Computers and Electrical Engineering 000 (2018) 1–12

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection☆

S.P. Rajamohana[a,b,*], K. Umamaheswari[a,b]

[a] Department of Information Technology, Coimbatore, Tamilnadu, India
[b] PSG College of Technology, Coimbatore, Tamilnadu, India

ARTICLE INFO

Article history:
Received 11 August 2016
Revised 1 February 2018
Accepted 8 February 2018
Available online xxx

Keywords:
Review spam classification
Feature subset selection
Naive Bayes
kNN and SVM

ABSTRACT

Currently, the masses are interested in sharing opinions, feedbacks, suggestions on any discrete topics on websites, e-forums, and blogs. Thus, the consumers tend to rely a lot on product reviews before buying any products or availing their services. However, not all reviews available over internet are authentic. Spammers manipulate the reviews in their favor to either devalue or promote products. Thus, customers are influenced to take wrong decision due to these spurious reviews, i. e., spammy contents. In order to address this problem, a hybrid approach of improved binary particle swarm optimization and shuffled frog leaping algorithm are proposed to decrease high dimensionality of the feature set and to select optimized feature subsets. Our approach helps customers in ignoring fake reviews and enhances the classification performance by providing trustworthy reviews. Naive Bayes (NB), K Nearest Neighbor (kNN) and Support Vector Machine (SVM) classifiers were used for classification. The results indicate that the proposed hybrid method of feature selection provides an optimized feature subset and obtains higher classification accuracy.

1. Introduction

In current times, the amount of content available to the user on the internet is rapidly increasing [1]. While purchasing the product or availing services customers generally tend to make a decision relying solely on the information available in the review sites [2]. However, there is a limited quality control for these available data. This limitation invites people to post spurious reviews on the websites in order to either promote or demote the products [3]. Such individuals are known as opinion spammers. The positive spam reviews about a product may lead to financial gains and would help to increase the popularity of the product [4]. Similarly, negative spam reviews are posted with the intention of defaming a product or services [5]. Recently, the problem of spam or fake reviews has been on the rise, and many such cases have been released in the news. Hence, there arises a necessity of finding the authenticity of these reviews. Feature selection (FS) is a technique in which a subset of features are selected from the original dataset [6]. It is mainly used to build more robust learning models and to reduce the processing cost. The main purpose of feature selection is to reduce the number of features to increase both the performance of the model and the accuracy of classification [7]. FS can be examined as a search into a state space. Thus, a full search can be performed in all the search spaces traversed. However, this approach is not feasible in case of a very large number of features. Hence, a heuristic search deliberates those features, which have not yet been

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. O. Bayat.

* Corresponding author at: Department of Information Technology, Coimbatore, Tamilnadu, India.
E-mail addresses: spr@ity.psgtech.ac.in (S.P. Rajamohana), uma@ity.psgtech.ac.in (K. Umamaheswari).

https://doi.org/10.1016/j.compeleceng.2018.02.015
0045-7906/

selected at each iteration, for evaluation. A random search creates random subsets within the search space that can be evaluated for importance of classification performance. Due to their randomized nature, meta-heuristics such as particle swarm optimization (PSO), evolutionary algorithms (EA), bat algorithm (BA), ant colony optimization (ACO) and genetic algorithm [8,9] are widely used for feature selection. When the feature space is high dimensional, selecting the optimal feature subset using traditional optimization methods have not proven to be effective. Therefore, meta-heuristic algorithms are used extensively for the appropriate selection of features. Two types of feature selection methods, namely the filter method and wrapper method can be incorporated for selecting subset of features. The filter model analyzes the intrinsic properties of data without involving the use of any learning algorithms [9] and can perform both subset selection and ranking. Though ranking involves identifying the importance of all the features, this method is more specifically used as a pre-process method since it selects redundant features. The wrapper model unlike other filter approaches considers the relationship between features [10]. This method initially uses an optimizing algorithm to generate various subsets of features and then uses a classification algorithm to analyze the subsets generated.

A rule-based approach was investigated to detect fake reviews in which the unexpected rules were defined to detect unusual behaviors of reviewers [11]. The study used an dataset available from Aamazon to identify spam activities. The N-gram method was applied to detect negative deceptive opinion [12]. Gold standard negative spam dataset which contains 400 reviews of 20 hotels in Chicago was used. The unigram and bigram features were trained by Support Vector Machine (SVM) classifiers. The results revealed that, the N-gram based SVM classifier achieved 86% accuracy in surpassing human judges. Two kinds of N-gram methods namely the character n gram (BON) and the word n-gram (BOW) were proposed to detect fake reviews [5]. Naive Bayes (NB) classifier was used for classifying both positive and negative reviews. The experimental results showed that the NB classifier achieved better results for positive reviews. Further, the SVM method was found to show better results in classifying deceptive and truthful negative reviews. The authors claimed that the BON showed better robustness when compared to BOW as it provided superior results with a small training dataset.

The content duplication technique was preferred for identifying the fake review [13]. Both duplicate and near-duplicate reviews were considered in training data set. Furthermore, two different techniques for spam detection were considered in the test dataset. The authors illustrated the content-based features which include 3 categories of reviews. Firstly, similarity of a review with the author's and other reviews on the target products. They also elucidate reviewer's centric features based on the burst patterns. The Probabilistic language model was developed to generate a similarity score between the reviews [14]. This approach evaluates the possibility of one review that are derived from the other. To detect the content similarity, they compared a couple of reviews by Kullback–Leibler. In addition to that Kullback–Leibler divergence measure calculates the spam score for every review. SVM was chosen for spam classification to classify both spam and ham reviews. They have achieved 81% precision in their method for detecting spam reviews.

Stylometric features, characterized either as lexical or syntactic representation were used for identifying review spam. While the lexical features represent the character or word-based features, the syntactic feature denotes the reviewers writing style at each sentence level. Graph-based methodology, the graph comprising three nodes: namely the review, the reviewer and store was applied for detecting review spammers [15,16]. It establishes the inter-relationships between two nodes, which is achieved by evaluating following: the credibility of the reviewer, the honesty of the reviews and the reliability of the store. In this case agreement score is calculated based on the user rating. The reliability of the store depends on the credibility of its reviewer's comments.

The existing works investigated the traditional feature selection techniques such as bag of words, bag of nouns, linguistic features, weighted PCA, keyword spotting and the machine learning algorithm for reviewing spam classification. However, till date no attempts have been made to use hybrid evolutionary algorithms for reviewing spam classification. The evolutionary algorithms have been applied for different applications such as scheduling, power system, and wireless sensor networks. This is the first study that utilizes evolutionary algorithms for classifying reviews into spam and ham. FS plays a major role in classification. Hence, lot of researchers primarily focus on statistical measures to choose the features. However, these methods do not furnish an appropriate solution space. The search space size has increased exponentially corresponding to the number of features in a given data set. Traditional feature selection techniques involve larger number of features. Although all of them are not required during classification, substantial number of irrelevant and redundant features tend to affect the overall performance of the classifier.

2. Proposed model

The proposed methodology uses evolutionary algorithms for FS in order to obtain the feature subset for achieving better accuracy of classification and identification of fake reviews. It consists of four phases namely, preprocessing, feature extraction and feature subset selection using hybrid iBPSO and SFLA and classification. The block diagram of the proposed system is illustrated in Fig. 1.

2.1. Data preprocessing

The data preprocessing phase consists of four phases- tokenization, stop words removal, stemming, and SentiWordNet. First, tokenization process is applied to convert the strings into tokens. Hence, each document is divided into tokens. After the tokenization process, the stop words are eliminated from the dataset. Following this stemming is applied to select the

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

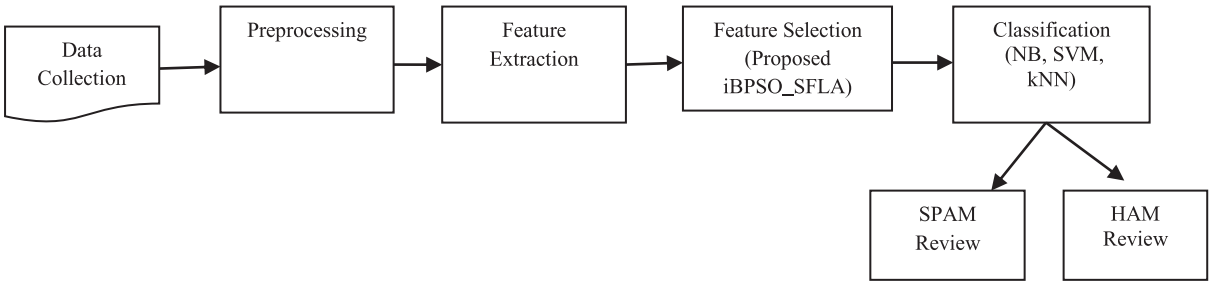

Fig. 1. Block diagram of the proposed iBPSO and SFLA.

root word from the word. Finally, the SentiWordNet is utilized for extracting the features [17]. The aim of SentiWordNet is to provide an extension for Word Net, in such a manner that all synsets can be associated with a value concerning the negative, positive or objective connotation. The positive and the negative scores for review are determined by calculating the average of the positive and negative scores. Then objective score will be calculated using Eq. (1). If the objective score is less than the threshold, the words are eliminated; else, the words are taken for further processing.

$$\boldsymbol{obj_{score}} = 1 - (\boldsymbol{Positivescore} + \boldsymbol{Negativescore}) \tag{1}$$

2.2. Feature extraction

The term frequency (TF) denotes the number of occurrences of each word in the document and d is calculated using the following Eq. (2). The inverse document frequency reduces the weight of words that repeatedly occurs, thus boosting the weight of the lesser frequent words in the document. Inverse document frequency (IDF) is calculated using Eq. (3). TF-IDF is used in information retrieval and text mining [18]. TF-IDF is a product with two statistics namely, term frequency and IDF, and is calculated using Eq. (4).

$$\mathbf{f}(\mathbf{t},\ \mathbf{d}) = \frac{\mathbf{f_t}(\mathbf{d})}{\max_{\mathbf{w}\in\mathbf{d}} \mathbf{f_d}(\mathbf{w})} \tag{2}$$

where w is the maximum weight of any raw term *t* which is present in the whole document *d*. *f*(*t*, *d*)frequency of term in each document d.

$$\mathbf{IDF}(\mathbf{t},\ \mathbf{D}) = \mathbf{log}\frac{|\mathbf{D}|}{|\{\mathbf{d}\ \in \mathbf{D};\ \mathbf{t}\ \in \mathbf{d}\}|} \tag{3}$$

|D|, the total number of documents which are present in the dataset, |{d∈D, t∈d}| denotes the number of documents.

$$\mathbf{TFIDF} = \mathbf{f}(\mathbf{t},\ \mathbf{d}) * \mathbf{IDF}\ (\mathbf{t},\ \mathbf{D}) \tag{4}$$

2.3. Particle swarm optimization

The particle swarm optimization (PSO) algorithm [30] is synonymous with the behavior of birds to flock and fish to school. PSO is applied to wide range to various fields such as task scheduling, fuzzy systems, control and power systems, and classification. According to PSO, the population is considered as a swarm. In a swarm, each individual is represented as a particle. A swarm contains a number of particles (n) and each particle implies a candidate solution on d dimensional search space. Every single particle is associated with a particular velocity. In the population, the *i*th particle is represented using position P_i by (p_{i1}, $p_{i2,}$ …, p_{id}) and velocity V_i by (v_{i1}, v_{i2}, …, v_{id}). every particle moves in the search space to obtain an optimal solution. The movement of each particle is directed by pbest position and gbest position. Each candidate solution is considered as pbest such as (pbesti1, $pbest_{i2}$. $pbest_{id}$). The entire swarm's best position is denoted by gbest such as ($gbest_{i1}$, $gbest_{i2,}$ …, $gbest_{id}$). A fitness value used to evaluate the best position of the particle. The current position, velocity of the ith particle are upgraded using the following Eqs. (5) and (6).

$$\boldsymbol{V_{(t+1)}} = \boldsymbol{W} \times (\boldsymbol{V_t} + \boldsymbol{C}_1 \times \boldsymbol{rand}(0,1) \times \boldsymbol{pBest_t}) - (\boldsymbol{currentvalue_t} + \boldsymbol{C}_2 \times \boldsymbol{rand}\ (0,\ 1) \tag{5}$$

$$\boldsymbol{currentvalue}_{\ (t+1)} = \boldsymbol{currentvalue_t} + \boldsymbol{V}_{\ (t+1)} \tag{6}$$

V (t + 1) represents the particle's former velocity, and V (t) indicates the particle's updated velocity. C1 and C2 are exists as constants. Factor W denotes the inertia weight that range between [0. 0, 1. 0] to bring the impact of the former velocity under control [27]. The current value (t + 1) and current value (t) are updated, former positions of the particles, respectively in Eq. (6). The standard PSO was basically developed for continuous optimization problems [19]. With the aim of feature selection, the real time valued version of PSO algorithm has been extended to either binary or discrete space, which propose

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

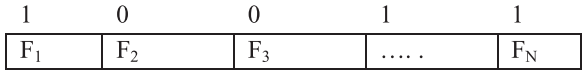

Fig. 2. Solution Representation.

Algorithm 1 Feature selection using iBPSO.

Step 1: Initialize the population randomly for iBPSO.
Step 2: Compute the fitness values for each particle.
Step 3: Compute velocity using linearly decreasing inertia weight and convergence factor in Eqs. (11) and (12).
Step 4: For each particle, estimate pbest and gbest.
Step 5: The fitness value is then compared with overall best value of the population's.
If the current value is better than the gbest, then update gbest.
Step 6: Update the particle's position according to Eqs. (7) and (8) and the current value $_{(t+1)}$ of the new population can be generated.
Step 7: Repeat step (2) until the convergence criteria is met or reaches the utmost number of iterations.

a binary version of PSO (BPSO) [26,28]. In DPSO, each particle tends to progress in a discrete search space. In case of BPSO algorithm, each particle position is restricted between [.,. 0, 1]. To standardize all the real valued velocities between [.,. 0, 1], sigmoid function is applied using Eqs. (7) and (8).

$$\boldsymbol{S(v)} = \frac{1}{1+\boldsymbol{e^{-x}}} \tag{7}$$

In BPSO, each particle is updated using Eq. (8)

$$\boldsymbol{X_i} = \begin{cases} \mathbf{1}, & if\,\boldsymbol{rand}\,() > s(v) \\ \mathbf{0} & otherwise \end{cases} \tag{8}$$

Rand () function denotes a random number ranging from [. . 0 to 1]. In Eqs. (5) and (6), the updated positions of the particle are normalized using the function S (v), where v refers to the particle's updated velocity. If S (v) is greater than the randomly generated number, then its position value of Xi represents {1}, implying that the features that are selected is required for the next update. If S (v) is lesser than a randomly produced number, then the position value of Xi represents {0}, which specifies that the features will not be considered. Cost can be cut down by reducing the computational time, which can be done by setting 500 iterations with the population size of 50. The acceleration parameters, C1 and C2, are set to 2, and the inertia weight parameter is set initially to 0.48, as in [17].

2.4. Feature selection using iBPSO

2.4.1. Candidate solution representation

In iBPSO, each particle position values are considered as a binary bit string which signifies the total number of features (N). If the particle position value of the feature is 1 then, the features are selected; else, they are not selected as represented in Fig. 2.

2.4.2. Objective function

The primary objective of this study is to facilitate the improvisation of the classification accuracy. The accuracy values of NB and kNN classifiers are used as fitness functions [24] for the proposed hybrid approach. The fitness function, fitness (x) is determined using the formula:

$$\boldsymbol{Fitness}\,(\boldsymbol{x}) = \boldsymbol{Accuracy}\,(\boldsymbol{x}) \tag{9}$$

where accuracy (x) refers to the Naive Bayes classification accuracy and the feature subset selection of training data set which is represented by x. The existing works have either used the convergence factor λ or the inertia weight W [25,7,19,26]. Inertia weight plays a vital role in exploration and exploitation. Hence, in the proposed iBPSO Linearly Decreasing Inertia Weight (LDIW) method has been combined with the convergence factor λ as shown in Eq. (10). The convergence factor λ, is calculated using Eq. (11). In LDIW, W_{start} and W_{end} refers to the starting and ending values, t is the iterator over all iterations, T_{max} is the maximum number of iterations as shown in Eq. (12). The Feature selection using iBPSO is explained in Algorithm 1.

$$\boldsymbol{V}_{(t+1)} = \boldsymbol{\lambda}\ \left(\boldsymbol{w_t} \times \left(\boldsymbol{V}_{(t)} + \boldsymbol{C}_1 \times \boldsymbol{rand}(\mathbf{0},\ \mathbf{1})\right) \times \boldsymbol{pBest}_{(t)}\right) - \ \left(\boldsymbol{currentvalue_t} + \boldsymbol{C}_2 \times \boldsymbol{rand}(\mathbf{0},\ \mathbf{1})\right) \tag{10}$$

$$\boldsymbol{\lambda} = \frac{2}{\left|2 - \boldsymbol{c} - \sqrt{\boldsymbol{c}}*\boldsymbol{c} - 4*\boldsymbol{c}\right|} \tag{11}$$

$$\boldsymbol{w_t} = (\boldsymbol{w_{start}} - \boldsymbol{w_{end}})\left(\frac{\boldsymbol{T_{max}}^{-t}}{\boldsymbol{T_{max}}}\right) + \boldsymbol{w_{end}} \tag{12}$$

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

Algorithm 2 Feature selection using hybrid iBPSO _SFLA.

Step 1: Initialize a population of each particle's position and velocity, randomly on search space D.
Step 2: For every particle, compute fitness function.
Step 3: Pbest and Gbest values are obtained for entire population.
Step 4: Compare the fitness value with the population's overall pbest value. If the current pbest value is better than gbest, then reset to the current particles value.
Step 5: Change the particle's velocity and position according to Eqs. (13) and (14) respectively.
Step 6: Loop back to step 2 until a criterion (convergence rate reaches the maximum number of iterations) is met.
Step 7: The possible solution to P's Population; is defined by group of virtual frogs (n).
Step 8: Frogs are sorted in a descending order based on their fitness value and is partitioned into subsets known as memeplexes (m).
Step 9: Frogs i are stated as $X_i = (X_{i1}, X_{i2}, \ldots, X_{is})$, where S denotes the number of variables.
Step 10: Each memeplex, with worst and best fitness of the frog is represented as X_w and X_b, respectively.
Step 11: The best fitness is recognized as X_g.
Step 12: Frog with worst fitness is to be improved according to the Eqs. (13) and (14).

2.5. Feature selection using SFLA

Shuffled frog leaping algorithm (SFLA) aggregates the benefits of genetic-based meta heuristic algorithm and the social behavior of the PSO. It contains the population of a set of frogs are broken down into subsets known as memeplexes [20]. These different memeplexes can be different frogs culture with each of them performing a local search. Among memeplex, each individual frog within a memeplex has ideas, that can be influenced by the other frogs, eventually leading the memetic evolution. After a specific number of evolution, ideas are shared among the memeplexes during shuffling. The local search and the shuffling processes continue until the convergence criteria is met [21]. The initial population, of F frogs is generated randomly. For S-dimensional problems, a frog i is denoted as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iS})$. The frogs are then sorted in a descending order, based on their fitness value. The whole population is split into m memeplexes. Each contains n frogs, that is ($P_{m\times n}$). Frog f1 moves towards the memeplex (M1), following which frog f2 moves to the memeplex (M2), the *m*th frog goes to the *m*th memeplex and hence frog m + 1 moves back to the memeplex (M1) and so on [22]. Within every memeplex, the best and the worst finesses of the frogs identified as X_b and X_w, respectively. Further, the frog with the overall best fitness is represented as X_g. A process that is similar to PSO is used to in each cycle to improve only those frogs with the worst fitness. Hence, the position of a frog that has the worst fitness value is adjusted according to the following Eqs. (13 and 14). The steps involved in feature selection using hybrid iBPSO_SFLA is explained in Algorithm 2.

Change in the position of frog is denoted by,

$$(\boldsymbol{D_i}) \;=\; \boldsymbol{rand}()* \;(\boldsymbol{X_b}.\; \boldsymbol{X_w}) \tag{13}$$

The Position of new frog is denoted as follows:

$$\boldsymbol{X_w} = \boldsymbol{current\;\; position}\; \boldsymbol{X_w} + \boldsymbol{D_i} \tag{14}$$

($D_{max} \geq D_i \geq -D_{max}$), rand () function which represents a random number between 0 and 1. D_{max} is the maximum change in a frog's position. It replaces the worst frog corresponding to the global best frog, where X_g replaces X_b. If there is no improvisation, then a new solution is randomly created to replace the frog. The calculations are then continued for a certain number of iterations [27].

2.6. Training process

Hybrid feature selection approach of iBPSO and SFLA algorithm is used for the improved feature subset selection. For the training process, 80% of the reviews (1280 instances) are taken into consideration. In iBPSO, the earliest population of each particle is created aimlessly in the S-dimension search area. Every particle signifies candidate solution to the problem. Entire swarm is referred to as a population. The particles are expressed as a binary string n which represents a total number of the features. If the position value corresponding to that feature is 1, it indicates the selected features, or else it can be taken as indicating the non-selected features. The fitness function is calculated using accuracy. These fitness values are considered as the pbest values of each particle. Among the pbest values, the highest value is taken as the gbest value. Following this, the position and velocity values are updated for the next iteration. The above steps are performed until the current iteration reaches the greatest iteration value. Finally, the optimized features are selected. An illustrative example for iBPSO is given in Fig. 3.

In hybrid iBPSO and SFLA, the population that contains the optimized feature set of iBPSO is provided as an input to the SFLA algorithm. In SFLA, each frog depicts a candidate solution. The populations are defined by a group of virtual frogs, which are sorted out in a descending order, based on those fitness value. The whole population is broken down into m memeplex. Frogs with the lowest and highest fitness value are known as X_w and X_b. The frog with the highest fitness value is the gbest value denoted as X_g. A frog with gbest value is considered as an optimized feature subset. An illustrative example proposed hybrid iBPSO and SFLA is provided in Fig. 4.

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

	f_1	f_2	f_3	f_4	f_5	....	. . .	f_{n-1}	f_n
$Particle_1^k$	1	0	0	1	1	0	1	0	0
$Particle_2^k$	0	0	1	1	0	0	0	0	1
$Particle_3^k$	1	0	1	0	1	0	1	0	0
. . .	1	1	0	0	1	1	0	0	1
...	0	1	0	1	1	0	1	0	1
$Particle_{N-1}^k$	1	0	1	0	0	1	0	1	1
$Particle_N^k$	1	0	0	0	0	1	1	0	1

Fig. 3. Population Initialization.

$Frog_3^k$	1	0	1	0	1	0	1	0	0
. . .	1	1	0	0	1	1	0	0	1
...	0	1	0	1	1	0	1	0	1
$Frog_{N-1}^k$	1	0	1	0	0	1	0	1	1
$Frog_N^k$	1	0	0	0	0	1	1	0	1

Fig. 4. Frog Initialization.

2.7. Testing process

For the testing purposes, the remaining 20% of the reviews (320 instances) were taken into consideration. NB and kNN algorithm were implemented for calculating the fitness function. Considering the population size from 1 to n, the features from F_1 to F_n as shown in training process.

2.8. Classification

2.8.1. kNN

First up, 1-nearest neighbor (1-nn) is employed for classification. This 1-nn method [29,31] is easily implemented without the need for any optimization. A training sample of N vectors $X_j=(x_{j1}, ..., x_{jd})$, $j=1, ..., N$ is assumed. In the above representation, d refers to the number of features that have been selected and x_{jk} is the description of observation j on feature k. In the 1-nn classifier, an unknown observation $z_i=(z_{i1}, ..., z_{id})$ is classified based on its Euclidean distance using Eq. (15). After calculating the Euclidean distance, zi is put into the class containing its nearest training observation. The kNN method is an extension of the 1-nn method, where k-nearest neighbors are taken into consideration instead of a single neighbor as in 1-nn.

$$\boldsymbol{D_{ij}} = \sqrt{\sum_{\boldsymbol{k=1}}^{\boldsymbol{d}} |\boldsymbol{z_{ik}} - \boldsymbol{x_{ik}}|} \tag{15}$$

2.8.2. NB

The NB classifier makes use of a probabilistic technique to predict a class for every case of data set. NB is one of the most popular text classification method used in many applications namely email spam detection, email sorting, document categorization, content detection, language detection, and sentiment classification [23]. Although it is frequently outdone by other procedures such as Random Forest, Max Entropy, Support Vector Machine etc., the NB classifier is very effective because, it is less rigorous in terms of computational cost. The working process of Naive Bayes is as follows:

Let T be the training sample. Each sample has category labels. A Sample set has totally m classes: $C_1, C_2, ..., C_m$. Every sample is depicted by an n-dimensional vector system design X is denoted as $\{x_1, x_2, ...,x_n\}$, and each vector represents n attributes $A_1, A_2, ..., A_n$. The different ways of calculating the probability of the class are explained below.

1. Given a simple X, the classifier predicts that X belongs to the highest posterior probability of class. If and only if $P(C_i|X) > P(C_j|X)$, $1 <= i, j <= m$, X is predicted to belong to class C_i. According to the Bayes' theorem, the probability is calculated as in Eq. (16).

$$\mathbf{P}\left(\frac{\mathbf{C_i}}{\mathbf{X}}\right) = \frac{\mathbf{P}\left(\frac{\mathbf{X}}{\mathbf{C_i}}\right) * \mathbf{P}(\mathbf{C_i})}{\mathbf{P}(\mathbf{X})} \tag{16}$$

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

Since P (X) remains the same for the rest of classes, it only needs to find the largest P ($X|C_i$)P (Ci). The prior probability of class C_i is calculated. P (C_i) = si/s, si refers to the number of training samples of class Ci, and s is the total number of training samples. If the prior probability of class C_i is not known, it is assumed that the probability of these classes is equal, then P (C_1) = P (C_2) = … = P (C_m). Therefore, the problem is transformed into how to get maximum P ($X| C_i$).

2. If the data set that under certain condition attribute characteristic value is independent of each other. P ($X| C_i$) is calculated which has many attributes, The workload of calculating P ($X| C_i$) is very high. In order to reduce the computational overhead of P ($X|C_i$), assumption Eq. (17).

$$\mathbf{P}\left(\frac{\mathbf{X}}{\mathbf{C_i}}\right) = \prod_{\mathbf{k}=1}^{\mathbf{k}} \mathbf{P}\left(\frac{\mathbf{x_k}}{\mathbf{C_i}}\right) \tag{17}$$

3. Probability P ($x1|C_i$), P ($x2|C_i$), …, P ($x_n|C_i$) can be calculated from the above training data set. Here, *xk* refers to the attribute Ak of sample X.
4. For each class, it calculates P ($X|C_i$)P (C_i) and only if P ($X|C_i$)P (C_i) is maximum, the classifier prediction sample X belongs to class C_i.

2.8.3. SVM

SVM is a popular constructive learning technique, formally defined by a separating hyper plane It is possible to obtain a solution by making a non-linear transformation of the original input set into a high dimensional feature set, where an optimal separating hyper plane can be found [5]. Though SVMs are efficient in classification, they have certain limitations in terms of identifying the choice of kernel, speed and size in training and testing, computational complexity, and memory requirements. The whole process of support vector machine based classification begins with getting the data collection α = {d1, d2, d3, …, dn} where each data corresponds to a domain or category C = {c1, c2, c3, …, cn} and a feature space F = {f1, f2, f3, …, fn}. The sample data are identified and mapped to W = {w1, w2, w3, …, wn} where 'w' refers to the weight of the document. The feature vectors are then fed as input to the SVM classifier to train the system. The data are mapped as +/−1 based on the relevancy during training. The unclassified or the test vector is fed to the classifier system and the output is predicted once the system is trained.

2.8.4. Performance evaluation

The evaluation of the classification performance is based on three metrics namely accuracy, precision, and recall as defined in the following Eqs. (18)–(20).

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}} \tag{18}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{20}$$

3. Simulation results and discussion

The proposed hybrid iBPSO and SFLA algorithms were implemented using Java with Intel P4, 2. 66 GHz CPU; 16GB RAM in Windows XP Professional operating system environment. In this experiment, hybrid iBPSO and SFLA FS algorithms were implemented for selecting the optimized subsets from the review spam dataset. The stages of the proposed methods results are presented below.

3.1. Dataset description

The proposed method uses the dataset developed by Ott. et al. [12], which consists of 1600 reviews of the 20 most popular Chicago hotels that are organized as follows: 800 positive reviews, out of which 400 are truthful, and 400 are deceptive, and 800 negative reviews, out of which 400 are truthful, and 400 deceptive. From this review dataset, 80% (1280 instances) of the reviews were used for training and the remaining 20% (320 instances) used for testing with significant features. The average length of a single review was around 600 characters. The aim was to classify such reviews into two categories: truthful and deceptive reviews.

3.1.1. Example of a typical truthful review dataset

My husband and I stayed for two nights at the Hilton Chicago, and enjoyed every minute of it! The bedrooms are immaculate, and the linnens are very soft. We also appreciated the free wifi, as we could stay in touch with friends while staying in Chicago. The bathroom was quite spacious, and I loved the smell of the shampoo they provided-not like most hotel shampoos. Their service was amazing, and we absolutely loved the beautiful indoor pool. I would recommend staying here to anyone.

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

Table 1
iBPSO parameters setup.

Population size	50
Maximum iterations	500
C1	2
C2	2
Inertia weight (w)	0.48

Table 2
SFLA parameters setup.

Population size	50
No of Memeplexes	10
Size of Memeplex	5
No of iterations	500
D_{max}	50

Table 3
Comparison of the number of features with various feature selection techniques.

Techniques	No. of features
LSI	2433
SentiWordNet	1771
iBPSO	772
SFLA	723
Hybrid iBPSO and SFLA	642

Table 4
Comparison of the gbest values of iBPSO, SFLA and hybrid iBPSO and SFLA.

Algorithm	Global best value
iBPSO	0.8455
SFLA	0.8743
Hybrid iBPSO and SFLA	0.9291

3.1.2. Example of a typical deceptive review dataset

The Affinia Manhattan is fantastic! My husband and I stayed there when we went to visit my sister. I loved the room. It was one of the best hotel beds we have ever slept in. The view was incredible. Manhattan is one of the most beautiful places I have ever been. The staff was very helpful as well. They had no problem going out of their way to be helpful. I would suggest this hotel to anybody!

3.1.3. Parameter settings

The parameters used for the proposed iBPSO and SFLA are shown in the Tables 1 and 2. After initializing the parameters, the fitness function was calculated using classification performance and was used to assess the selected subspaces of features for each dataset. The training process was implemented using 10 fold cross validation method.

Prior to applying FS and classification methods, the review spam dataset was first preprocessed. After tokenizing a document, stop words were removed from the document. The number of features after stemming is decreased. The number of features obtained at every step are shown in Table 3. The well-known sentiment lexicon known as SentiWordNet and contains about 10,000 words approximately with both positive and negative score. Accordingly, after applying SentiWordNet, the feature count was reduced to 53,648. Moreover, the duplicate features, were removed as well. Thus, the feature count was reduced to 1771. In iBPSO based feature selection, the initial parameters such as the number of features, population size, and the number of iterations are provided as inputs. Each particle position values are then randomly initialized between . 0 and . 1. The fitness values for each particle was computed using classification accuracy. where the pbest and gbest values were noted. The velocity and the position values were updated for achieving maximum iteration. After running 500 iterations, the final gbest values with their corresponding features are identified as the optimized feature subset. In this hybrid iBPSO and SFLA, the optimized feature subset of iBPSO was provided as an input to the SFLA. In SFLA, the features are sorted according to their fitness values and are further divided into memeplexes, as specified. After calculating the fitness of each frog, the gbest values are calculated for each memeplex. The final gbest value of the proposed hybrid iBPSO and SFLA are identified as an optimized feature subset. The results of hybrid iBPSO and SFLA are shown in Table 4. A comparison of the results of iBPSO and SFLA revealed that the proposed hybrid iBPSO and SFLA provided better results. The total number of features selected in iBPSO and SFLA and hybrid iBPSO and SFLA are given in Table 3.

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

Table 5
Spam and ham features.

Pleased, wonderful, Comfortable, Amazing, Loved, Great, Marveling, Fabulous, Damage, Horrible, Discomfort, Ignorance, Refused, Lacked, Complaint, Absurd, Blatantly, Worst.

Table 6
Number of reviews classified as spam or ham.

Reviews	Naive Bayes	kNN	SVM
SPAM	711	622	537
HAM	889	978	1063

Table 7
Performance analysis of classification accuracy using different inertia weight.

Different inertia weight	iBPSO	SFLA
W	0.8884	0.8617
$\lambda + W$	0.8909	0.8896
LDIW	0.9106	0.9139
LDIW $+ \lambda$	0.9318	0.9181

Table 8
Number of iterations compared against classification accuracy.

Number of iterations	Accuracy	
	NB	kNN
100	0.8252	0.7738
200	0.8528	0.8338
300	0.8922	0.8433
400	0.92,212	0.8634
500	0.9344	0.8894

Table 9
Performance analysis of classification accuracy using various feature selection techniques.

Techniques	Feature selection techniques		
	BPSO	SFLA	iBPSO_SFLA
NB	82.5	85.5	94.97
kNN	85.05	87.07	92.12
SVM	88.38	89.84	91.25

A sample subset of features containing both ham and spam to classify the review into spam or not spam is shown in Table 5.

After selecting the optimized feature subsets, the reviews are classified into spam and ham using the Naive Bayes and kNN. Table 6 depicts the reviews classified as spam and ham.

Different inertia weight was applied and LDIW was combined with the convergence factor λ in iBPSO as shown in Table 7. The results obtained show that the iBPSO achieves quick convergence and they have been compared with the existing methods and classified using Naive Bayes, kNN and SVM. From the results obtained, it can be observed that the iBPSO based feature selection has attained the optimized feature subset, which improves classification accuracy.

For the purpose of comparisons, the results of Naive Bayes, kNN and SVM are included using hybrid approach. Feature selection plays such a significant role in classifying the spam reviews. By using evolutionary algorithms, an optimized feature subset was obtained, thus increases the classification accuracy. The accuracy values obtained for iterations 100, 200, 300, 400, and 500 are shown in Table 8. In iBPSO, different inertia weights were used. The hybrid iBPSO and SFLA were applied to select the optimized feature subset. It can be observed that Naive Bayes achieves better results than kNN.

From the results, it can be said that the hybrid approach using Naive Bayes Classifier has the highest classification accuracy at 94. 97%, when compared against kNN and SVM. The number of deceptive reviews detected using Naive Bayes classifier was higher when compared to the number of deceptive reviews detected using the kNN and SVM classifiers, as shown in Table 9.

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

Table 10
Comparison of evaluation metrics such as accuracy, precision, and recall for the three methods used.

Techniques	Measures		
	Accuracy (%)	Precision (%)	Recall (%)
Naive Bayes	94.97%	86.10%	78.50%
kNN	83.56%	77.28%	75.45%
SVM	74.26%	69.39%	64.78%

A comparison of the values obtained for accuracy, precision, and recall values for the proposed hybrid approach using Naive Bayes, kNN, and SVM are represented in Table 10. From the results, it can be inferred that Naive Bayes produces the highest and better classification accuracy than the other two methods.

4. Conclusion

Feature selection is critical to the performance improvement for a classification. Hence, it is important to discard the irrelevant and, noisy features from a given dataset that would decrease the classification accuracy. A number of methodologies have been adopted to select the best feature subset., In this investigation, an hybrid approach was applied for selecting the optimized feature subset. This hybrid methodology efficiently reduces the feature subset size due to randomization, which in turn improves the accuracy of the classifier. Moreover, the results when compared against the existing feature selection techniques, indicate that the proposed feature selection technique offers classification accuracy and is efficient. Furthermore, the method classifies the reviews into spam and ham reviews efficiently. Thus, it can be concluded that the Naive Bayes classifier shows better classification performance than the kNN and SVM classifiers.

Acknowledgments

The authors would like to acknowledge the efforts from Dr. Biswapriya B. Misra [ORCID ID: 0000-0003-2589-6539], Assistant Professor, Internal Medicine, Wake Forest Baptist Medical Center, Winston-Salem, NC, USA for extensive help in editing the current version of the manuscript for language issues.

References

[1] Li J, Ott M, Cardie C, Hovy EH. Towards a general rule for identifying deceptive opinion spam. In: ACL, vol. 1; 2014. p. 1566–76.
[2] Mukherjee A, Venkataraman V, Liu B, Glance N. What yelp fake review filter might be doing?. Seventh international AAAI conference on weblogs and social media; 2013.
[3] Banerjee S, Chua AY. Applauses in hotel reviews: genuine or deceptive?. In: Science and information conference (SAI), 2014. IEEE; 2014. p. 938–42.
[4] Ahmad I, e Amin F. Towards feature subset selection in intrusion detection. In: Information technology and artificial intelligence conference (ITAIC), 2014 IEEE 7th joint international. IEEE; 2014. p. 68–73.
[5] Fusilier DH, Montes-y-Gómez M, Rosso P, Cabrera RG. Detecting positive and negative deceptive opinions using PU-learning. Inf Process Manage 2015;51(4):433–43.
[6] Vieira SM, Mendonça LF, Farinha GJ, Sousa JM. Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. Appl Soft Comput 2013;13(8):3494–504.
[7] Abdul-Rahman S, Bakar AA, Mohamed-Hussein ZA. Optimizing big data in bioinformatics with swarm algorithms. In: Computational science and engineering (CSE), 2013 IEEE 16th international conference on. IEEE; 2013. p. 1091–5.
[8] Chuang LY, Li JC, Yang CH. Chaotic binary particle swarm optimization for feature selection using logistic map. In: Proceedings of the international conference of engineers and computer scientists; 2008.
[9] Nakamura R, Pereira L, Costa K, Rodrigues D, Papa J. BBA: a binary bat algorithm for feature selection. In: Conference on graphics, patterns and image, Ouro Preto; 2012. p. 22–5.
[10] Xue B, Zhang M, Browne WN. Particle swarm optimization for feature selection in classification: a multi – objective approach. IEEE Trans. Cybern. 2013;43(4):1656–71.
[11] Jindal N, Liu B, Lim EP. Finding unusual review patterns using unexpected rules. In: Proceedings of the 19th ACM international conference on Information and knowledge management. ACM; 2010. p. 1549–52.
[12] Ott M, Choi Y, Cardie C, Hancock JT. Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies-volume 1; 2011. p. 309–19.
[13] Lin Y, Zhu T, Wu H, Zhang J, Wang X, Zhou A. Towards online anti-opinion spam: spotting fake reviews from the review sequence. In: Advances in social networks analysis and mining (ASONAM), 2014 IEEE/ACM international conference on. IEEE; 2014. p. 261–4.
[14] Lai CL, Xu KQ, Lau RY, Li Y, Jing L. Toward a language modeling approach for consumer review spam detection. In: e-Business engineering (ICEBE), IEEE 7th international conference on. IEEE; 2010. p. 1–8.
[15] Shojaee S, Murad MAA, Azman AB, Sharef NM, Nadali S. Detecting deceptive reviews using lexical and syntactic features. In: Intelligent systems design and applications (ISDA), 2013 13th international conference on. IEEE; 2013. p. 53–8.
[16] Wang G, Xie S, Liu B, Philip SY. Review graph based online store review spammer detection. In: Data mining (ICDM), IEEE 11th international conference on; 2011. p. 1242–7.
[17] Asghar MZ. Detection and scoring of internet slangs for sentiment analysis using SentiWordNet.. Life Sci J 2014;11(9).
[18] Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies-volume 1. Association for Computational Linguistics; 2011. p. 142–50.
[19] Soliman OS, Elhamd EA. Classification of hepatitis C virus using modified particle swarm optimization and least squares support vector machine. Int J Sci Eng Res 2014;5(3):122.
[20] Eusuff M, Lansey K, Pasha F. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Eng Optim 2006;38(2):129–54.

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

[21] Afzalan E, Taghikhani MA, Sedighizadeh M. Optimal placement and sizing of DG in radial distribution networks using SFLA. Int J Energy Eng 2012;2(3):73–7.
[22] Farahani M, Movahhed SB, Ghaderi SF. A hybrid meta-heuristic optimization algorithm based on SFLA. In: 2nd international conference on engineering optimization, September; 2010. p. 6–9.
[23] Rajamohana SP, Umamaheswari K, Karthiga R. Sentiment Classification based on LDA using SMO Classifier. Int J Appl Eng Res 2015;10(55):1045–9.
[24] Chuang LY, Yang CH, Li JC. Chaotic maps based on binary particle swarm optimization for feature selection. Appl Soft Comput 2011;11(1):239–48.
[25] Lin SW, Ying KC, Chen SC, Lee ZJ. Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Syst Appl 2008;35(4):1817–24.
[26] Inbarani HH, Azar AT, Jothi G. Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. Comput Methods Programs Biomed 2014;113(1):175–85.
[27] Jadidoleslam M, Bijami E, Amiri N, Ebrahimi A, Askari J. Application of shuffled frog leaping algorithm to long term generation expansion planning. Int J Comput Electr Eng 2012;4(2):115.
[28] Bharti KK, Singh PK. Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering. Appl Soft Comput 2016;43:20–34.
[29] Cervantes A, Galván IM, Isasi P. AMPSO: a new particle swarm method for nearest neighborhood classification. IEEE Trans Syst Man Cybern Part B 2009;39(5):1082–91.
[30] Kennedy J, Eberhart R. PSO optimization. In: Proc. IEEE Int. Conf. Neural Networks, vol. 4. Piscataway, NJ: IEEE Service Center; 1995. p. 1941–8.
[31] Duda RO, Hart PE. Pattern recognition and scene analysis. New York: Wiley; 1973.

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015

S. P. Rajamohana is an Assistant Professor in the Department of Information Technology at the, PSG College of Technology, Coimbatore, India. She completed her Master's in Information Technology from the same institution and is currently pursuing her PhD in Information and Communication Engineering from PSG College of Technology, Anna University, Chennai. Her research interests include review spam classification and evolutionary algorithms.

K. Umamaheswari, Professor & Head in the Department of Information Technology, at the PSG College of Technology, India and has completed her Bachelor's and Master's in Computer Science and Engineering in 1989 and 2000 respectively and PhD degree from Anna University in 2010. She has 22 years of teaching experience and more than 100 Publications in international and national journals and conferences. Her research interests include data mining, cognitive networks and information retrieval.

Please cite this article as: S.P. Rajamohana, K. Umamaheswari, Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.015