Chapter 5

Feature Selection Model for Incomplete Data

Millions of data is being generated everyday from different sources like internet, sensors, etc. Noise, human error in measurement, lack of proper communication, etc leads to presence of irrelevant and redundant features, missing values in the dataset gathered. Hence, it is necessary to preprocess the datasets before analysing the performance of learning algorithms. Handling missing values [28, 94, 113] and feature selection are the two important aspects that must be paid heed for enhancing classification performance of pattern recognition problems and reducing the computational complexity. However, all the past studies have focussed on these approaches individually.

Ignoring the instances containing missing values [162] or replacing the missing values with known statistical parameters like mean, median, global constant, etc are some common place techniques to deal with datasets containing missing values. However, these methodologies degrade the performance by removing the information that might be present in the instance containing missing values. Estimating the

missing values i.e. missing data imputation is an another option to deal with this problem. Various machine learning techniques like Neural Networks [55], Nearest neighbour methods [134, 135], Support Vector Machines [60], Clustering [92] and Biclustering [31] have been applied to impute missing values from the information contained in the dataset itself. Liu et. al. [98] has employed classification based impuation estimating the missing values using k nearest neighbour and self organising map according to context. Several fuzzy approaches have been established for imputation purpose and have produced very effective results [124, 126]. Sefidian et. al. has computed missing data by integrating a regression model with grey-based fuzzy c-means and feature selection based on mutual information and has reported efficient results [126]. Shu et. al. has [130] employed rough set for feature selection in incomplete data environment.

5.1 Fuzzy Rough Model for Feature Selection and Missing Value Imputation

One of the efficient way for enhancing classification accuracy and simultaneously reducing complexity is to first impute the missing values and then perform feature selection. Missing values can be estimated by using the information available from dataset itself. Employing the subset of features of dataset containing missing values will save much time by reducing the efforts involved in evaluating all the features.

5.1.1 Feature Grouping

An effective way to find most closely related features to a given feature is to form feature grouping. Correlation between the features is employed to form the grouping and is given by the following formula:

$$cor(a,b) = \frac{\sum_{x\in U}(a(x)-m_a)(b(x)-m_b)}{\sqrt{\sum_{x\in U}(a(x)-m_a)^2 \sum_{x\in U}(b(x)-m_b)^2}} \tag{5.1}$$

where $a, b \in A$ and m_a is mean of feature a. The value of $cor(a,b)$ measures the degree of closeness between features a and b. It ranges from -1 to $+1$. Negative value of correlation coefficient is also considered as being correlated, i.e. $correlation(a,b) = \mid cor(a,b) \mid$. In the proposed work, the correlation of the feature containing missing value with all the remaining features is calculated and the one with maximum value is denoted as $correlation_{max}$. The features whose correlation is greater than $c\%$ (some constant) of $correlation_{max}$ are employed for missing value imputation.

5.1.2 Missing Value Imputation

Instead of utilising the whole dataset for imputation, a subset of data containing the most correlated features, C' (as given in Section 5.1.1) and k nearest (or most similar) instances having the same decision class is employed. The procedure for imputing missing value is shown in Algorithm 5.1.2. The fuzzy lower and upper approximation employed here are defined as follows:

$$\mu_{S\downarrow_{C'}M}(x) = \inf_{y\in N} I(\mu_{S_{C'}}(x,y), \mu_{S_M}(x,y)) \tag{5.2}$$

$$\mu_{S\uparrow_{C'}M}(x) = \sup_{y\in N} T(\mu_{S_{C'}}(x,y), \mu_{S_M}(x,y)) \tag{5.3}$$

where M is the feature containing missing value and is regarded as decision feature, N consists of k nearest instances and having same actual decision class (considering whole dataset) as missing value containing instance. $S_{C'}(x,y)$ calculates the similarity between x and y using feature contained in C'. The crux of the algorithm is

to compute fuzzy lower approximations of all the instances $y \in N$. The weighted average of decision feature M's value in set N is utilised as final prediction considering the value of lower approximation as weights. In case when sum of lower approximations is 0, only the average of decision feature's value is used.

Algorithm 5.1.2: Fuzzy Missing Value Imputation

Input: U: Dataset containing missing values, C: Set of all features;

for all $x \in U$ and $ContainsMissing(x)$ **do**

for all $M \in A$ and $IsMissing(M(x))$ **do**

$C' \leftarrow CorrelatedFeature(M)$

$N \leftarrow kNearestneighbour(x, C', k)$

$num \leftarrow 0, den \leftarrow 0$

for all $y \in N$ **do**

$low \leftarrow \mu_{S\downarrow_{C'}M}(y)$

$num \leftarrow num + low * M(y)$

$den \leftarrow den + low$

end for

if $den > 0$ **then**

$M(x) \leftarrow (num/den)$

else

$M(x) \leftarrow \sum_{z \in N} M(z)/ \mid N \mid$

end if

end for

end for

In the algorithm, $ContainMissing(x)$ checks whether the instance x contains any

missing value. $IsMissing(M(x))$ returns true if feature M contains a missing value for instance x, $CorrelatedFeature(M)$ returns features correlated with M as described in Section 5.1.1. While $kNearestneighbour(x, C', k)$ returns the k nearest neighbour of x with respect to features in C'.

The similarity $S_{C'}(x, y)$ is calculated as:

$$S_{C'}(x, y) = \min_{a \in C'} \mu_{S_a}(x, y) \tag{5.4}$$

where equaion (5.5) is used for computing $\mu_{S_a}(x, y)$.

$$\mu_{S_a(x,y)} = \max(\min(\frac{(a(y) - (a(x) - \sigma_a))}{\sigma_a}, \frac{((a(x) + \sigma_a) - a(y))}{\sigma_a}), 0) \tag{5.5}$$

where σ_a denotes the standard deviation of the feature a. This value is calculated between non-missing values of instances x and y, simply ignoring the missing value. The entire methodology is illustrated via a toy example (table 5.1). Missing values are denoted by '?'.

Instance 2 contains two missing values. For feature 1 of instance 2. The algorithm starts by finding the correlated feature to 1.

$$cor(1, 2) = -0.4082,$$

$$cor(1, 3) = -0.2998,$$

$$cor(1, 4) = 0.7736$$

All the features whose absolute value of correlation coefficient is larger than say 50% i.e. $\frac{0.7736}{2} = 0.3868$ are used for imputation. Therefore, features 2 and 4 are employed. Decision class of the missing instance is 1, so only instances belonging to class 1 are used to find nearest neighbour. Similarity between instance 2 with missing value is calculated with instances $\{1, 5, 6\}$ using formula given in equation (5.5).

$$\mu_{S_{\{2,4\}}}(2, 1) = 0.9757, \mu_{S_{\{2,4\}}}(2, 5) = 0.9514, \mu_{S_{\{2,4\}}}(2, 6) = 0.9757,$$

For this example, k is set to 3. So, the three nearest neighbours of instance 2 are

$\{1, 5, 6\}$. The reduced dataset consisting of $\{1, 5, 6\}$ instances and $\{2, 4\}$ features is employed to calculate lower approximation.

$$\begin{aligned}
S \downarrow_{\{2,4\}} M_2(1) &= \min\Big(\max\big(1 - \mu_{S_{\{2,4\}}}(1,5), \mu_{S_{\{1\}}}(1,5)\big), \\
&\quad \max\big(1 - \mu_{S_{\{2,4\}}}(1,6), \mu_{S_{\{1\}}}(1,6)\big)\Big) \\
&= \min\Big(\max\big(1 - 0.6904, 0.9271\big), \\
&\quad \max\big(1 - 0.7936, 0.9757\big)\Big) \\
&= 0.9271, \\
S \downarrow_{\{2,4\}} M_2(5) &= \min\Big(\max\big(1 - \mu_{S_{\{2,4\}}}(5,1), \mu_{S_{\{1\}}}(5,1)\big), \\
&\quad \max\big(1 - \mu_{S_{\{2,4\}}}(5,6), \mu_{S_{\{1\}}}(5,6)\big)\Big) \\
&= \min\Big(\max\big(1 - 0.6904, 0.9271\big), \\
&\quad \max\big(1 - 0.8968, 0.9514\big)\Big) \\
&= 0.9271, \\
S \downarrow_{\{2,4\}} M_2(6) &= \min\Big(\max\big(1 - \mu_{S_{\{2,4\}}}(6,1), \mu_{S_{\{1\}}}(6,1)\big), \\
&\quad \max\big(1 - \mu_{S_{\{2,4\}}}(6,5), \mu_{S_{\{1\}}}(6,5)\big)\Big) \\
&= \min\Big(\max\big(1 - 0.7936, 0.9757\big), \\
&\quad \max\big(1 - 8968, 0.9514\big)\Big) \\
&= 0.9514
\end{aligned}$$

So, the estimated value of missing feature is

$$\begin{aligned}
&\frac{\begin{array}{r} S \downarrow_{\{2,4\}} M_2(1) * M_2(1) + S \downarrow_{\{2,4\}} M_2(5) * M_2(5) \\ + S \downarrow_{\{2,4\}} M_2(6) * M_2(6) \end{array}}{S \downarrow_{\{2,4\}} M_2(1) + S \downarrow_{\{2,4\}} M_2(5) + S \downarrow_{\{2,4\}} M_2(6)} \\
&= \frac{0.9271 * 0 + 0.9271 * 3 + 0.9514 * 1}{0.9271 + 0.9271 + 0.9514} \\
&= 1.3304
\end{aligned}$$

When most of the feature values are missing for an instance, then such an instance will not contribute much to classification accuracy. To tackle this issue, instances

TABLE 5.1: Toy Example

Features / **Instances**	a_1	a_2	a_3	a_4	**D**
x_1	0	1	1	10	1
x_2	?	2	0	?	1
x_3	0	0	4	20	2
x_4	0	?	2	15	2
x_5	3	0	1.5	25	1
x_6	1	1	2.5	20	1

containing more than certain percentage of missing values are ignored for further consideration.

5.1.3 Search Heuristic for Finding Reduct

After imputing the missing values, there is a need for an efficient and effective search strategy for reducing the dataset dimensionality without performing exhaustive search [96].

Monarch Butterfly Optimization (MBO) [118, 157] is a search heuristic based on the migratory behaviour of monarch butterflies. Whole population of monarch butterfly is divided into Land 1 and Land 2. Each child butterfly is generated for next generation via migratory operator from Land 1 or Land 2. In order to head towards optimal solution, only the one posing better fitness out of parent or child is passed onto next generation and the other one is discarded. Monarch butterflies lie in Land 1 from April to August (5 months) and from September to March (7 months) in Land 2. Let N be total number of monarch butterflies, p proportion of which stay in Land 1 i.e. $p * N = N_1$ while the remaining $N - N_1 = N_2$ lies in Land 2. The

value of p is set to $\frac{5}{12}$. Each monarch butterfly is represented by m (number of features) dimensional binary vector where 1 or 0 represents presence or absence of corresponding feature respectively. The migration behaviour of butterflies in Land 1 is expressed by migration operator while that in Land 2 by butterfly adjusting operator.

In case of migration operator, a variable $rd = rand * period$ is calculated where $period$ is set to 1.2 (as 12 months of a year), $rand$ is a random number generated from uniform distribution. If $rd \leq p$, then jth element of newly generated butterfly is updated via equation (5.6) else via equation (5.7).

$$b_{i_1,j}^{g+1} = b_{rd_1,j}^{g} \tag{5.6}$$

where $b_{i_1,j}^{g+1}$ and $b_{rd_1,j}^{g}$ denotes the jth element of i_1th butterfly for generation $g+1$ and jth element of rd_1 butterfly randomly drawn from Land 1 respectively.

$$b_{i_1,j}^{g+1} = b_{rd_2,j}^{g} \tag{5.7}$$

where $b_{rd_2,j}^{g}$ denotes the jth element of rd_2 butterfly randomly drawn from Land 2. The entire methodology of migration operator is depicted in Algorithm 5.1.3 below.

Algorithm 5.1.3: Migration Operator

for $i_1 = 1$ to N_1 **do**

  for $j = 1$ to m **do**

    Generate $rand$ randomly from uniform distribution

    $rd = period * rand$

    if $rd \leq p$ **then**

      Randomly select rd_1 butterfly from Land 1

      $b_{i_1,j}^{g+1} = b_{rd_1,j}^{g}$

    else

      Randomly select rd_2 butterfly from Land 2

$b_{i_1,j}^{g+1} = b_{rd_2,j}^{g}$

end if

end for

if $Fit_{b_{i_1}^{g+1}} < Fit_{b_{i_1}^{g}}$ **then**

$b_{i_1}^{g+1} = b_{i_1}^{g}$

end if

end for

The fitness of newly generated butterfly $b_{i_1}^{g+1}$ is compared with $b_{i_1}^{g}$ and the one with higher fitness survives and is passed onto next generation.

In butterfly adjusting operator, jth element of i_2th butterfly is assigned the jth element of b_{Best}, best butterfly (with best fitness) in Land 1 and Land 2 if randomly generated number $rand \leq p$ otherwise a jth element of a random butterfly rd_3 from Land 2 is assigned to $b_{i_2,j}^{g+1}$. Further when $rand > Bar$ then the value of $b_{i_2,j}^{g+1}$ is inverted (from 1 to 0 or 0 to 1) if $\frac{S_{max} * (Levy() - 0.5)}{g^2} > 0$, where Bar is the butterfly adjusting rate, $Levy()$ indicates the walk step that a monarch butterfly takes while S_{max} is the maximum walk step that is permissible in one step and g is the current generation. Positive value of $\frac{S_{max} * (Levy() - 0.5)}{g^2}$ inverts the value thereby encouraging exploration while a negative value divert towards exploitation process. A crossover operator is introduced for full utilization of butterfly population [158]. An updated version of butterfly is created as

$$b_{i_3,j}^{g+1} = b_{i_2,j}^{g+1} * (1 - cr) + b_{i_2,j}^{g} * cr \tag{5.8}$$

where cr is the crossover rate given by following formula:

$$crossover = 0.8 + 0.2 * \frac{Fit_{b_{Best}} - Fit_{b_{i_2}^{g}}}{Fit_{b_{Best}} - Fit_{b_{Worst}}} \tag{5.9}$$

$$cr = \begin{cases} 1 & random\ number < crossover \\ 0 & otherwise \end{cases} \tag{5.10}$$

And the one with greator fitness is kept and other is discarded out of $b_{i_2}^{g+1}$ and $b_{i_3}^{g+1}$. Algorithm 5.1.3 below illustrates the entire methodology.

Algorithm 5.1.3: Butterfly Adjusting Operator

for $i_2 = 1$ to N_2 **do**

for $j = 1$ to m **do**

Generate $rand$ randomly from uniform distribution

if $rand \leq p$ **then**

$b_{i_2,j}^{g+1} = b_{Best,j}^{g}$

else

Randomly select rd_3 butterfly from Land 2

$b_{i_2,j}^{g+1} = b_{rd_3,j}^{g}$

if $rand > Bar$ and $\frac{S_{max} * (Levy() - 0.5)}{g^2} > 0$ **then**

$b_{i_2,j}^{g+1} = \sim b_{i_2,j}^{g+1}$ i.e. its value is inverted

end if

end if

end for

$b_{i_3,j}^{g+1} = b_{i_2,j}^{g+1} * (1 - cr) + b_{i_2,j}^{g} * cr$ i.e. generate new butterfly where cr is given by equation (5.10)

if $Fit_{b_{i_2}^{g+1}} < Fit_{b_{i_3}^{g+1}}$ **then**

$b_{i_2}^{g+1} = b_{i_3}^{g+1}$

end if

end for

After proposing the methodology for generating the new generation by migration and butterfly adjusting operator, MBO begins by evaluating the fitness of each

butterfly using equation (5.11). Positions of all butterflies are updated step by step until some termination condition is satisfied.

$$Fit_i = \alpha \times \gamma_{Red}(D) + \beta \times \frac{\mid m \mid - \mid Red \mid}{\mid m \mid} \tag{5.11}$$

where $\mid m \mid$ is the total number of features in the dataset, $\mid Red \mid$ is the number of bits set in butterfly b_i. The two parameters α and β govern the importance of classification performance and subset length respectively, such that $\alpha = 1 - \beta, \alpha \in [0, 1]$. The entire search heuristic is depicted in Algorithm 5.1.3 below.

Algorithm 5.1.3: Monarch Butterfly Optimization

Input: $maxGen$: maximum number of generations; N: total number of monarch butterflies; N_1, N_2: number of butterflies in Land 1 and Land 2 respectively; S_{max}: maximum step size; Bar: butterfly adjusting rate; $period$: migration period; p: migration ratio;

Evaluate fitness of each monarch butterfly

while $g < maxGen$ **do**

Sort all the butterflies according to their fitness

Divide the population into Land 1 and Land 2

for $i_1 = 1$ to N_1 **do**

Generate the new butterflies using Algorithm 5.1.3 for Land 1

end for

for $i_2 = 1$ to N_2 **do**

Generate the new butterflies using Algorithm 5.1.3 for Land 2

end for

Combine the newly generated populaion of Land 1 and Land 2

Evaluate the fitness of newly generated population

$g = g + 1$

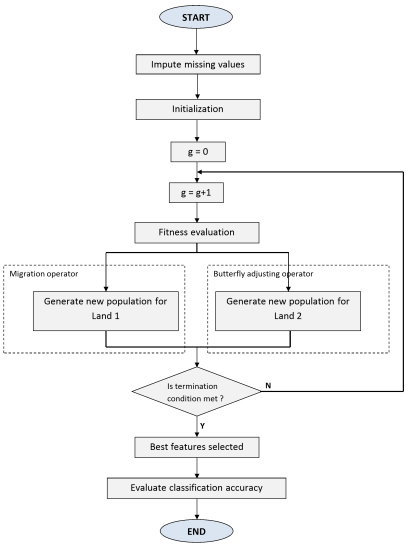


FIGURE 5.1: The flowchart of proposed model

end while

Return butterfly with maximum fitness

The entire methodology of the proposed approach is depicted in Figure 5.1.

5.2 Experimentation

The experimental evaluation needed to show the effectiveness of the proposed approach is detailed in this section. The various parameters for the proposed approach are set as follows: The maximum step size for MBO $S_{max} = 1.0, Bar = 5/12$ (Butterfly adjusting rate), migration ratio $p = 5/12$ and $period = 1.2$ (migration period) is chosen. The value of maximum generation $maxGen$ is set to 50 and 30 monarch butterflies are considered i.e. $N = 30$. Further, α and β used in fitness computation are set to 0.9 and 0.1 respectively. The default parameter settings are employed.

TABLE 5.2: Benchmark datasets

Dataset	Instance	Feature	Class
Glass	214	9	6
Appendicitis	106	7	2
Wine	178	13	3
Seeds	210	7	3
Leaf	340	14	30
Lymphography	148	18	4
Lenses	24	4	3
Flags-religion	194	28	8
Fertility-diagnosis	100	9	2
Zoo	101	16	7
Dbworld-bodies	64	4702	2
Statlog-german-credit	1000	20	2

5.2.1 Results

Series of experiments are conducted to illustrate the performance of our proposed approach. A comparison with existing missing value imputation and feature selection methods is undertaken to demonstrate the effectiveness of our approach. All the eleven complete datasets are taken from UCI machine learning repository [108] and are summarised in table 5.2 while the ten datasets containing missing values (as shown in table 5.3) are from UCI, open ML [148] and open MV net[1].The results are validated using 10×10 fold cross validation technique, wherein features are selected during each training fold and results are validated for the test set employing the reduced set of features. Two classifiers namely $kNN(k = 3)$ [131] and SVM [117] are employed to evaluate performances and the corresponding highest results are made bold-faced.

[1] `https://openmv.net/tag/missing-data`

TABLE 5.3: Benchmark datasets containing missing values

Dataset	Instance	Feature	Missing Value Percentage	Class
Breast-cancer-wisconsin	699	10	0.23	2
Bands	539	19	5.38	2
Cleveland	303	13	0.15	5
Sick	3772	28	2.17	2
Hepatitis	155	19	5.67	2
Housevotes	435	16	5.63	2
Mammographic	961	5	3.37	2
Class grades	99	5	0.81	5
Travel times	205	9	0.43	2
Dermatology	366	34	0.06	6

5.2.1.1 By employing Parameter variation

To decide the values the parameters involved in the proposed approach, a number of experiments are performed by varying the parameters over range of values. The parameters employed are level of induced missing values, k (number of nearest neighbour used for imputing missing values), percentage of correlated features $c\%$ and percentage of missing values in an instance to ignore the same. The best values of these parameters are determined and employed for subsequent experimentation.

By varying the level of induced missing Values One of the efficient way to illustrate the effectiveness of the proposed imputed feature selection approach is to randomly induce missing values at varying percentage and compare performance with original dataset. Eleven datasets are employed for which 10%, 20%, 25% and 30% values are randomly missed. The number of features selected and the corresponding accuracy are depicted in table 5.4 and 5.5 and Figure 5.2. For 10% missing values level, the proposed approach with imputing the missing values and reducing

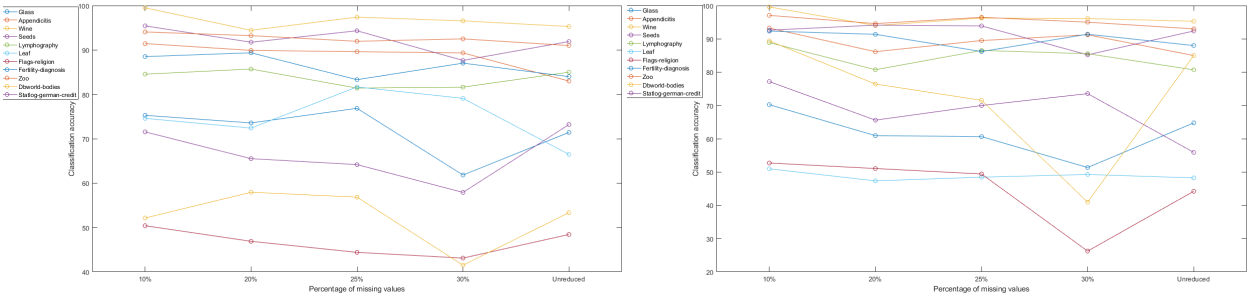
TABLE 5.4: Number of features selected by varying percentage of missing values

Dataset	**10%**	**20%**	**25%**	**30%**
Glass	6.8	7	7	1
Appendicitis	5	5.3	3	5.9
Wine	6	6.8	6	6.2
Seeds	4.4	4.9	4.9	4.0
Lymphography	7.5	7.4	7.7	7.8
Leaf	11.0	12.1	12	12
Flags-religion	14.6	13.5	13	1
Fertility-diagnosis	6.8	6.2	6.4	6.1
Zoo	5.0	5.2	5.2	5.3
Dbworld-bodies	1830.6	1826.4	1817.8	1809.8
Statlog-german-credit	11.6	1.0	1.0	1.0

data dimensionality enhances classification performance for nearly all the datasets and the difference is negligible for the ones showing similar performances. All the experiments are therefore conducted at 10% missing level. At 20% and 25% missing level, similar classification accuracy is seen for most of the datasets except for Leaf at 25% which might have resulted because of noise. Glass, Seeds, Flags-religion, Dbworld-bodies and Statlog-german-credit demonstrate a bit decrease in accuracy at 30% missing level that might have resulted because of selection of too few features by MBO feature selection algorithm and the removal of most of the correlated and important information by missing level for these datasets. However, for other datasets there is significant increase in classification performance than original illustrating that not only the proposed model imputes missing values accurately but also enhances performances by selecting apt features.

TABLE 5.5: Comparison of proposed approach on varying percentage of missing values

Dataset		10%	20%	25%	30%	Complete dataset
Glass	KNN	75.27±2.28	73.56±2.99	**76.85±3.52**	61.82±5.39	71.42±8.97
	SVM	**70.23±4.76**	60.93±2.22	60.63±3.87	51.35±4.66	64.76±9.57
Appendicitis	KNN	**91.44±3.66**	89.91±3.12	89.62±0.61	89.34±3.70	83.00±9.48
	SVM	**93.25±1.87**	86.15±4.68	89.51±1.33	91.23±4.11	85.00±7.07
Wine	KNN	**99.50±0.45**	94.43±1.24	97.40±1.43	96.56±1.32	95.29±6.67
	SVM	**99.60±0.59**	94.01±2.67	96.26±1.51	96.11±0.87	95.29±5.40
Seeds	KNN	**95.43±0.45**	91.75±1.41	94.33±2.63	87.65±2.20	91.90±5.04
	SVM	92.62±1.45	**94.13±1.39**	93.9±0.99	85.21±4.91	92.38±4.60
Lymphography	KNN	84.55±4.04	**85.71±3.05**	81.41±2.49	81.63±1.54	85.00±10.35
	SVM	**88.88±3.26**	80.74±2.17	86.54±2.06	85.55±0.89	80.71±12.16
Leaf	KNN	74.58±5.26	72.40±1.13	**81.67±1.67**	79.10±1.44	66.47±5.03
	SVM	**50.96±3.35**	47.34±3.73	48.45±2.43	49.26±1.51	48.23±9.32
Flags-religion	KNN‘	**50.39±4.84**	46.88±2.98	44.39±1.52	43.10±6.00	48.42±10.46
	SVM	**52.70±2.85**	51.05±2.79	49.41±3.49	26.23±6.02	44.21±9.98
Fertility-diagnosis	KNN	88.51±4.93	**89.37±3.92**	83.31±1.94	87.04±6.23	84.00±13.39
	SVM	**92.34±3.40**	91.37±1.41	86.20±1.56	91.40±4.24	88.00±13.16
Zoo	KNN	**94.06±1.91**	93.23±2.64	91.96±4.55	92.50±5.74	91.00±8.75
	SVM	**97.06±2.08**	94.59±2.05	96.49±2.11	95.02±3.56	93.00±8.23
Dbworld-bodies	KNN	52.13±7.23	**57.94±10.29**	56.83±3.90	41.50±10.30	53.33±20.48
	SVM	**89.39±2.65**	76.43±4.53	71.54±5.79	40.95±10.92	85.00±5.27
Statlog-german-credit	KNN	71.54±1.10	65.51±1.69	64.17±3.07	57.90±7.60	**73.20±5.32**
	SVM	**77.17±0.08**	65.58±1.76	69.98±0.81	73.55±2.24	55.90±13.85



KNN SVM

FIGURE 5.2: Graphical visualization showing comparison of the classification accuracy on varying percentage of missing values

By varying the value of k (number of nearest neighbour for missing value imputation) The exerimentation is done over range of values of k and henceforth the best value is determined. Since, k is used for imputing missing values, the number of features selected is identical for all the datasets as shown in table 5.6. Maximum classification accuracy is obtained for $k = 3$ with similar performance for some datasets except for Dbworld-bodies (table 5.7). The results imply that nearest three

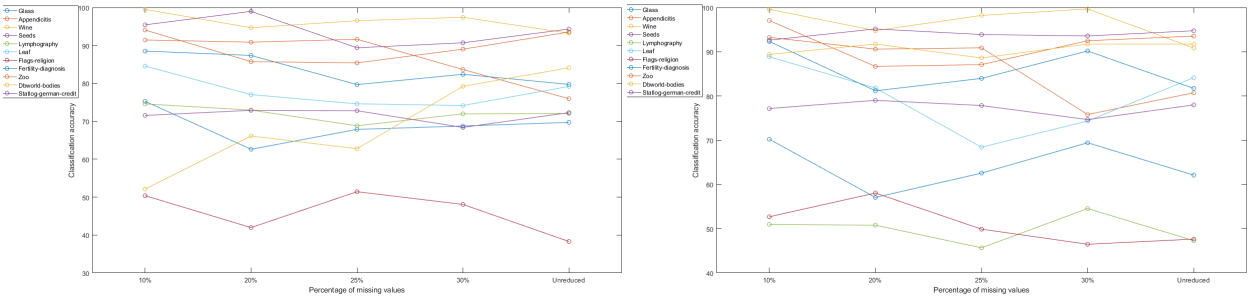
TABLE 5.6: Number of features selected by varying the value of k (number of nearest neighbour for missing value imputation

Dataset	$k = 3$	$k = 5$	$k = 7$	$k = 10$	$k = 15$
Glass	6.8	6.8	6.9	7.0	7.0
Appendicitis	5.0	5.0	5.1	4.2	5.0
Wine	6.0	6.4	6.4	6.3	6.5
Seeds	4.4	4.5	5.0	4.9	5.0
Leaf	11	10.9	11.9	12.8	11.6
Lymphography	7.5	7.8	7.7	8.5	7.7
Flags-religion	14.6	13.2	14.6	14.4	14.9
Fertility-diagnosis	6.8	6.4	6.5	6.5	6.1
Zoo	5.0	6.0	4.8	4.9	4.7
Dbworld-bodies	1830.6	1827.7	1830.6	1827.7	1827.7
Statlog-german-credit	11.6	11.9	12.0	12.0	11.8

TABLE 5.7: Comparison of proposed approach on varying the value of k (number of nearest neighbour for missing value imputation)

Dataset		$k = 3$	$k = 5$	$k = 7$	$k = 10$	$k = 15$
Glass	KNN	**75.27±2.28**	62.62±2.65	67.86±3.69	68.75±3.14	69.71±2.82
	SVM	**70.23±4.76**	57.07±2.95	62.57±2.08	69.43±4.56	62.12±1.41
Appendicitis	KNN	91.44±3.66	90.86±2.29	**91.63±3.17**	83.65±5.29	75.96±5.92
	SVM	**93.25±1.87**	90.59±2.11	90.88±3.50	75.81±5.81	80.73±2.02
Wine	KNN	**99.50±0.45**	94.67±2.82	96.53±1.38	97.43±1.45	93.24±2.13
	SVM	99.60±0.59	94.77±2.13	98.22±1.49	**99.68±0.58**	90.85±3.93
Seeds	KNN	95.43±0.45	**98.99±1.51**	89.38±2.26	90.70±0.95	94.33±2.69
	SVM	92.62±1.45	**95.16±1.77**	93.89±1.45	93.57±1.54	94.75±2.42
Leaf	KNN	**74.58±5.26**	72.95±3.19	68.81±1.67	71.96±2.98	72.06±1.44
	SVM	50.96±3.35	50.8±2.47	45.68±2.12	**54.56±2.04**	47.31±1.09
Lymphography	KNN	**84.55±4.04**	77.03±5.80	74.60±4.07	74.16±4.14	79.19±3.12
	SVM	**88.88±3.26**	81.81±2.55	68.43±2.11	74.35±2.90	84.15±6.12
Flags-religion	KNN‘	50.39±4.84	41.96±1.45	**51.43±2.67**	48.07±2.54	38.30±5.50
	SVM	52.70±2.85	**58.07±8.33**	49.90±3.11	46.49±4.34	47.65±2.46
Fertility-diagnosis	KNN	**88.51±4.93**	87.36±3.20	79.67±2.42	82.39±1.89	79.73±1.73
	SVM	**92.34±3.40**	81.17±4.82	83.98±2.47	90.18±3.76	81.72±2.84
Zoo	KNN	**94.06±1.91**	85.72±3.62	85.40±3.22	89.01±3.22	93.54±2.48
	SVM	**97.06±2.08**	86.69±4.33	87.09±3.88	92.49±1.61	93.54±2.48
Dbworld-bodies	KNN	52.13±7.23	66.10±7.96	62.77±7.15	79.22±6.70	**84.10±6.64**
	SVM	89.39±2.65	**91.74±4.73**	88.60±2.64	**91.74±4.73**	**91.74±4.73**
Statlog-german-credit	KNN	71.54±1.10	**72.89±0.76**	72.78±2.23	68.38±2.15	72.28±1.38
	SVM	77.17±0.08	**79.02±0.91**	77.85±3.20	74.67±0.60	77.99±1.06

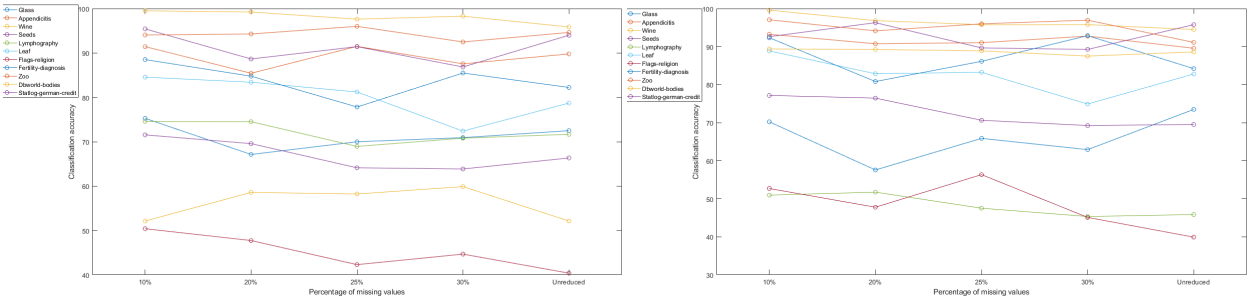
instances are most crucial for imputing missing values. The visual representation for better understanding is given in Figure 5.3.



KNN SVM

FIGURE 5.3: Graphical visualization showing comparison of the classification accuracy by varying the value of k (number of nearest neighbour for missing value imputation)

By varying the percentage of correlated features employed for obtaining reduced dataset using proposed approach The number of features used to impute missing values is estimated by finding $c\%$, i.e. features whose correlation is greater than $c\%$ of $correlation_{max}$. The value of $c\%$ is varied from 50% to 90% in step of 10, as recorded in table 5.8 and 5.9 and visualized in Figure 5.4. As the value of $c\%$ increases, the number of features used for imputation decreases. Highest accuracy is achieved for $c\%$ set to 50% and is henceforth used for experimentation. High performance for other values of $c\%$ for most of datasets suggest that those very features are most significant for imputation.



KNN SVM

TABLE 5.8: Number of features selected by varying the percentage of correlated features employed for obtaining reduced dataset using proposed approach

Dataset	**Percentage of correlated features employed for obtaining reduced dataset using proposed approach**				
	50%	**60%**	**70%**	**80%**	**90%**
Glass	6.8	6.8	7.1	6.7	7.6
Appendicitis	5.0	5.0	4.9	5.5	4.8
Wine	6.0	7.0	6.9	6.4	6.7
Seeds	4.4	4.0	5.0	4.6	5.0
Leaf	11	10.9	11.4	11.0	11.0
Lymphography	7.5	7.6	7.9	8.1	7.6
Flags-religion	14.6	14.3	13.7	14.3	13.4
Fertility-diagnosis	6.8	5.8	6.4	6.6	6.3
Zoo	5.0	5.3	4.6	5.0	4.4
Dbworld-bodies	1830.6	1827.7	1827.7	1814.9	1830.6
Statlog-german-credit	11.6	11.9	1.0	1.0	1.0

TABLE 5.9: Comparison of proposed approach on varying the percentage of correlated features employed for obtaining reduced dataset using proposed approach

Dataset		**Percentage of correlated features employed for obtaining reduced dataset using proposed approach**				
		50%	**60%**	**70%**	**80%**	**90%**
Glass	KNN	**75.27±2.28**	67.13±2.49	69.99±3.2	70.92±2.96	72.49±5.42
	SVM	70.23±4.76	57.58±1.31	65.88±2.62	62.92±2.30	**73.49±1.77**
Appendicitis	KNN	91.44±3.66	85.43±2.09	**91.49±4.77**	87.56±2.81	89.79±0.99
	SVM	**93.25±1.87**	90.76±3.28	91.05±6.17	92.77±1.32	89.58±1.14
Wine	KNN	**99.50±0.45**	99.23±0.57	97.62±1.33	98.30±0.76	95.86±1.07
	SVM	**99.60±0.59**	96.84±1.73	95.77±1.67	95.79±1.71	94.50±1.40
Seeds	KNN	**95.43±0.45**	88.65±1.83	91.43±4.06	86.85±2.29	94.00±1.47
	SVM	92.62±1.45	**96.30±2.13**	89.67±3.35	89.27±1.00	95.77±0.61
Leaf	KNN	**74.58±5.26**	74.51±1.84	68.94±1.32	70.80±1.00	71.67±1.3
	SVM	50.96±3.35	**51.74±2.71**	47.52±2.03	45.33±1.46	45.85±0.60
Lymphography	KNN	**84.55±4.04**	83.41±1.55	81.21±3.55	72.39±6.61	78.71±5.14
	SVM	**88.88±3.26**	82.88±2.54	83.26±2.72	74.91±3.11	82.81±3.91
Flags-religion	KNN‘	**50.39±4.84**	47.73±2.10	42.30±1.87	44.67±4.70	40.38±5.77
	SVM	52.70±2.85	47.78±3.35	**56.35±3.5**	45.10±3.07	39.91±6.01
Fertility-diagnosis	KNN	**88.51±4.93**	84.77±3.22	77.82±3.21	85.49±4.01	82.23±3.95
	SVM	92.34±3.40	80.83±4.75	86.14±2.69	**92.89±1.63**	84.22±3.03
Zoo	KNN	94.06±1.91	94.28±2.49	**96.00±1.69**	92.49±2.98	94.63±2.62
	SVM	**97.06±2.08**	94.15±2.62	96.00±1.69	96.96±1.94	91.12±2.61
Dbworld-bodies	KNN	52.13±7.23	58.58±6.55	58.22±6.80	**59.89±4.38**	52.13±7.23
	SVM	**89.39±2.65**	89.25±6.46	88.90±6.70	87.52±3.96	88.60±2.64
Statlog-german-credit	KNN	**71.54±1.10**	69.57±1.66	64.13±0.67	63.86±1.29	66.34±2.34
	SVM	**77.17±0.08**	76.46±1.09	70.64±1.03	69.25±4.29	69.54±0.26

TABLE 5.10: Number of features selected by varying the percentage of missing entries required in an instance to ignore the instance

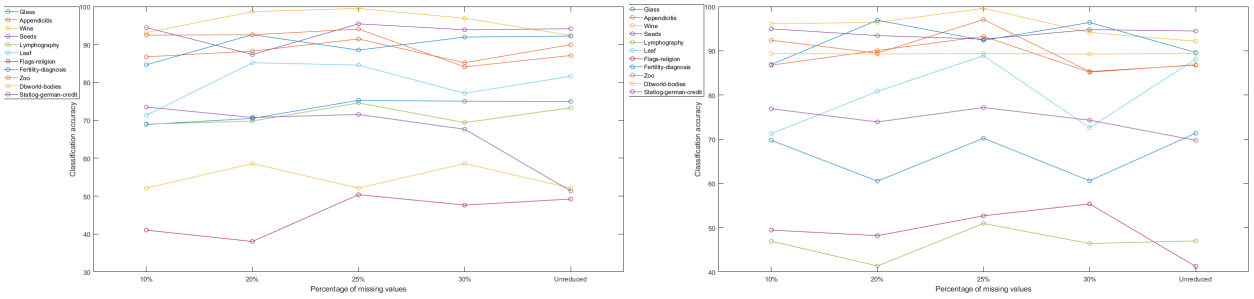
Dataset	**Percentage of missing entries required in an instance to ignore the instance**				
	50%	**60%**	**70%**	**80%**	**90%**
Glass	6.9	6.5	6.8	6.3	7.1
Appendicitis	5.0	4.9	5.0	4.2	4.0
Wine	7.0	6.0	6.0	7.0	6.2
Seeds	4.7	4.8	4.4	4.9	5.0
Leaf	11.6	11.7	11	11.3	11.0
Lymphography	7.4	7.4	7.5	8.2	7.7
Flags-religion	14.1	15.0	14.6	14.3	14.3
Fertility-diagnosis	6.7	6.7	6.8	6.3	6.7
Zoo	5.6	4.8	5.0	5.5	5.4
Dbworld-bodies	1830.6	1827.7	1830.6	1827.7	1830.6
Statlog-german-credit	11.9	11.9	11.6	12.1	1.0

FIGURE 5.4: Graphical visualization showing comparison of the classification accuracy by varying percentage of correlated features employed for obtaining reduced dataset using proposed approach

By varying the percentage of missing entries required in an instance to ignore the instance The percentage of missing value in an instance to ignore the instance is varied and tabulated in table 5.10 and 5.11. The number of features selected by feature selection is nearly same for all datasets. The one giving highest performance for most of the datasets i.e. 70% is chosen. High performance for few datasets at 50% or 60% suggests that those instances might be outliers that are ignored thereby enhancing accuracy. Variation of accuracy is visualized in Figure 5.5.

TABLE 5.11: Comparison of proposed approach on varying the percentage of missing entries required in an instance to ignore the instance

Dataset		Percentage of missing entries required in an instance to ignore the instance				
		50%	**60%**	**70%**	**80%**	**90%**
Glass	KNN	68.89±4.58	70.50±3.96	**75.27±2.28**	75.02±3.18	74.95±2.03
	SVM	69.75±3.00	60.55±3.98	70.23±4.76	60.61±3.02	**71.37±4.94**
Appendicitis	KNN	86.74±2.68	88.24±3.21	**91.44±3.66**	85.18±3.15	89.94±4.41
	SVM	86.74±2.68	90.06±2.19	**93.25±1.87**	85.16±2.88	86.83±2.81
Wine	KNN	93.12±1.28	98.65±0.75	**99.50±0.45**	96.90±1.82	92.40±1.80
	SVM	96.10±1.97	96.41±1.96	**99.60±0.59**	94.13±1.35	92.15±3.55
Seeds	KNN	94.44±1.58	87.34±3.13	**95.43±0.45**	93.88±1.61	94.13±1.76
	SVM	**94.95±0.84**	93.43±1.32	92.62±1.45	94.78±1.49	94.46±1.68
Leaf	KNN	69.00±2.16	69,79±1.73	**74.58±5.26**	69.39±1.13	73.31±1.57
	SVM	46.95±1.01	41.36±1.13	**50.96±3.35**	46.43±2.41	47.03±0.75
Lymphography	KNN	71.33±5.48	**85.17±3.24**	84.55±4.04	77.11±5.21	81.62±8.16
	SVM	71.27±3.20	80.82±2.41	**88.88±3.26**	72.62±2.36	88.11±5.58
Flags-religion	KNN‘	41.03±5.76	38.04±5.13	**50.39±4.84**	47.64±6.45	49.23±3.34
	SVM	49.47±3.46	48.20±7.03	52.70±2.85	**55.38±10.69**	41.23±7.25
Fertility-diagnosis	KNN	84.62±2.77	**92.57±4.44**	88.51±4.93	91.93±3.61	92.22±4.53
	SVM	86.90±3.91	**96.91±2.00**	92.34±3.40	96.39±2.26	89.61±1.67
Zoo	KNN	92.45±4.60	92.53±1.95	**94.06±1.91**	84.09±3.18	87.08±6.93
	SVM	92.37±4.64	89.46±4.36	**97.06±2.08**	85.31±3.59	86.75±6.77
Dbworld-bodies	KNN	52.11±7.23	**58.58±6.55**	52.13±7.23	58.58±6.55	52.13±7.23
	SVM	**89.39±2.65**	89.25±6.46	**89.39±2.65**	89.25±6.46	**89.39±2.65**
Statlog-german-credit	KNN	**73.50±2.41**	70.74±2.11	71.54±1.10	67.64±0.88	51.36±3.63
	SVM	76.88±1.25	73.91±1.28	**77.17±0.08**	74.30±0.72	69.76±1.05



KNN SVM

FIGURE 5.5: Graphical visualization showing comparison of the classification accuracy by varying percentage of missing entries required in an instance to ignore the instance

5.2.1.2 Comparison with Existing Missing Value Imputation method

The proposed missing data imputation methodology is compared with five other imputation techniques namely Ignore, K-means, KNN, Most-common and SVM-impute. All these previous imputation are implemented in KEEL software [1]. The

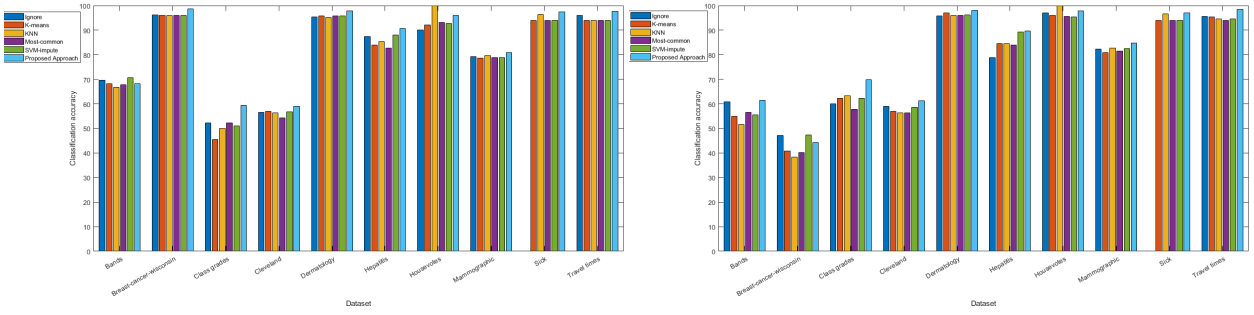
TABLE 5.12: Comparison with other missing value imputation algorithms

Dataset		Ignore	K-means	KNN	Most-common	SVM-impute	Proposed approach
Breast-cancer-wisconsin	KNN	96.17±2.51^{2}	96.08±2.56^{4}	95.94±1.33^{6}	96.08±3.05^{5}	96.08±2.05^{3}	**98.06±0.90**^{1}
	SVM	47.05±9.40^{2}	40.72±5.08^{4}	38.26±5.64^{6}	40.14±6.73^{5}	**47.39±19.03**^{1}	44.30±2.09^{3}
Bands	KNN	69.72±8.92^{2}	68.11±5.44^{4}	66.79±5.12^{6}	67.73±6.12^{5}	**70.75±6.11**^{1}	68.12±2.86^{3}
	SVM	60.83±17.39^{2}	54.90±6.85^{5}	51.69±8.58^{6}	56.60±8.93^{3}	55.47±5.70^{4}	**61.47±1.80**^{1}
Cleveland	KNN	56.55±8.63^{4}	57.00±6.93^{2}	56.33±10.11^{5}	54.33±9.55^{6}	56.67±10.54^{3}	**59.10±3.01**^{1}
	SVM	58.96±8.36^{2}	57.00±6.93^{4}	56.33±10.23^{6}	56.33±9.08^{5}	58.66±8.49^{3}	**61.32±3.32**^{1}
Sick	KNN	All removed^{6}	93.87±1.37^{4}	96.44±0.68^{2}	93.89±1.71^{3}	93.87±1.57^{5}	**97.47±4.35**^{1}
	SVM	All removed^{6}	93.87±1.37^{4}	96.55±0.97^{2}	93.89±1.71^{3}	93.87±1.57^{5}	**97.08±1.87**^{1}
Hepatitis	KNN	87.50±11.78^{3}	84.00±11.84^{5}	85.33±8.77^{4}	82.66±14.12^{6}	88.00±8.19^{2}	**90.62±3.60**^{1}
	SVM	78.75±13.24^{6}	84.66±12.59^{4}	84.55±11.35^{5}	84.00±10.03^{3}	89.33±5.62^{2}	**89.61±2.98**^{1}
Housevotes	KNN	90.00±5.44^{6}	92.09±3.98^{5}	**100.0±0.00**^{1}	93.25±4.16^{3}	92.79±3.86^{4}	95.97±1.88^{2}
	SVM	96.95±2.93^{3}	96.04±1.91^{4}	**100.0±0.00**^{1}	95.58±2.55^{5}	95.34±2.45^{6}	97.92±0.95^{2}
Mammographic	KNN	79.15±3.26^{3}	78.54±3.90^{6}	79.68±2.87^{2}	78.85±2.98^{4}	78.85±5.19^{5}	**80.89±2.15**^{1}
	SVM	82.28±3.85^{4}	80.83±3.15^{6}	82.81±3.61^{2}	81.45±3.76^{5}	82.50±4.33^{3}	**84.83±1.32**^{1}
Class grades	KNN	52.22±16.60^{2}	45.55±16.10^{6}	50.00±16.76^{5}	52.22±17.41^{3}	51.11±15.88^{4}	**59.36±3.64**^{1}
	SVM	60.00±14.99^{5}	62.22±16.72^{3}	63.33±14.86^{2}	57.77±15.53^{6}	62.22±18.29^{4}	**69.86±3.24**^{1}
Travel times	KNN	96.11±4.57^{2}	94.00±4.59^{3}	94.00±5.16$^{4.5}$	94.00±6.58^{6}	94.00±5.16$^{4.5}$	**97.75±1.55**^{1}
	SVM	95.55±3.51^{2}	95.50±3.68^{3}	94.50±5.50^{4}	94.00±5.16^{6}	94.50±5.98^{5}	**98.49±0.84**^{1}
Dermatology	KNN	95.42±3.85^{5}	95.83±3.00^{3}	95.27±2.63^{6}	95.83±2.69^{2}	95.83±3.27^{4}	**97.78±0.59**^{1}
	SVM	95.71±3.36^{6}	96.94±2.04^{2}	96.11±2.34^{5}	96.11±1.94^{4}	96.12±2.98^{3}	**97.97±0.98**^{1}
Average Rank	KNN	3.5	4.2	4.15	4.3	3.55	1.3
	SVM	3.8	3.9	3.9	4.5	3.6	1.3
F statistics	KNN	5.18					
	SVM	5.27					

experimental results are shown in table 5.12. Missing values are imputed for the datasets of table 5.3 and the resulting dataset is evaluated for classification accuracy. The obtained results clearly demonstrate that the proposed imputation approach outperforms the existing imputation methodologies. For Breast-cancer-wisconsin and Bands datasets SVM-impute has performed better for a classifier while for Housevotes, KNN has achieved highest accuracy but the increase is insignificant on comparison with number of times proposed missing value imputation approach has outperformed other works that can be justified using statistical hypothesis testing. The value of $F(5, 45) = 2.43$ at $\alpha = 5\%$ level for significance, therefore the null hypothesis is rejected using Freidman test, i.e. six algorithms are statistically different. For Bonferroni Dunn test, $q_{0.05} = 2.576$ so $Cd_{0.05} = 2.15$. Hence, proposed approach is statistically better than other missing imputation techniques for both the classifiers at 5% level of significance.

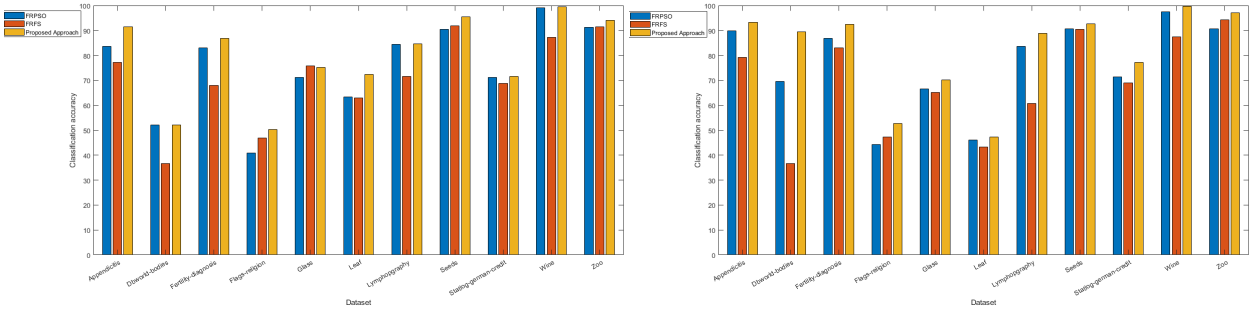
TABLE 5.13: Number of features selected on comparison with other state of art feature selection algorithms

Dataset	FRPSO	FRFS	Proposed approach
Glass	9	7.7	**6.8**
Appendicitis	7	5.9	**5**
Wine	10	7.7	**6**
Seeds	7	6.0	**4.4**
Lymphography	11.5	8.8	**7.5**
Leaf	14	11.3	**11**
Flags-religion	19.1	14.7	**14.6**
Fertility-diagnosis	9	**6.0**	6.8
Zoo	7.2	6.8	**5**
Dbworld-bodies	2274.9	2277.1	**1830.6**
Statlog-german-credit	14.8	14.4	**11.6**



KNN SVM

FIGURE 5.6: Graphical visualization showing comparison of the classification accuracy with other missing value imputation algorithms



KNN SVM

FIGURE 5.7: Graphical visualization showing comparison of the classification accuracy with other feature selection algorithms

TABLE 5.14: Comparison of classification accuracies with other state of art feature selection algorithms

Dataset		**FRPSO**	**FRFS**	**Proposed approach**
Glass	KNN	71.20±3.63[3]	**75.85±4.95**[1]	75.27±2.28[2]
	SVM	66.48±2.04[2]	65.14±3.55[3]	**70.23±4.76**[1]
Appendicitis	KNN	83.68±2.30[2]	77.30±2.33[3]	**91.44±3.66**[1]
	SVM	89.93±1.22[2]	79.13±2.83[3]	**93.25±1.87**[1]
Wine	KNN	99.10±0.60[2]	87.15±5.66[3]	**99.50±0.45**[1]
	SVM	97.47±1.81[2]	87.50±2.55[3]	**99.60±0.59**[1]
Seeds	KNN	90.39±2.16[3]	91.95±1.11[2]	**95.43±0.45**[1]
	SVM	90.62±2.67[2]	90.45±1.88[3]	**92.62±1.45**[1]
Lymphopgraphy	KNN	84.46±1.09[2]	71.62±9.03[3]	**84.55±4.04**[1]
	SVM	83.63±1.54[2]	60.77±6.49[3]	**88.88±3.26**[1]
Leaf	KNN	63.30±2.38[2]	62.92±2.17[3]	**72.42±3.62**[1]
	SVM	46.11±2.97[2]	43.19±1.62[3]	**47.22±3.68**[1]
Flags-religion	KNN	40.95±4.91[3]	46.96±2.99[2]	**50.39±4.84**[1]
	SVM	44.30±1.85[3]	47.35±2.63[2]	**52.70±2.85**[1]
Fertility-diagnosis	KNN	83.06±1.96[2]	68.02±8.93[3]	**88.51±4.93**[1]
	SVM	86.80±2.03[2]	83.05±2.82[3]	**92.34±3.40**[1]
Zoo	KNN	91.17±1.04[3]	91.38±2.14[2]	**94.06±1.91**[1]
	SVM	90.74±1.30[3]	94.31±3.27[2]	**97.06±2.08**[1]
Dbworld-bodies	KNN	52.06±2.40[2]	36.69±8.68[3]	**52.11±7.23**[1]
	SVM	69.55±4.98[2]	36.69±8.68[3]	**89.39±2.65**[1]
Statlog-german-credit	KNN	71.11±1.82[2]	68.74±1.78[3]	**71.54±1.10**[1]
	SVM	71.47±4.06[2]	69.06±4.45[3]	**77.17±0.08**[1]
Average Rank	KNN	2.36	2.54	1.09
	SVM	2.18	2.81	1.0
F statistics	KNN	16.64		
	SVM	54.20		

5.2.1.3 Comparison with Other Feature Selection Algorithms

This section details the comparison of proposed approach with other state of art feature selection algorithms. The fuzzy rough particle swarm optimization (FRPSO) [104] and the fitting model for fuzzy rough feature selection (FRFS) with particle swarm optimization [154] are employed for comparison with proposed approach at 10% missing level. The experimental results are shown in table 5.13 and 5.14 and figure 5.7. The number of features selected by FRPSO and FRFS is high for all of

the datasets except Fertility-diagnosis. There is a significant increase in classification accuracy for all the datasets except for Glass for a single classifier with the proposed approach. However, the decrease in classification accuracy for Glass is negligible. Enhancing the classification performance while imputing missing values and then selecting relevant non-redundant features clearly demonstrates the superiority of the proposed approach. For statistical testing, $M = 11, N = 3$, so the value of $F(2, 20) = 3.49$ is used. The null hypothesis is rejected implying the significant difference between algorithms. Here, $q_{0.05} = 2.241$ such that $Cd_{0.05} = 1.0$. The null hypothesis is again rejected by Bonferroni Dunn test for both classifiers demonstrating the superiority of the proposed approach.

5.3 Summary

Databases with missing values are very common in numerous industrial and research areas. Missing values are incorporated in datasets due to incorrect measurements, non-response in surveys, faulty entering of data, malfunctions of instruments, and experimental errors in the laboratories, etc.

The current work employs missing data imputation methodology followed by feature selection using fuzzy rough set. It provides an effective way to pre-process data and reduce the dimensionality thereby enhancing classification performance and reducing computational complexity. The idea of missing data estimation and instance ignorance are combined for data imputation. Monarch butterfly optimization search heuristic was employed for fuzzy rough set based feature selection to achieve optimal results. The effectiveness of the proposed approach is demonstrated on various benchmark datasets. The performance of the proposed technique is assessed using well known SVM and kNN classifiers. Moreover, the proposed approach is compared

against state-of-the-art methods. Finally, Freidman and Bonferoni Dunn tests are performed to demonstrate the significance of the proposed approach. An efficient data pre-processor like the one proposed in this chapter is required for attaining high performance in pattern recognition problems. To the best of our knowledge, none of the previous approaches have addressed this combined issue.

Since the most correlated features are employed for imputation of missing values, the imputation technique in general is quite efficient. Moreover, monarch butterfly optimisation decreases the computational time as only the most relevant subset are evaluated for quality avoiding the need for any exhaustive, forward or backward search. However, data size may increase both in terms of features and instances. Feature selection or instance selection alone cannot handle the ever increasing size and dimensionality of dataset. Both the aspects of data reduction must be taken into consideration for enhancing classification accuracy (as discussed in next chapter).

****