Pertanika J. Sci. & Technol. 26 (3): 1355 - 1372 (2018)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Detection of Spam Using Particle Swarm Optimisation in Feature Selection

Surender Singh[1,2]* and Ashutosh Kumar Singh[2]

[1]*Department of Information Technology, Maharaja Surajmal Institute of Technology, New Delhi, India*
[2]*Department of Computer Applications, National Institute of Technology, Kurukshetra, Haryana, India*

ABSTRACT

Spamming is a major issue in the area of web search. There are many features (Link & Content based) which are used for spam and non-spam classification. This paper recommends CFS+PSO, which takes the advantages of swarm behaviour (uses randomness and global communication between particles) and Correlation Based Feature Selection Technique (CFS). The objective of feature selection is to build logical model with improved performance in time and accuracy. The performance of CFS+PSO is evaluated on WEBSPAM-UK2006 with Multilayer Perceptron (MLP), Naïve Bayes, Support Vector Machine (SVM), J48 & AdaBoost. Experimental results show great decline in existing features and computational time while increases in the accuracy measures (F_1 Score and AUC).

Keywords: Content and link based features, correlation based feature selection, data mining, filter and wrapper model, particle swarm optimization, spam

INTRODUCTION

Web spamming (which is known as spamdexing) is recognised as one of the main problems of search engines (Gyongyi & Garcia-Molina, 2005). Nowadays, information retrieval (IR) is a main concern of search engine industries. Spam not only corrupts the search quality but along with it, weakens the trust of users in a particular search engine and leads to phishing (Webber, Maria de Fátima, & Hepp, 2012). The manipulation can be done in different forms like adding content spam and link spam. Content spam is a common area for spammers because search engines (like Google & Yahoo!) use models which are based on the rank and content of websites. Primarily based on the web document structure, content spamming is subdivided into five categories, namely, title, body, meta-tags, anchor text, and URL

ARTICLE INFO

Article history:
Received: 29 December 2017
Accepted: 30 March 2018

E-mail addresses:
surenderbhanwala333@gmail.com (Surender Singh)
ashutosh@nitkkr.ac.in (Ashutosh Kumar Singh)
*Corresponding Author

ISSN: 0128-7680

spamming (Gyongyi & Garcia-Molina, 2005). In link spamming, the spammer creates a large number of links to a page just to increase the link based rank.

Many conventional algorithms are developed for spam detection but infeasible due to the dynamic growth of the Web (Becchetti, Castillo, Donato, Baeza-Yates, & Leonardi, 2008). However, machine learning (ML) algorithms give better results due to their ability to study the necessary patterns (Goh & Singh, 2015). Along with ML algorithms, feature selection (FS) shows a crucial part in the success of spam detection. Most machine learning strategies or algorithms degrade in execution when executed with most features that are not essential or repetitive for anticipating the desired results (Li, Li, & Liu, 2017).

In a broad way, FS is divided into the following: filter technique, wrapper technique and hybrid technique (Yu & Liu, 2004). Statistical analysis is required for features without any ML structure while the ML model is assumed in wrapper method to confirm (validate) the learning performance of the particular model (Guyon & Elisseeff, 2003; Dash & Liu, 1997). The hybrid model or technique takes the strengths of both models (Huang, Cai, & Xu, 2007). The representation of all three models or techniques are given in Figure 1.

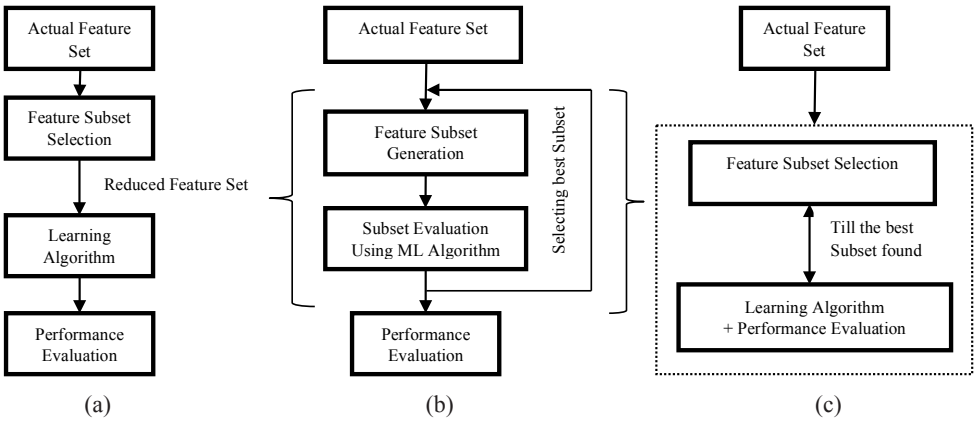


Figure 1. (a) Filter Technique (b) Wrapper Technique (c) Hybrid Technique

The best (optimum) features can be created and the search process is carried out in some ways, like SFS (sequential forward search), SBS (sequential backward search) and Bidirectional search. In SFS (Guan, Liu, & Qi, 2004; Reunanen, 2003), the search process starts with a blank set and adds features successfully but SBS search starts with a complete set and then eliminates features (Gasca, Sanchez, & Alonso, 2006; Hsu, Huang, & Dietrich, 2002). Bidirectional selection search starts with both sides and increases and eliminates features at the same time (Caruana & Freitag, 1994). There is an alternative search called complete search but is not possible with huge number of features (like in web spam area). According to Pudil, Novovicova and Kittles (1994), an additional search algorithm, known as floating search is proposed because of nesting effect in SFS or SBS.

Many of the above search techniques (or strategies) use local instead of global search (Kabir, Shahjahan, & Murase, 2012). These algorithms suffer from computational complexity due to partial search over the features set (space). Thus, most of the research is carried out on metaheuristics (nature inspired) algorithms (Ke, Feng, & Ren, 2008). The two primary segments of metaheuristic algorithms are determination of best solutions and randomisation. The former affirms that solutions will join to optimality while the latter keeps away the solution being stuck at local optima and increase the differing potentials of solutions. To accomplish these objectives, researchers have tried many metaheuristic methods including Firefly algorithms (Yang & He, 2013), Harmony algorithms (Ramos, Souza, Chiachia, Falcao, & Papa, 2011), Bee algorithms (Karaboga & Basturk, 2007), GA (Yang & Honavar, 1998), Ant Colony Optimisation (Kabir, Shahjahan, & Murase, 2009), Simulated Annealing (Filippone, Masulli, & Rovetta, 2006) and Wolf-Search algorithms (Song, Fong, & Tang, 2016; Tang, Fong, Yang, & Deb, 2012) in solving feature subset selection in various problems.

In this work, a CFS+PSO technique is suggested, which is combination of correlation based feature selection approach and particle swarm optimisation methodology that uses a hybrid search approach in feature space of web-spam area. The key emphasis of this technique is to create subsets of significant features of lesser size. This method exploits swarm intelligence in search strategy which combines with filter technique. The objective is to search the global best solution among the current best. Also, hybrid techniques are proficient in discovering a better answer, when a single method is frequently restricted with insufficient solution.

The rest of the paper is sequenced in five segments. Related work regarding feature selection and search methods are explored in segment 2. Segment 3 explains the CFS+PSO technique. Experimental and parameters setting are discussed in segment 4. The outcome results are displayed in segment 5. Finally, the conclusion with future scope is described in segment 6.

RELATED WORK

Basically, FS is a method of rejecting the irrelevant features to enhance the performance (time & accuracy) of ML algorithms. FS can be categorised mainly in two types: feature subset determination and feature ranking based on how the features are joined for assessment. Space and computational complexity is high in feature subset selection approach because it evaluates the individual feature subset with a feature selection metric (correlation or consistency) using any one of the searching techniques (Bolón-Canedo, Sánchez-Maroño, & Alonso-Betanzos, 2013). In feature ranking, each feature is ranked using selection metric such as chi-square feature evaluation, information gain, gain ratio attribute assessment and symmetric uncertainty. Then, top ranking attributes or features are selected as significant features by some threshold value. Space and time complexity is less compared to subset selection for this approach. Furthermore, FS algorithms can be divided in three classes: filter, wrapper and hybrid. Filter is done by ranking of features and determination of feature subset while wrapper creates subsets by use of any search technique, then assesses these sets using machine learning classifiers (Dash & Liu, 1997). Run time and search overhead is increased in comparison with filter technique.

Neural Network (NN), Support Vector Machines (SVM) and Boosting Algorithm can be used as a classification function in wrapper or filter method (Breiman, 1996, 2001; Cortes & Vapnik, 1995; Haykin & Lippmann, 1994; Quinlin, 1993).

In actual fact, filter methods are fast and easy to implement because of the assessment of features without any model presumed between outputs and inputs of the data. Chow and Huang (2005) present an idea of mutual information. PCA method is implemented to remove redundancy between the segments of high-dimensional vector data which empowers a lower-dimensional data without main loss of information (Kambhatla & Leen, 1997). Abdulla and Kasabov (2003) designed a model where features are reduced by refining the dominance effects.

The new FS technique is based on chi-square statistical measure (CHIR) given by Li, Luo and Chung (2008). Song, Ni and Wang (2013) propose a clustering technique in which features are separated into clusters, then highly illustrative feature that is intensely connected to objective classes is selected from clusters to build a new subset of features. Sotoca and Pla (2010) have applied hierarchical clustering technique.

Searching also shows a very critical role in finding of significant features from a given dataset for any feature selection method. Different types of sequential or metaheuristic algorithms are proposed for the searching problem. Sequential search is applied by many researchers (Gasca et al., 2006; Guan et al., 2004; & Hsu et al., 2002). Uğuz (2011) proposed Information Gain (IG) and GA & PCA (feature selection and extraction methods). Shunmugapriya and Kanmani (2017) applied a hybrid technique, which takes the benefits of ACO as well as Artificial Bee Colony (ABC) techniques to optimise FS. Some bio-inspired metaheuristic algorithms are also invented which includes Firefly (Yang, 2009), Cuckoos (Yang & Deb, 2009) and Bats (Yang, 2010).

PROPOSED METHODOLOGY

Feature selection method includes four steps, which are defined as: (a) subset generation - selection of an initial point (feature subset) because it can influence the search direction, (b) evaluation function - this step assesses the subset produced in the previous step by using filter or wrapper approach, since previous approach is autonomous of the induction algorithm while wrapper techniques use induction algorithm for assessing the weight of highlight subsets, (c) stopping criteria - a stopping point must be chosen because dependent upon the valuation policy, a feature selector may leave including (or expelling) features (elements) when the quality value of a present feature subset is not increasing, and (d) validation methodology - validation technique is to check whether the feature subset choice is substantial or not. Usually the result of the original feature set is compared with the feature set chosen in the previous step as input to some induction algorithm utilising datasets (Dash & Liu, 1997).

Feature Selection (CFS)

CFS algorithm selects features depending on correlation based heuristic assessment function (Hall, 1999). The preference of the assessment function is towards subsets whose features are extremely related with the class but independent of other features. Unrelated features are disregarded on the fact that they show less association with the class while other features are

separated out as they will be very much linked with at least some of the features. CFS evaluation (or assessment) function (Senliol, Gulgezen, Yu, & Cataltepe, 2008) is as:

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (1)$$

where M_s is the heuristic "function" of a feature subset S containing k features, $\overline{r_{cf}}$ is the mean feature-class correlation (f∈ S), and $\overline{r_{ff}}$ is the average of feature-feature intercorrelation. $k\overline{r_{cf}}$ denotes predictiveness of the class with a set of features where $\sqrt{k + k(k-1)\overline{r_{ff}}}$ shows redundancy between the features. The above condition is the fundamental of CFS and the set of features with the highest value found during the process is utilised to reduce the size of both the training and testing set. Figure 2 demonstrates the steps of the CFS algorithms.

Input:
S – Data set with features F, |F| = total no of features
M_s – Heuristic merit to be maximized
SG – Successor Generation operator
Output:
P = Starting Point (F);
Solution = Best of P according to M_s;
Repeat
P = Search_Strategy (P, SG (M_s), F);
F'= Best of P according to M_s;
If M_s (F') > M_s (Solution) or (M_s (F') = M_s (Solution) and |F'| < |Solution|) then Solution = F';
Until Stop (M_s, P).

Figure 2. Steps involved in CFS Algorithm

Optimisation Strategy

Optimisation is dominant in many applications. The main idea behind optimisation is to minimise the cost (computation, time and resources) and to maximise the performance and proficiency. Due to the various constraints in real world applications, we have to find optimal solutions. PSO was developed in 1995, based on the swarm nature to monitor the particles for searching global best solutions. However, PSO has many resemblances with GA and Virtual Ant Algorithms but instead of using crossover (or mutation), it takes advantage of global communication between the particles.

PSO searches for space of an objective function by adjusting the paths of individual particles. The movement of particles can be described by two main segments: stochastic and deterministic. The particles are involved towards the position of $\boldsymbol{gb}^*$ (global best) and their best location ($\boldsymbol{x}_i^*$), while in the interim it tends to move arbitrarily. When the particles find a position which is superior to any earlier found position, it updates it as the new current best for that particle. The main objective is to search the globally best solution between all the existing best solutions until there is no more improvement. The main steps of PSO is summarised in algorithm shown in Figure 3. Each particle has a position in the search area, which is denoted by $x_i = (x_1, x_2, x_3, x_4 \ldots x_n)$. Particles scan or move in search space for the best solutions. Every particle has a velocity, which is denoted as $v_i = (v_1, v_2, v_3, v_4 \ldots v_n)$. During this movement, each particle refreshes (updates) its velocity and position according to its own and its neighbour's experience (Eq. 2 & Eq. 3).

$$v_i^{t+1} = w * v_i^t + c_1 * r_1 * (gb^* - x_i^t) + c_2 * r_2 * (x_i^* - x_i^t) \quad (2)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3)$$

Here t signifies the count of iteration and w is inertia weight in the optimisation process, which is for controlling the influence of the previous velocities on the current velocity. The c_1 and c_2 are learning parameters (or acceleration constants) and r_1 and r_2 are random values distributed in between 0 and 1. The algorithm ends when a predetermined condition is achieved, which can be a good fitness value or a maximum number of iterations (Xue, Zhang, & Browne, 2013).

Start
Objective function F(x) where x= $x_1, x_2, x_3, \ldots\ldots x_D$
Initialize locations x_i and velocity v_i of n particles.
Initial minimum $f_{min}^{t=0}$= min {f(x1), f(x2), ...f(xn)} where t=0
While *(stopping criterion)*
t= t+1 (counter)
For *(loop over n particles & all D dimensions)*
Generate new velocity v_i^{t+1} using $v_i^{t+1} = w * v_i^d + c_1 * r_1 * (gb^* - x_i^t) + c_2 * r_2 * (x_i^* - x_i^t)$
Calculate new locations $x_i^{t+1} = x_i^t + v_i^{t+1}$
Evaluate objective functions at new locations x
Find the current minimum f_{min}^{t+1}
End for
Find the current best x_i^* & current global best gb^*
End while
Output the results x_i^* & gd^*
End

Figure 3. Algorithm for Particle Swarm Optimisation

CFS+PSO Algorithm

In this segment, CFS+PSO algorithm is recommended, which is used to assess the importance and the dismissal of the selected feature subset. CFS+PSO utilises correlation based feature technique to form the fitness functions and assessment of integrity of the reduced feature subset. For a feature subset X with m features, X = (x_1, x_2, x_3, x_4……. x_m), CFS assesses mean of association between feature-class and average of intercorrelation between feature-feature to decrease the classification error or increase accuracy (by using Eq.1). The subset of independently good features may not be the best combination because of redundancy between features.

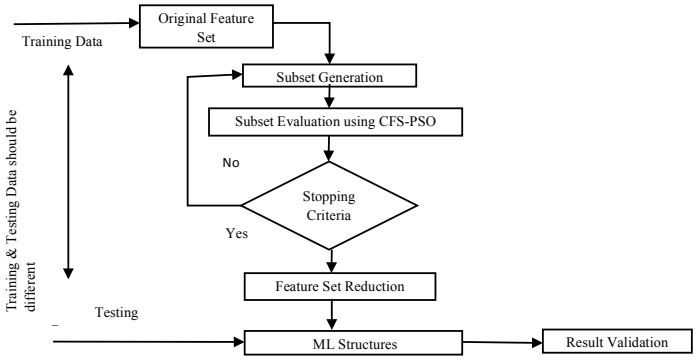


Figure 4. Correlation based feature selection method with PSO

The removal of redundant features can increase performance due to the reduction of the dimensionality. In PSO, every solution of the problem is denoted by a particle, which can be represented by array (or vector). Particles move in search space to search for the best solutions and during this movement, every particle can remember its best experience with its neighbours. So, all particles search for the optimal (best) answer by updating the position of each particle, based on its best experience and its nearby particles. Steps involved in CFS+PSO algorithm is shown in Figure 5.

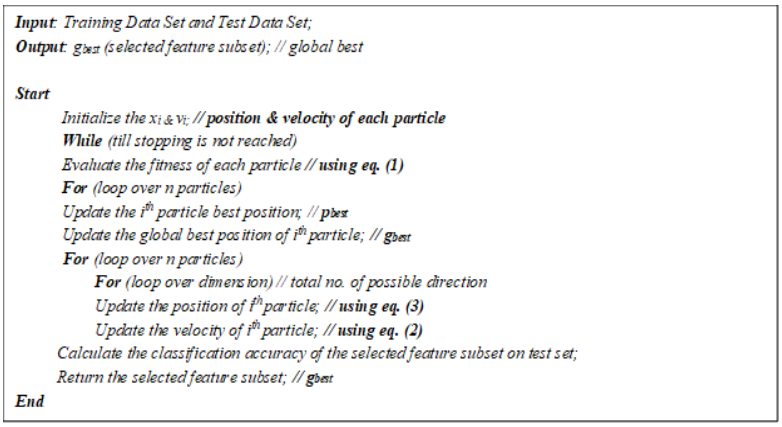


Figure 5. Algorithm for correlation based feature selection with PSO

BENCHMARK AND PARAMETER SETTING

In this paper we use publicly available benchmark WEBSPAM UK-2006, which consists of 777, 410, 46 pages in 11,402 hosts in the UK domain ("WEBSPAM UK-2006", 2006). Features are distributed as content and link-based. In Table 1, A denotes the content-based features (Ntoulas, Najork, Manasse, & Fetterly, 2006). There are a total of 96 full content-based features which are denoted by B (Castillo, Donato, Gionis, Murdock, & Silvestri, 2007). Label C represents the link-based features. The transformed link-based features are designated by set D. This conversion works better for classification than the raw link-based features. More detailing on these features is described by (Becchetti, Castillo, Donato, Leonardi, & Baeza-Yates, 2006). For experimental purpose, the authors combined some feature sets, like A+C and B+D (Singh & Singh, 2018; Goh & Singh, 2015; Goh, Singh, & Lim, 2013; Singh, Kumar, & Leng, 2011;).

Table 1
Distribution of features in dataset (WEBSPAM UK-2006)

Total Features	Name of feature Set
24	Content Based Features (A)
96	Full Content Based Features (B)
41	Link Based Features (C)
138	Transformed Link Based Features (D)
65	Content + Link Based Features (A+C)
234	Full Content + Transformed Link Based Features (B+D)

The classification (training and testing set) in spam database is presented in Table 2.

Table 2
Classification of Spam and Ham in Benchmark

	Benchmark	
	Training Set (Set 1)	Testing Set (Set 2)
Spam	553	1250
Ham (Non-Spam)	3810	601
Total	4363	1851

For the classification, the following algorithms were used: Naïve-Bayes, SVM, J48, MLP and AdaBoost. The Naive Bayes is a basic version of Bayes formula with strong independent assumptions between features and concludes which class a unique instance belongs to. J48 is the implementation of algorithm C4.5, and its predecessor, that summarises training data in the form of a decision tree. Random forest builds multiple decision trees and outputs the class which is mean prediction of the individual trees. SVM classifier gives high-dimension features using hyperplanes which provide the largest minimum distance to divide data points between classes. MLP is a feedforward neural network model that maps the weighted inputs to the output of each neuron using multi-weights connections. AdaBoost also called Adaptive Boosting is used for building strong classifiers with linear combination of weak classifiers.

Evaluation Criteria

After construction of the classifier, it must be assessed for accurateness. Effective estimation is also significant because without knowing the expected accuracy, it cannot be used in real-world problems. Confusion matrix for binary classification (spam or non-spam) was used. The main measure is classification accuracy which is totally correct prediction divided by total cases in data set. But here, the authors took two other evaluation criteria - area under the ROC (AUC) and F_1 score (F-measure) because it cannot be said that any one of the classifiers is strictly better than the other. ROC is 2-dimensional graphs where FP rate (FPR) and TP rate is plotted on X and Y axis respectively. It represents trade-off between costs (FP) and benefits (TP). F_1 score is the harmonic mean of precision and recall (Eq. 4). Precision is correct positive cases divided by total positive predicted cases whereas recall is termed as the count of correct positive predictions divided by the total count of positives cases.

$$F - measure = 2.\frac{precesion * recall}{precesion + recall} \tag{4}$$

Parameter Setting

Parameter selection is one of the most essential parts of any algorithm. The parameters in PSO are taken according to the settings suggested by Clerc and Kennedy (2002). The detailed

settings are shown as follows: $c1 = c2 = 1.496$, $w = 0.7298$. Stopping criteria is taken as 30 (maximum iteration) in this experiment. Swarm (particles) size also affects performance of PSO because with few particles, it tends to confine with local maxima while the use of too many particles degrades (slows down) the algorithm. So, the authors took the swarm size as half of the total features used. While using NB, J48 and AdaBoost classifiers default parameters were considered. In SVM, radial basis function was taken because it gives value to each point based on its distance from the origin or a fixed centre. In the MLP structure, learning rate and momentum is taken as 0.3 and 0.2 respectively. The total number of epoch used for training is 500, validation threshold for testing is 20 and the number of features used in feature set is used as hidden neurons.

RESULTS AND DISCUSSION

For each feature set in WEBSPAM UK-2006, the authors ran CFS-PSO algorithm for the selection of optimal features. The reduction of features in link based feature set (C) was 88% (maximum) and 68% (minimum) for full content based feature set (B). Reduction for other feature sets is shown in Table 3. To test the performance of CFS-PSO (in terms of accuracy), five classification algorithms were applied: Naïve Bayes, J48, AdaBoost, SVM and MLP. After that results were compared using original number of features.

Table 3
Optimal selection of features after applying CFS+PSO

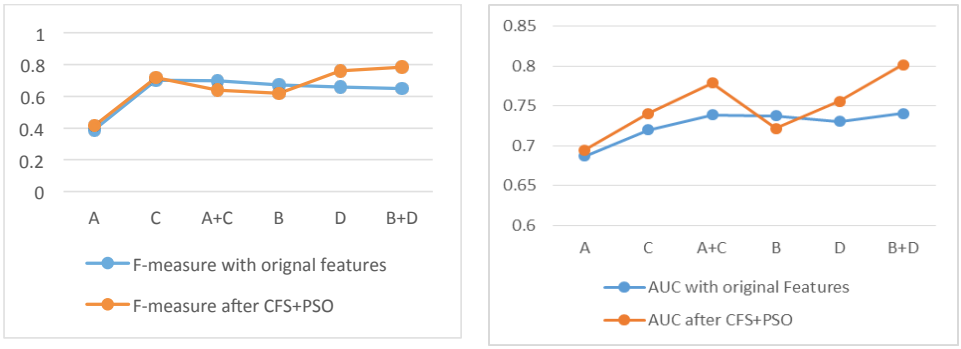
Label	Original features	Optimal features after CFS+PSO	Reduction in features
A	24	6	75%
C	41	5	88%
A+C	65	12	82%
B	96	31	68%
D	138	40	71%
B+D	234	59	75%

Evaluation of CFS+PSO for Accuracy Parameters

F_1 Score and AUC with existing (original) and optimum features (after applying CFS+PSO) for Naïve Bayes classifier is shown in Table 4. The maximum improvement of F1 Score (with optimal features) is found for feature set B+D (21.7%) followed by 15.53% in case of feature set D, 6.41% for feature set A and 2.29% for feature set C. Naïve Bayes showed improvement in AUC and also 8.23% in case of B+D feature set while 5.41%, 3.42%, 2.92% and 1.16% was seen for A+C, D, C and A feature sets respectively. But for feature set B, F measure and AUC (with optimal features) are decreased by 7.61% and 2.17% respectively (shown in Figure 6).

Table 4
F_1 score and AUC with existing (Original) and optimum features for Naïve Bayes classifier

Label	Naïve Bayes Classifier			
	F_1 score with existing features	AUC with existing features	F_1 score after CFS+PSO	AUC after CFS+PSO
A	0.39	0.687	0.415	0.695
C	0.7	0.72	0.716	0.741
A+C	0.699	0.739	0.64	0.779
B	0.67	0.738	0.619	0.722
D	0.657	0.731	0.759	0.756
B+D	0.647	0.741	0.784	0.802



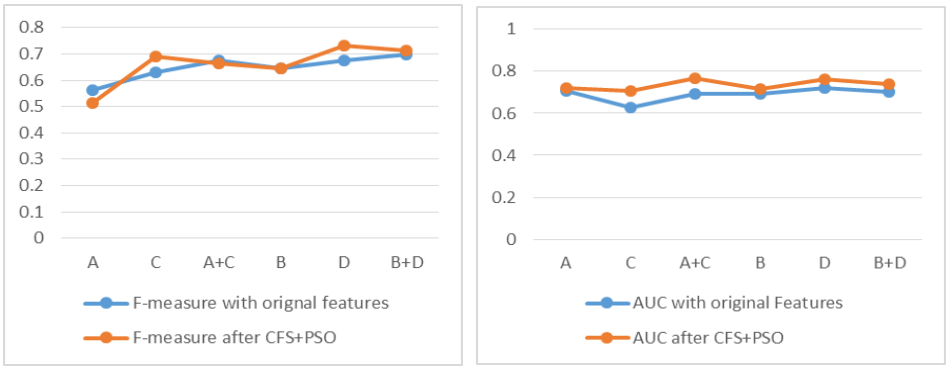
(a) (b)

Figure 6. (a) F_1 score (F-measure) for NB (b) AUC for NB

For J-48, AUC (with optimal features) is increased with maximum improvement of 12.60% in case of link based features, as shown in Table 5). F1 score is improved by 9.54% for link based, 8.43% for transformed link based and 2.30% for B+D feature sets while decreased for A+C and content based feature sets (Figure 7).

Table 5
F_1 score and AUC with existing and optimum features for J-48 classifier

Label	J48 Classifier			
	F_1 score with existing features	AUC with existing features	F_1 score after CFS+PSO	AUC after CFS+PSO
A	0.563	0.706	0.514	0.719
C	0.629	0.627	0.689	0.706
A+C	0.677	0.695	0.665	0.768
B	0.646	0.694	0.646	0.717
D	0.676	0.723	0.733	0.761
B+D	0.697	0.701	0.713	0.74



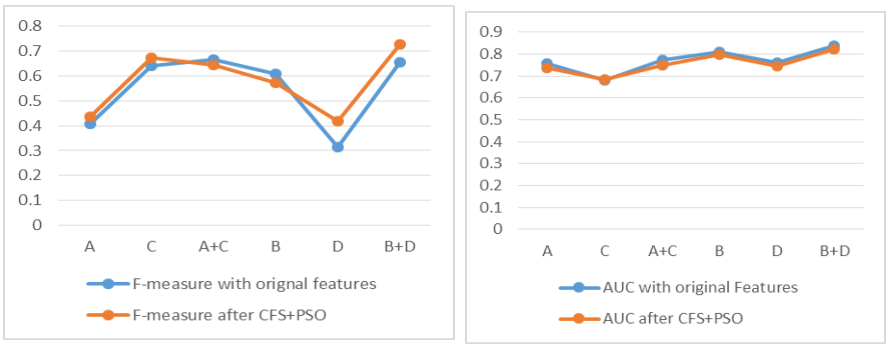
(a) (b)

Figure 7. (a) F_1 score (F-measure) for J48 (b) AUC for J48

With AdaBoost, F1 score increased by 33.02% for transformed link based feature (denoted by D) followed by minor enhancement of 4.67% for link based feature set while decreased by 3.44% and 6.23% for A+C and full content feature set respectively (Figure 8 and Table 6). However, there is no improvement in AUC by CFS+PSO (with exception of C feature set).

Table 6
F_1 Score and AUC with existing (Original) and optimum features for AdaBoost classifier

Label	AdaBoost Classifier			
	F_1 score with existing features	AUC with existing features	F_1 score after CFS+PSO	AUC after CFS+PSO
A	0.406	0.759	0.438	0.738
C	0.643	0.68	0.673	0.685
A+C	0.668	0.773	0.645	0.748
B	0.61	0.811	0.572	0.799
D	0.315	0.763	0.419	0.744
B+D	0.655	0.84	0.726	0.822



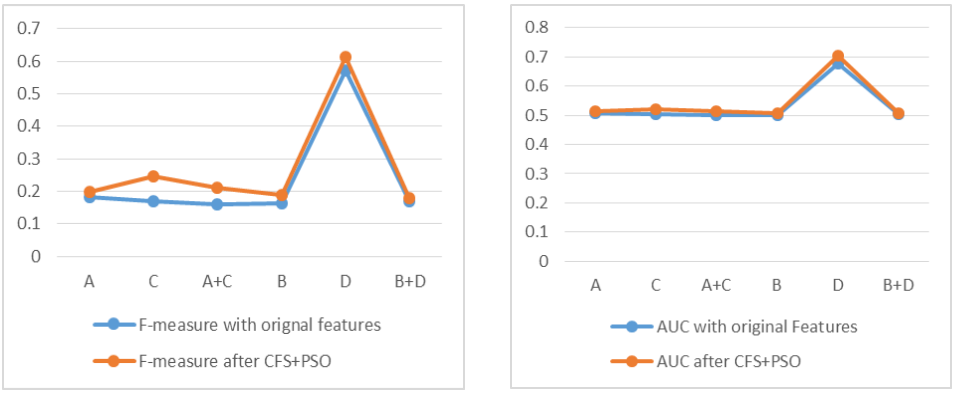
(a) (b)

Figure 8. (a) F_1 score (F-measure) for AdaBoost (b) AUC for AdaBoost

AUC and F1 Score for every feature set is increased for SVM with optimum features (Figure 9 and Table 7). The maximum improvement in AUC and F1 Score is 3.78% and 45.83% for C feature set respectively.

Table 7
F_1 score and AUC with existing (Original) and optimum features for SVM classifier

Label	SVM Classifier			
	F_1 score with existing features	AUC with existing features	F_1 score after CFS+PSO	AUC after CFS+PSO
A	0.182	0.506	0.199	0.514
C	0.168	0.503	0.245	0.522
A+C	0.159	0.5	0.21	0.515
B	0.163	0.501	0.189	0.509
D	0.573	0.678	0.613	0.703
B+D	0.169	0.504	0.18	0.507



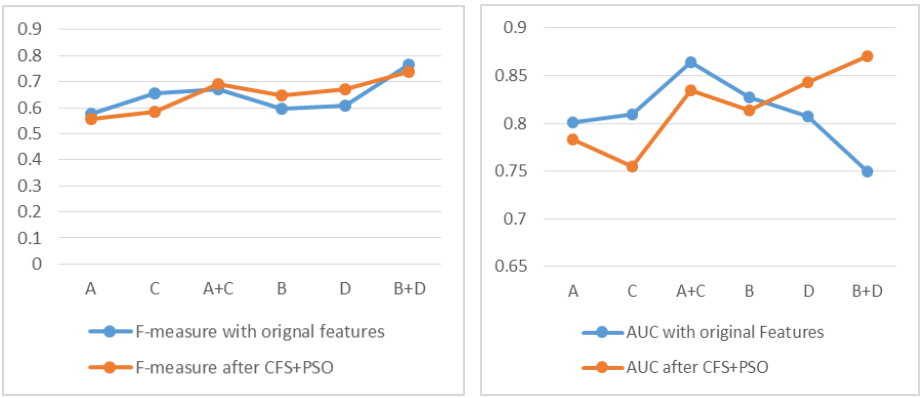
(a) (b)

Figure 9. (a) F_1 score (F-measure) for SVM (b) AUC for SVM

In case of MLP, the maximum enhancement of F1 Score is for A+C (2.38%), full content based (8.57%) and transformed link based feature set (10.38%) while there is a decrease for all other feature sets. AUC is increased by 16.13% and 4.46% only for B+D and D feature set.

Table 8
F_1 score and AUC with existing (Original) and optimum features for MLP classifier

Label	MLP Classifier			
	F_1 score with existing features	AUC with existing features	F_1 score after CFS+PSO	AUC after CFS+PSO
A	0.576	0.801	0.557	0.783
C	0.655	0.81	0.586	0.755
A+C	0.673	0.864	0.689	0.835
B	0.595	0.827	0.646	0.814
D	0.607	0.807	0.67	0.843
B+D	0.764	0.75	0.739	0.871



(a) (b)

Figure 10. (a) F_1 score (F-measure) for MLP (b) AUC for MLP

Analysis of Time Complexity

The complexity of the algorithm is reduced due to the removal of irrelevant and redundant features. Hence the computational time for every classifier is also reduced. But here the authors have only shown the analysis of SVM and MLP because they are computationally very expensive in comparison to other classifiers. From Table 9, it can be seen that time taken by classifier with optimal features (by applying CFS-PSO) is less than the time taken by the same classifier with original features for every feature set.

Table 9
Computational time (in seconds) to build a model with / without CFS-PSO for SVM and MLP classifier

Label	SVM classifier		MLP classifier	
	Computational time with existing features	Computational time with optimal features	Computational time with existing features	Computational time with optimal features
A	7.48	1.16	26.06	4.9
C	58.75	15.64	62.41	2.83
A+C	67.3	24.37	143.11	9.73
B	71.73	32.35	320.13	44.17
D	23.83	8.89	662.28	70.5
B+D	108.14	46.7	1864.53	138.78

CONCLUSION

In this paper, the authors integrated CFS and PSO and comparison was done on two accuracy measures (AUC and F_1 Score) using five different classifiers The maximum increase of AUC for B+D feature set is 16.13% and 8.23% in MLP and NB respectively but in case of J48 classifier it is improved by 12.60% for link based feature set. F_1 score is improved by 33.02% in transformed link based features using AdaBoost but AUC is decreased for all feature sets except link based features. In the current analysis, CFS+PSO technique is best suited for SVM as both accuracy parameters are increased for every feature set and computational cost is also very low. In future, other meta-heuristic algorithms would be considered and comparison of this technique with wrapper and hybrid approach can also be made.

ACKNOWLEDGEMENTS

This paper is an extension of the paper published in *The 6th International Conference on Smart Computing and Communications*, 7-8 December 2017. National Institute of Technology, Kurukshetra, India.

REFERENCES

Abdulla, W., & Kasabov, N. (2003). Reduced feature-set based parallel CHMM speech recognition systems. *Information Sciences, 156*(1-2), 21-38.

Becchetti, L., Castillo, C., Donato, D., Baeza-Yates, R., & Leonardi, S. (2008). Link analysis for Web spam detection. *ACM Transactions on the Web, 2*(1), 1-42.

Becchetti, L., Castillo, C., Donato, D., Leonardi, S., & Baeza-Yates, R. A. (2006). Link-based characterization and detection of web spam. In *Second International Workshop on Adversarial Information Retrieval on the Web* (pp. 1-8). Seattle, Washington, USA: ACM.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and Information Systems, 34*(3), 483-519.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123-140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32.

Caruana, R., & Freitag, D. (1994). Greedy attribute selection. In *Machine Learning Proceedings* (pp. 28-36). New Brunswick, NJ, USA: ACM.

Castillo, C., Donato, D., Gionis, A., Murdock, V., & Silvestri, F. (2007). Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 423-430). Amsterdam, Netherlands: ACM.

Chow, T., & Huang, D. (2005). Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. *IEEE Transactions on Neural Networks, 16*(1), 213-224.

Clerc, M., & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58-73.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(1-4), 131-156.

Filippone, M., Masulli, F., & Rovetta, S. (2006). Supervised classification and gene selection using simulated annealing. In *International Joint Conference on Neural Networks* (pp. 3566-3571). Vancouver, BC, Canada: IEEE

Gasca, E., Sánchez, J., & Alonso, R. (2006). Eliminating redundancy and irrelevance using a new MLP-based feature selection method. *Pattern Recognition, 39*(2), 313-315.

Goh, K. L., & Singh, A. K. (2015). Comprehensive literature review on machine learning structures for web spam classification. *Procedia Computer Science, 70*, 434-441.

Goh, K. L., Singh, A. K., & Lim, K. H. (2013). Multilayer perceptrons neural network based web spam detection application. In *IEEE China Summit and International Conference on Signal and Information Processing* (pp. 636-640). Beijing, China: IEEE.

Guan, S., Liu, J., & Qi, Y. (2004). An incremental approach to contribution-based feature selection. *Journal of Intelligent Systems, 13*(1), 15-42.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*(Mar), 1157-1182.

Gyongyi, Z., & Garcia-Molina, H. (2005). Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web* (pp. 1-9). Chiba, Japan: ACM.

Hall, M. A. (1999). *Correlation-based feature selection for machine learning.* (Unpublished doctoral dissertation). University of Waikato, Hamilton, New Zealand. Retrieved from https://www.cs.waikato.ac.nz/~mhall/thesis.pdf

Haykin, S., & Lippmann, R. (1994). Neural networks, a comprehensive foundation. *International Journal of Neural Systems, 5*(4), 363-364.

Hsu, C. N., Huang, H. J., & Dietrich, S. (2002). The ANNIGMA-wrapper approach to fast feature selection for neural nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 32*(2), 207-212.

Huang, J., Cai, Y., & Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters, 28*(13), 1825-1844.

Kabir, M., Shahjahan, M. & Murase, K. (2012). A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Applications, 39*(3), 3747-3763.

Kabir, M. M., Shahjahan, M., & Murase, K. (2009). An efficient feature selection using ant colony optimization algorithm. In *International Conference on Neural Information Processing* (pp. 242-252). Berlin, Heidelberg: Springer.

Kambhatla, N., & Leen, T. (1997). Dimension reduction by local principal component analysis. *Neural Computation, 9*(7), 1493-1516.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459-471.

Ke, L., Feng, Z., & Ren, Z. (2008). An efficient ant colony optimization approach to attribute reduction in rough set theory. *Pattern Recognition Letters, 29*(9), 1351-1357.

Li, Y., Li, T., & Liu, H. (2017). Recent advances in feature selection and its applications. *Knowledge and Information Systems, 53*(3), 551-577.

Li, Y., Luo, C., & Chung, S. M. (2008). Text clustering with feature selection by using statistical data. *IEEE Transactions on Knowledge and Data Engineering, 20*(5), 641-652.

Ntoulas, A., Najork, M., Manasse, M., & Fetterly, D. (2006). Detecting spam web pages through content analysis. In *Proceedings of the 15th International Conference on World Wide Web* (pp. 83-92). New York, USA: ACM.

Pudil, P., Novovičová, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters, 15*(11), 1119-1125.

Quinlan, J. R. (1993). *C4. 5: Programming for machine learning.* San Mateo, CA: Morgan Kaufmann Publishers.

Ramos, C. C., Souza, A. N., Chiachia, G., Falcão, A. X., & Papa, J. P. (2011). A novel algorithm for feature selection using harmony search and its application for non-technical losses detection. *Computers and Electrical Engineering, 37*(6), 886-894.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research, 3*(Mar), 1371-1382.

Senliol, B., Gulgezen, G., Yu, L., & Cataltepe, Z. (2008). Fast Correlation Based Filter (FCBF) with a different search strategy. In *23rd International Symposium on Computer and Information Sciences* (pp. 1-4). Istanbul, Turkey: IEEE

Shunmugapriya, P., & Kanmani, S. (2017). A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid). *Swarm and Evolutionary Computation, 36*, 27-36.

Singh, A. K., Kumar, R., & Leng, A. G. K. (2011). An experimental study on spam detection algorithms. In *Proceeding of IEEE Conference TENCON* (pp. 1382-1385). Bali, Indonesia: IEEE.

Singh, S., & Singh, A. K. (2018). Web-spam features selection using CFS-PSO. *Procedia Computer Science, 125*, 568-575.

Song, Q., Fong, S., & Tang, R. (2016). Self-adaptive wolf search algorithm. In *5th IIAI International Congress on Advanced Applied Informatics* (pp. 576-582). Kumamoto, Japan: IEEE.

Song, Q., Ni, J., & Wang, G. (2013). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering, 25*(1), 1-14.

Sotoca, J. M., & Pla, F. (2010). Supervised feature selection by clustering using conditional mutual information-based distances. *Pattern Recognition, 43*(6), 2068-2081.

Tang, R., Fong, S., Yang, X. S., & Deb, S. (2012). Wolf search algorithm with ephemeral memory. In *Seventh International Conference on Digital Information Management* (pp. 165-172). Macau, China: IEEE.

Uğuz, H. (2011). A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based Systems, 24*(7), 1024-1032.

Webber, C. G., Maria de Fátima, W., & Hepp, F. S. (2012). Testing phishing detection criteria and methods. In *Frontiers in Computer Education* (pp. 853-858). Berlin, Germany: Springer.

WEBSPAM-UK2006. (2006). *WEBSPAM-UK2006 (previous dataset)*. Retrieved January 10, 2018, from http://chato.cl/webspam/datasets/uk2006/.

Xue, B., Zhang, M., & Browne, W. N. (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics, 4*3(6), 1656-1671.

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. In *Feature extraction, construction and selection* (pp. 117-136). Boston, MA: Springer.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *International Symposium on Stochastic Algorithms* (pp. 169-178). Berlin, Germany: Springer.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization* (pp. 65-74). Berlin, Germany: Springer.

Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *Nature and Biologically Inspired Computing* (pp. 210-214). Coimbatore, India: IEEE.

Yang, X. S., & He, X. (2013). Firefly algorithm: Recent advances and applications. *International Journal of Swarm Intelligence, 1*(1), 36-50.

Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research, 5*(Oct), 1205-1224.