EMITTER International Journal of Engineering Technology
Vol.2, No.2, December 2014
ISSN: 2443-1168

Dimensionality Reduction Algorithms on High Dimensional Datasets

Iwan Syarif

Politeknik Elektronika Negeri Surabaya
Jl. Raya ITS, Sukolilo, Surabaya 60111, Telp:+62-31-5947280, Fax:+62-31-5946114
E-mail: iwanarif@pens.ac.id

Abstract

Classification problem especially for high dimensional datasets have attracted many researchers in order to find efficient approaches to address them. However, the classification problem has become very complicatedespecially when the number of possible different combinations of variables is so high. In this research, we evaluate the performance of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) as feature selection algorithms when applied to high dimensional datasets.Our experiments show that in terms of dimensionality reduction, PSO is much better than GA. PSO has successfully reduced the number of attributes of 8 datasets to 13.47% on average while GA is only 31.36% on average. In terms of classification performance, GA is slightly better than PSO. GA-reduced datasets have better performance than their original ones on 5 of 8 datasets while PSO is only 3 of 8 datasets.

Keywords: feature selection, dimensionality reduction, Genetic Algorithm (GA), Particle Swarm Optmization (PSO).

1. DIMENSIONALITY REDUCTION

In the various applications of machine learning and data mining, the use of high dimensional datasets with hundreds of thousands of features is not unusual[1]. In other words, modern data sets are very often in high dimensional space. Extracting knowledge from huge data requires new approaches. The more complex the datasets, the higher the computation time and the harder they are to be interpreted and analysed. Therefore, classification on high dimensional data has become a recurring problem; since it occurs in various data mining applications for which a decision step is necessary.

The impact of high dimensionality on classification is poorly understood[2]. Many datasets such as microarray, DNA, proteomics, etc. have thousands or more features while the sample size (number of instances) is typically tens or less than hundred. Most of basic classifiers break down when the dimensionality is high. Miller reported that there is a well-known phenomenon that a prediction model built from thousands of attributes *(m)*but has a relatively small sample size *N)* can be quite unstable[3].

The above problem reveals the importance of dimensionality reduction on high dimension data classification. Dimensionality reduction is a process for reducing the number of random variables under consideration. There are some advantages of dimensionality reduction[4]:

- Most machine learning and data mining techniques may not be effective for high-dimensional data
- Query accuracy and efficiency degrade rapidly as the dimension increases
- Lower computational cost
- Help avoid over-fitting (training on highly-related features rather than contingent ones)

There are a lot of dimensionality reduction techniques but they can be divided into two categories: feature selection and feature extraction which explained in the following section.

2.1. FEATURE EXTRACTION

In feature extraction, all available variables are used and the data is transformed using a linear transformation to a reduced dimension space. Its main goal is to replace the original variables by a smaller set of underlying variables [5]. Principal Component Analysis (PCA)is one of the most widely used feature extraction techniques for data analysis and compression. PCA can be used to reduce the dimensionality of a data set by finding new variables which are smaller than the original but still retains most of the original data set information [4][6].PCA derives new variables that are linear combinations of the original variables by finding a few orthogonal linear combinations of the original variables with the largest variance [7]. The new variables, called principal components (PCs), are uncorrelated and are in decreasing order of importance. So, the goal of PCA is to find a set of directions that maximizes the variances of the original data.

Another variant of PCA is Independent Component Analysis (ICA) which is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. Draper et. al. [8] have compared the performance of PCA and ICA in the face/image recognition problems. They reported that ICA outperforms PCA on visible light image, but on the other hand PCA outperforms ICA in another different type of images.

2.2. FEATURE SELECTION

Feature selection is a popular technique used to find the most important and optimal subset of features for building powerful learning models. An efficient feature selection method can eliminate irrelevant and redundant data; hence it can improve the classification accuracy [9][10][11].

Approaches for feature selection can be categorized into two models, namely a filter model and a wrapper model. The wrapper model (Kohavi& John, 1997) applies the classifier accuracy rate as the performance measure. Some researchers have concluded that if the purpose of the model is to minimize the classifier error rate, and the measurement cost for all the features is equal, then the classifier's predictive accuracy is the most important factor.

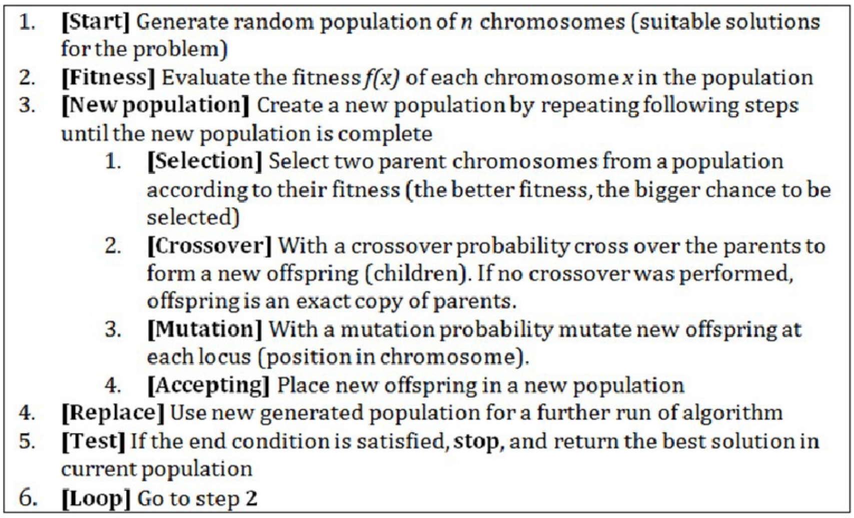
There are a lot of feature selection techniques, but in this paper we only select two algorithms: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). We decided to use GA and PSO in this research because actually feature selection is a kind of optimisation problems where GA and PSO have been proven and widely used by many researchers [9][10][12][13][14].

2.2.1. Genetic Algorithms

The Genetic Algorithm (GA) has been applied to many function optimization problems and has been shown to be good in finding optimal and near optimal solutions. The GA uses three main types of rules at each step to create the next generation from the current population:

- *Selection rules* select the individuals, called *parents*, that contribute to the population at the next generation.
- *Crossover rules* combine two parents to form children for the next generation.
- *Mutation rules* apply random changes to individual parents to form children.

The basic GA algorithm is shown in Figure 1 below.



```
1. [Start] Generate random population of $n$ chromosomes (suitable solutions for the problem)
2. [Fitness] Evaluate the fitness $f(x)$ of each chromosome $x$ in the population
3. [New population] Create a new population by repeating following steps until the new population is complete
    1. [Selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    2. [Crossover] With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
    3. [Mutation] With a mutation probability mutate new offspring at each locus (position in chromosome).
    4. [Accepting] Place new offspring in a new population
4. [Replace] Use new generated population for a further run of algorithm
5. [Test] If the end condition is satisfied, stop, and return the best solution in current population
6. [Loop] Go to step 2
```

Figure 1.GA pseudo-code[1]

[1] http://www.obitko.com/tutorials/genetic-algorithms/ga-basic-description.php

GA can be applied in feature selection because this problem has an exponential search space. The detail information of feature selection using GA is explained in [9].

2.2.2. Particle Swarm Optimizations

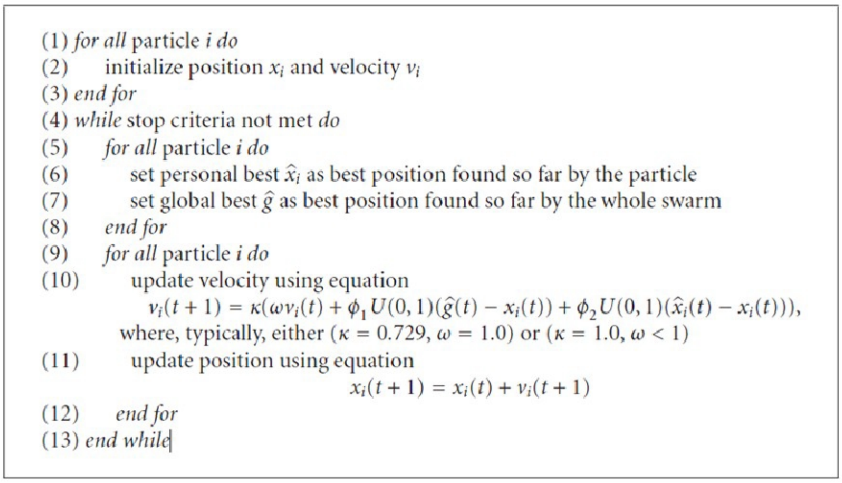
Particle swarm optimization (PSO) is an evolutionary algorithm technique that was first developed by Kennedy and Eberhart (1995) and is inspired by the behaviour of bird flocking to reach destination not completely known. PSO is powerful, easy to implement and computationally efficient [15]. PSO is also an effective and flexible technique to explore the search space of a problem[16]. Like other evolutionary algorithms, PSO performs searches using a population (called swarm) of individuals (called particles) that are updated from iteration to iteration[10]. To discover the optimal solution, each particle changes its searching direction according to two factors, its own best previous experience (called personal best or pbest) and the best experience of the whole swarms (called global best or gbest). The local best of a particle can be considered as the cognitive part while the global best particle is considered as the social part [16][10][17].

Each particle in the swarm represents one possible solution to the problem. At first, the swarm of particles are given a random initial location and velocity and are updated based on these following equations:

$$v_{i,j}^{t+1} = \omega v_{i,j}^{t} + c_1 r_1 (p_{i,j} - x_{i,j}^{t}) + c_2 r_2 (p_{g,j} - x_{i,j}^{t}) \quad (1)$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \quad (2)$$

Where *x* is the position of the particle *i,v* is its velocity, *j* is the dimension, *t* is time and ω is the inertial weight which represents how much of the previous velocity is retained while exploring. C_1 and c_2 are learning factor, r_1 and r_2 are weighting parameters, $p_{i,j}$ is local best while $p_{g,j}$ is global best particle. For each iteration the fitness of each particle is calculated, the personal best and global best are also updated using Equation 1 and Equation 2. Once the termination criteria being achieved, PSO will have good fitness, a set number of generations or a convergence factor such as a threshold for minimum population change. The PSO algorithm is described more details in the Figure 2.



```
for all particle i do
    initialize position $x_i$ and velocity $v_i$
end for
while stop criteria not met do
    for all particle i do
        set personal best $\hat{x}_i$ as best position found so far by the particle
        set global best $\hat{g}$ as best position found so far by the whole swarm
    end for
    for all particle i do
        update velocity using equation
            $v_i(t+1) = \kappa(\omega v_i(t) + \phi_1 U(0,1)(\hat{g}(t) - x_i(t)) + \phi_2 U(0,1)(\hat{x}_i(t) - x_i(t)))$,
        where, typically, either $(\kappa = 0.729, \omega = 1.0)$ or $(\kappa = 1.0, \omega < 1)$
        update position using equation
            $x_i(t+1) = x_i(t) + v_i(t+1)$
    end for
end while
```

Figure 2.PSO pseudo-code[18]

3. SYSTEM DESIGN

Our dimensionality reduction module is shown in Figure 3below. Various high dimensional dataset are processed by four different algorithms to get the most important features.

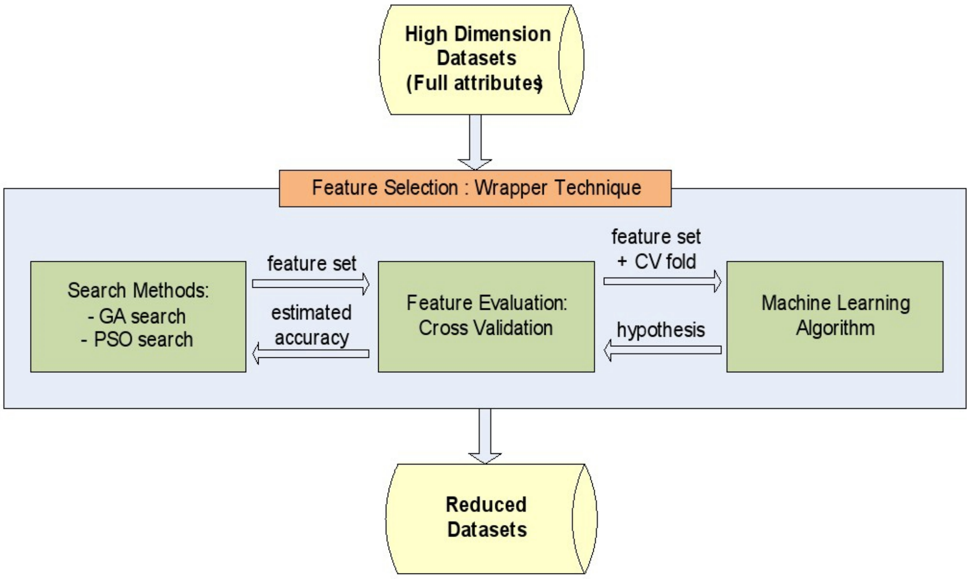

High Dimension Datasets (Full attributes)
Feature Selection : Wrapper Technique
Search Methods:
- GA search
- PSO search
feature set
estimated accuracy
Feature Evaluation: Cross Validation
feature set + CV fold
hypothesis
Machine Learning Algorithm
Reduced Datasets

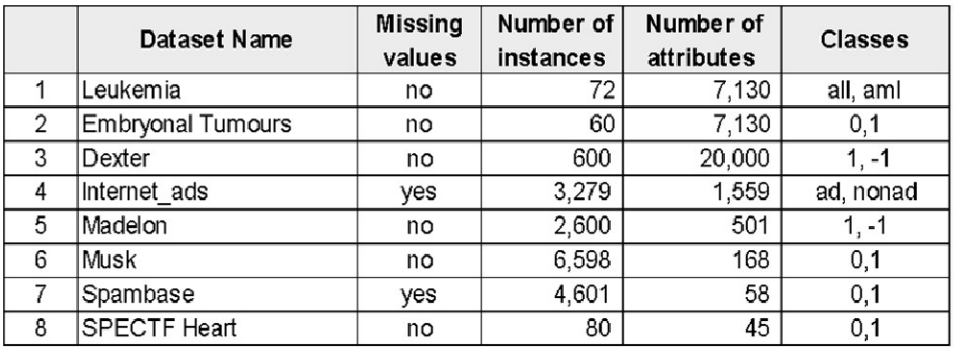
Figure 3. Feature Selection using GA and PSO

We used four basic machine learning algorithms: naive Bayes (NB), k Nearest Neighbour (k-NN), decision tree (DT) and rule induction (RI) to do classification on 8 original datasets, GA-reduced datasets and PSO-reduced datasets.

3.1. DATASETS

Weselected 8 high dimensional datasets which publicly available on UCI Machine Learning repository[2] as shown in Table 1 below.

Table 1. Highdimensional datasets



	Dataset Name	Missing values	Number of instances	Number of attributes	Classes
1	Leukemia	no	72	7,130	all, aml
2	Embryonal Tumours	no	60	7,130	0,1
3	Dexter	no	600	20,000	1, -1
4	Internet_ads	yes	3,279	1,559	ad, nonad
5	Madelon	no	2,600	501	1, -1
6	Musk	no	6,598	168	0,1
7	Spambase	yes	4,601	58	0,1
8	SPECTF Heart	no	80	45	0,1

From 8 datasets, there are 2 datasets (internet_ads and spambase) have missing values and there are 2 datasets have unbalanced data (internet_ads and musk).

3.2. PERFORMANCE MEASUREMENT

The metric used to evaluate the performance of classifier is given below.

Table 2. Performance metric[19]

		Predicted Label	
		Positive	**Negative**
Actual Label	**Positive**	True Positive (TP)	False Negative FN)
	Negative	False Positive (FP)	True Negative (TN)

The accuracy rate and false positive rate are measured using the Equation 3 and Equation 4 below.

$$TruePositiveRate = \frac{TP}{TP + FN} \qquad (3)$$

$$FalsePositiveRate = \frac{FP}{FP + TN} \qquad (4)$$

Many researchers use accuracy and false positive rate as performance measurement for classification problems, but other researchers [19][20][21]argue that accuracy and false positive rates are not enough and simply using accuracy results can be misleading. They suggest accuracy, precision, recall and ROC curve as better performance measurement methods.

[2]https://archive.ics.uci.edu/ml/datasets.html

Precision is the percentage of positive predictions that are correct. Recall or sensitivity is the percentage of positive labelled instances that were predicted as positive. Specificity is the percentage of negative labelled instances that were predicted as negative. Accuracy is the percentage of correctly classified instances over the total number of instances.

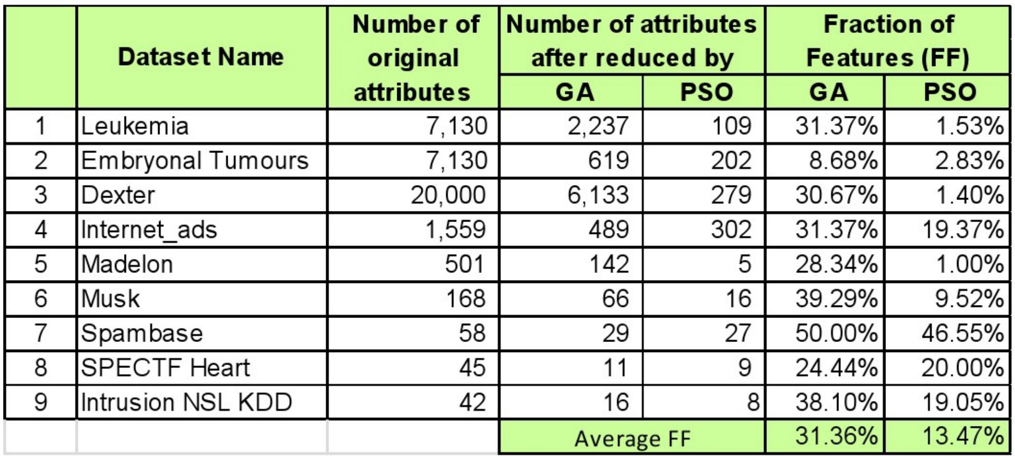
Table Error! No text of specified style in document..Classification performance measurement

Measurement	Formula
Precision	$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$
Recall /Sensitivity	$Recall/Sensitivity = \frac{\text{TP}}{\text{TP} + \text{FN}}$
Selectivity	$Selectivity = \frac{\text{TN}}{\text{FP} + \text{TN}}$
Accuracy	$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$
F-Measure	$F - Measure = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

4. EXPERIMENTAL RESULTS

We applied both GA and PSO to eight different datasets and the results are shown in Table 4.

Table 4. Feature Selection using GA and PSO



	Dataset Name	Number of original attributes	Number of attributes after reduced by		Fraction of Features (FF)	
			GA	PSO	GA	PSO
1	Leukemia	7,130	2,237	109	31.37%	1.53%
2	Embryonal Tumours	7,130	619	202	8.68%	2.83%
3	Dexter	20,000	6,133	279	30.67%	1.40%
4	Internet_ads	1,559	489	302	31.37%	19.37%
5	Madelon	501	142	5	28.34%	1.00%
6	Musk	168	66	16	39.29%	9.52%
7	Spambase	58	29	27	50.00%	46.55%
8	SPECTF Heart	45	11	9	24.44%	20.00%
9	Intrusion NSL KDD	42	16	8	38.10%	19.05%
			Average FF		31.36%	13.47%

Table 4shows that PSO reduced the number of attributes much better than GA in all datasets. The average fraction of feature (FF) of GA is 31.36% while PSO is 13.47%. In 4 of 8 datasets, PSO has successfully reduced the number of attributes to less than 5% of their original attributes (embryonal tumours 2.83%, leukemia 1.53%, dexter 1.40% and madelon 1.00%). However, fraction of features (FF) is not the only performance indicator of feature selection algorithms. FF is useless if the selected subsets have less accuracy than the original ones. Therefore, we need to find dimensionality algorithms which can reduce the number of attributes while in the same time maintain or improve the accuracy.

In the first experiment, we apply four basic classifiers: naïve Bayes (NB), k-Nearest Neighbour (k-NN), decision tree (DT) and rule induction (RI) to the original datasets and the results are shown below.

Table 5. Classification performance of original datasets

Data set	Original dataset	NB	k-NN	DT	RI
	#attributes	F-measure	F-measure	F-measure	F-measure
Leukemia	7,130	98.31%	89.70%	78.73%	83.40%
Embryonal Tumours	7,130	74.41%	67.06%	58.68%	74.05%
Dexter	20,000	81.39%	86.63%	86.79%	83.23%
Internet_ads	1,559	98.20%	91.45%	86.18%	95.00%
Madelon	501	59.05%	64.90%	64.29%	73.32%
Musk	168	93.67%	97.15%	92.57%	95.16%
Spambase	58	82.90%	85.62%	92.59%	93.05%
SPECTF Heart	45	79.49%	67.53%	79.69%	61.90%

In Table 5.we can see that k-NN achieves the best result only on 1 dataset (Musk) with F-measure=97.15% while naive Bayes (NB) gives the best accuracy on three datasets (leukemia, embryonaltumours and internet_ads). Decision Tree(DT)achieves the bestresults in 2 datasets (dexter and SPECTF heart) and rule induction (RI) is the best on madelon and spambase.

In the second experiment, we applied four basic classifiers to the eight datasets which have been reduced by GA and the results are shown in Table 6.

Table 6. Classification performance of GA-reduced datasets

Data set	Reduced by GA	NB	k-NN	DT	RI
	#attributes	F-measure	F-measure	F-measure	F-measure
Leukemia	2,237	98.31%	78.55%	81.16%	82.45%
Embryonal Tumours	619	65.42%	78.82%	58.58%	81.00%
Dexter	6,133	73.30%	60.04%	87.16%	81.84%
Internet_ads	489	98.07%	78.49%	88.32%	95.02%
Madelon	142	59.35%	65.23%	64.29%	68.15%
Musk	66	95.23%	96.48%	91.96%	95.93%
Spambase	29	80.34%	90.33%	91.69%	92.90%
SPECTF Heart	11	88.50%	74.57%	73.24%	73.52%

NB and RI achieved the best results on 3 of 8 datasets while k-NN and DT achieved the best results on 1 dataset only. Even though with much less attributes,4 of 8 datasets have better classification performance than using full attributes. In the embryonaltumours dataset, rule induction (RI) has successfully increased the F-measure from 74.41% to 81.00% but with less attributes (from 7,130 original attributes reduced to 619 attributes). In dexter dataset, decision tree (DT) was slightly increased the F-measure from 86.79% to 87.16% but with 31% of original attributes (the number of attributes was reduced from 20,000 to 6,133). In SPECTF heart dataset, naïve Bayes (NB) has also significantly increased the F-measure from 79.69% to 88.50% with 25% attributes (the number of attributes has been reduced by GA from 45 to 11).

However, feature selection using GA does not always improve the classification performance. All four classifiers were unable to improve the classification performance in 4 datasets: internet_ads (its F-measure was slightly decreased from 98.20% to 98.07%), madelon (its F-measure was dropped from 73.32% to 68.15%), musk (its F-measure wasslighlty decreased from 97.15% to 96.48%) and spambase (its F-measure wasslighlty decreasedfrom 93.05% to 92.90%).

Finally, in the third experiment we applied the same classifiers to the eight datasets that have been reduced by PSO and the results are shown in Table 7.The Table 7 shows that NB achieved the best classification performance on 4 of 8 datasets, followed by DT (3 datasets) and k-NN (1 dataset). In this experiment, RI was not as good as other algorithms.

Table 7. Classification performance of PSO-reduced datasets

Data set	Reduced by PSO	NB	k-NN	DT	RI
	#attributes	F-measure	F-measure	F-measure	F-measure
Leukemia	109	96.55%	89.10%	69.12%	70.61%
Embryonal Tumours	202	65.40%	70.74%	76.61%	68.34%
Dexter	279	73.13%	73.98%	44.56%	70.72%
Internet_ads	302	97.77%	73.18%	96.96%	95.12%
Madelon	5	60.24%	64.25%	64.29%	63.07%
Musk	16	99.92%	96.45%	91.65%	95.29%
Spambase	27	90.29%	90.91%	92.92%	92.25%
SPECTF Heart	9	85.89%	81.42%	77.77%	76.82%

PSO has successfully significantly reduced the number of attributes to 13% on average (GA is only 31% on average).Unfortunately this technique does not always improve or maintain the classification accuracy because there were only 3 of 8 datasets have their classification performance improved. The accuracy of embryonaltumourdataset was improved from 74.41% to 76.61% with only 8% attributes, musk dataset was improved from 97.15% to 99.92% with 39% attributes and SPECTF heart dataset was improved from 79.69% to 85.89% with 24% attributes. The other 5 datasets (leukemia, dexter, internet_ads, madelon and spambase) have their classification performance slightly reduced from 0.13% (spambase) to 12.81% (dexter).

6. CONCLUSION

In terms of dimensionality reduction, PSO is much better than GA. PSO has successfully reduced the number of attributes of 8 datasets to 13.47% on average while GA is only 31.36% on average. The most extreme casesare in dexter dataset where PSO reduced the number of attributes to 1.40% (from 20,000 to 279 attributes), and in the madelon dataset where PSO reduced the number of attribute to 1% (from 501 to 5 attributes). In terms of classification performance, GA is slightly better than PSO. GA-reduced datasets have better performance than their original ones on 5 of 8 datasets while PSO is only 3 of 8 datasets. Overall, both GA and PSO are very good in reducing the number of attributes as well as maintaining the classification performance.

REFERENCES

[1] A. C. Braun, U. Weidner, and S. Hinz, **Classification in High-Dimensional Feature Spaces #x2014;Assessment Using SVM, IVM and RVM With Focus on Simulated EnMAP Data**, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 5, no. 2, pp. 436 –443, Apr. 2012.

[2] J. Fan and Y. Fan, **High dimensional classification using features annealed independence rules**, *Ann Stat.*, 2008.

[3] A. J. Miller, *Subset selection in regression*. Boca Raton: Chapman & Hall/CRC, 2002.

[4] I. Fodor, **A Survey of Dimension Reduction Techniques**, 2002.

[5] F. S. Tsai and K.-L. Chan, **Dimensionality reduction techniques for data exploration**, in *2007 6th International Conference on Information, Communications Signal Processing*, 2007, pp. 1–5.

[6] Y. Chen, Y. Li, X.-Q. Cheng, and L. Guo, **Survey and taxonomy of feature selection algorithms in intrusion detection system**, in *Proceedings of the Second SKLOIS conference on Information Security and Cryptology*, Berlin, Heidelberg, 2006, pp. 153–167.

[7] S. K. Dandpat and S. Meher, **Performance improvement for face recognition using PCA and two-dimensional PCA**, in *2013 International Conference on Computer Communication and Informatics (ICCCI)*, 2013, pp. 1–5.

[8] B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge, **Recognizing faces with PCA and ICA**, *Comput. Vis. Image Underst.*, vol. 91, no. 1–2, pp. 115–137, Jul. 2003.

[9] I.-S. Oh, J.-S. Lee, and B.-R. Moon, **Hybrid genetic algorithms for feature selection**, *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 26, no. 11, pp. 1424 –1437, Nov. 2004.

[10] A. S. J. Tjiong and S. T. Monteiro, **Feature selection with PSO and kernel methods for hyperspectral classification**, in *2011 IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 1762 –1769.

[11] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, **An Improved Particle Swarm Optimization for Feature Selection**, *Engineering*, vol. 8, no. 2, pp. 924–928, 2006.

[12] R. Malhotra, N. Singh, and Y. Singh, **Genetic Algorithms: Concepts, Design for Optimization of Process Controllers**, *Comput. Inf. Sci.*, vol. 4, no. 2, p. p39, 2011.

[13] K. Roy and P. Bhattacharya, **Improving Features Subset Selection Using Genetic Algorithms for Iris Recognition**, in *Artificial Neural Networks in Pattern Recognition*, L. Prevost, S. Marinai, and F. Schwenker, Eds. Springer Berlin Heidelberg, 2008, pp. 292–304.

[14] V. Kachitvichyanukul, **On Comparison of Three Evolutionary Algorithms: GA, PSO and DE**, *Ind. Eng. Manag. Syst.*, vol. 11, no. 3, pp. 215–223, Sep. 2012.

[15] C.-L. Huang and J.-F. Dun, **A distributed PSO–SVM hybrid system with feature selection and parameter optimization**, *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1381–1391, Sep. 2008.

[16] M. A. Schuh, R. A. Angryk, and J. Sheppard, **Evolving Kernel Functions with Particle Swarms and Genetic Programming**, in *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference, 2012*, Marco Island, Florida, 2012, pp. 80–85.

[17] M. Korürek and B. Doğan, **ECG beat classification using particle swarm optimization and radial basis function neural network**, *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7563–7569, Dec. 2010.

[18] A. Moraglio, C. D. Chio, J. Togelius, and R. Poli, *Geometric Particle Swarm Optimization*. 2008.

[19] J. Davis and M. Goadrich, **The relationship between Precision-Recall and ROC curves**, in *Proceedings of the 23rd international conference on Machine learning*, New York, NY, USA, 2006, pp. 233–240.

[20] S. B. Kotsiantis, **Supervised Machine Learning: A Review of Classification Techniques**, in *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, Amsterdam, The Netherlands, The Netherlands, 2007, pp. 3–24.

[21] N. Williams, S. Z, and G. Armitage, **A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification**, *Comput. Commun. Rev.*, vol. 30, 2006.