

Particle Swarm Optimization based Feature Selection

Neha
Department of CSE
GJUS & T, Hisar
India

Jyoti Vashishtha
Department of CSE
GJUS & T, Hisar
India

ABSTRACT

Feature Selection is a pre-processing step in knowledge discovery from data (KDD) which aims at retrieving relevant data from the database beforehand. It imparts quality to the results of data mining tasks by selecting optimal feature set from larger set of features. Various feature selection techniques have been proposed in past which, unfortunately, suffer from unavoidable problems such as high computational cost and getting stuck into the local optima. Evolutionary algorithms such as Particle Swarm Optimization (PSO) possess immense abilities to explore a large search space and rarely fall into local optima thus making them a nice choice for feature selection. In this paper, we have explored pros and cons of traditional and PSO based feature selection techniques and suggested some effective changes in existing approaches.

Keywords

Particle Swarm Optimization (PSO), Evolutionary Algorithm (EA), Feature Selection (FS).

1. INTRODUCTION

Classification is an important task of data mining in which a model is constructed from training set and this model is used to classify unseen data. Typically, a real world dataset comprising of irrelevant and redundant features create problem in classification and make classification a really challenging task [1], [2]. Feature Selection is capable of tackling this challenge by selecting relevant features and improving the accuracy of classification and other data mining tasks. Feature Selection is a necessary preprocessing technique in data that eliminates irrelevant, redundant and noisy data [3]. Due to easy availability of hardware, there is tremendous increase in amount of data and it is not easy to extract valuable information from such a huge data and hence the need of feature selection [4]. Feature selection is a five step process namely: (i) Initialization (ii) Subset discovery (iii) Subset evaluation (iv) Stopping Criteria and (v) Validation of result [5], [6]. First step is initialization of Feature Selection process with all features. Second step involves subset discovery that employs several search strategies like complete, sequential, random and heuristic search methods to generate candidate feature subsets. Third step is subset evaluation, which uses evaluation function to select better subset. Fourth step is stopping criteria, which is based on either generation procedure or selection of necessary features. Last step is validation of result in which different tests are carried out to validate the selected subsets. Feature Selection is a difficult task due to large search space and interaction among features [5], [7], [8]. A dataset with N features will have 2^N subsets. For large value of N, exhaustive search is impractical. Thus we have Wrapper, Filter and Embedded methods-three broad categories of Feature Selection as opposed to Brute Force method [5], [9], [10]. Wrapper Method applies learning algorithm while Filter method uses different measures like information distance to select relevant subset. Embedded Approach combines the

advantages of both wrapper and filter Approach. These traditional Feature Selection Methods, however, suffer from various problems such as high computational cost and getting stuck into the local optima. Therefore, global search techniques are required to do feature selection. Evolutionary computation techniques are well known for their global search ability and have been successfully applied for feature selection [11], [12], [13]. In comparison to other evolutionary algorithms, PSO is computationally less expensive and converge more quickly [12], [14]. It has offered promising solutions to feature selection problem [15], [16], [17].

Rest of the paper is organized as follows: In section 2 we have explored working of PSO and different variants of it. In Section 3 we have described feature selection and different techniques used for it. In Section 4 we have explored different PSO algorithms, used for feature selection. Section 5 discusses future scope and suggests possible modifications in existing approaches. Finally Section 6 concludes the paper.

2. PARTICLE SWARM OPTIMIZATION

Particle Swarm optimization is a nature inspired and a meta-heuristic technique that simulates the behavior of bird's flocking, proposed by Kennedy and Eberhart in year 1995 [18]. PSO algorithm uses a population of randomly generated particles having associated velocity and position, where each particle corresponds to randomly generated solution [19]. PSO searches for optimal solution by iteratively changing the velocity and position based on flying experience of own particle and of group toward gbest and pbest location in successive iteration. gbest corresponds to best fitness value of population that any particle has achieved while pbest corresponds to best fitness value of particle that it has achieved so far. Velocity and Position of particle is changed as per equations (i) & (ii) respectively:

$$v_{id} = v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id}) \dots \dots (i)$$
$$x_{id} = x_{id} + v_{id} \dots \dots (ii)$$

Equation (i) has three parts: 1) Momentum, 2) Cognitive and 3) Social [20]. *Momentum* part states that velocity of a particle can't be changed quickly. Each particle updates their velocity based on its previous velocity, and distance of current position from gbest and pbest location. *Cognitive part*, c_1 describes particle's learning from its own flying experience while *Social part*, c_2 represents particle's learning from flying experience of group. Stopping criteria of this algorithm is either good fitness value or maximum number of iterations. At the end, optimal or near optimal solution is obtained. Particle's velocity is limited to maximum velocity, V_{max} on each dimension. Low V_{max} value causes solution to get trapped in local optima where as high V_{max} value cause particle to fly past good solution. As V_{max} constraints the exploration of

PSO, a modification to above algorithm was suggested by Y. Shi *et al.* [21] to overcome this problem. A new parameter called inertia weight (w), was added to the original PSO algorithm to balance global and local search.

$$v_{id} = w * v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id}) \dots \dots \dots (iii)$$

$$x_{id} = x_{id} + v_{id} \dots \dots \dots (iv)$$

Larger and smaller value of w corresponds to global and local search respectively. Suitable value of w ensures balance between global and local exploration and exploitation. Next development came in the form of constriction factor (K) which ensures convergence of PSO algorithm [20]. With the introduction of constriction factor, updated equation becomes:

$$v_{id} = K * (v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id})) \dots \dots \dots (v)$$

$$x_{id} = x_{id} + v_{id} \dots \dots \dots (vi)$$

With
$$K = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}$$

Where,

$$\varphi = c_1 + c_2, \varphi < 4 \dots \dots \dots (vii)$$

Under the condition that value of inertia weight, w in equation (iii) set equal to K and values of c_1 and c_2 following condition (vii), then equation (v) and (iii) become mathematically equivalent which can be considered as a special case of PSO algorithm with inertia weight. PSO was initially designed to solve real value problem and further, extended to Discrete/binary space to solve discrete problem where velocity was squashed using logistic function [20], [22]. Also, other different PSO variants like Bare Bone PSO [23], Boolean BPSO [32], were proposed to solve the discrete problem. Sousa *et al.* [24] used both binary and continuous representation of (PSO) for classification. Three PSO variants namely Discrete Particle Swarm Optimizer (DPSO), Linear Decreasing Weight Particle Swarm Optimizer (LDWPSO) and Constricted Particle Swarm Optimizer (CPSO) have been compared with Genetic Algorithm (GA) and Tree Induction Algorithm (J48). Experimental results demonstrate that PSO is competitive with tree induction as well as other evolutionary algorithms.

3. FEATURE SELECTION

A number of Feature Selection methods have been used in past and these methods can be broadly categorized as: filter approach, wrapper approach and embedded approach. Filter Methods avoid using classification algorithm and also do not consider the effect of interaction among features [25], [38]. As opposed to filter approach, Wrapper approach uses learning algorithm to select feature subset and often achieves better result than filter approach [14]. Third, Embedded methods do feature selection by taking advantages of both wrapper and filter methods [26]. It considers feature selection as a part of classifier.

3.1 Wrapper Approach

Wrapper approach employs learning algorithm as a part of evaluation function and uses classifier accuracy as fitness

measure [6], [8]. Most of Wrapper Algorithms are categorized as: Exact methods, Greedy Sequential Subset Selection method, Partitioning methods, Mathematical Programming Methods and Meta-heuristic methods [27]. Commonly used greedy methods are Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) which perform feature subset selection sequentially [26], [27]. However, these two algorithms completely differ in their working. SFS starts with empty set of features and candidate features are sequentially added while SBS starts with all the possible features and candidate features are sequentially removed until there is no further improvement in classification performance. However, these algorithms suffer from nesting effect meaning that addition or removal of a feature cannot be undone. To overcome the nesting effect, combination of SFS and SBS, called Plus-1-take-away-r method is used [3]. This method integrates the features of both SFS and SBS by following r step of SBS to the l steps of SFS. However, selection of suitable values of l and r is also problematic since l and r are user-defined parameters. Floating search methods are the other possible solutions to the nesting problem [3], [27]. Sequential Forward Floating Selection (SFFS) and Sequential Backward Floating Selection (SBFS) are two such methods which resemble Plus-1-take-away-r method with dynamic control. They are capable of selecting or removing the features at different stages of feature selection procedure until desired numbers of features are obtained. However, these sequential floating techniques suffer from the problem of getting stuck in local optima.

3.2 Filter

Filter approach is independent of any learning algorithm. They provide a general view of feature space. Moreover, computational cost of filter approach is less as compared to wrapper approach. It uses several different measures like distance, information and dependency to select features [21], [23]. These evaluation measures are described as follow:

3.2.1 Distance Measures

This measure is based on separability of instances. Feature A is preferred than another feature B if A generate a greater difference among two-class conditional probabilities than B . A and B are indistinguishable if the difference is zero [26].

3.2.2. Information Measures

This measure uses information gain to select features. The information gain from a feature X is defined as the difference between the prior uncertainty and expected posterior uncertainty using X [26]. Feature with more information gain value is preferred than low information gain value.

3.2.3. Dependence Measures

This measure uses correlation of feature with class to select the features. A feature having highest correlation with class is preferred.

Relief is a filter approach based algorithm which uses feature relevance property to rank the features and a threshold to select feature subset [9], [26]. However, the selection of threshold value is a disadvantage of this algorithm. It considers only relevant features rather than redundant features. Another filter algorithm, FOCUS starts with all features and uses exhaustive approach to get optimal subset [28]. Being an exhaustive method, it suffers from high computational cost.

3.3 Embedded Approach

Embedded Approach combines the advantages of both wrapper and filter Approach [9]. Feature selection module is embedded inside the classifier and hence the name. Three different kinds of embedded methods are: (a) Pruning Methods (b) Models with build-in mechanism (c) Regularization models. Pruning Method initially trains a model by using all features and then removes features by setting value of their coefficient to 0. It uses SVM as classifier. Build in mechanism based embedded method uses ID3 and C4.5 as classifier. Third, Regularization Method has objective of minimizing fitting error and keeping coefficient values small. Features with coefficient value 0 are removed. This method performs better than the rest two.

4. FEATURE SELECTION USING PSO

The traditional feature selection techniques rarely provide satisfactory results for large dataset. By using them we obtain either optimal or computationally effective feature subset but not the both. Many evolutionary search techniques like GA, PSO etc. have been used in past for Feature Selection due to their global search ability. Few of the works discussing feature selection using PSO have been appraised in this subsection. Wang *et al.* [28] proposed feature selection method based on PSO and rough set theory as imprecision, uncertainty can be easily tackled by rough set theory. Experimental results show that PSO is a good choice for rough set based feature selection. But drawback of using standard rough set theory is that it consumes most of the running time. Therefore, probabilistic rough set theory with binary PSO (BPSO) has been used by Liam *et al.* to select optimal feature subset [29]. Two algorithms namely PSOPRS, PSOPRSN were introduced based on filter approach. PSOPRS uses probabilistic rough set theory in fitness function whereas PSOPRSN adds number of attributes in fitness function to get similar classification performance by reducing the number of attributes. The proposed algorithm performs better than PSORS which is based on BPSO and original rough set theory and two traditional dimension reduction algorithms (cfsF, cfsB). Two new algorithms (BPSO-P, BPSO-G) are developed by same author on the basis of BPSO and information theory to find optimal feature subset [30]. Fitness function in these algorithms considers both relevance and redundancy to select best feature subset. BPSO-P uses mutual information of pair of features as a measure while BPSO-G uses entropy of group of features to select feature subset. Experiment results show that both algorithms achieve better classification accuracy provided they use suitable weight. However, this algorithm has not been compared with other algorithms. BPSO can easily get stuck in local optimum solution after several generations. Therefore, several strategies have been adopted to improve the performance of BPSO. Chuang *et al.* [31] proposed an approach which is based on resetting value of the parameter gbest, if its value remains unchanged after several iterations. To evaluate particle's fitness value, KNN method along with LOOCV has been used using gene expression data. Experimental results illustrate that modified approach results in better classification accuracy. Yang *et al.* [32] proposed use of Boolean operator to reset the gbest parameter. If the gbest fitness remains identical after three successive generations then Boolean function is used to replace old gbest with new gbest fitness value. Proposed modifications lead to better results as compared to GA and original BPSO. Chuang *et al.* [33] embedded chaotic map

with BPSO (CBPSO) which adjusts the inertia weight in order to prevent BPSO from getting stuck in local optima. Chuang *et al.* [34] introduced Catfish effect to enhance the performance of BPSO. Those Particles having worst fitness in a number of consecutive iterations are replaced by new particles. Catfish-BPSO simplifies process of Feature selection and outperforms BPSO and deterministic algorithms. PSO has also been used in classification problems having one dimensional search space. Wang *et al.* [35] proposed real valued PSO for problems having one dimensional search space. Real valued PSO is used rather than BPSO because BPSO has demerit of getting trapped in local optima. An optimal feature subset is expected from real valued PSO because one dimensional search space is relatively less complicated. Multi-swarm PSO (MSPSO), proposed by Liu *et al.* [36], uses support vector with F-measure to get optimal feature subset. MSPSO is very effective in the sense that it outperforms other competing algorithms like standard PSO, grid search and genetic algorithm (GA). However MSPSO is computationally costlier because of complicated communication rules that it uses. Sahu *et al.* [37] associated clustering method with feature selection to make the selection process more effective. In such an approach, clustering of dataset is performed to obtain a feature subset which is later used as an input to an existing PSO algorithm to further refine the selection. Xue *et al.* [38] used PSO first time for multi-objective feature selection problem. Continuous PSO is opted for feature selection rather than binary PSO because in BPSO, particle's position is changed according to velocity. However, Standard PSO considers both velocity and current position when changes are made to particle's position. In this paper, two new methods NSPSOFS, CMDPSOFS are proposed for feature selection based on multi-objective PSO. NSPSOFS uses the concept of non-dominated sorting [39] and CMDPSOFS uses the concept of crowding, mutation and dominance [40] to find non-dominated solutions to get optimal feature subset. Experimental results show that. NSPSOFS produces similar or sometimes worse results as compared to evolutionary algorithms. However, CMDPSOFS outperforms all other method on classification performance.

Table 1: List of Appraised Paper

Feature Selection Approach	PSO Variant	Paper Appraised
Wrapper Approach	Continuous PSO	[1],[12],[13],[14],[15], [17]
	Binary PSO	[7],[8],[16],[22],[23],[27],[31],[32],[33],[34],[35]
Filter Approach	Continuous PSO	[10]
	Binary PSO	[28],[29],[30],[37]

Table 2: Brief Summary of Appraised Papers

PSO Variants	Proposed Algorithm	Approach used	Comparison with	Experiment Results
Continuous PSO	PSO+SVM [1]	PSO+ SVM, ten-fold cross validation	GA+SVM	High classification Accuracy rate, more appropriate parameter and subset
Binary PSO	PSOMP, PSOTFS [7]	PSO+ statistical clustering, KNN classifier with k=5	GFFS(Greedy forward feature selection algorithm)	Better classification performance, avoid premature convergence
Guassian PSO	GPSO [8]	Guassian distribution +BPSO+ Statistical clustering	LFS (Linear forward selection),GSBS (greedy stepwise backward selection),PSOFS (Standard PSO based feature selection)	Better classification performance in term of number of features and classification accuracy
Continuous PSO	SCTSOFS [12]	PSO+ Size controlled approach, KNN with k=1	PSOFS (Standard PSO based feature selection), SFS (Sequential Forward Selection), SBS (Sequential Backward Selection)	Less Computational Cost, better classification performance
Continuous PSO	PSO+SVM [13]	PSO+SVM, PSO for optimization purpose, SVM for verification	GA+SVM, Comparison among PSO with different weight carried out	Better Classification performance than GA+ SVM, Less classification error for PSO with global and local acceleration of 1.5 and threshold of 0.7 than GA
Binary PSO	BPSO-Er, BPSO-ErNo, BPSO-2Stage [16]	Two stage approach , first stage for classification performance optimization + second stage for minimization of features	LFS(linear forward Selection), GSBS (greedy stepwise backward selection)	High classification performance
Continuous PSO	PSOArR, PSOArRWS [17]	Archive to store gbest + selection mechanism of gbest in standard PSO	PSOFS, LFS, GSBS, CMDPSOFS	Improved classification performance
Probability based binary PSO	PBPOfs [22]	Flipping probability, new position updating equation in BPSO, 5NN	Standard BPSOFS, LFS, GSBS	Less computation time, better classification performance
Binary Bare bone PSO	Binary BPSO [23]	Reinforced memory strategy +uniform combination +1NN classifier	SGA (Simple GA algorithm), HGA(Hybrid GA algorithm), BPSO (binary PSO), CBPSO (chaotic map based binary PSO)	Best average classification accuracy
Binary PSO	Discrete PSO with Adaptive Feature Subset Selection [27]	Modified BPSO with extending social learning to update the velocity of the particles + logistic regression model	Tabu search and scatter search	Better classification accuracy and computational performance
Improved Binary PSO	IBPSO [31]	KNN+LOOCV, resetting value of the parameter gbest if its value remains unchanged after several iterations.	NON- SVM methods contain KNN, NN, and PNN. MC-SVM methods contain OVR, OVO, DAG, WW, CS.	Smaller feature subsets
Boolean Binary PSO	B-BPSO [32]	Boolean algebra operation in original BPSO, gbest value replaced by new value if its value remain identical during successive 3 generation	GA, Binary PSO	High classification accuracy with less features
Binary PSO	CBPSO [33]	Chaotic map include logistic	SFS(Sequential Forward	Especially, BPSO

		Map and tent Map with BPSO, adjustment in the inertia weight in order to prevent BPSO from getting stuck in local optima	Selection), PTA (plus-1 take-away-r), SFFS(Sequential forward floating selection),SGA(simple GA), HGA(Hybrid GA), BPSO(standard BPSO)	with tent map have high classification accuracy
Binary PSO	Catfish BPSO [34]	Catfish effect +BPSO, replace Particles with worst fitness in a number of consecutive iterations by new particles, KNN	SFS(Sequential Forward Selection), PTA (plus-1 take-away-r), SFFS(Sequential forward floating selection),SGA(simple GA), HGA(Hybrid GA) BPSO(standard BPSO)	Outperform all methods
Multi Swarm PSO	IFS [36]	IFS approach that combines parameter optimization and feature selection	GA+SVM, PSO+SVM	Increased classification accuracy rate
Multi-objective PSO	NPSOFS, CMDPSOFS [38]	PSO + crowding, mutation, ϵ - dominance, 10 fold cross validation, KNN with k=5	LFS (Linear Forward Selection), GSBS(Greedy stepwise backward selection), ErFS(Commonly used PSO algorithm), 2SFS (PSO with a Two-Stage Fitness Functions), NSGA-II, SPEA2(Strength Pareto evolutionary algorithm 2), PAES(Pareto archived evolutionary strategy)	Better classification performance using fewer features and less computational time

Table 3, gives a comparison of classification accuracies of feature selection using traditional and different PSO methods for vehicle and sonar datasets. Test problems are classified as large scale problems having number of feature greater than 50 and medium scale problems having number of features between 20 and 49. Vehicle dataset consists of 18 features and 846 instances and come under test group for medium scale problem while Sonar dataset has 60 features and 208 instances which are defined for large scale problems.

Table 3: Classification Accuracy Comparison of traditional FS and PSO based FS Methods

Algorithm	Classification Accuracy	
	Vehicle Dataset	Sonar Dataset
SFS	68.20	89.90
SFFS	70.80	93.75
PBPSOfs	84.84	87.30
Binary BPSO	75.54	96.08
CBPSO	75.06	95.57
Catfish BPSO	75.06	96.92
Binary PSO	74.70	94.47
MSPSO	89.66	93.71

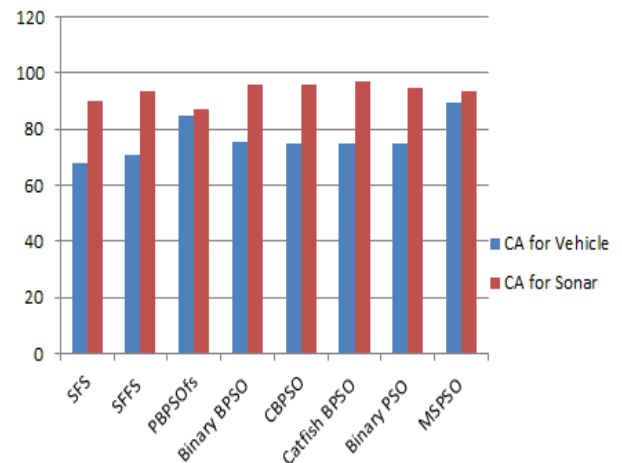


Fig 1: Distribution of CA using Traditional and PSO based FS methods

Figure 1 shows the classification accuracies on vehicle and sonar datasets using SFS [33], SFFS [33], PBPSOfs [22], Binary BPSO [23], CBPSO [33], Catfish BPSO [34], Binary PSO [34], MSPSO [36]. It is evident from the figure that feature selection using particle swarm optimization gives better classification accuracy as compare to traditional FS methods (SFS, SFFS). Hence, Particle swarm optimization based feature selection methods are more effective.

5. FUTURE SCOPE

PSO has been widely used for feature selection to improve classification performance. Further, a lot of work is being done using multi-objective PSO for feature selection to improve classification performance and to reduce number of features selected as well. Most of the existing multi-objective feature selection based on PSO algorithms use binary tournament selection to select gbest and uniform and non-uniform mutation. There is a scope to further reduce search

space for better classification accuracy if enhanced selection and mutation procedures are being used.

6. CONCLUSION

In this paper, traditional feature selection methods and evolutionary algorithms such as PSO for feature selection has been investigated. Several PSO variants are available in research literature using wrapper and filter approach to make the feature selection process more effective. After thorough exploration, it has been concluded that PSO based algorithms are quite efficient for selecting optimal feature subset. Existing techniques, however, can be further modified to achieve better results.

7. ACKNOWLEDGMENTS

Authors would like to thank Amarnath Pathak for his huge support and suggestions.

8. REFERENCES

- [1] Lin, S. W., Ying, K. C., Chen, S. C. and Lee, Z. J. 2008. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications*, Vol. 35, No. 4, pp. 1817-1824.
- [2] Vashishtha, J., Kumar, D., Ratnoo, S. and Kapila. 2011. Mining comprehensible and interesting rules: a genetic algorithm approach. *International Journal of Computer Applications*, Vol. 31, No. 1, pp.39-47.
- [3] Yusta, S. C. 2009. Different meta-heuristic strategies to solve the feature selection problem. *Pattern Recognition Letters*. Vol. 30, No. 5, pp. 525-534.
- [4] Han, J., Kamber, M. and Pei, J. 2011. *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann.
- [5] Tran, B., Xue, B. and Zhang, M. 2014. Overview of Particle Swarm Optimization for Feature Selection in Classification. In *Proceeding of 10th International Conference on Simulated Evolution and Learning*, Springer, Dunedin, New Zealand, pp. 605-617.
- [6] Visalakshi, S. and Radha, V. 2015. Wrapper based Feature Selection and Classification for Real Time Dataset. *International Journal of Emerging Technologies in Computational and Applied Sciences*, Vol. 3, pp. 306-311.
- [7] Lane, M. C., Xue, B., Liu, I. and Zhang, M. 2013. Particle Swarm Optimization and Statistical Clustering for Feature Selection. *Advances in Artificial Intelligence, Lecture Notes in Computer Science* Heidelberg, Springer, Vol. 8272, pp. 214-220.
- [8] Lane, M. C., Xue, B., Liu, I. and Zhang, M. 2014. Gaussian Based Particle Swarm Optimization and Statistical Clustering for Feature Selection. *Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science*, Heidelberg, Springer, Vol. 8600, pp. 133-14.
- [9] Chandrashekar, G. and Sahin, F. 2014. A survey on feature selection methods. *Computers & Electrical Engineering*, Elsevier. Vol. 40, No. 1, pp. 16-28.
- [10] Guan, J., Han, F. and Yang, S. 2013. A new gene selection method for microarray data based on PSO and informativeness metric. *Intelligent Computing Theories and Technology*, Lecture Notes in Computer Science, Heidelberg, Springer, Vol. 7996, pp. 145-154.
- [11] Pathak, A. and Vashishtha, J. 2015. Classification Rule and Exception Mining Using Nature Inspired Algorithms. *International Journal of Computer Science and Information Technologies*. Vol. 6, No. 3, pp. 3023-3030.
- [12] Yeoman, T. B., Xue, B. and Zhang, M. 2015. Particle Swarm Optimization for Feature Selection: A Size-Controlled Approach. In *Proceedings of the 13th Australasian Data Mining Conferences on Research and Practice in Information Technology*, Sydney, Australia Vol. 168.
- [13] Azevedo, G. L., Cavalcanti, G.D. and E CB Filho, C. 2007. An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting. In *Proceedings of 2007 Congress on Evolutionary Computation*, Singapore, IEEE, pp. 3577-3584.
- [14] Xue, B., Zhang, M. and Browne, W. N. 2012. Multi-objective Particle Swarm Optimization for Feature Selection. In *Proceedings of 14th Annual Conference on Genetic and Evolutionary Computation*, ACM, Philadelphia, Pennsylvania, USA, pp. 81-88.
- [15] Mohemmed, A. W., Zhang, M. and Johnston, M. 2009. Particle Swarm Optimization based Adaboost for face detection. In *Proceedings of 2009 Congress on Evolutionary Computation*, Trondheim, IEEE, pp. 2494-2501.
- [16] Xue, B., Zhang, M. and Browne, W. N. 2012. New fitness functions in binary particle swarm optimization for feature selection. In *Proceedings of 2012 Congress on Evolutionary Computation*, Brisbane, QLD, IEEE, pp. 1-8.
- [17] Xue, B., Qin, A.K. and Zhang, M. 2014. An archive based particle swarm optimization for feature selection in classification. In *Proceedings of 2014 Congress on Evolutionary Computation*, Beijing, IEEE, pp. 3119-3126.
- [18] Kennedy, J. and Eberhart, R. 1995. Particle swarm optimization. In *Proceeding of International Conference on Neural Networks*. IEEE, Vol. 4, pp. 1942-1948.
- [19] Eberhart, R. and Shi Y. 2007. *Computational Intelligence*, 1st Edition, Morgan Kaufmann..
- [20] Shi, Y. 2004. Particle Swarm Optimization. In *Proceeding of International Conference on Neural Networks Society IV*, IEEE, No. 1, pp. 8-13.
- [21] Shi, Y. and Eberhart, R. 1998. A Modified Particle Swarm Optimizer. In *Proceeding of International Conference on Evolutionary Computation*, Anchorage, AK, IEEE, pp. 69-73.
- [22] Xue, B., Nguyen, S. and Zhang, M. 2014. A New Binary Particle Swarm Optimization Algorithm for Feature Selection. In *Proceedings of 17th European Conference on Applications of Evolutionary Computation*, Granada, Spain, Springer, pp. 501-513.
- [23] Zhang, Y., Gong, D., Hu, Y. and Zhang, W. 2015. Feature selection algorithm based on bare bones particle swarm optimization. *Neurocomputing*, Elsevier, Vol. 148, pp. 150-157.

- [24] Sousa, T., Silva, A. and Neves A. 2004. Particle Swarm based Data Mining Algorithms for classification tasks. *Parallel Computing*, Elsevier, Vol. 30, No. 5–6, pp. 767–783.
- [25] Xue, B., Zhang, M. and Browne, W.N. 2014. Particle swarm optimization for feature selection in classification: Novel initialization and updating mechanisms. *Applied Soft Computing*, Elsevier, Vol. 18, pp. 261–276.
- [26] Dash, M. and Liu, H. 1997. Feature selection for classification. *Intelligent data analysis*, Elsevier, Vol. 1, No. 3, pp. 131–156.
- [27] Unler, A. and Murat, A. 2010. A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research*, Elsevier, Vol. 206, No. 3, pp. 528–539.
- [28] Wang, X., Yang, J., Teng, X., Xia, W. and Jensen, R. 2007. Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters*, Elsevier, Vol. 28, No. 4, pp. 459–471.
- [29] Cervante, L., Xue, B., Zhang, M. and Shang, L. 2012. A Dimension Reduction Approach to Classification Based on Particle Swarm Optimization and Rough Set Theory. In *Proceedings of 25th Australasian Joint Conference on Artificial Intelligence*, Sydney, Springer, pp. 313-325.
- [30] Cervante, L., Xue, B., Zhang, M. and Shang, L. 2012. Binary particle swarm optimization for feature selection: A filter based approach. In *Proceedings of 2012 Congress on Evolutionary Computation*, Brisbane, QLD, IEEE, pp. 1–8.
- [31] Chuang, L.Y., Chang, H.W., Tu, C.J. and Yang, C.H. 2008. Improved Binary PSO for Feature Selection Using Gene Expression Data. *Computational Biology and Chemistry*, Elsevier, Vol. 32, No. 1, pp. 29–38.
- [32] Yang, C.S., Chuang, L.Y., Ke, C.H. and Yang, C.H. 2008. Boolean binary particle swarm optimization for feature selection. In *Proceedings of 2008 Congress on Evolutionary Computation*, Hong Kong, IEEE, pp. 2093–2098.
- [33] Yang, C.S., Chuang, L.Y., Li J.C., and Yang, C. H. 2008. Chaotic maps in binary particle swarm optimization for feature selection. In *Proceedings of Conference on Soft Computing in Industrial Applications*, Muroran, IEEE, pp. 107–112.
- [34] Chuang, L.Y., Tsai, S.W. and Yang, C.H. 2011. Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Systems with Applications*, Elsevier, Vol. 38, No. 10, pp. 12699–12707.
- [35] Wang, J., Zhao, Y. and Liu, P. 2010. Effective feature selection with Particle Swarm Optimization based one-dimension searching. In *3rd International Symposium on Systems and Control in Aeronautics and Astronautics*, IEEE, pp. 702–705.
- [36] Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X. and Wang, S. 2011. An Improved Particle Swarm Optimization for Feature Selection. *Journal of Bionic Engineering*, Science Direct, Vol. 8, No. 2, pp. 191–200.
- [37] Sahu, B. and Mishra, D. 2012. A Novel Feature Selection Algorithm using Particle Swarm Optimization For Cancer Microarray Data. *Procedia Engineering*, Elsevier, Vol. 38, pp. 27–31.
- [38] Xue, B., Zhang, M. and Browne, W.N. 2013. Particle swarm optimization for feature selection in classification: A multi-objective approach. *Cybernetics*, IEEE, Vol. 43, No. 6, pp. 1656-1671.
- [39] Deb, K., Pratap, A., Agarwal, S. and Meyerivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation*, IEEE, Vol. 6, No. 2, pp. 182–197.
- [40] Sierra, M. R. and Coello, C. A. C. 2005. Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ϵ -Dominance. *Lecture notes in computer science*, Guanajuato, Mexico, Springer, pp. 505–519.