Optimal Feature Subset Selection Using Differential Evolution and Extreme Learning Machine

Bharathi P T[1], Dr. P. Subashini[2]

[1]Research Scholar, Department of Computer Science, Avinashilingam Institute for Home Science And Higher Education for Women, Coimbatore, India

[2]Professor, Department of Computer Science, Avinashilingam Institute for Home Science And Higher Education for Women, Coimbatore, India

Abstract: ***Feature selection problem often occurs in pattern recognition and more specifically in classification. Features extracted from feature extraction methods could contain a large number of feature set. In original feature set, some of them can prove to be irrelevant, redundant and even unfavorable to classification accuracy. So it is essential to remove these type of features, which in turn leads to dimensionality reduction and could eventually improve the classification accuracy. The objective of feature selection is carried out in three steps. Firstly, improving the prediction performance of the predictors, secondly for providing faster and more cost-effective predictors, and finally providing a better understanding of the underlying process that generated the data. Considering river ice images, analyzing the different types of ice features and their characteristics is complex in nature. Hence, in this work feature extraction is carried out by computing Gray Level Co-occurrence Matrix (GLCM) features for 0^0, 45^0, 90^0 and 135^0 and feature subset selection is performed with Differential Evolution Feature Selection (DEFS) algorithm. DEFS, utilizes the DE float number optimizer in a combinatorial optimization problem like feature selection. DEFS feature selection method highly reduces the computational cost while at the same time proves to present a powerful performance and provides 93% accuracy with the features selected. Proposed method, Extreme Learning Machine (ELM) combined with DEFS (ELM_DEFS) technique selects best feature subset from the original feature set. Selected feature set is used for better simplification and training the classifier, to classify river ice types correctly. Features selected from the proposed method reduce 65% of the features and provide 97.78% accuracy for river ice images.***

Keywords: Feature Selection, Gray Level Co-occurrence Matrix, Differential Evolution Feature Selection, Extreme Learning Machine.

1. Introduction

Feature selection is an active research area in computer science. It has been a fruitful field of research and development since 1970s to till date in pattern recognition, machine learning and data mining [1]. Feature selection is defined as a process to select the best optimal subset of M features from the original set of *N* features, so that the new feature subset space is optimally reduced based on some evaluation criterion. When the dimensionalities of a domain expand, then the number of feature set *N* also increases. Finding the best feature subset is usually difficult.

Feature selection is a basic problem in many different areas, such as document classification, forecasting, object recognition, bioinformatics, and or in modeling of complex technological processes. Datasets with hundreds of thousands of features are not uncommon in such applications. All features could be important for some problems, but for some other problems a small subset of features is usually relevant.

Feature selection reduces the dimensionality of feature space, which removes redundant, irrelevant and or noisy data. Feature selection is carried out in two main purposes. Initially, it makes a training set and then applying a classifier more efficient by decreasing the size of the original feature set. This is of particular importance for classifiers, which are expensive to train a classifier. Secondly, feature selection often increases classification accuracy by eliminating redundant and irrelevant noise features. As a consequence feature selection can help us to avoid over fitting [2]. Feature selection algorithms are divided into filters, wrappers and embedded approaches. Filters approaches evaluate quality of selected features, independent from the classification algorithm. Wrapper approaches require application of a classifier to train the given feature set to evaluate this quality. Embedded approaches perform feature selection during learning of optimal parameters [3]. Some classification algorithms have inherited the ability to focus on relevant features and ignore irrelevant ones.

Researchers have studied various aspects of feature selection methods. In which, search is a key topic in the study of feature selection methods. Another important aspect is how to measure the goodness of a feature subset. According to class information available in the data set, there are supervised and unsupervised feature selection approaches.

Texture Analysis is an efficient measure to estimate the roughness, structural orientation, smoothness or regularity differences of diverse regions in an image scene. Textures are extracted using two categories of feature extraction methods, they are first order statistics and second order statistics. Second order statistics feature extraction methods are, Gray Level Co-occurrence Matrix (GLCM) [4] and Gray Level Run Length Matrix (GLRLM) [5]. In most of the studies related to Ice Classification only GLCM method is used. In this work, first order statistics and second order statistics are considered. The list of various features extracted under each category is shown in Table 1.

Table 1: Features Extracted from River ice image

Methods		*Features Extracted*
First order statistics		Mean, Standard Deviation, Variance, Skewness, Kurtosis
Second order statistics	*GLCM*	Autocorrelation, Contrast, correlation, Energy, Entropy, Homogeneity, Sum of Squares, Sum Average, Sum Variance, Sum Entropy, Difference entropy, Difference variance, Information measures of correlation (1), Information measures of correlation (2), Inverse difference, Inverse difference normalized, Inverse difference moment normalized
	GLRLM	Short-run emphasis (SRE), Long-run emphasis (LRE), Gray level Non-uniformity (GLN), Run Percentage (RP), Run-length Non-uniformity (RLN), Low gray-level emphasis (LGRE), High Gray-level emphasis (HGRE)

The present work is organized as follows: Section 2 describes feature subset selection structure. Section 3 describes DEFS and ELM methods for feature selection. Section 4 describes the proposed methodology ELM based DEFS (ELM_DEFS) for optimal feature selection and section 5 discusses experimental results obtained with feature subset selection methods. Finally, section 6 concludes with some final remarks and all the references been made for completion of this work

2. Feature Subset Selection Structure

General architecture for most of the feature selection algorithms consists of four basic steps. The steps involved are subset generation, evaluation, stopping criterion and result validation. Feature selection algorithms initially consists of original feature set, evaluate it and loop until an ending criterion is satisfied. Finally, the subset found is validated by the classifier algorithm on real dataset.

Subset Generation: Subset generation is a search procedure, which generates subsets of features for evaluation. Forward addition and backward elimination are the two families for subset generation methods. Forward addition starts with an empty subset and keeps on adding features after features by local search and backward elimination consists of entire feature set and keeps on eliminating one by one based on some search condition. Non-deterministic search like evolutionary search is often used to build the subsets. The total number of candidate subsets is 2^N, where N is the number of features in the original data set, which makes exhaustive search through the feature space infeasible with even moderate N.

Subset Evaluation: Each subset generated by the generation procedure needs to be evaluated by an evaluation criterion and compared with the previous best subset with respect to this criterion. If new best subset is found then it replaces the previous best subset. A simple step for evaluating a subset is to consider the performance of the classifier algorithm when it runs with that subset.

Stopping criteria: Without a suitable stopping criterion, the feature selection process will run exhaustively before it stops. A feature selection process may stop, if one of the following reasonable criteria is satisfied [15]:

- *Predefined numbers of features is selected*
- *Predefined number of iterations is reached*
- *In case of addition (or deletion) of a feature fails to produce a better subset*
- *Optimal subset according to the evaluation criterion is obtained*

Validation: The resultant best feature subset needs to be validated by carrying out different tests on both the selected subset and the original set and comparing the results using artificial data sets and/or real-world data sets.

Feature subset selection model normally incorporates a search approach for exploring the space of feature subsets. Genetic Algorithm (GA), Simulated Annealing (SA) can be used to explore better search space. In recent years, there has been a growing interest in evolutionary algorithms for diverse fields of Science and Engineering. The differential evolution feature selection algorithm (DEFS) is a relatively novel optimization technique to solve numerical-optimization problems. In this work, Extreme Learning Machine (ELM) combined with DEFS technique is used for feature subset selection.

3. DEFS and ELM for Feature Selection

From past few decades, it has been observed that Artificial Neural Networks (ANN) play a major role in image classification and pattern recognition applications. It is because of their generalization and conditioning requirement of minimal training points and faster convergence time. ANNs are found to perform better and results in faster output in comparison with that of the conventional classifiers. Selection time incurred due to preprocessing speed delay is the limitations and to increase classification accuracy, more training data is utilized in comparison with that of testing data are found in conventional classifiers. ANN is to be addressed with improving the training performance and better classification accuracy are noted in neural network architecture. The limitations of conventional classifier are overcome by using ELM, which handles the training for single hidden layer feed forward neural networks.

2.1 Conventional Extreme Learning Machine (ELM)

ELM Classifier is a single hidden layer feed forward neural network. The weights for the input layer, hidden layer and biases are randomly assigned without any training process. Moore-Penrose inverse and norm least square solution are used for calculating the output weights which reduces the training time of the ELM network [11], [13]. ELM is best match for larger training samples. This classifier is compared with that of the conventional neural network classifiers using the classification rate for river ice images and to classify ice types. Figure 1, shows the basic ELM architecture.

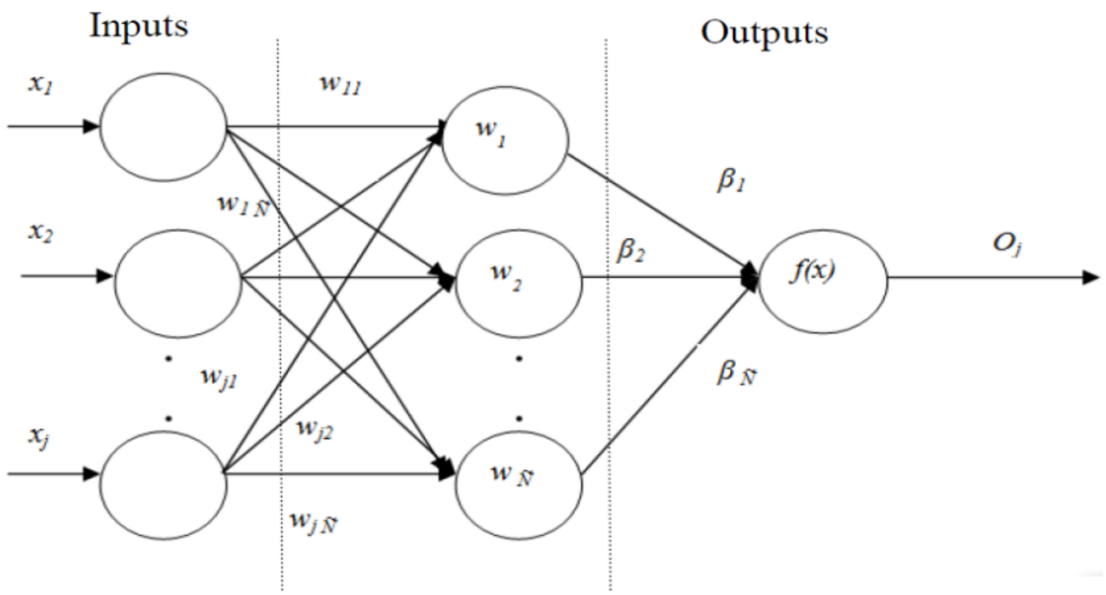

Figure 1: ELM Architecture

Conventional ELM classifier algorithm is given as:
Given a training set $N = \{(x_i, t_i) | x_i \in R_n, t_i \in R_m, i = 1, \ldots, N\}$, kernel function *f(x)*, and hidden neuron $\tilde{N}$.

Step1: Select activation function and number of hidden neurons Ñ for the given problem.
Step 2: Assign arbitrary input weight w_i *and bias* b_{i1} *, i= 1,…, H*
Step 3: Calculate output matrix H at the hidden layer

$$\sum_{i=1}^{\tilde{N}} \beta_i f(w_i, x_j + b_i) = t_j, j = 1, \ldots, N \qquad (1)$$

$$H = f \cdot (w \oplus x + b)$$

Step 4: Calculate the output weight β as:

$$\beta' = H^{\dagger} T \qquad (2)$$

where $H^{\dagger}$ *is the Moore-Penrose generalized pseudo-inverse of hidden layer output matrix.*

3.2 Differential Evolution Feature Selection (DEFS)

DEFS is a population based simple optimization method with parallel and direct search, easy to use, good convergence, and fast implementation properties. Like genetic algorithms (GA), DEFS employs crossover and mutation operators as selection mechanisms. An important difference among GA and DEFS is that, GA works on the crossover operator which provides an exchange of information required for building better solutions but DE algorithm fully works on the mutation operation as its central procedure. The mutation operation is based on the differences of randomly sampled pairs of solutions within the population.

The first step in DEFS optimization method is to generate a population of NP members each of D-dimensional real valued parameters, where NP is the population size and D is the number of parameters to be optimized. The key idea behind DEFS method for generating trial parameter vectors are by adding the weighted difference vector between two population members x_{r1} and x_{r2} to a third member x_{r0}. The following equation shows how to merge three different randomly selected vectors to create a mutant vector, $V_{i,g}$ from the current generation g:

$$v_{i,g} = x_{r0,g} + F \times (x_{r1,g} - x_{r2,g}) \qquad (3)$$

where $F \in (0,1)$ is a scale factor that control the rate at which the population evolve. The index g indicates the generation to which a vector belongs. In totaling up, each vector is assigned a population index $'i'$ which runs from 0 to $NP-1$. Parameters inside vectors are indexed with $'j'$, which operates from 0 to $D-1$. In addition, DEFS employs uniform crossover in order to build testing vectors out of parameter values that have been copied from two different vectors. DEFS crosses each vector with a mutant vector, as specified in Eq. (4):

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } rand(0,1) \le C_r \quad or \\ x_{j,i,g} & otherwise \end{cases} \qquad (4)$$

where $u_{j,i,g}$ is the jth dimension from the ith trial vector along the current population g. The crossover probability $0 \le C_r \le 1$, is a user defined value that controls the fraction of parameter values that are copied from the mutant [8, 9].

In order to utilize the float number optimizer of DEFS in feature selection, a number of modifications have been suggested by Rami N. Khushaba [7]. Like all population-based optimizers DEFS attacks the starting point problem by sampling the objective function at multiple, randomly chosen initial points as original population. Thus an original population matrix of size $(NP \times DNF)$ containing NP randomly chosen initial vectors

x_i, $i = \{0, 1, 2, \dots, NP-1\}$ is created, where DNF is the desired number of features to be selected. In DEFS method, the search space is limited between 1 to total number of features (NF). Next step is to generate a set of new feature vectors from the original population as mutant population in which each vector is indexed with a number from 0 to $NP-1$.

For each position in the original population matrix, a mutant vector is formed by adding the scaled difference between two randomly selected population members to a third vector, according to Eq. (3). Unlike the original DE [6, 14] that uses a constant scale factor, DEFS method allows the scale factor to change dynamically as follows:

$$F = \frac{C_1 \times rand}{\max\left(x_{j,r1,g}, x_{j,r2,g}\right)} \quad (5)$$

where c$_1$ is a constant smaller than 1. The result of this is to allow the NP members to oscillate within bounds without crossing the optimal solutions and thereby aid them to find enhanced points in the optimal region. Additionally, a system constant with specification is implemented as

$$x_{j,i,g} = \begin{cases} NF & if\ x_{j,i,g} > NF \\ 1 & if\ x_{j,i,g} < 1 \end{cases} \quad (6)$$

In the selection stage, trial vector competes against the population vector of the same index x_0 then the corresponding position in the population matrix will contain either the trial vector u_0 or the original vector x_0 depending on which one of them achieved a better fitness solution. The procedure repeats until each of the NP population vectors have competed against a randomly generated trial vector. Once the last experimental vector has been tested then the survivors of the NP pair wise competition members will become parents for the next production in the evolutionary cycle.

Due to the fact that a real number optimizer is being used, nothing will prevent two dimensions from settling at the same feature indexes. In order to overcome such a problem, a roulette wheel weighting scheme is utilized [10]. In this scheme, a cost weighting is implemented in which the probabilities of each feature are calculated from the distribution factors associated with it. The distribution factor of feature f_j within the current generation g, is referred as $FD_{j,g}$ which is calculated using below equation:

$$FD_{j,g} = a_1 \times \left(\frac{PD_j}{PD_j + ND_j}\right) + \frac{NF - DNF}{NF} \times \left(1 - \frac{\left(PD_j + ND_j\right)}{\max\left(PD_j + ND_j\right)}\right) \quad (7)$$

where PD$_j$ is the number of times that feature f_j has been used in the good subsets. ND$_j$ is the number of times that feature f_j has been used in the less competitive subsets. NF is the total number of features, a_1 is a suitably chosen positive constant that reflects the importance of features in PD and DNF. Divide the estimated distribution factors for the current and next iterations by the maximum value i.e., $FD_g = FD_g / \max\left(FD_g\right)$ and $FD_{g+1} = FD_{g+1} / \max\left(FD_{g+1}\right)$. Compute the relative difference according to the following equation:

$$T = \left(FD_{g+1} - FD_g\right) \times FD_{g+1} + FD_g \quad (8)$$

The above equation provides higher weights to features that make obvious improvement in the current iteration in comparison to the previous one. Next add some sort of randomness in this process to avoid selecting the same features every time, to emphasize the importance of unseen features

$$T = T - 0.5 \times rand\left(1, NF\right) \times \left(1 - T\right) \quad (9)$$

For the remaining iterations, the distribution factors are updated as $FD_g = FD_{g+1}$, within each iteration and FD$_{g+1}$ hold recently computed values in each iteration. These steps are repeated until a stopping criterion is attained or a predetermined generation number is reached.

4. ELM based DEFS (ELM_DEFS) for optimal feature selection

Proposed methodology combines the concept of ELM for optimizing the weights in DEFS feature selection method. In DEFS, the population matrix will contain either the trial vector, u$_0$ or the original vector x$_0$, depending on which one of them achieved a better fitness is nothing but lower error rate is calculated by using the ELM network. The input weights and bias weights are used to increase the generalization performance and the conditioning of the ELM network [14]. ELM_DEFS enables in selecting the best feature set with higher accuracy. Steps involved in the proposed methodology are as follows:

Algorithm:

Step 1: input original feature set.
Step 2: Initialization and parameter setting
Step 3: Initialize positions with a set of input weights and hidden biases:
[W_{11}, W_{12},…, W_{1n} ,…, W_{21}, W_{22}, …, W_{2n}, …, b_1, b_2, … , b_H]. These will be randomly initialized within the range of [-1, 1] on D dimensions in the search space.
*Step 4: **while** termination condition not met **do***
*Step 5: **for all** population member — vector x_i **do***
Step 6: Merge three different randomly selected vectors to create a mutant vector, $v_{i,g}$

$$v_{i,g} = x_{r0,g} + F \times \left(x_{r1,g} - x_{r2,g}\right)$$

Step 7: create mutant vector $v_{i,g}$
Step 8: crossover x_i and $v_{i,g}$ to create trial vector $u_{i,g}$ by the equation given below

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & if\ rand(0,1) \le C_r \quad or \\ x_{j,i,g} & otherwise \end{cases}$$

Step 9: The scale factor to change dynamically is calculate as follows:

$$F = \frac{C_1 \times rand}{\max\left(x_{j,r1,g}\ ,x_{j,r2,g}\right)}$$

Step 10: Additionally, a system constant with specification is implemented as

$$x_{j,i,g} = \begin{cases} NF & if\ x_{j,i,g} > NF \\ 1 & if\ x_{j,i,g} < 1 \end{cases}$$

Step 11: For each member in the group, the respective output final weights in ELM network is computed by using the equation given below

$$\beta' = H^{\dagger}T$$

Step 12: Mean Square Error (MSE) of each member, is evaluated as the fitness function which is given below:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_k^i - d_k^i\right)^2$$

where the term y_k and d_k are the errors of actual output and target output of the k^{th} output neuron of i^{th} sample.

Step 13: Calculate distribution factor of feature f_j within the current generation g, is referred as $FD_{j,g}$ which is given in below equation:

$$FD_{j,g} = a_1 \times \left(\frac{PD_j}{PD_j + ND_j}\right) + \frac{NF - DNF}{NF} \times \left(1 - \frac{\left(PD_j + ND_j\right)}{\max\left(PD_j + ND_j\right)}\right)$$

Step 14: Compute relative difference according to the following equation:

$$T = \left(FD_{g+1} - FD_g\right) \times FD_{g+1} + FD_g$$

*Step 15: **end for***
*Step 16: **for all** population member—vector **x_i** **do***
*Step 17: **if** $f(u_i) \leq f(x_i)$ then*
*Step 18: **$x_i \leftarrow u_i$***
*Step 19: **end if***
*Step 20: **end for***
*Step 21: **end while***

5. Results and Discussions

In this work, initially DEFS is used to select appropriate features for dimensionality reduction and feature subset selection process after the features extracted using GLCM and GLRLM. Once the DEFS algorithm selects optimal features then the selected features are given as input to the classifier to classify them into different river ice class types. In the proposed ELM_DEFS methodology, the weights and bias of ELM network is used to optimize the error rate (i.e. to minimize the classification error rate) of DEFS method for better simplification and classification of river ice types.

The real datasets are collected from Stuttgart, a city in Germany. The real dataset images contain all the River Ice types. River ice types considered are, Open water, Frazil pans, Pancake ice, Freeze over border ice, Juxtaposed ice, Light Consolidated ice and Heavy Consolidated ice are considered for classification purpose. Some sample images are shown in Figure 2. MATLAB is the simulation tool employed for implementing the proposed work.

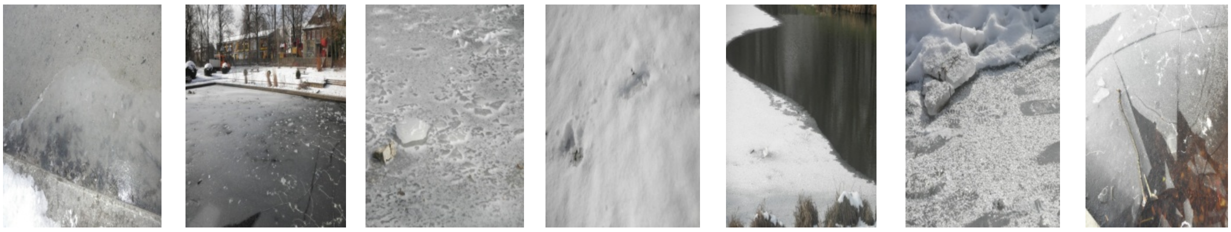

Figure 2: Sample River Ice images

Features selected from DEFS method are given as input to the classifier with the accuracy of 82%, 93% and 60% for 5 features, 10 features and 15 features respectively. DEFS method provides 93% accuracy with 10 features. From Table 2, most of the GLCM features are considered as relevant features, so only GLCM features are considered in further steps of this work.

Table 2: Feature subset selected by DEFS feature selection method

Selected Features	*DEFS*
5 Features	Contrast, Correlation, Homogeneity, Sum Average, Difference Entropy
10 Features	Mean, Contrast, Correlation, Energy, Entropy, Homogeneity, Sum Average, Sum Entropy, Short run emphasis, Run Percentage
15 Features	Mean, Skewness, Kurtosis, Inverse Difference Moment, Contrast, Correlation, Energy, Entropy, Homogeneity, Sum Average, Sum Variance, Sum Entropy, Difference Entropy, Short run emphasis, High Gray Level Run Emphasis

Spatial dependence of gray-level values is captured by GLCM method. The GLCM features are calculated for 0^0, 45^0, 90^0 and 135^0 and a distance scale factor of 1. GLCM features considered are - Autocorrelation, Contrast, Energy, Entropy, Correlation, Homogeneity, Sum of Squares, Sum Average, Sum Variance, Sum Entropy, Difference variance, Difference entropy, Information measures of correlation1, Information measures of correlation2, Inverse difference, Inverse difference normalized, Inverse difference moment normalized. These features are extracted at four angles as shown in Table 3, making a total of 68 features.

Table 3: Texture Feature Directions

Directions of Texture features extracted		
	Degree	*Offset*
1	0^0	0,1
2	45^0	-1,1
3	90^0	-1,0
4	135^0	-1,-1

Among 68 features, the best five features selected by DEFS and the proposed ELM_DEFS techniques and their feature values are shown in Table 4 and Table 5 respectively. The five features selected from DEFS method are Contrast, Correlation, Homogeneity, Sum Average and Difference Entropy. The best five features selected by the proposed ELM_DEFS method are Autocorrelation, Entropy, Difference variance, Information measures of correlation2 and Inverse difference moment normalized.

Table 5: GLCM feature values for best five features of the ELM_DEFS method.

Class	*Auto Correlation*	*Entropy*	*Difference variance*	*Information measures of correlation2*	*Inverse difference moment normalized*
1	1.014313	0.074141	0.007764	0.122644	0.999807
2	1.69156	0.592112	0.008869	0.791331	0.999777
3	6.86882	0.733175	0.266869	0.810456	0.99579
4	1.940826	0.398382	0.025669	0.711789	0.999541
5	14.46128	0.817725	1.121868	0.807214	0.98857

Table 4: Best five features selected by DEFS and the proposed ELM_DEFS methods.

Techniques	*Selected GLCM Features*
DEFS	Contrast, Correlation, Homogeneity, Sum Average, Difference Entropy
ELM_DEFS	Autocorrelation, Entropy, Difference variance, Information measures of correlation2, Inverse difference moment normalized

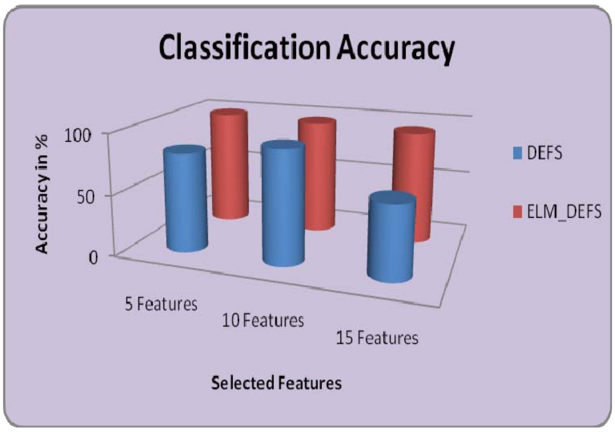

Figure 3: Selected features selected from DEFS and Proposed ELM_DEFS method with classification accuracy

The two methods are classified by using Probabilistic Neural Networks (PNN) classifier. The main advantage in using the PNN classifier is that it classifies the data with minimum number of training vectors and achieves good classification accuracy. Table 6 shows a confusion matrix calculated from the classification results of 45 samples with 5 classes and each class consists of 9 samples. Based on this confusion matrix the sensitivity, specificity and accuracy are shown in Figure 4.

The proposed methodology, ELM_DEFS increases classification accuracy with reduced feature set consists only the features extracted from GLCM method with four angles making a total of 68 features. ELM_DEFS selects five most relevant features with the accuracy of 97.78%, 95.56% and 93.18% for 5 features, 10 features and 15 features respectively when compared with DEFS method. From Figure 3, the result is apparent that with five features 97.78% accuracy is obtained for the proposed method.

The entries have the following meanings, True Negative (TN) is the number of correct negative predictions, False Positive (FP) is the number of incorrect positive predictions, False Negative (FN) is the number of incorrect negative predictions, True Positive (TP) is the number of correct positive predictions [16].

Table 6: Confusion Matrix for five-class classification problem

		Predicted class j by classifier				
	True class i	*1*	*2*	*3*	*4*	*5*
Actual classes	*1*	9	0	0	0	0
	2	0	9	0	0	0
	3	0	0	8	0	1
	4	0	0	0	9	0
	5	0	0	0	0	9

Accuracy, Precision, Recall/Sensitivity, Specificity and F-Measure are the performance metrics used for a classifier to test its performance.

Accuracy is the proportion of true results (both TP and TP) in the population:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Precision is the fraction of retrieved instances that are relevant:

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

Recall or Sensitivity relates to the test's ability to identify positive results:

$$Recall/Sensitivity = \frac{Number\ of\ TP}{Number\ of\ TP+Number\ of\ FN} \quad (12)$$

Specificity relates to the test's ability to identify negative results:

$$Sensitivity = \frac{Number\ of\ TN}{Number\ of\ TN+Num\ ber\ of\ FP} \quad (13)$$

F - Measure is a measure that combines precision and recall is the harmonic mean of precision and recall:

$$F-Measure = 2 \cdot \frac{Recall \cdot Precision}{Recall+Precision} \quad (14)$$

Table 7 shows the classification performance metrics for the features selected from DEFS and ELM_DEFS methods. Figure 4 compares the performance measures of the DEFS and ELM_DEFS methods in terms of classification accuracy, sensitivity and specificity measures.

Table 7: Performance evaluation metrics

Performance Metrics	*DEFS (%)*	*ELM_DEFS (%)*
Accuracy	93	**97.78**
Precision	93	**97.78**
Recall/Sensitivity	82.8	**90**
Specificity	82.4	**1**
F-Measure	95	**97.2**

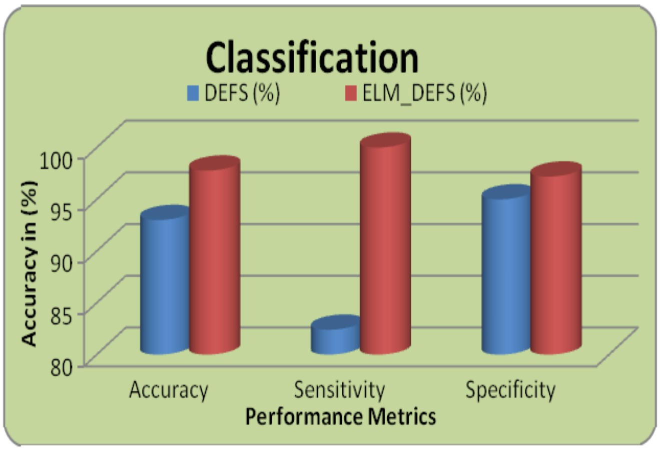

Figure 4: Classification accuracy, sensitivity and specificity measures for DEFS and Proposed ELM_DEFS method

6. Conclusion

The key issue in the development of pattern recognition of river ice type classification is the formation of feature extraction analogy, feature subset selection and the classifier. This paper presents a novel feature subset selection algorithm based on a combination of extreme learning machine and differential evolution feature selection optimization techniques. Simulation result shows that the proposed feature extraction and feature subset selection method works fine and produces optimal number of feature set with higher classification rate when compared with other methods. The five features selected from DEFS method provide a classification accuracy of 82%. In the proposed methodology, the input weights and bias weights of the ELM network is used to optimize the fitness function of the DEFS method. The five optimal features selected from ELM_DEFS method provide 97.78% classification accuracy with reduced feature subset. Thus, the proposed methodology in this research work is developed for river ice feature subset selection to reduce the computational cost and time of the classifier.

References

[1] Isabelle Guyon and Andre Elisseeff, "An Introduction to Variable and Feature Selection", Journal of Machine Learning Research 3, Pg. No.1157-1182, 2003.

[2] L.Ladha and T.Deepa, "Feature Selection Methods and Algorithms", International Journal on Computer Science and Engineering (IJCSE), Pg. No. 1787-1797Vol. 3 No. 5 May 2011.

[3] Yvan Saeys et. al., "A review of feature selection techniques in bioinformatics", BIOINFORMATICS REVIEW, doi:10.1093/bioinformatics/btm344, Vol. 23 no. 19, pages 2507–2517, 2007.

[4] Haralick R.M., Shanmugam K. and Dinstein I. "Textural Features for Image Classification", IEEE Trans. Syst. Man Cybern, Vol.3, pp. 610-621, 1973.

[5] Xiaoou Tang, "Texture Information in Run-Length Matrices", IEEE TRANSACTIONS ON IMAGE PROCESSING: VOL. 7, NO. 11, NOVEMBER 1998.

[6] Goran Martinovic, et. al., "A Differential Evolution Approach to Dimensionality Reduction for Classification needs", Int. J. Appl. Math. Comput. Sci., 2014, Vol. 24, No. 1, 111–122, DOI: 10.2478/amcs-2014-0009.

[7] Rami N. Khushaba and et. al., "Feature subset selection using differential evolution and a statistical repair mechanism", Expert Systems with Applications 38, Pg. No. 11515–11526, 2011.

[8] Ahmed Al-Ani and et. al., "Feature subset selection using differential evolution and a wheel based search strategy", Swarm and Evolutionary Computation 9, Pg. No.15–26, 2013.

[9] Utpal Kumar Sikdar et al.," Differential Evolution based Feature Selection and Classifier Ensemble for Named Entity Recognition", Proceedings of COLING 2012: Technical Papers, pages 2475–2490, COLING 2012, Mumbai, December 2012.

[10] Haupt, R. L and Haupt, S. E, Practical genetic algorithms, 2nd edition, A John Wiley & Sons, Inc., Publication 2004.

[11] R. Rajesh and J. Siva Prakash, "Extreme Learning Machines - A Review and State-of-the-art", International Journal of Wisdom Based Computing, Vol. 1(1), 2011.

[12] Huang G.B., Zhu Q.Y. and Siew C.K. "Extreme learning machine: Theory and applications", Neurocomputing,Vol.70, No.1, pp.489-501, 2006.

[13] Guang-Bin Huang et. al., "Extreme Learning Machine for Regression and Multiclass Classification", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 42, NO. 2, APRIL 2012.

[14] K.Kothavari and S.N.Deepa, “A Hybrid DE-RGSO-ELM for Brain Tumor Tissue Categorization in 3D Magnetic Resonance Images”, Journal of Mathematical Problems in Engineering, (Articles in Press).

[15] Jasmina NOVAKOVIĆ et. al., “Toward Optimal Feature Selection Using Ranking Methods and Classification Algorithms”, DOI: 10.2298/YJOR1101119N, Yugoslav Journal of Operations Research, Pg. No 119-135, Vol. 21, 1st November 2011.

[16] POWERS, David M. W, “Evaluation: from Precision, Recall and F-Measure to Roc, Informedness, Markedness & Correlation”, Journal of Machine Learning Technologies, ISSN: 2229-3981 & ISSN: 2229-399X, Volume 2, Issue 1, pp-37-63, 2011.

Author Profile

Bharathi P. T, Ph.D Research Scholar, Dept. of Computer Science, Avinashilingam Institute of Home Science and Higher Education for Women, Coimbatore, India. Have 6 years of teaching experience. Area of interest: Digital Image Processing, Neural Networks and Pattern Recognition.

Dr. P. Subashini, Associate Professor, Dept. of Computer Science, Avinashilingam Institute of Home Science and Higher Education for Women, Coimbatore, India. Have 21 years of teaching and research experience. Her research has spanned a large number of disciplines like Image analysis, Pattern recognition, neural networks, and applications to Digital Image processing. Under her supervision she has seven research project of worth one crore from various funding agencies like DRDO, DST and UGC.