features. It is observed that in all the dataset, the number of feature obtained by the proposed method (eGA) are invariably less than the reported average number of features [9]. These methods of [9] have not identified irrelevant features. Whereas, through proposed method irrelevant/noisy and redundant features could be identified.

Proposed method produces the optimal feature set, detects irrelevant/noisy and redundant features, and it is able to select most informative features.

2.6 Application in Short Term Price Forecasting

This subsection focuses on feature selection for short-term price forecasting for power distribution systems. The method proposes the decision tree algorithm combined with genetic algorithms for feature selection for price forecasting. Later, these selected features are used with decision trees to forecast the prices. The usefulness of proposed algorithm is

Table 2.7: Average Reduct sizes using different methods

Dataset	UFRFS	B-UFRFS	FRFS	B-FRFS	Proposed method Average(Best) reduct
Ecoli	6	6	6	6	**5.6(5)**
Glass	7.1	7.1	9	8.4	**5.4(5)**
Wisconsin	8.1	9	6.9	6.8	**4.2(3)**
Cleveland	10.4	11.4	7.6	7.8	**5.0(2)**

established by comparing the forecasts obtained using full feature set with that of reduced feature set. This application of eGA based feature selection may lead to some new insights on the type and the seasonality of features and their effects on the electricity prices.

The feature selection approach was applied to short term price forecasting problem in a following manner.

- Using combination of GA and J48 for feature selection in price forecasting problem.
- Using J48 Classifier for prediction of Australian price data.
- Season-wise feature selection is attempted to draw certain insight on number and type of feature effecting the price in different seasons.

The method uses eGA and decision tree classifier for feature selection and process of feature selection has been performed weekly.

It is observed that certain features are selected more number of times than the others depending on the season and also that for the same system in a year the number of required feature can be as low as 2 to as large as 19.

The Mean Absolute Percentage Error (MAPE) has been calculated day-wise, week-wise and season-wise Without Feature Selection (WoFS) and With Feature Selection (FS). It is established from the result that proposed feature selection (FS) method provides better forecast accuracy of electricity prices in comparison to that using full set.

For the short term price forecasting, data is collected from Australian Energy Market Operator for New South Wales. The data consists of load and price of all seasons from Jan 2014 to June 2016. The weather data (wind speed, temperature and humidity) of Sydney City is taken from www.weatherzone.com/au.

Forecast accuracy after forming training set and testing set data, the daily forecast using J48 classifier is computed. Mean Absolute Percentage Error (MAPE) is calculated without and with feature selection, for entire week and for all the seasons viz. winter, spring and summer.

The training set was taken on the concept of similar weeks. The dataset corresponds to the five similar weeks of the months of the last year and the past week of the same year. For example, if the feature selection is to be performed for the week of 1-7 July 2015, the training set would include the data corresponding to 24-30 June 2015, 1-7 July 2014, 24-30 June 2014, 17-23 June 2014, 8-14 July 2014, 15-21 July 2014. The major advantage of feature selection is that the method can be employed for conducting feature analysis, as to which feature is effecting the forecast or consumption patterns more significantly.

The forecast obtained for 1-7 July, 2015 is shown in Table 2.8. The MAPE for this week with feature selection is 8.61 using 14 selected features, whereas without feature selection it is 9.06 (i.e. with 31 features). The forecast obtained for 1-7 September, 2015 is shown in Table 2.8, the MAPE for this week with feature selection is 11.23 using 15 selected features, whereas without feature selection it is 11.70. It is observed that despite very small number of feature selected, the results are better compared to that of without feature selection.

The data from June 2015 to august 2015 falls under summer, September 2015 to November 2015 falls under spring, and data from December 2015 to February 2016 falls under winter. The result is further obtained weekly without feature selection and with feature selection.

Table 2.8: MAPE for representative weeks with and without feature selection

S. No.	Season	Month	Week 1 (1-7)			Week 3(15-21)			Mean of all 4 weeks	
			Without FS	With Proposed FS		Without FS	With Proposed FS		Without FS	With Proposed FS
			(31 features)	No. of Selected		(31 features)	No. of Selected		(31 features)	
			MAPE	Features	MAPE	MAPE	Features	MAPE	MAPE	MAPE
1	Winter	June	15.69	12	11.81	8.62	9	8.26	11.54	10.73
2		July	9.06	14	8.61	15.64	19	14.52	12.17	10.86
3		Aug	12.31	13	13.25	9.39	17	11.64	11.15	11.62
Winter Mean			12.35		11.22	11.21		14.14	11.62	11.07
4	Spring	Sep	11.7	15	11.23	10.25	13	10.92	12.48	11.27
5		Oct	11.64	8	10.49	9.76	15	10.3	9.9	9.58
6		Nov	11.96	6	9.37	25.3	14	25.81	15.14	14.46
Spring Mean			11.77		10.36	15.1		15.68	12.51	11.77
7	Summer	Dec	13.27	12	11.67	25.48	9	17.61	17.01	13.3
8		Jan	7.9	14	7.44	13.95	14	13.04	12.44	10.99
9		Feb	8.93	6	8.14	12.31	4	11.92	12.73	13.14
Summer Mean			10.03		9.08	17.24		14.19	14.06	12.48
Over all Mean			11.38		10.22	14.52		13.78	12.73	11.77

Table 2.9: Comparison of proposed results with other methods

MAPE of different models for Summer season				
ANN	SVM	Regression	J48 WoFS	J48 with FS
13.87	10.8	9.98	10.42	**8.86**

When MAPE is compared season wise it is observed that after using feature selection we can get lower MAPE (11.07) than that of without selecting features (11.62) for winter season. In spring, feature selection yields MAPE of 11.77 as compared to 12.51 without feature selection. In summer, feature selection yields MAPE of 12.48 as compared to 14.06 without feature selection. The application of feature selection can lead to improved price forecasting.

2.7 Conclusions

The method of eGA produces the optimal feature set and using the proposed method of feature selection the redundant and irrelevant/noisy features are identified. Results of the experiments shows that the proposed method is able to select the most informative features in terms of classification accuracy.

Present study considered 4 UCI datasets to validate the efficacy of the proposed approach of feature selection using eGA. An analysis of irrelevant/noisy features has also been performed. In later chapters, fuzzy rough set based fitness functions have been used, hence to compare the performance of fuzzy rough set with respect to the feature selection method which use classification accuracy as fitness function, this study has been done.

In the real life application in price forecasting, a decision tree method with feature selection is presented for predicting the electricity prices. The method uses eGA and decision tree classifier for feature selection and process for feature selection has been performed weekly. The results explain the efficacy of feature selection method. It is observed that certain features are getting selected frequently than others depending upon season. The Mean Absolute Percentage Error (MAPE) has been calculated for, with and without feature selection, using predicted data for the whole year. It is established from the results that feature selection method provides better forecast accuracy of electricity prices in comparison to that of using full set.

For the sake of comparison, we have included other methods which have been used for price forecasting in power systems. The forecast accuracy (MAPE) of the decision tree based classifier J48 is better than other methods implemented for the same data. The comparison also shows that feature selection is an effective way to improve the forecast accuracy [62].

J48 method without feature selection gives results comparable to other methods namely ANN, SVM and Regression methods. However, when feature selection is incorporated with J48 method the accuracy is enhanced appreciably.

The methods having classification accuracy as fitness function, calculate the classification accuracy repeatedly, hence become computationally intensive. Further, these methods do not use interdependencies existing among dataset objects, which may be helpful in giving better reducts. In view of this, in the later chapters rough set and fuzzy rough set based fitness functions have been used.