Engineering Journal issn: 0125-8281

Article

Feature Selection for Text and Image Data Using Differential Evolution with SVM and Naïve Bayes Classifiers

Abhishek Dixit[1,a,*], **Ashish Mani**[2,b], **and Rohit Bansal**[3,c]

1 Department of Computer Science, Amity School of Engineering and Technology, Amity University, Noida, U.P., India
2 Department of EEE, Amity School of Engineering and Technology, Amity University, Noida, U.P., India
3 Department of Management Studies, Rajiv Gandhi Institute of Petroleum Technology, Rae Bareli, U.P., India
E-mail: [a]abhishekdixitg@gmail.com (Corresponding Author), [b]amani@amity.edu, [c]rohitbansaliitr@gmail.com

Abstract. Classification problems are increasing in various important applications such as text categorization, images, medical imaging diagnosis and bimolecular analysis etc. due to large amount of attribute set. Feature extraction methods in case of large dataset play an important role to reduce the irrelevant feature and thereby increases the performance of classifier algorithm. There exist various methods based on machine learning for text and image classification. These approaches are utilized for dimensionality reduction which aims to filter less informative and outlier data. Therefore, these approaches provide compact representation and computationally better tractable accuracy. At the same time, these methods can be challenging if the search space is doubled multiple time. To optimize such challenges, a hybrid approach is suggested in this paper. The proposed approach uses differential evolution (DE) for feature selection with naïve bayes (NB) and support vector machine (SVM) classifiers to enhance the performance of selected classifier. The results are verified using text and image data which reflects improved accuracy compared with other conventional techniques. A 25 benchmark datasets (UCI) from different domains are considered to test the proposed algorithms. A comparative study between proposed hybrid classification algorithms are presented in this work. Finally, the experimental result shows that the differential evolution with NB classifier outperforms and produces better estimation of probability terms. The proposed technique in terms of computational time is also feasible.

Keywords: Feature selection, support vector machine, differential evolution, naïve Bayesian, global optimization.

ENGINEERING JOURNAL Volume 24 Issue 5
Received 13 October 2019
Accepted 3 July 2020
Published 30 September 2020
Online at https://engj.org/
DOI:10.4186/ej.2020.24.5.161

1. Introduction

Since last few years, feature selection process is implemented in different domains such as in application with large size of databases, DNA microarray analysis, image classification, biometric applications and text classification. As the data is increasing day by day in all these applications which raises the challenges related to data analysis in terms of valuable feature selection [1]. Feature selection process makes it easy to extract useful information from large volume of data by underlying tools and methods without compromising the quality of dataset. Feature selection approach is highly recommendable for choosing relevant data among large set of data which contains irrelevant information. Theoretical attempts in this direction can lead to pessimistic conclusion and in this exponentially many data points are needed to produce good features which is troublesome task. Tabli [2] suggested that comprehensive search find all the possibilities of feature sets while in approximate search the focus is on high quality solutions but does not mean to assure an optimal solution therefore, metaheuristics algorithms are used.

As more and more digital applications are increasing and so their usages are also increasing therefore data is also increasing. This large dataset will create serious problems with various machine learning systems in terms of scalability and performance. As an example, data with numerous amounts of features sets may have large amount of superfluous and irrelevant information, and this will degrade the outcomes of learning algorithms. This brings the importance of feature analysis so that problems in high dimensional data can be addressed [3]. Some of the challenges which deals with huge data and number of instances are referred from Liu's [4] work. Two models are mainly used to categorize feature selection algorithm namely filter model and wrapper model [5]. In the filter-based model usual attributes of the training data to select features with no involvement of any learning algorithm. On the other side, wrapper model selects features with one learning algorithm which is already predetermined. Also, based on the result of the learning algorithm it produces best selected features. But this model is computationally of high cost because it requires a classifier to learn a hypothesis [6]. However, in case of high dimensional data set, the filter model is preferred over the other. Variables are selected so that irrelevant information can be filtered out from the data. But if a machine learning system use irrelevant variables then this leads to poor generalization. To improve this problem various classification algorithm such as principal component analysis, SVM etc. can be used. After some successful feature selection criteria, an algorithm must be developed to find useful features by means of any model. Otherwise, the selection of feature subsets becomes a NP hard problem. To overcome such issues hybrid model can be used which uses best features of both the model discussed above. In these models or algorithms, first, goodness feature is obtained from data so that best subset for a provided cardinality can be selected. After this, cross validation is exploited for the selection of best feature subset through dissimilar cardinalities. Such type of model work as merger of filter and wrapper algorithms to produce best performance measures with learning algorithm [7] [8]. Various hybrid approaches are suggested in the literature to estimate the best features with optimal computational cost. Pourhashemi et al. [9] proposed a hybrid approach Chi Squared (Chi2) and Random Tree wrapper as feature selection approach and SVM, NB etc as classifiers to improve the accuracy of these classifier.

As discussed above, in many classifications problems large number of feature sets create problems to achieve best results. Therefore, to overcome issues related to performance, several classifiers are proposed in the literature. E.g., a generic technique for classification of text dataset is to divide the problem into separate binary classification problems. Then apply all binary classifiers and combine their predictions into a single decision. The outcomes using this strategy produces best possible outcomes. On the other side, images contain thousands of pixels as data in multiple colour channels. The correlation and relationship among pixels can be used to categorize a class. From image and signals, features are extracted as representative of each object and its class to produce best feature set. Many feature selection algorithms are tested on these kinds of datasets. Some of them are discussed in this section.

Wilson and Martinez [10] proposed three classifiers for handling nominal and continuous attributes which are named as heterogeneous value difference metric (HVDM), the interpolated value difference metric (IVDM), and the windowed value difference metric (WVDM). In their experiment, they tested their technique on 48 applications and obtained higher classification accuracy compared with other datasets consist of both nominal and continuous attributes. For text classification Ragas and Koster [11] proposed four text classification algorithms. These algorithms are known as Rocchio's algorithm, the simple Bayesian classifier, the sleeping experts and winnow. Their algorithms are tested on Dutch corpus collected from various newspapers. The performance of these algorithms is compared based on learning speed and error rate. Shin et al. [12] showed that the performance of kNN classifier can be improved if we remove the irrelevant features from the training data. Doing this also performs 10% better than Centroid-based classifier. Later, Danesh et al. [13] implemented a supervised classification approach of text data set. In their method, document is characterized as vectors and treats each component as a specific word. They used voting method and OWA operator and decision template for combining various classifiers such as Naive-Bayes, k-NN and Rocchio. From the results we can observe that 15% classification error is reduced as measured on 2000 training data from 20 newsgroups dataset. Buddeewong and Kreesuradej [14] proposed an approach for enhancing the prediction accuracy of association rule-based classifier by categories (ARC-BC). Their algorithm is based on two types of frequent item sets.

The initial recurrent item sets that is L_k comprise all term with no overlapping with other categories. The subsequent recurrent item sets, OL_k have all features which is overlapping to other categories. They also propose an operation to join the second frequent item sets. Their results show good performance for the proposed classifier. Trappey et al. [15] implemented a document (patent) classification and searching algorithm using neural network. The classification algorithm initiates by taking out key phrases as per their frequency in the data and determines significance of such key phrases. Then, to find the similarities, a correlation analysis is applied between key phrases. Based on their higher correlation, classification into smaller set of phrases has been achieved. At last, back propagation network model is used as classifier. Their results show an improvement in terms of document classification compared with other similar approaches implemented in this area. In 2009, Li and Park [16] improved the work done by Trappey et al. [15]. They proposed an improved back propagation neural network and proved that their method is best for reduction in the dimension of the data and therefore produces best results. In this approach to increase the Back-propagation network model the new learning phase evaluation back propagation neural network (LPEBP) is proposed. They used singular value decomposition (SVD) to minimize the dimension and built a latent semantics between terms. Their experimental results show that the LPEBP is faster than the traditional BPNN. Donghui and Zhijing [17] suggested a new hybrid approach to improve the performance of text categorization. They combine hidden markov model (HMM) and support vector machine (SVM) to improve the results. HMMs are used to extract the features and that feature vector is used by SVM to normalize so that SVM can successfully classify the texts. They proved that their results are effective and produces higher accuracy. For multi label learning Reyes et al. [18] presented three variants of ReliefF algorithm. The name of their algorithms is ReliefF-ML, PPT-ReliefF and RReliefF-ML respectively. PPT-ReliefF employed a problem transformation approach where multi-label problem is changed to a single-label problem. The other approaches are ReliefF-ML and RReliefF-ML that transforms the classic ReliefF algorithm in order to handle directly the multi-label data. The outcomes of this algorithm are justified using many nonparametric statistical tests and verifies the effectiveness of the proposed multi-label learning. Recently, Pereira et al. [19] represented a study based on multi-label classification for feature selection. They provided a comprehensive study related to categorization of the feature selection techniques that have been created for the multi-label classification setting.

Recently many researchers have used evolutionary algorithms in the area of classification, feature selection and dimensionality reduction of large set of data. Initially, Yang and Honavar [20] proposed a novel approach using a genetic algorithm for multi criteria optimization problem of feature subset selection. Feasible results were shows for feature subset. Oh et al. [21] proposed a feature selection approach using hybrid variant of genetic algorithm. Local search operations are embedded with hybrid genetic algorithms to fine tune the searching and further, their requirement of efficiency and timing are analysed. Their technique showed better convergence power compared with classical algorithms. Their result demonstrates that the proposed technique is superior to simple genetic algorithm and sequential search algorithms. Wang et al. [22] proposed a new feature selection approach using rough set and particle swarm optimization (PSO). Rough sets are used for feature selection and like genetic algorithms PSO is a new evolutionary approach to provide solution to problem space. They find optimal regions and features using PSO. This is an attractive technique for feature selection because it discovers best features within subset space. Compared with GAs, PSO does not require complex operators such as crossover and mutation, but only primitive and simple mathematical operators are needed. Their results show that PSO is efficient for rough set-based feature selection. Derrac et al. [23] proposed an evolutionary algorithm for data dimensionality reduction. They used instance and feature selection. The instance selection is carried out using a steady state genetic algorithm in combination with fuzzy rough set. Then, interesting features are selected to increase the evolutionary search process. These proposed algorithms show improvement in classification performance however using evolutionary approaches still faces the problems of premature convergence and high complexity.

To overcome these problems, many researchers applied Differential evolution algorithm for feature optimization. Differential evolution algorithm is a stochastic, population-based algorithm proposed by Storn and Price [24]. The DE has been used in Recently in 2016, Onan et al. [25] proposed a static classifier selection-based approach using majority forward search, voting error and multi objective differential evolution algorithm. Their algorithm integrates following as the base learners: Bayesian logistic regression, naïve Bayes, linear discriminant analysis, logistic regression, and support vector machines. Their experimental results used for investigation of various classification tasks such as sentiment analysis, software defect prediction, credit risk modelling, spam filtering, and semantic mapping, demonstrate that the presented approach in the paper can forecast better in comparison to conventional ensemble learning approaches for example AdaBoost, bagging, random subspace, and majority voting. More basic concepts on evolutionary computation can be studied from [26]. Another approach proposed by Hancer et al. [27] in which is based on multi objective DE for feature selection. All these algorithms use classical DE approach where a fixed mutation strategy is used with tuning of control parameters. However, tuning of control parameters is a time consuming and fixed parameter setting approach is limited to specific problems.

In order to solve above problem with DE, different variants are proposed by choosing adaptive mutation and tuning of parameter approach such as. Zhang et al. [28] proposed self-learning approach of DE for feature optimization. Hancer [29] proposed a new multi-objective approach of DE to find the homogeneous clusters by optimizing the feature set. Alswaitti et al. [30] proposed a variance-based DE with new crossover strategy to increase the convergence rate. Author links open overlay panel. Tarkhaneh et al. [31] proposed an improved mutation strategy for feature selection. However, all these algorithms depend on history for finding the randomness and thus lack specific guidance.

To overcome the above issues and to maintain the exploration and exploitation capability in DE, A novel self-adaptive mutation strategy is proposed for feature selection and further hybridized with NB and SVM classifiers to improve the accuracy in data classification. The overall goal of this paper is first to propose a self-adaptive DE for feature selection. Secondly combine this algorithm with NB and SVM classifier to improve the accuracy of these classifier. Third to compare both the hybrid variants of classifiers and investigate the performance of both these classifiers.

Our approach is different from the previous approaches as in our approaches $DE/rand/1$ mutation strategy is used whereas in our approach $DE/rand/2$ mutation strategy is used with self-archival approach for control parameters calculation. This is done to improve the convergence speed and calculation precision. Secondly, in previous approaches feature selection and clustering are performed independently and therefore it is difficult to find out if the selected set of features are applicable for the obtained clustering partitions. Whereas in our hybrid strategy both the feature selection and clustering are performed simultaneously, and this ensure that the n optimal number of clusters are preserved and also reducing the irrelevant features.

The proposed algorithms DENB and DESVM are applied on 25 UCI text datasets and compared with other classical SVM and NB classifiers as well as 2 new variants of classifiers proposed by Diab et al. [32] for investigating the performance. The results show that our proposed hybrid classifier is giving better results as compared to these existing approaches. In addition to this we have also evaluated our approaches on blood cancer image datasets. The results show that our approaches are giving better results as compared to conventional approaches and when compared with each other DENB is giving better performance.

This paper is mainly categorized into 4 sections. Section 2 discuss Section 2 presents preliminaries and proposed algorithm. Section 3 demonstrate results and comparative analysis. The last section 4 discusses the conclusion and future aspects.

2. Related work

In this section, a discussion about the relevant techniques for the proposed algorithm are described. After that, proposed hybrid algorithm is summarized using these techniques. Two algorithms are proposed to increase the performance of feature selection using the consistency evaluation criterion.

2.1. Naïve Bayesian Learning (NBL)

The naïve Bayesian classifier is simple, efficient but at the same time it is extremely sensitive to feature selection. Use of this will improve the features from a set of attributes. The explanation of this algorithm is as follow.

In a data training set, each instance is represented as an attribute values of vector element$\langle v_1, v_2, v_j \ldots\ldots v_k \rangle$, where $j = 1, \ldots\ldots, k$ and v_j represent the j^{th} attribute value. For a novel overlooked example of the form $\langle v_1, v_2, v_j \ldots\ldots v_k \rangle$, naïve Bayesian allocates a new class say d_{pred} that has a maximum conditional probability value which is obtained by using Eq. (1).

$$d_{pred} = argmac_{d \in D} \frac{P(v_1, v_2 \ldots\ldots v_k | d).p(d)}{p(v_1, v_2 \ldots\ldots v_k)} \tag{1}$$

In Eq. (1), D is all classes set, For class d, $p(d)$ is probability value, probability $p(v_1, v_2 \ldots\ldots v_k)$ is attributes to $1,2, \ldots\ldots\ldots, k$ on the $v_1, v_2, v_i \ldots\ldots v_k$ and $P(v_1, v_2 \ldots\ldots v_k | d)$ is the probability for all the attributes on values for instance of d .

Naïve Bayesian algorithm assume that the occurrences of each word value is autonomous among the class values. Therefore, the probability of aggregation satisfies the class *d*. The probability of various features is represented in terms of Eq. (2). Also using Eq. (2), new class d_{pred} can be modelled as Eq. (3).

$$p(v_1, v_2 \ldots\ldots v_k | d) = \prod_{i=1}^{n} p(a_i/d) \tag{2}$$

$$d_{pred} = argmac_{d \in D} p(d). \prod_{i=1}^{n} p(a_i/d) \tag{3}$$

where training data set might evaluate $p(d)$ and $p(a_i/d)$. The conditional probability will be 0 If any value of the probability term is 0. In order to evade this situation Equation (4) can be implemented with Laplace smoothing.

$$p(a_i/d) = \frac{count(a_i, d) + 1}{count(d) + |v|} \tag{4}$$

Here, $count(a_i, d)$ represents the is attribute i count containing a_i in class *d* of training data set, $|v|$ is attribute i values and $count(d)$ represents count of class d instances. This is used to create a novel noise free training set using these equations. In the proposed work, we implemented multinomial naïve Bayes. In Multinomial Naïve Bayesian [33] document is represented by utilizing the vector of words. In the presented work Eq. (3) is modelled and implemented as shown in Eq.(5).

$$d_{pred} = argmac_{d \in D} [log\ p(d) + \sum_{i=0}^{n} f_i\ log\ p(v_i/d)\ ] \tag{5}$$

where f_i is the count of frequency word in the dataset prepared for training the model.

2.2. Support Vector Machine (SVM)

For the classification, pattern recognition and regression analysis, Vapnik [34] proposed the SVM classification approach. This support vector machine is a supervised learning algorithm which carries data model from nonlinear to a high dimensional space mapping for finding optimal hyperplane defined by various support vectors. In order to evaluate the decision function, this classifier is based on support vectors and work with linear models by utilizing the non-linear class boundaries.

Let's say a nonlinear function $\varphi(x)$. This function is used for mapping the X as a training set to a linear feature space which is also high dimensional. and$w * \varphi(x) + b = 0$ is the classifier hyperplane. Using $f(x) = sign(w * \varphi(x) + b)$ as a decision function. In a similar manner nonlinear problem can also be solved into linear. Equation (6) denotes the optimization n numbers of problems in nonlinear classification hyperplane.

$$\begin{matrix} min \\ w, b, \varepsilon \end{matrix} \ \frac{1}{2} w^T w + C \sum_{i=1}^{n} \varepsilon_i$$

$$s.t.\ y_i(w^T * \varphi(x_i) + b) \geq 1 - \varepsilon_i \quad \varepsilon_i \geq 0,\ i = 1,2, \dots n \tag{6}$$

here, C is penalty parameter showing the more acceptance of misclassification. ε_i is non-negative slack variable. By introducing Lagrange formulation, the non-liner classifier can be converted into liner classifier and is represented as Eq. (7).

$$\begin{matrix} min \\ \alpha \end{matrix} \ \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{n} \alpha_i,$$

$$s.t. \textstyle\sum_{j=1}^{n} \alpha_i y_i = 0,\ \ 0 \leq \alpha_i \leq C,\ \ i = 1,2, \dots n \tag{7}$$

In this equation, α_i is the support value or weight if $0 \leq \alpha_i \leq c$, x_i is the support vector while the kernel function is defined as $K(x_i, x_j) = \varphi(x_i) * \varphi(x_j)$. The support machine will find both the centers x_i and weight α_i and obtain unique solution when choosing parameter C.

2.3. Differential Evolution Algorithm (DE)

Differential evolution algorithm is an exploratory, population based, stochastic algorithm utilized to solve continuous and discrete optimization problems. Differential evolution algorithm was proposed by Storn and Price [24] in 1997 as exploratory algorithm for finding an optimal solution in a large search space. In the evolution algorithm, firstly, solution vectors are randomly initialized and further improved by using differential operators. These operators are described in terms of various steps such as fitness evaluation, mutation, crossover and selection. The implementation is done as per following steps:

2.3.1. Initialization

In the first step of this algorithm, initialization of candidate solution say X is done in search space and solution vector in differential evolution can be represented as$X_i = \{x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{iD}\}$. These solution vectors can be represented as Eq. (8).

$$x_{ij} = x_j^{min} + R(0,1)(x_j^{max} - x_j^{min}) \tag{8}$$

where $i = \{1,2, \dots, NP\}$ and NP is the populations size, $j = \{1,2, \dots D\}$ and D is the search space dimensionality. (0, 1) is the uniformly distributed random. Predefined maximum and minimum values of parameter j are defined asx_j^{min} and x_j^{max}.

2.3.2. 2.3.2. Mutation

Mutation is the foundation for differential evolution algorithm and is a major benchmark for assessing the efficiency of various DE algorithms. To develop the search space, each solution space has to go through mutation strategy. During this stage, new vectors are generated as weighted difference and are added with existing vector. Let us support that $X_{i,G}$ is a target vector and $V_{i,G+1}$ is a mutant vector. This mutant vector is generated using Eq. (9).

$$V_{i,G+1} = X_{r1,G} + F.(X_{r2,G} - X_{r3,G}) \tag{9}$$

where r_1, r_2, r_3 are the arbitrary keys belongs to [1, NP]; $r_1 \neq r_2 \neq r_3$ and scaling factor is represented as F . The value is in the range $[0, 2]$. This factor expends $X_{r_{2,G}}$ and $X_{r_{3,G}}$ difference.

2.3.3. 2.3.4. 2.3.3. Crossover

Let's say $U_{i,g+1}$ is a trail vector which is generated by implemented crossover in uniform manner. In this step, the target vector and mutant vector are mixed on the basis of crossover rate. This is done to increase the diversity and written as Eq. (10).

$$U_{i,j,g+1} = \begin{Bmatrix} V_{i,j,g+1} \ if\ (rand_{j,i} \leq C_r\ or\ j = j_{rand}) \\ X_{i,j,g} \quad otherwise \end{Bmatrix} \tag{10}$$

With $j = 1, 2 \dots .d$, and $rand_{j,i}$ is an equally distributed random number from 0 to 1, $C_r \in [0,1]$ is crossover probability. This parameter controls fractional parameters values which are derived from the mutant vector.

2.3.4. Selection

Selection process starts with comparing the fitness values of each individual vectors and chooses the best optimized value for the next generation. If trial vector at generation $g + 1\left(U_{j,g+1}\right)$ produces more cost value compare to target vector, then trial vector replaces target vector in next generation. Therefore, the target vector (X is obtained using Eq. (11).

$$X_{j,g+1} = \begin{cases} U_{j,g+1} & if\ f\left(U_{j,g+1}\right) \le f\left(X_{j,g+1}\right) \\ X_{j,g+1} & otherwise \end{cases} \tag{11}$$

where $f(X)$ defines the objective function with decision variable X and $j = 1, 2, \ldots\ldots NP$. The procedure of mutation, recombination and selection done to achieve a stopping condition.

In the recent years it is observed that the researchers implemented differential evolution in machine learning techniques particularly in the field of text classification. The techniques are effective to produce good results and that is the reason this work is inspired by DE

2.4 The Proposed Algorithms

This section presents proposed algorithms for text and image feature classification. As like supervised feature selection method, our proposed algorithms generate candidate subset followed by evaluation process based on specified criteria. Finally, best features are computed.

2.4.1 DE to Fine Tune Naïve Bayes and SVM Classifier

DE is proposed to fine-tune the Naïve Bayesian and SVM classifier by optimizing large set of features. The main objective to implement the algorithms is to obtain better classification accuracy. In this paper, our algorithms are labelled as DENB and DESVM. Both the algorithms identifying the subgroup of most valuable features and evaluates classification accuracy. The implementation of DE start with the selection of n arbitrary elements for finding best using iterative approach. Then, NB and SVM are used for every solution to check the performance. Selected fit feature elements considered by DE are further used to produce next generation solution. Likewise, the accuracy for obtaining best features are achieved. This analysis or process stays till the convergence of the performance of NB and SVM meet. The DESVM and DENB algorithm steps are as shown below.

Proposed algorithm DENB and DESVM

```
Input: training samples, classes
Output: final solution as set of better NB or SVM

Generate population by using lower bound,upper bound
and random number
Set Generation =25, Cr=.9, Fmin, Fmax, Lb and Ub, th=0.6

Initialize sample text dataset and classes;

% Efficiency on full samples
  temp=true (1, size(samples,2));
  objective original = apply classical SVM or NB classifiers

% Dimension or Number of variables

D=size(samples,2)

% generate population

 for i=1: Np
     Calculate the performance of SVM or NB on full set

    % calculate objective function

    $f_{obj} = \sum_{i=1}^{NP} alpha * \left(\frac{\#trial\ vector}{Total\ Feature}\right) + (1 - alpha) * (1 - perf_{classifier})$

end

$for\ gen = 1: generation$
    $for\ k = 1: NP$
    Generate three random numbers r1, r2and r3 where   r1≠r2≠r3

% Calculate mutation:
        $F = F_{min} + (F_{max} - F_{min}) * rand(1, D)$

        $Y_{i,G+1} = X_{r1,G} + +F.\left(X_{r2,G} - X_{r3,G}\right) + F.\left(X_{r4,G} - X_{r5,G}\right)$

% (Apply trial vector/crossover
        $U_{i,j,\,g+1} = \left\{\begin{matrix} V_{i,j,g+1}\ if\ (rand_{j,i} \le C_r\ or\ j = j_{rand} \\ X_{i,j,g}\ \ otherwise \end{matrix}\right\}$

Calculate the performance of SVM or NB on selected set

    $f_{obj} = \sum_{i=1}^{NP} alpha * \left(\frac{\#trial\ vector}{Total\ Feature}\right) + (1 - alpha) * (1 - perf_{classifier})$

newVal<=prevVal
    performance = newPerfVal;
 end
besteff= performance;
features = nnz(pop(idx,:)>th)
end
end
```

In the proposed scheme a new self-adaptive mutation approach is proposed which can improve the convergence speed. We have used $DE/rand/2$ mutation strategy and the mutation operator choose the best operator from the 3 random vectors. The value of scaling factor F is selected as per below equation

$$F = F_{min} + (F_{max} - F_{min}) * rand(1, D) \tag{12}$$

$$Y_{i,G+1} = X_{r1,G} + +F.\left(X_{r2,G} - X_{r3,G}\right) + F.\left(X_{r4,G} - X_{r5,G}\right) \tag{13}$$

In each iteration the value of mutation vector is changed as per the lower bound Lb and upper bound Ub values

$$Y_{i,G+1} = \begin{cases} Y_{i,G+1} & if\ Y_{i,G+1} < Lb \\ Y_{i,G} & if\ Y_{i,G+1} > Ub \end{cases} \tag{14}$$

where $Lb\ and\ Ub$ are calculated as

$$\begin{aligned} Lb &= zeros(1, D) \\ Ub &= ones\ (1, D) \end{aligned} \tag{15}$$

The trail and crossover vectors are chosen as per below.3.3.1

$$U_{i,j,\,g+1} = \begin{Bmatrix} V_{i,j,g+1}\ if\ (rand_{j,i}\ \leq C_r\ or\ j = j_{rand} \\ X_{i,j,g}\quad otherwise \end{Bmatrix} \tag{16}$$

where, $U_{i,g\,+1}$ is a trail vector which is generated by implemented crossover in uniform manner.

This approach can generate the diverse mutant vector. when $Y_{i,G+1} < Lb$ mutation vector is chosen from next generation and when $Y_{i,G+1}\ .Ub$ mutant vector is chosen from the current population. Convergence sped is also improved by this solution. When $Y_{i,G+1} < Lb$ then directly $Y_{i,G+1}$ is set as the mutant vector. The added parameter $rand$ in calculating the scaling factor will ensure that the value should always be greater than 0 and this helps the generation to fall into local optima.

Although our new proposed DE variant can optimise the feature subset, but this may still have the redundant feature subset due the fact that the basic fitness function will not try to minimise the feature subset. This require a new formula for fitness function in order to achieve the maximum classification accuracy.

$$f_{obj} = \sum_{i=1}^{NP} alpha * \left(\frac{\#trial\ vector}{Total\ Feature}\right) + (1 - alpha) * (1 - perf_{classifier})$$

where

$$perf_{classifier} = SVM/NB \tag{17}$$

In the above equation $alpha$ is taken as constant value 0.15. $trial\ vector$ is the feature selected from feature selection approach for the new generation. $Total\ feature$ is the length of the selected the features.

From the above equation we can see the relative importance of $alpha\ and\ (1 - alpha)$ in setting up the feature set and classifier's output. As classifier's performance is always important as compared to the feature set the value of $(1 - alpha) > alpha$. As it is evident that that redundancy (number of features) are always greater than the relevancy (Classifier's output). Therefore, to balance this trial vector is divided by length of vectors. By doing this the selection of redundant feature subset is solved. But this may have a problem of choosing lower feature subset having lower classifier's performance. To solve this, we propose another approach where in the initial process, we first optimize the classifier's performance and later full feature set is added to the fitness function. In the final stage the archived outcome of initial stage which make sure that the feature set is minimized during evolutionary process and based on the higher performance of classifier. This help the evolutionary process to look for the optimization of larger feature subset with higher classifier's performance.

3. Experimental Results

3.1. Experimental Settings

The performance and effectiveness of the proposed algorithms are evaluated using 25 datasets obtained from UCI repository [35]. Missing values were simply ignored. The classification accuracy of both the algorithms for every dataset is obtained by 100 iteration run over 30 time. The experiment is performed using MATLAB with system configuration as 2.11 GHz, Intel ® Core™ i7-8650U CPU@1.90GHz and 16GB RAM. Different parameters as shown in Table 1 are used during the implementation phase of this work.

Table 1. Different experimental parameters with their setting values.

Defined Parameters	Setting value
Population (NP)	50
No of Iteration	100
Crossover rate	0.9
Threshold	0.6
Min Scaling Factor	0.5
Max scaling Factor	1.0
alpha	0.15

3.2. Result Analysis

For analysing the results for DESVM and DENB, 25 UCI text datasets are chosen. Also, the proposed algorithms are evaluated on image dataset. For image analysis, blood cancer image dataset consist of 231 images is used. We applied Friedman's test in proposed experimental study. The values are estimated using Microsoft excel for calculating the p values. This is shown in Table 5 and Table 6. The information related to chosen 25 datasets are shown in Table 2.

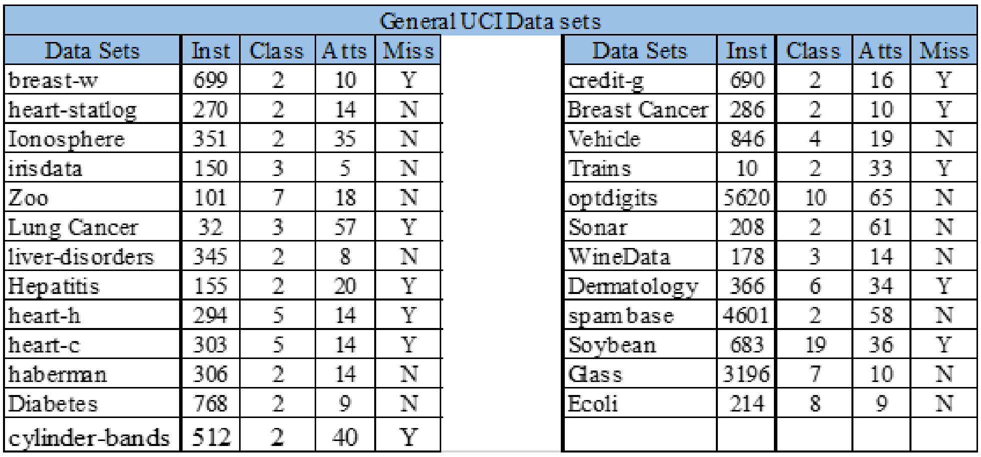
For text-based classification, the mutation factor will increase the accuracy and improve the issues related to local optima. This make DE closer to random search to improve the search accuracy. In our experiment, the mutation factor is calculated by taking minimum and maximum mutation factor as .5 and 1 and further estimated random values of mutation for each iteration. The cross over rate is taken as 0.9 because using this algorithm produces best result. To increase likelihood of finding global optimum and solution space the number of generations are limited to 100 and population size to 50. Classification accuracy is recorded after running the algorithm 30 times for all values.

Table 3 shows the percentage accuracy of the proposed DESVM and DENB algorithm compared with

DOI:10.4186/ej.2020.24.5.161

work done by Diab and Hindi [32]. From the demonstrated results, proposed algorithm is showing better results for feature-based classification. From the results it can be deduced that our proposed algorithm is giving better classification accuracy when the data sets are high in comparison to less data sets. This is because high dimensional data sets have ability for good feature selection and better capability to train the data model. This also shows that use of differential evolution algorithm DE for feature selection is showing better results and giving optimized feature subset to classifiers to improve the classification accuracy. Over 20 datasets, the proposed algorithm is producing better than NB-MPDE while produced best accuracy compared with classical SVM and NB. DESVM is giving best result of 99.1 on Dermatology data set whereas DENB shows better accuracy of 99.05 on Ionosphere data. Figure 1 is a box plot representation for all five algorithms tested on UCI text datasets. From the experimental results we can see that DESVM and DENB methods have higher efficiency for larger feature set than SVM, NB, NBMPDE. For this reason, we further applied both DESVM and DENB on image data.

Table 2. Chosen datasets to evaluate results.



General UCI Data sets										
Data Sets	Inst	Class	Atts	Miss		Data Sets	Inst	Class	Atts	Miss
breast-w	699	2	10	Y		credit-g	690	2	16	Y
heart-statlog	270	2	14	N		Breast Cancer	286	2	10	Y
Ionosphere	351	2	35	N		Vehicle	846	4	19	N
irisdata	150	3	5	N		Trains	10	2	33	Y
Zoo	101	7	18	N		optdigits	5620	10	65	N
Lung Cancer	32	3	57	Y		Sonar	208	2	61	N
liver-disorders	345	2	8	N		WineData	178	3	14	N
Hepatitis	155	2	20	Y		Dermatology	366	6	34	Y
heart-h	294	5	14	Y		spambase	4601	2	58	N
heart-c	303	5	14	Y		Soybean	683	19	36	Y
haberman	306	2	14	N		Glass	3196	7	10	N
Diabetes	768	2	9	N		Ecoli	214	8	9	N
cylinder-bands	512	2	40	Y						

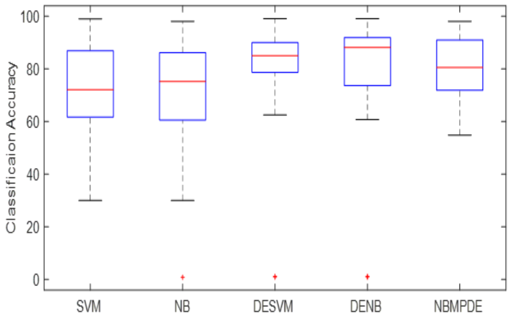
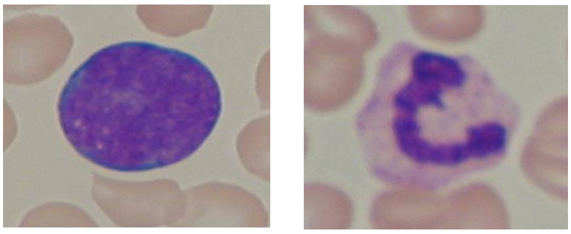


Fig. 1. A Box plot representation for best classification accuracy of SVM, NB, DESVM, DENB, NBMPDE using chosen datasets.



Class 1 Class 2

Fig. 2. Image clusters obtained from blood cancer dataset.

The proposed algorithms are further evaluated for image data. Blood cancer dataset is taken for testing the accuracy of classical SVM, NB and proposed DESVM and DENB. A sample images of two cells clusters is shown in Fig. 2. One cluster is having cancerous images and another with non-cancerous images. The clustering of image sets is done using k means algorithm which is computationally simple and faster. Then the segmented datasets are used as input for various algorithms. The comparative performances of various algorithms are shown in Table 4. For the considered image dataset, DESVM classification accuracy is 98.55% compared with classical SVM accuracy of 86.96%. On the other hand, DENB produces 100% accuracy as compare to simple NB accuracy of 95.65%. This establishes that the proposed methods are effective for DE based hybrid classification. Also, for better understanding a box plot representation w.r.t classification accuracy is shown in Fig. 3. It shows that DENB performs significantly better in comparison to other models.

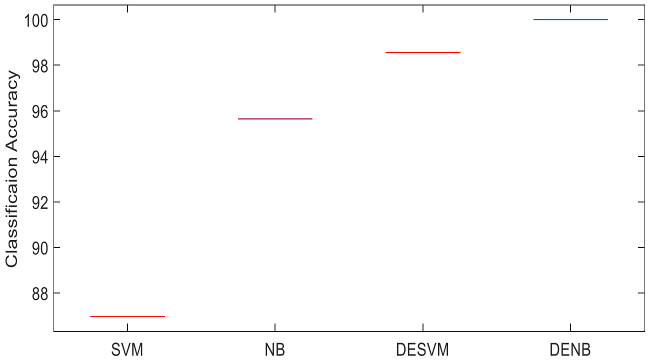


Fig. 3. Box Plot for classification Accuracy of SVM, NB, DESVM, DENB using image dataset.

In this work, a comparison based on execution time for all the presented algorithms using all datasets is also presented. This is shown in Table 5. The timings are shown in minutes for each method to terminate. The worst time for DESVM is 285.3658 min and DENB is 270.3515 min. The results also indicate that the meta-heuristics algorithms perform slowly as compared to classical methods like NB and SVM. Therefore, it can be concluded that NBMPDE has the worst execution time (average) of 67.15 followed by DESVM of 24.42 min and finally DENB of 19.15 min.

Table 3. Comparing best efficiency results with SVM, NB, NBDE, SVMDE and NB-MPDE.

Data Sets	**SVM%**	**Proposed DESVM %**	**NB%**	**Proposed DENB %**	**NB-MPDE%**
breast-w	99.04	99.04	98.09	99.04	96.43
heart-statlog	88.89	88.89	81.48	88.89	84.82
Ionosphere	95.238	96.2	90.48	99.05	92.09
irisdata	84.44	84.44	1	1	91.01
Zoo	89.66	1	86.21	93.1	91
Lung Cancer	30	90	30	90	66.67
liver-disorders	60.58	62.5	60.58	61.538	54.84
Hepatitis	61.7	78.72	59.57	76.6	86.96
heart-h	70.46	86.36	75	87.5	81.52
heart-c	60	63.33	62.22	63.33	84.5
haberman	72.53	73.63	73.63	73.63	77.57
Diabetes	1	1	0.8	1	0.7
cylinder-bands	37.65	91.36	85.89	91.98	71.85
credit-g	66.25	83.09	67.15	85.51	74.1
Breast Cancer	83.59	88.51	81.48	90.12	71.69
Vehicle	71.43	85.71	50	60.71	69.14
Trains	50	1	1	1	70
optdigits	75.67	82.65	75.881	83.49	89.73
Sonar	62.903	91.94	50	91.94	74.52
WineData	1	1	98.11	1	1
Dermatology	97.29	99.1	88.29	97.3	96.09
spam base	71.59	83.77	84.42	87.23	94.7
Soybean	86.96	88.04	86.96	91.3	87.46
Glass	78.46	83.07	75.39	89.23	76.28
Ecoli	76.24	80.21	71.29	77.23	79.6

Table 4. Comparing efficiency results with SVM, DESVM, NB and DENB.

Image Data Set	SVM%	DESVM%	NB %	DENB %
Blood Cancer	86.96	98.551	95.65	100

Table 5. Average execution time comparison.

Data Sets	SVM	DESVM	NB	DENB	NBMPDE
breast-w	0.016275945	3.09383	0.0053795	1.824026667	1.9844
heart-statlog	0.005542645	3.25744667	0.0056584	2.277996667	1.0509
Ionosphere	0.025063483	6.15933167	0.0143597	4.921103333	6.3357
irisdata	0.0014053	1.216139	0.0024203	0.800299167	0.2796
Zoo	0.00184987	2.25108833	0.0150537	6.41616	1.8919
Lung Cancer	0.000756421	1.196296	0.0210496	6.708303333	1.284

liver-disorders	0.007282228	5.39469833	0.0051627	2.509405	0.5725
Hepatitis	0.012535853	1.73717333	0.0065701	2.146113333	1.1567
heart-h	0.014955667	2.43758833	0.0055638	1.298925	2.7913
heart-c	0.031909117	4.88777667	0.0102377	3.641766667	2.8929
haberman	0.004981557	1.4005495	0.0015534	0.512814667	0.3027
Diabetes	0.000529656	0.49058317	0.002674	0.8975955	1.7585
cylinder-bands	0.033358917	8.11419167	0.0111716	3.079285	11.982
credit-g	0.008001742	4.65826	0.0130308	3.451463333	7.8464
Breast Cancer	0.005513282	3.09383	0.0071114	1.563721333	0.748
Vehicle	0.011408195	4.43814167	0.0103748	4.076883333	11.74
Trains	0.00053008	0.32924867	0.0091031	1.820838333	0.1618
optdigits	0.114917917	245.698756	0.244369	145.6987667	1372.3
Sonar	0.01924225	2.22871667	0.0205217	10.02882167	8.3695
WineData	0.002822402	2.51297667	0.0071925	2.41239	1.0916
Dermatology	0.007536052	6.25724667	0.0300603	10.62367167	17.716
spam base	0.051831817	285.3658	0.360805	270.3515	172.23
Soybean	0.002835475	3.01467667	0.0054439	1.498255833	108.94
Glass	0.005343313	4.582695	0.0089511	3.836873333	1.9171
Ecoli	0.00601983	5.567305	0.0088914	2.481975	2.5761
cars	0.030784333	25.4946333	0.0120861	2.904455	5.9351

Table 6. Friedman test statistics.

N	20
Chi sq	271.26
Df	4
Asymptotic significance	1.7E-57

Table 7. Rank of different algorithms.

Strategies	Mean rank on best value
SVM	3.36
DESVM	1.92
PSOSVM	3.96
DEPSOSVM	1.76
NB-MPDE	2.84

Table 8. Feature subset reduction on all the benchmark data.

Data Sets	Method	Avg feature subset
breast-w	DENB, DESVM	5.4,5.6
heart-statlog	DENB, DESVM	6.1,6.7
Ionosphere	DENB, DESVM	6.2,6.8
irisdata	DENB, DESVM	4,4
Zoo	DENB, DESVM	6.5,7.8
Lung Cancer	DENB, DESVM	20.2,21,1
liver-disorders	DENB, DESVM	4,4
Hepatitis	DENB, DESVM	8.2,8.2
heart-h	DENB, DESVM	6.8,7
heart-c	DENB, DESVM	6.8,7
haberman	DENB, DESVM	6.8,7
Diabetes	DENB, DESVM	8,8
cylinder-bands	DENB, DESVM	17.9,18.5
credit-g	DENB, DESVM	5.8,6.1
Breast Cancer	DENB, DESVM	6.8,7
Vehicle	DENB, DESVM	5.1,5.3
Trains	DENB, DESVM	5.1,5.2
optdigits	DENB, DESVM	5.1,5.1
Sonar	DENB, DESVM	5.1,5.0
WineData	DENB, DESVM	5.1,5.1
Dermatology	DENB, DESVM	5.1,5.2
spam base	DENB, DESVM	5.1,5.3
Soybean	DENB, DESVM	5.1,5.4
Glass	DENB, DESVM	5.1,5.5
Ecoli	DENB, DESVM	5.1,5.6

Table 8 shows the details of total and average feature subset reduced on benchmark functions by our proposed algorithm. From the results we can observe that the total feature subset is reduced to less than the 1/3rd of total feature set. Wang et al. [36] suggested the measurement of feature reduction. As suggested by this approach feature should be reduced to fixed 50% of total feature set which is not the case with the proposed approach and our algorithm is showing far more better results in comparison to other algorithms. This is due to the fact that in wrapper approaches classifier; s is used for feature selection whereas in our approach feature selection is separate process and, in our approach classier is not used for feature selection. Also, as we can see from the results, our approach is showing better results in terms of classifier's accuracy. Therefore, from the results we can say that this proposed approach should be accepted as another filter selection approach for fine Tuning of Text and Image Data using Differential Evolution with SVM and Naïve Bayes

From all the experimental results the overall performance of our proposed algorithm is showing better results in comparison of other algorithms. We can conclude from the experiments that performance of our proposed algorithms DESVM and DENB are giving optimized feature subset for both text and image data sets and in turn showing better classification accuracy for text and image datasets.

4. Conclusion

This paper presents hybrid algorithm using differential evolution (DE) with naïve bayes and SVM classifier to improve accuracy of classification. The proposed methods are tested on 25 text and one image dataset. All the features are optimized using DE and feed forward to SVM and NB classifier for better results. The empirical result shows that our feature selection approach integrated with SVM and NB classifiers is able to improve the overall efficiency of the classifiers. Among the two classifiers NB and SVM, NB integrated with DE as feature selection approach is giving better estimate of probability terms. The computation time for analyzing huge data is also better for proposed algorithms compared with other conventional classifiers. Overall, we can conclude that our proposed hybrid algorithms fulfil the objectives of the presented work. In future, we want to explore our proposed algorithm to fine tune the video data. We also want to explore the other evolutionary algorithms to fine tune the classifiers. Hybridization of DE with other evolutionary algorithm to fine tune the data set will also be interesting area.

References

[1] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Recent advances and emerging challenges of feature selection in the context of big data," *Knowledge-Based Systems*, vol. 86, no. 2015, pp. 33–45, 2015.

[2] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009.

[3] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *20th International Conference on Machine Learning (ICML-03)*, 2003.

[4] H. Liu, H. Motoda, and L. Yu, "Feature selection with selective sampling," in *19th International Conference on Machine Learning,* 2002.

[5] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *18th Conference on Machine Learning*, 2001.

[6] P. Langley, "Selection of relevant features in machine learning," in *AAAI Fall Symposium on Relevance*, 1994.

[7] A. Y. Ng, "n feature selection: Learning with exponentially many irrelevant features as training examples," in *Fifteenth International Conference on Machine Learning*, 1998.

[8] E. Xing, M. Jordan, and R. Karp, "Feature selection for high-dimensional genomic microarray data," in *Eighteenth International Conference on Machine Learning,* 2001.

[9] S. M. Pourhashemi, "E-mail spam filtering by a new hybrid feature selection method using chi2 as filter and random tree as wrapper," *Eng. J*, vol. 18, no. 3, pp. 123–134, 2014.

[10] D. R. Wilson and T. R. Martinez, "Improved Heterogeneous Distance Functions," *Journal of Artificial Intelligence Research*, vol. 6, no. 1, pp. 1–34, 1997.

[11] H. Ragas and C. H. A. Koster, "Four text classification algorithms compared on a Dutch corpus," in *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Melbourne, Australia, 1998.

[12] K. Shin, A. Abraham, and S. Y. Han, "Improving kNN text categorization by removing outliers from training set," in *International Conference on Intelligent Text Processing and Computational Linguistics, Springer,* Berlin, Heidelberg, 2006.

[13] A. Danesh, B. Moshiri, and O. Fatemi, "Improve text classification accuracy based on classifier fusion methods," in *10th International Conference on Information Fusion, Quebec, Que.,* Canada, 2007.

[14] S. Buddeewong and W. Kreesuradej, "A new association rule-based text classifier algorithm," in *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, Hong Kong, China, 2005.

[15] A. J. C. Trappey, F.-C. Hsu, C. V. Trappey, and C.-I. Lin, "Development of a patent document classification and search platform using a back-propagation network," *Expert Systems with Applications*, vol. 31, no. 4, pp. 755–765, 2006.

[16] C. H. Li and S. C. Park, "An efficient document classification model using an improved back propagation neural network and singular value decomposition," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3208–3215, 2009.

[17] C. Donghui and L. Zhijing, "A new text categorization method based on HMM and SVM," in *2nd International Conference on Computer Engineering and Technology,* Chengdu, China, 2010.

[18] O. Reyes, C. Morell, and S. Ventura, "Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context," *Neurocomputing*, vol. 168, pp. 168–182, 2015.

[19] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, "Categorizing feature selection methods for multi-label classification," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 57–78, 2018.

[20] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," Computer Science Technical Reports, Iowa State University, USA, 1997.

[21] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* vol. 26, pp. 1424–1437, 2004.

[22] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle

swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459–471, 2007.

[23] J. Derrac, C. Cornelis, S. Garcíac, and F. Herrera, "Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection," *Information Sciences*, vol. 186, no. 1, pp. 73–92, 2012.

[24] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization,* vol. 11, no. 4, p. 341–359, 1997.

[25] A. Onan, S. Korukoğ,, and H. Bulut, "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification," *Expert Systems with Applications*, vol. 62, no. 2015, pp. 1–15, 2016.

[26] T. Baeck, D. Fogel, and Z. Michalewicz, *Evolutionary Computation 1: Basic Algorithms and Operators*. New York, USA: CRC Press, 2000.

[27] E. Hancer, B. Xue, and M. Zhang, "Differential evolution for filter feature selection based on information theory and feature ranking," *Knowledge-Based Systems*, vol. 000, pp. 1–17, 2017.

[28] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Information Sciences*, vol. 507, pp. 67-85, 2020.

[29] E. Hancer, "A new multi-objective differential evolution approach for simultaneous clustering and feature selection," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103307, 2020.

[30] M. Alswaitti, M. Albughdadi, and N. A. M. Isa, "Variance-based differential evolution algorithm with an optional crossover for data clustering," *Applied Soft Computing*, vol. 80, pp. 1–17, 2019.

[31] O. Tarkhaneh and I. Moser, "An improved differential evolution algorithm using Archimedean spiral and neighborhood search based mutation approach for cluster analysis," *Future Generation Computer Systems*, vol. 101, pp. 921–939, 2019.

[32] D. M. Diab and K. M. E. Hindi, "Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification," *Applied Soft Computing*, vol. 54, no. 2017, pp. 183-199, 2017.

[33] L. Jiang, Z. Cai, D. Wang, and H. Zhang, "Improving tree augmented naive Bayes for class probability estimation," *Knowledge-Based Systems*, vol. 26, pp. 239–245, 2012.

[34] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[35] R. Info. (n.d.). UCI Machine Learning Repository [Online]. Available: https://archive.ics.uci.edu/ml/datasets.html. [Accessed 20 May 2018].

[36] Z. Wang, M. Li, Juanzi, and Li, "A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure," *Information Sciences*, vol. 307, pp. 73–88, 2015.

Abhishek Dixit, photograph and biography not available at the time of publication.

Ashish Mani, photograph and biography not available at the time of publication.

Rohit Bansal, photograph and biography not available at the time of publication.