How to cite this article:

Swesi, I. M. A. O., & Bakar, A. A. (2019). Feature clustering for pso-based feature construction on high-dimensional data. *Journal of Information and Communication Technology, 18*(4), 439-472.

FEATURE CLUSTERING FOR PSO-BASED FEATURE CONSTRUCTION ON HIGH-DIMENSIONAL DATA

[1]Idheba Mohamad Ali Omer Swesi & [2]Azuraliza Abu Bakar
[1]Faculty of Accounting, University of Al-Jabar Al-Gharbi, Libya
[2]Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia, Malaysia

ana180611@yahoo.com; azuraliza@ukm.edu.my

ABSTRACT

Feature construction (FC) refers to a process that uses the original features to construct new features with better discrimination ability. Particle Swarm Optimisation (PSO) is an effective search technique that has been successfully utilised in FC. However, the application of PSO for feature construction using high dimensional data has been a challenge due to its large search space and high computational cost. Moreover, unnecessary features that were irrelevant, redundant and contained noise were constructed when PSO was applied to the whole feature. Therefore, the main purpose of this paper is to select the most informative features and construct new features from the selected features for a better classification performance. The feature clustering methods were used to aggregate similar features into clusters, whereby the dimensionality of the data was lowered by choosing representative features from every cluster to form the final feature subset. The clustering of each features are proven to be accurate in feature selection (FS), however, only one study investigated its application in FC for classification. The study identified some limitations, such as the implementation of only two binary classes and the decreasing accuracy of the data. This paper proposes a cluster based PSO feature construction approach called ClusPSOFC. The Redundancy-Based Feature Clustering (RFC) algorithm was applied to choose the most informative

Received: 29 March 2018 **Accepted**: 24 June 2019 **Published:** 26 September 2019

features from the original data, while PSO was used to construct new features from those selected by RFC. Experimental results were obtained by using six UCI data sets and six high-dimensional data to demonstrate the efficiency of the proposed method when compared to the original full features, other PSO based FC methods, and standard genetic programming based feature construction (GPFC). Hence, the ClusPSOFC method is effective for feature construction in the classification of high dimensional data.

Keywords: Particle swarm optimisation, feature construction, genetic programming, classification, high-dimensional data.

INTRODUCTION

Classification is a concept that is applied in the area of data mining and machine learning to classify a class based on the predefined set of classes. However, the classification algorithm fail to produce the desirable results when the data space representation (defined by a set of features) has poor quality (Xue, Zhang, Dai, & Browne, 2013; Dai, Xue, & Zhang, 2014). Therefore, feature transformations that include feature construction (FC) and feature selection (FS) are suggested to improve the quality of the input space. FS refers to a process that selects a subset of informative features from the original data (Dash & Liu, 1997, Swesi & Bakar, 2017). On the other hand, FC refers to the process that selects the informative features and combines them to produce new features that would allow for better discrimination of the problem (Tran, Xue, & Zhang, 2016a; Elola et al., 2017). These processes are conducted because the constructed features have the ability to identify hidden relationships that exist between the original features, particularly when a better classification performance are not achieved from the original features. There are three types of FS and FC approaches: wrapper, filter, and embedded (Chandrashekar & Sahin, 2014; Chen, Zhang, & Xue, 2017). The wrapper approach applies a classifier that serves as an evaluation criteria, while the filter approach does not employ the use of a classifier. The wrapper approach produces better results than the filter approach, however, at the expense of higher computational time. On the other hand, the embedded approach is almost similar to the wrapper approach, as both approaches evaluate models using a learning algorithm. However, the former is faster with regards to computational time (Tran, Zhang, & Xue, 2016b). FS method searches for a good subset of features, given 2^N possible subsets. FC method searches for good features, chooses the appropriate operators, and combines the features.

Moreover, FC requires a bigger search space than FS, and therefore requires a powerful search technique to construct the high-level features.

Evolutionary computation (EC) approaches are global search techniques that are widely utilised in many fields. Genetic programming (GP) is a successful evolutionary algorithm for FC that has the ability to build mathematical expressions, based on tree representation (Tran, Xue, & Zhang, 2016a; Yazdani, Shanbehzadeh, & Hadavandi, 2017; Chen, Zhang, & Xue, 2017; Mahanipour, Nezamabadi-pour, & Nikpour, 2018). A modified Balanced Cartesian Genetic Programming feature extractor (MBCGP-FE) method has been introduced by Yazdani, Shanbehzadeh, and Hadavandi (2017). Experimental results of eight datasets suggested that the proposed method improves the performance by constructing new informative fractures. However**,** the method required high computational time when applied to high dimensional data. Furthermore, the authors also noted the presence of noise in the data that led to the construct of ineffective features that may have affected the classification performance. The solution to this problem was addressed by Mahanipour, Nezamabadi-pour, and Nikpour (2018), that employed a fuzzy rough quick reduct for selecting informative features, and applied GP to construct new features. The results obtained from the five University of California Machine Learning Repository (UCI) datasets supported the effectiveness of the proposed method. However, the experiments were conducted on datasets that contained a small number of features, i.e. not more than 500 features, which suggested that the proposed method may not be effective for high-dimensional data. Recent studies have also proposed to use of GP based embedded FC method to improve the performance of symbolic regression (Chen, Zhang, & Xue, 2017). The performance of the proposed method was evaluated on six datasets, and demonstrated better generalisation ability than the standard GP. However, the proposed method deals with limitations such as overfitting, and the datasets used in the experiments did not reflect the high dimensionality.

Particle swarm optimisation (PSO) is a form of EC technique that was inspired from the behaviour of bird flocking and fish schooling. In comparison to other EC techniques such as GP and genetic algorithm (GA), PSO can converge more quickly and is computationally less expensive. Over the past decade, the algorithm has been extensively employed for FS (Banka & Dara, 2015; Gunasundari, Janakiraman, & Meenambal, 2016; Zhang, Gong, Sun, & Guo, 2017), but has limited use for FC (Xue, Zhang, Dai, & Browne, 2013; Dai, Xue, & Zhang, 2014). However, one possible drawback from the work by Xue, Zhang, Dai, and Browne (2013) was that it implemented a long FC process that selected a large number of features to construct new ones. A maximum of 500 features were used in the experimental datasets for the two studies (Xue, Zhang, Dai, & Browne, 2013; Dai, Xue, & Zhang, 2014). Furthermore, the use of the entire features in both studies that included redundant and irrelevant features, may have resulted in the degradation of the performance. Therefore, to conduct further investigations on the applicability of PSO for FC, a new approach was proposed. The proposed approach is expected to manage datasets with high dimensionality of features, remove redundant and irrelevant features, and select the prominent features for the feature construction process In data mining, clustering refers to the task of grouping a set of instances into clusters. This differs from feature clustering since feature clustering combines similar features into one cluster (Tran, Xue, & Zhang, 2017; Moradi & Rostami, 2015). Based on the resultant clusters, one or more features can be selected from each cluster to form the resulting feature subset. The clustering of features have evidently achieved better performance in numerous FS methods (Tran, Xue, & Zhang, 2017; Sahu & Mishra, 2012; Jaskowiak & Campello, 2015; Gupta, Gupta, & Sharma, 2016). Nonetheless, there has been insufficient studies conducted on the use of feature clustering techniques in the field of FC. This paper presents a new approach that applies feature clustering for PSO-based FC within the classification of high-dimensional data, known as ClusPSOFC. Feature clustering is implemented in this approach to reduce the dimensionality, improve classification performance, and select prominent features. The evaluation of this method was conducted on six UCI datasets and six microarrays datasets, whereby the results obtained were compared to PSO based feature construction (PSOFC) (Xue , Zhang, Dai, & Browne; 2013), PSO based feature construction using array representation (PSOFCArray), PSO based feature construction using pair representation (PSOFCPair), and genetic programming based feature construction (GPFC). Thereafter, the following issues are addressed:

i. The effectiveness of the clustering algorithm to automatically group features into clusters.
ii. Investigating the performance of the PSOFC approach after applying the clustering algorithm against other PSO based FC methods and the GP based FC method.
iii. The usefulness of ClusPSOFC in choosing a lower number of features than other PSO based FC approaches and GP based FC method.
iv. Investigating the performance of combining the original and constructed features with regards to accuracy.

The rest of this paper is organised in the following manner: the overall background information, methodology of a proposed feature construction method, followed by the experimental results and their discussion. Finally, the last section concludes the paper with some remarks for future directions.

BACKGROUND

Particle Swarm Optimization

The population-based stochastic optimisation technique, PSO was first developed by Eberhart and Kennedy (1995). The social behaviour of fish schooling and bird flocking served as their inspiration. The algorithm begins with an initial random population that is referred to as swarm of particles. Each particle serves as a candidate solution for the main problem, and is processed as a part in n-dimensional space. Through PSO evolution, all the particles have a tendency to move towards better search space positions until an optimal solution is achieved. For every particle i, a vector $(x_i = x_{i1}, x_{i2}, \dots, x_{iD})$ represents a position, whereby a vector $(v_i = v_{i1}, v_{i2}, \dots, v_{iD})$ is defined as the velocity. The dimensionality of the search space is represented by '*D'*. Each particle's velocity and position is updated, by using Equations (1) and (2) respectively. The best position achieved by each particle is known as *pbest*, while the best position achieved by the whole swarm is known as *gbest*.

$$v_{id}^{t+1}=w * v_{id}^{t}+c_1 * r_1 * [p_{id}-\ x_{id}^{t}]+c_2 * r_2 * [p_{gd}-\ x_{id}^{t}] \tag{1}$$

$$x_{id}^{t+1}=x_{id}^{t}+v_{id}^{t+1} \tag{2}$$

Where v_{id}^{t} and x_{id}^{t} represent the velocity and the position respectively, of particle at iteration t in the dimension *d*. Furthermore, pbest and gbest positions are denoted as and respectively. *W* represents the inertia weight used to regulate the balance between the exploration and the exploitation. c_1, and c_2 refer to the acceleration constants, r_1 and r_2 refer to the random numbers that are uniformly distributed between 0 and 1, v_{id}^{t+1} and $\in [-vmax, vmax]$. The PSO was originally developed to handle the continuous optimisation problems. To expand the application of PSO, Eberhart and Kennedy (1997) designed another version of the PSO, known as BPSO. The BPSO tackles discrete optimisation problems and perform feature selection. A binary bit string encodes the position of the particles in BPSO, where each bit represents a feature; i.e. if the value of the bit is 1, it indicates a selected feature, whereas a bit value of 0 indicates a non-selected feature. A sigmoid transfer function is applied to transform the real-value velocities into probability values that ranges between (0, 1), while the position of every particle is updated using the formula below:

$$x_{id}^{t+1}=\begin{cases} 1, & \text{if rand( )<s(}v_{id}^{t+1}\text{)} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Where $s(v_{id}^{t+1})=\frac{1}{1+e^{-v_{id}^{t+1}}}$ (4)

In Equation 4, represents sigmoid transformation, while *rand* is a random number that is selected from the uniform distribution in [0, 1].

Related Work

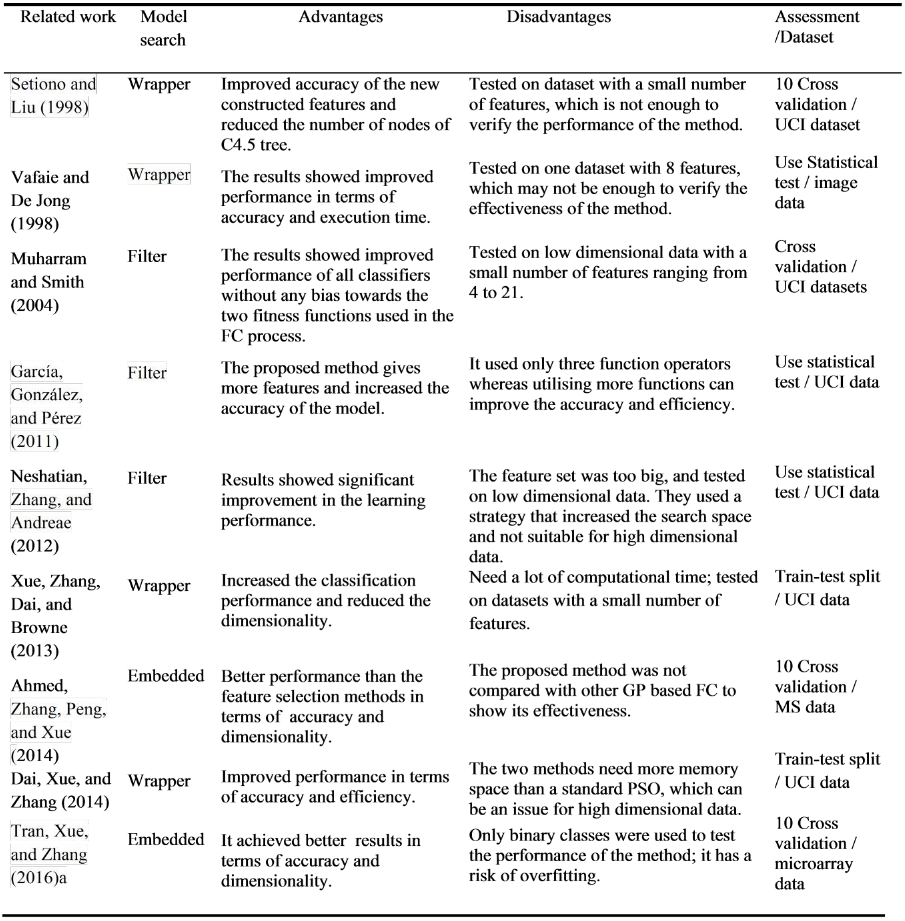
Feature construction methods are rarely cited in literature compared to feature extraction and feature selection methods. Within the literature on FC, an earlier work was proposed by Setiono and Liu (1998). The study formulated an automatic system to build compound features for both discrete and continuous data. Based on the results, there were improvements in the performance of the neuronal network using both artificial and real world datasets. In a subsequent study, research conducted by García, González, and Pérez (2011) proposed a novel technique that applied a set of predetermined functions over the input variables to examine if the combination of the attributes would provide additional information regarding the classification performance as compared to a single attribute. The experiments were carried out using 9 UCI databases comprised of 2-60 features, and demonstrated that the proposed technique significantly increased prediction accuracy.

Various algorithms have been designed to enhance the learning concept by utilising different feature construction methods. However, most of these algorithms are based on GP, due to its effectiveness to construct programs and expressions. For example, a feature selection and construction method was proposed by Vafaie and De Jong (1998). The proposed method used a GP to achieve FC, and implemented GA to further reduce the number of features through FS. Based on the experimental results, the proposed algorithm enhanced the classification performance and/or lessen the number of features that were required. However, the wrapper approaches typically have a high computational time. To address this issue, FC techniques using the filter approaches were proposed due to their low computational cost. In a study, Muharram and Smith (2004) proposed a filter based GP approach for FC using two univariate feature selection metrics; i.e. information gain and gini index, as fitness functions. The study evaluated the proposed method on five UCI datasets, and noted that there was an improvement in the classification performance using the constructed features. In contrast to the single FC technique proposed by Muharram and Smith (2004), a GP-based filter for constructing multiple features was presented by Neshatian, Zhang, and Andreae (2012). In this method, the construction of new features is coupled with the application of the entropy-based fitness function. Furthermore, the decomposable objective function was used to construct the multiple features. Experimental results show that the newly constructed features have the ability to improve the learning performance. Although these studies have displayed promising results, further studies are still require to investigate their application on high dimensional data. Recently, GP-based FC methods were proposed to handle high dimensional data (Ahmed, Zhang, Peng, & Xue, 2014; Ahmed, Zhang, Peng, & Xue, 2016; Tran, Xue, & Zhang, 2016a; Tariq, Eldridge, & Welch, 2018). In one study, Ahmed, Zhang, Peng, and Xue (2014) introduced a GP-based FC that constructed multiple features using all possible subtrees with the best agents. The study was implemented by using eight mass spectrometry datasets, and the results suggested that the constructed features achieved better performance in comparison to the original features. In contrast to the previous study that used GP to only construct multiple features, Tran, Xue, and Zhang (2016a) introduced an embedded GP that was used to construct both single and multiple features. A single feature was constructed from the entire tree while the multiple features were built using all possible subtrees. The study was carried out using seven high dimensional data. Based on the results, the proposed method significantly enhanced the classification accuracy and reduced the dimensionality. However, the method lowered the classification performance, since the datasets had large number of features which contained redundant or irrelevant features.

PSO has been used to address a broad range of problems, and have successfully solved feature selection (Rutkowski, 2008; Banka & Dara, 2015; Xue, Zhang, & Browne, 2014; Jain, Jain, & Jain, 2018). However, for feature construction, only three works were proposed (Xue, Zhang, Dai, & Browne, 2013; Dai, Xue, & Zhang, 2014; Mahanipour & Nezamabadi-pour, 2017). A PSO-based feature construction (PSOFC) was first proposed by Xue, Zhang, Dai and Browne (2013). In this approach, BPSO was used to select the low level features followed by a set of operators. A local search was then performed to combine these features to produce a new one. Based on the experimental results obtained from the seven UCI datasets, the proposed algorithm was able to construct a single new feature with better classification performance. However, one of the problems of the PSOFC is that the feature construction process becomes longer if vast amounts of features were included, as the local search evaluates all the operators to find the optimal feature for each of the selected features. Hence, the process requires a longer computational time when the number of selected features are large. In their second work, Dai, Xue, and Zhang (2014) introduced PSOFCArray and PSOFCPair that employed two representations: array representation and pair representation. According to the results, it was discovered that these methods were able to enhance the classification performance. Nevertheless, the PSOFCPair was useful in determining if the feature was chosen. Other than that, it was possible to discern the selected operators which may not be ideal for both feature and operator selection. Furthermore, by using one dimension in the particle to determine the selection of both features and operators, this may limit the search of the best combination to construct a new feature with better classification performance. Recently, Mahanipour and Nezamabadi-pour (2017) modified the two approaches in Dai, Xue, and Zhang (2014) by applying the forward feature selection (FFS) method to reduce the dimensionality. The two modified approaches were then used to construct the new features. The results showed an increase in the classification performance. However, the experiments were only performed on datasets with a small number of features, that ranged between 14 and 500.

Table 1

Summary of EC based FC methods



Related work	Model search	Advantages	Disadvantages	Assessment /Dataset
Setiono and Liu (1998)	Wrapper	Improved accuracy of the new constructed features and reduced the number of nodes of C4.5 tree.	Tested on dataset with a small number of features, which is not enough to verify the performance of the method.	10 Cross validation / UCI dataset
Vafaie and De Jong (1998)	Wrapper	The results showed improved performance in terms of accuracy and execution time.	Tested on one dataset with 8 features, which may not be enough to verify the effectiveness of the method.	Use Statistical test / image data
Muharram and Smith (2004)	Filter	The results showed improved performance of all classifiers without any bias towards the two fitness functions used in the FC process.	Tested on low dimensional data with a small number of features ranging from 4 to 21.	Cross validation / UCI datasets
García, González, and Pérez (2011)	Filter	The proposed method gives more features and increased the accuracy of the model.	It used only three function operators whereas utilising more functions can improve the accuracy and efficiency.	Use statistical test / UCI data
Neshatian, Zhang, and Andreae (2012)	Filter	Results showed significant improvement in the learning performance.	The feature set was too big, and tested on low dimensional data. They used a strategy that increased the search space and not suitable for high dimensional data.	Use statistical test / UCI data
Xue, Zhang, Dai, and Browne (2013)	Wrapper	Increased the classification performance and reduced the dimensionality.	Need a lot of computational time; tested on datasets with a small number of features.	Train-test split / UCI data
Ahmed, Zhang, Peng, and Xue (2014)	Embedded	Better performance than the feature selection methods in terms of accuracy and dimensionality.	The proposed method was not compared with other GP based FC to show its effectiveness.	10 Cross validation / MS data
Dai, Xue, and Zhang (2014)	Wrapper	Improved performance in terms of accuracy and efficiency.	The two methods need more memory space than a standard PSO, which can be an issue for high dimensional data.	Train-test split / UCI data
Tran, Xue, and Zhang (2016)a	Embedded	It achieved better results in terms of accuracy and dimensionality.	Only binary classes were used to test the performance of the method; it has a risk of overfitting.	10 Cross validation / microarray data

Based on the studies presented above, further investigation is needed to test the performance of PSO for FC. Some of the significant issues that need to be addressed includes the improvement of the PSO efficiency and the investigation of its application potential in datasets that contain large numbers of features. Although PSO has been applied to feature construction, current works are only limited to the datasets that has a small number of features (a few hundred). However, no research that applies PSO for FC using high dimensional data has been conducted. Furthermore, applying PSO to the original data may not be useful in the construction of new features as the original data may have irrelevant or redundant features. Moreover, there is a high probability that PSO would choose unimportant features for the feature construction process. Therefore, these limitations would adversely affect the classification performance. Table 1 shows a list of the aforementioned EC based FC methods, with their advantages and disadvantages.

Feature Clustering

Clustering is considered a major task in data mining. The goal of this method is to group similar objects into clusters. Different clustering techniques were introduced and various measures were utilised to assess the similarities between objects (Xu & Tian, 2015; Wong, 2015; Jabbar, Ku-Mahamud, & Sagban, 2018). Recently, there has been a growing interest in the application of clustering algorithms to aggregate similar features into clusters. This is referred to as feature clustering, and is subsequently used to accomplish feature selection (Roth & Lange, 2003). Feature clustering is a powerful technique that is used to lower the dimensionality of data by grouping similar features into the same cluster. One or more features from each clusters are eventually chosen to form the final subset. Various clustering methods and diverse techniques that studies the outcomes (features) of each clusters were proposed (Nguyen, Xue, Liu, & Zhang, 2014; Sardana, Agrawal, & Kaur, 2016).

The classification of high dimensional data is challenging due to the nature and high dimensionality of the feature sets. These datasets are characterized by a large number of features (in thousands) and small number of samples that are less than one hundred, however many of these features would be irrelevant or redundant. Furthermore, by defining these attributes, the classification performance would be severely constrained, and would cause the run-time and classifier complexity to increase (Tran, Xue, & Zhang, 2016a). These problems can be handled by using an effective approach that obtains non-redundant and relevant features from high dimensional data. The proposed approach should reduce the search space before further exploration by the wrapper method (like PSO). Hence, rather than investigating the whole feature set, a small set consisting of relevant and non-redundant features are chosen and carried forward to PSO for effectual construction of new features.

The Correlation Coefficient (CC) is the most popular measure to evaluate the redundancy or dependency among features. Although CC has not been employed as much as mutual information, it offers a quantitative measurement that evaluates the strength of a linear relationship between two variables. In one study, Hsu and Hsieh (2010) replaced a distance measure with CC in k-means clustering algorithm. The experimental results on two datasets that contained hundreds of features indicated that the proposed method achieved better performance than one method, but performed worse than the other method that was suggested. Therefore, it is essential to apply the k-means algorithm to define the number of clusters that could influence the performance of the proposed method. In another study, a feature selection approach based on correlation and clustering was introduced (Kumari, Rajeswari, & Vaithiyanathan, 2015). The approach was conducted in two phases. The first phase eliminated irrelevant features based on the correlation between feature and class. The second phase removed redundant features by constructing a binary tree of the features that were previously relevant. The tree was then divided into clusters, and one representative feature from each cluster was chosen to constitute the final feature subset. The results demonstrated that the proposed algorithm has the ability to effectively remove redundant and irrelevant features, which resulted in attaining a small feature size and better classification accuracy.

Correlation coefficient gauges the number of correlation that are present among features, while Mutual Information (MI) emphasis the information that is obtained between the features. Symmetric uncertainty (SU) (Hall & Smith, 1998), is viewed as a normalized version of MI, and is used to identify non-redundant and relevant features. If a feature's SU is smaller than the thresholds, then it is considered irrelevant and is discarded. Conversely, the redundancy between two features is examined if the value of SU is high (Tran, Xue, & Zhang, 2017). The study by Song and his co-authors (Song Ni, & Wang, 2013) combined the SU and MST tree to cluster features. First, SU was used to eliminate irrelevant features. Thereafter, a MST was created by using the relevant features identified by SU. The results from 35 high dimensional data indicated that the proposed method performed better than the other methods. Recently, Tran and colleagues combined SU and CC to develop a new cluster algorithm known as Redundancy Feature Clustering (RFC) (Tran, Xue, & Zhang, 2017). They introduced a cluster-based GP feature construction (CGPFC) method that uses RFC algorithm to improve the GP's performance. Results from eight gene expression data suggested that the proposed method performed better than the slandered GP and the original data. In summation, feature clustering approaches are beneficial to FS and FC as it removes redundant and irrelevant features. However, no research in literature has investigated feature clustering in PSO for feature construction. By applying the RFC algorithm, this study proposes the utilisation of feature clustering to assemble features into clusters. From each cluster, the best features will be chosen for feature construction.

PROPOSED METHOD

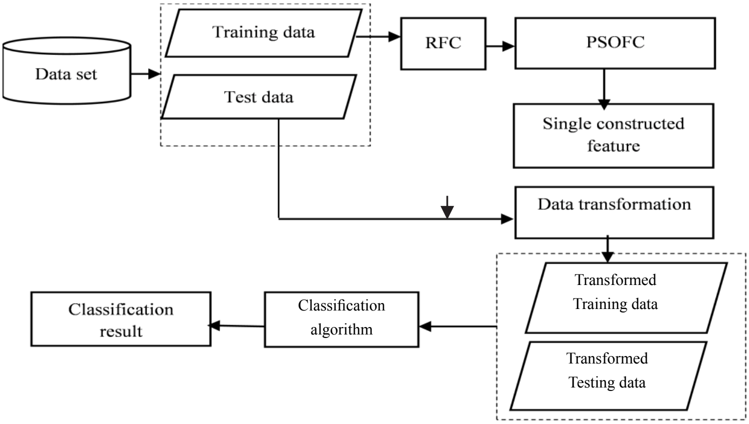


Figure 1. The proposed ClusPSOFC approach.

In this section, the proposed cluster-based PSOFC (ClusPSOFC) approach for high dimensional data is introduced. Figure 1 shows the overall structure of this approach. The main objective of this approach is to combine a PSO based FC with a cluster algorithm to reduce the dimensionality of the data, remove redundancy among features, and improve the PSO performance.

From Figure 1, the redundancy-based feature clustering (RFC) algorithm is utilized to group features into clusters. Then, the best feature from each cluster is selected to form the final feature subset that is applied to PSO to construct a new feature. Based on the newly constructed feature, the new training data and test data is then created by removing features that were not selected in the feature construction process. Finally, a classification algorithm is generated from the transformed training data, and a classifier is applied to the transformed test data to produce the final classification results. In this method, the RFC algorithm utilises a filter measure to cluster features, while PSOFC is based on the wrapper approach. Further details on the two methods are explained in the following sub-sections.

Redundancy-Based Feature Clustering Method

The first stage of the proposed method is the application of the RFC which combines groups of similar (redundant) features into the same cluster. Over the years, various clustering techniques have been suggested, however, the k-means approach is the most popular, due to its simplicity and effectiveness. However, one of the limitation of k-means is the requirement to specify the number of clusters in advance, which can be difficult especially for data with high features dimension. If the number of clusters are not defined appropriately, this may lead to the grouping of redundant or uncorrelated features. For feature clustering, the number of clusters is not important as the number of clusters in instance clustering. Thus, instead of clustering features based on a predetermined number of clusters, an automatic feature clustering approach is required to automatically group features into clusters. In Tran, Xue, and Zhang (2017), the proposed RFC algorithm is significant and was adopted for this study, which will be employed as a redundancy concept that has been the focus of current literature. Unlike the number of clusters, the correlation or redundancy between a pair of features can vary from 0 to 1, where 0 is an indication of no correlation between features, while 1 indicates full correlation.

RFC is a simple approach that ensures all features found in the same cluster are considered redundant if their correlation is higher than the threshold. Hence, this approach assembles features that have a redundancy level that is greater than the predefined threshold value. Given two features; A and B with a CC higher than the threshold, these features will then be combined into the same cluster. In such case, there will be an automatic determination of the number of clusters. Furthermore, by employing this approach, it is possible to produce clusters that contains only features with correlations that are equal to or higher than the predefined redundancy threshold.

The major processes of the RFC algorithm are presented in Algorithm 1. At the start, all features that are irrelevant are removed in Part 1. This work assumes that a feature is irrelevant if it is incapable of providing information regarding its class. SU is an appropriate method for feature relevancy, therefore, a feature with a SU value that is higher than the threshold is classified as a strong relevant feature. Equation 5 is used to calculate the SU between target class C and feature X. SU assigns a value between 0 and 1, which indicates no correlation and full correlation respectively.

$$\mathrm{SU(X,C)} = 2\left[\frac{\mathrm{Gain(X|C)}}{\mathrm{H(X)+H(C)}}\right] \tag{5}$$

$$\mathrm{Gain(X|C)} = \mathrm{H(X)} - \mathrm{H(X|C)} \tag{6}$$

Where H (X|C) represents the conditional entropy for X given C, and H(X) is the entropy of a discrete random variable X.

After determining all relevant features, these features are ranked based on the SU values whereby the first feature on the list is the most relevant feature. The first while loop selects the next feature *f* from the list to form the first cluster. Then, the remaining features in the list are scanned through the second while loop in its order of SU values, to add any feature that is correlated with *f*. In this step, CC was used to quantify the redundancy between features (Redundant feature removal) from Part 2. Although the correlation coefficient can only gauge the linear relation between variables, it has been proven to be successful in numerous feature selection methods (Tran, Xue, & Zhang, 2017; Hsu & Hsieh, 2010). The values of the CC is between 1 and -1, whereby the absolute value describes the correlation that exists between the two features. The CC for a pair of variables A and B is calculated using Equation (8):

$$\mathrm{CC(A,B)} = \left|\frac{\sum_{i=1}^{n} A_i B_i - n\bar{A}\bar{B}}{\sqrt{\sum_{i=1}^{n} A_i^2 - n\bar{A}^2}\sqrt{\sum_{i=1}^{n} B_i^2 - n\bar{B}^2}}\right| \tag{8}$$

If the CC value between two features is higher than the threshold, the features are considered redundant and are grouped into the same cluster. When a feature is added to a cluster, it is deleted from the list. Hence, all features are grouped into various clusters and the number of clusters are automatically defined based on the predetermined redundancy threshold. Finally, the second while loop returns all created clusters, from which the best feature from each cluster is selected to form the final feature subset that will be used as the input set for PSO based feature construction.

Algorithm 1: The Redundancy-Based Feature Clustering (RFC)

```
Input:  // D= (F1, F2 ….Fm, C)← the training data and its class label
           ϴ← The T-Redundancy threshold
Output: // Clusters of features
Part 1: // Irrelevant feature removal
        Step 1:    For every feature (Fi) in the training data (D) do
        Step 2:    T-Relevance ← SU (Fi, C) using Eq. (5) // Calculate the relevance
        Step 3:    If T-Relevance > 0, then
        Step 4:    Add the feature in the list F where F← F ∪ {Fi}// group relevant features
        Step 5:    Sort the features in (F) // sort all the relevant features based on their SU
Part 2:  // Redundant feature removal
        Step 6:    While (F ≠Φ) do
        Step 7:    Fi ←next feature in F
        Step 8:    Cluster ← {Fi}// add a feature Fi into cluster
        Step 9:    While (F ≠Φ) do
        Step 10:   Fj ←next feature in F
        Step 11:   T-Redundancy← CC (Fi, Fj) using Eq. (8)// Calculate the redundancy
        Step 12:   If (CC > ϴ), then
        Step 13:   Cluster← Cluster ∪{Fi}// add a feature Fi into cluster
        Step 14:   Clusters← Clusters ∪ Cluster
        Step 15:   Return clusters// return the created clusters whose best features will be
                   used for PSOFC
```

Proposed ClusPSOFC Algorithm

Basic PSO-based feature construction method

The PSOFC method is presented in Xue, Zhang, Dai, and Browne (2013). In this method, BPSO is used to select the number of low-level features from the original data so that a new feature can be constructed. In BPSO, each particle is represented by the n-bit binary string, where the value of '0' indicates that the feature was not chosen while the value of '1' indicates that the feature was chosen to construct a new set. As the aim of this method is to construct informative features, a set of operators are applied to the original features. However, the main challenge of using PSO for feature construction is that the PSO is incapable of directly selecting operators. Thus, operators are selected via the application of a local search that takes a longer time.

Fitness function

In this study, the proposed ClusPSOFC approach applies the wrapper method. Hence, various learning algorithms are utilised to evaluate the performance of the constructed features. To assess every PSO individually, the training set is transformed based on the feature constructed. Thereafter, the classification performance of the transformed training set is tested using the classification accuracy (CA). CA is essentially used as a fitness measure to guide the search, where the instance of the new constructed feature is classified as class 1 if its corresponding value is greater than ‘0’; otherwise, it is classified as class 2.

Overall ClusPSOFC algorithm

In this stage, after the features are aggregated into clusters, the best features are collected from each cluster so that they can be utilised to construct a new feature. This stage introduces the proposed ClusPSOFC method. Through the application of this technique, the RFC groups the features using a filter measure, while the PSOFC algorithm conforms to the wrapper method. Algorithm 2 describes the pseudo code for the proposed ClusPSOFC approach. First, the RFC is applied to build a set of clusters. Then, the best feature from each clusters are chosen to form the final feature subset that is used to generate the PSO individuals. The lines included in the while loop are used to construct a new feature, and the loop is performed until the stopping criterion is implemented. In these lines, the low level features are selected by BPSO and each particle represents a binary string, whereby features with the value ‘1’ are used to construct a new feature, while features with the value ‘0’ are discarded. After selecting the low level feature, a set of function operators are chosen to combine the selected features. These operators include four mathematical operators (+, –, *, and protected division %), and are applied to the selected features using a local search method. Subsequently, a new evolved feature of the best individual is produced.

Algorithm 2: The Pseudo Code of the Proposed ClusPSOFC Algorithm.

```
Require: Training data and test data, redundancy threshold θ;
Ensure:   gbest (single constructed feature), training and test classification performance;
Begin
    Call RFC algorithm; /*According to Algorithm 1*/
    Initialise individuals randomly utilising the best feature in each cluster of clusters;
    While Maximum number of generations is not reached do
        Group the low-level features selected by a particle;
        Choose function operators for constructing new features;
        Construct a new one feature for each particle i on training data;
        Evaluate the new constructed feature using the learning algorithm;
        Fitness ← accuracy using binary classification;
        For i=1 to Swarm Size do
            Update the personal best (pbest) of particle i;
            Update the global best (gbest) of particle i;
        End
            For i=1 to Swarm Size do
                For d=1 to Dimensionality do
                    Update the velocity of particle i using Equation 1;
                    Update the position of particle i using Equations 3 and 4;
                End
            End
    End
    Calculate the classification performance of the constructed feature based on the test set
    using NB and KNN as the learning algorithm;
    Return the position of gbest (the constructed feature);
    Return the training and testing classification performance;
End
```

EXPERIMENTS AND DISCUSSION

In this section, two sets of experiments were designed. The first set of experiments were conducted by utilising six datasets with low dimensionality of features (Experiment I), and the second set of experiments were performed using six datasets with high features dimension (Experiment II).

Experiment I

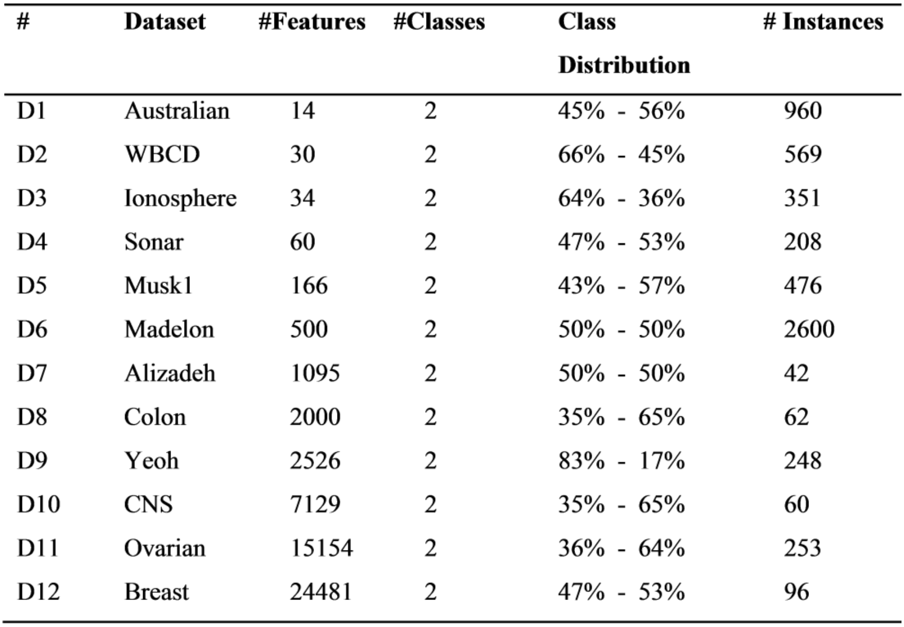
This experiment used six UCI datasets of low to medium features dimension (ranging between 14 and 500). The experiment was designed to examine and classify the datasets, and was conducted with the application of the proposed method in this study that adhered to the predefined PSO parameters. In addition, the performance of the ClusPSOFC method against other existing PSOFC methods were examined.

Experimental design

To study the performance of the ClusPSOFC algorithm, critical comparisons were conducted with three other PSO algorithms that are used for FC, namely PSOFC (Xue, Zhang, Dai, & Browne, 2013), PSOFCArray, and PSOFCPair (Dai, Xue, & Zhang, 2014). Six UCI datasets were downloaded from the UCI machine learning repository (Frank & Asuncion, 2010), and were used to test the performance of the algorithms. The main characteristics of the datasets are summarised in Table 1.

Table 1

Description of the datasets



#	Dataset	#Features	#Classes	Class Distribution	# Instances
D1	Australian	14	2	45% - 56%	960
D2	WBCD	30	2	66% - 45%	569
D3	Ionosphere	34	2	64% - 36%	351
D4	Sonar	60	2	47% - 53%	208
D5	Musk1	166	2	43% - 57%	476
D6	Madelon	500	2	50% - 50%	2600
D7	Alizadeh	1095	2	50% - 50%	42
D8	Colon	2000	2	35% - 65%	62
D9	Yeoh	2526	2	83% - 17%	248
D10	CNS	7129	2	35% - 65%	60
D11	Ovarian	15154	2	36% - 64%	253
D12	Breast	24481	2	47% - 53%	96

For all the algorithms used in Experiment 1, classification accuracy was calculated using the Naive Bayes (NB) classifier on the unseen test data. NB is a common classifier that is employed for its flexibility and efficiency. It has been shown to perform well in different types of problems because of the simplistic nature of the model. NB assumes the features are conditionally independent, for which the existence or absence of a feature does not affect the existence or absence of any other feature if the target class is given. In all datasets, 70% of the observations were randomly chosen for training, while the remaining 30% were allotted for testing. The parameters for all PSO based FC methods are set as follows: w = 0.7298, $vmax$ = 6.0, $c1$ = $c2$ = 1.49618, swarm size = 30, and maximum iteration = 100. The selection of the parameters are based on Xue, Zhang, Dai, and Browne (2013), and Dai, Xue, and Zhang (2014). The function set is made up of four basic arithmetic operators that are utilised to build new features, which are "+", "-", "*" and "/" (protected division). Furthermore, the redundancy threshold was set at 0.9, based on Tran, Xue, and Zhang (2017). The algorithms for each dataset ran for 50 independent times. A t-test was carried out as a statistical significance test to compare the classification performance of the various algorithms, with a significance level fixed at 0.05. The '+' or '-' operators signify if the classification performance of the proposed method is significantly better or worse compared to the other methods. '=' indicates that methods have similar classification performances. In general, the higher the number of '+' operators, the better the performance of ClusPSOFC.

Results and discussions

The performance of the features constructed by the proposed method was examined by comparing it against the original features, and those that were constructed by the other three PSO based FC methods. Table 2 shows the results for the PSOFC, PSOFCArray, PSOFCPair, and proposed ClusPSOFC. To ensure an unbiased comparison between the methods, all algorithms used the same parameters and was conducted in same number of times. From Table 2, 'All' denotes all the original features served as inputs for the classifier. In addition, 'CF' signifies the constructed feature that serves as the input for the classifier. "FCOrg" signifies the constructed feature that was combined with the original features, and was utilised for classification. Furthermore, the accuracy corresponding to 'All' for the proposed ClusPSOFC method was obtained after the application of the clustering algorithm. 'A', 'B', and 'Std' represent the average, best, and standard deviations respectively for the test accuracy obtained by NB on the constructed features that combined the original and constructed features over 50 runs. 'T' denotes the T-test results of the proposed method when compared to the other three methods. For each datasets, the best result was presented in bold.

Table 2

Classification accuracy of PSOFC, PSOFCArray, PSOFCPair and ClusPSOFC on the tested data sets using NB classifier

D#	Method	PSOFC			PSOFCArray			PSOFCPair			ClusPSOFC	
		B	A±Std	T	B	A±Std	T	B	A±Std	T	B	A±Std
D1	All	**85.51**			85.51			85.51			81.88	
	CF	**86.47**	62.97±10.2E0	+	76.33	55.21±5.02	+	59.9	53.93±1.84	+	79.95	**78.45±6.2E-3**
	CFOrg	88.41	**86.97±43.7E-2**	-	**88.89**	86.59±67.4E-2	-	85.99	85.34±45.1E-2	-	83.33	81.67±5.8E-3
D2	All	90.64			90.64			90.64			**97.66**	
	CF	61.4	61.4±35.1E-4	+	61.4	61.4±0	+	61.99	61.41±8.26E-2	+	**99.22**	**97.34±8.5E-3**
	CFOrg	90.64	90.64±32.7E-4	+	90.64	90.64±0	+	90.64	90.64±0	+	**98.44**	**98.00±2.4E-3**
D3	All	28.57			28.57			28.57			**92.45**	
	CF	83.81	77.56±1.71E0	+	87.62	82.32±1.56	+	8476.	80.86±2.36	+	**89.94**	**86.13±2.28E-2**
	CFOrg	28.57	28.57±14.3E-4	+	28.57	28.57±0	+	28.57	28.57±0	+	**94.34**	**92.99±7.9E-3**
D4	All	53.97			53.97			53.97			**86.24**	
	CF	68.25	68.25±0E0	+	47.62	47.62±0	+	47.62	47.62±0	+	**88.36**	**81.99±3.36E-2**
	CFOrg	33.33	33.33±1.42E-4	+	53.97	53.97±0	+	53.97	53.97±0	+	**87.30**	**84.40±1.25E-2**
D5	All	42.66			42.66			42.66			**73.08**	
	CF	59.44	58.43±62.8E-2	+	60.14	59.36±43.5E-2	+	60.14	59.38±36.6E-2	+	**77.27**	**67.40±4.43E-2**
	CFOrg	72.73	72.73±27.3E-4	+	72.73	72.73±0	+	72.73	72.73±0	+	**78.67**	**74.48±1.45E-2**
D6	All	49.49			49.49			49.49			**66.84**	
	CF	49.1	49.1±25.6E-4	+	49.49	49.49±0	+	49.49	49.49±0	+	**69.06**	**67.59±7.6E-3**
	CFOrg	55.38	55.38±46.2E-4	+	49.49	49.49±0	+	55.51	55.51±1.84E-2	+	**68.55**	**67.13±6.9E-3**

ClusPSOFC versus All features

Based on the results in Table 2, although a single feature constructed by ClusPSOFC was utilised, the NB was able to achieve similar or slightly better accuracy than the other methods that were incorporated. In most cases, the best accuracy (B) of the single constructed feature is usually higher, in contrast with the use of the ‘All’ features. In this particular instance, the best accuracy of the NB single constructed feature was observed to be better compared to utilising all the features on four out of six datasets. The highest improvements in the best accuracy is 4.19% on Musk1 dataset, and 2.22 on Madelon dataset. Based on these results, the proposed method is able to detect the hidden information that is held by the low features, and is capable of creating new features that can perform better than the original features.

Additionally, the evaluation of the performance for the feature construction methods that involved only one feature was carried out by adding the new feature into the original features, and subsequently combining the features for classification. Through the application of this process, the average accuracy of NB was observed to be similar or better than the original features in almost all the cases. Furthermore, the best accuracy (B) for the combination of both the original features and constructed features was observed to be higher compared to the original features on all the datasets. The highest improvement of the best accuracy is an increase of 5.59% on Musk1 dataset and 1.89% on Ionosphere dataset. Based on these findings, it is suggested that the combination of the constructed feature and original features provided useful information that increased the classification performance. Nevertheless, in some cases, the constructed feature may be considered as a redundant feature, whereby its combination with the original features may not affect the average accuracy as the accuracy did not decrease in the Australian and Ionosphere datasets.

Overall, a total of 24 comparisons were made between ClusPSOFC (using CF & CFOrg) and 'All' on 6 datasets, with the use of NB classifier. The feature constructed by ClusPSOFC revealed 14 successful attempts, 7 unsuccessful attempts, and 3 that were unchanged. These results suggest that the discriminating ability of the constructed feature by ClusPSOFC, is higher than the original feature set.

ClusPSOFC versus PSOFC, PSOFCPair and PSOFCArray

Based on Table 2 and Figure 2 on the comparison of the other PSO based methods, the ClusPSOFC-based NB learning algorithm that used constructed feature (CF) only, was able to attain higher results than PSOFCPair, PSOFC and PSOFCArray on all six datasets. The highest improvement in the average accuracy of the ClusPSOFC constructed feature is 36% on WBCD dataset, 34% on Sonar dataset, and 18% on Madelon. In the Australian dataset, the NB was 6.52% lower than PSOFC, however, its average accuracy was 15.48% higher. On the other hand, when the constructed feature was combined with the original features, ClusPSOFC performed better than all the other methods in five out of six datasets, while its performance in one of the dataset declined.

Out of the 90 comparisons conducted between ClusPSOFC and the other three PSO based methods that used NB classifier for the six datasets, the proposed method had 80 successful comparisons, while 10 were unsuccessful. This result is attributed to the capability of the PSOFC method that applies an inner loop to select the best operator. Thereafter, this led to an exhaustive search of all possible operators to determine the optimal operator for every feature, and is further refined to obtain an improved set of operators. Moreover, when the RFC algorithm is applied to the PSOFC, it chooses a subset of informative features and removes those that are irrelevant and redundant. Then, these features with the highest information are used as inputs for the PSOFC algorithm to build a new feature to attain a more accurate algorithm.

To further validate the significant difference between ClusPSOFC, PSOFCPair, PSOFC and PSOFCArray, a statistical test (t-test) was performed. From Table

2, it is observed that there is a significant difference between ClusPSOFC and the other three methods with regards to the classification accuracy. Therefore, this paper concludes that the ClusPSOFC method is better than PSOFC, PSOFCPair, and PSOFCArray in solving FC problems.

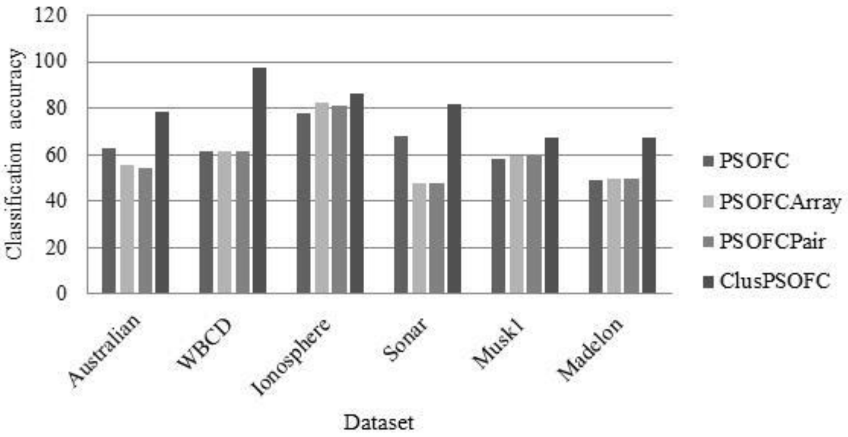


Figure 2. Comparison of average classification accuracy on constructed feature (CF).

Performance of ClusPSOFC for the selected features

In PSO based feature construction methods, a FS process is applied to select the informative features from the full features, to construct a new one. Although all the methods constructed a single feature, it is noted that the number of features used to create one new feature are different. Figure 3 shows that the proposed algorithm always chooses a significantly lower amount of features compared to the other methods. For the three datasets; Sonar, Musk1, and Madelon, ClusPSOFC method chose less than half of the amount of features that were used by PSOFCPair, PSOFC and PSOFCArray. This is due to the application of the RFC clustering algorithm that selected the informative features and reduced the dimensionality. When a small amount of features were used, the new features that was constructed by ClusPSOFC had a better classification performance, compared to the other methods that utilised all the features. Therefore, this paper concludes that the features chosen by ClusPSOFC demonstrate better discriminating ability compared to those used by other methods. Furthermore, when redundant features are eliminated, feature clustering improves the performance of PSO while reducing the computational cost.

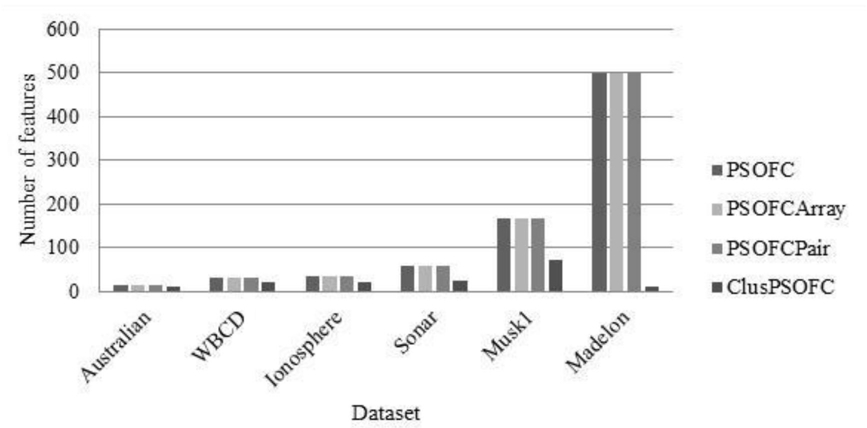


Figure 3. The size of selected features.

Computational time

Figure 4 shows the average computing time for the four methods in 50 independent runs, where time is expressed in seconds. As shown in the figure, the proposed algorithm displayed faster performance compared to the other methods in five out of the six datasets. This resulted from the lower number of features that were produced by the clustering algorithm, whereby all the features were utilised in the other methods. Therefore, the ClusPSOFC

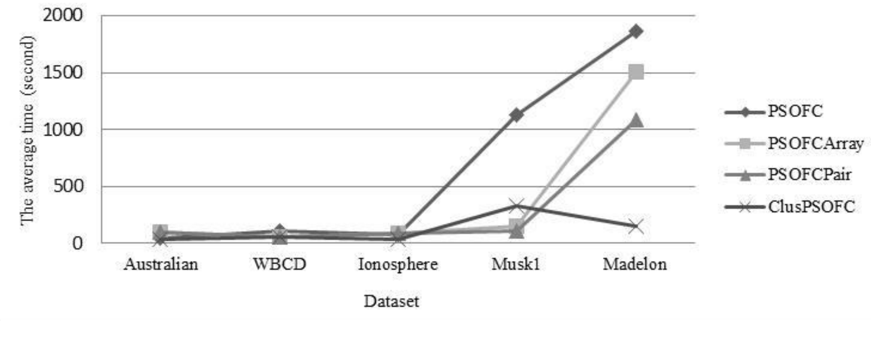


Figure 4. The computational time.

algorithm is very effective at applying a lower number of features that are generated by the clustering algorithm. The complexity of feature space is reduced to a searching space that is smaller, thus minimizing the computational effort to develop the classification algorithm.

In summary, Table 2 and Figures 2 to 4 suggest that feature clustering algorithm is effective in aiding the PSO algorithm to build a new feature that is capable of achieving better classification performance with a shorter computational time, compared to other PSO based FC methods.

Experiment II

This section presents the second set of experiments that are conducted using datasets of large features dimension (ranging between 1095 and 24481). First, the processes of the experiments including the data sets that were used throughout this study were highlighted, and proceeded to the preprocessing of the data sets. Experiment 2 is conducted to investigate the performance of the ClusPSOFC method against the GPFC method. GPFC method was employed as a comparison tool due to its wide application that utilises an algorithm for feature construction, especially in high dimensional data. In addition, this method creates a tree based representation that deals with function operations easily and directly. The tree based representation of GP is a flexible technique that allows for the construct of new features with higher discriminating power.

Experimental design

In this experiment, the performance of the proposed algorithm is compared with GPFC (Tran, Xue, & Zhang, 2016a), which is proven to achieve promising results in FC. For this purpose, six gene datasets which are; Colon, CNS, Ovarian, Breast, Alizadeh and Yeoh were used to examine the performance of the algorithms, where the characteristics of these datasets are presented in Table 1. The first four datasets are downloaded from http://csse.szu.edu.cn/staff/zhuzx/Datasets.html, while the last two datasets are obtained from Tran, Xue, and Zhang (2016). The comparisons between the two methods (i.e., GPFC and ClusPSOFC) are based on the average classification accuracy of the test set, which was calculated using K-Nearest Neighbour (5KNN). Both algorithms were put through 30 independent runs for every dataset, and a new feature was constructed from every run. A statistical significance test (t-test) was carried out to determine the significance of the results produced by the two methods. A 95% significance interval for the T-tests was set. The '+' or '-' operators indicated that the proposed method either showed a significantly better or worse classification performance than the other techniques, while the '=' operator indicated that both methods showed similar performances. In general, the higher the number of '+' operators, the better the performance of ClusPSOFC.

From Table 1, the microarray dataset consists of a lower sample number than the number of features, thus, becoming more challenging. Another challenge on the application of such datasets is that they generally consist of imbalanced data, wherein the sample distribution between the classes are not uniform. Therefore, to conduct a thorough data analysis, discretization was implemented due to the high level of noise generated while collecting data. Initially, every feature was required for normalisation to possess a 0 mean and unit variance. Thereafter, the features were discretised into 3 values (-2, 0 and 2) to remove noise from the data; as performed in Ding and Peng (2005), Fayyad and Irani (1993), Tran, Xue, and Zhang (2016). These values represent three states, which are under-expression, baseline, and over-expression of a gene. Specifically, a feature value will be transformed to 0 if the value lies in the interval of $[(\mu-\sigma)/2, (\mu+\sigma)/2]$, wherein (μ) and (σ) represent the mean and standard deviation respectively, for each feature values. Furthermore, the feature value will be transformed to 2 and -2, if the feature lies either to the right or left of the interval, respectively.

Results and discussions

Table 3 records the average test accuracy obtained for the different datasets by the proposed approach, ClusPSOFC and GPFC. From Table 3, 'All' refers to the original features, 'CF' refers to the constructed features, and 'CFOrg' refers to the combination of the constructed and all original features. Furthermore, 'B', 'A', and 'Std' refer to the best, average, and standard deviation of the accuracy respectively that are computed by KNN from 30 independent runs. 'T' denotes the T-test results for the proposed approach when compared to the GPFC approach. The best results for every datasets were highlighted in bold.

ClusPSOFC versus All features

Table 3

Classification accuracy of GPFC and ClusPSOFC on the tested data sets using KNN classifier

Dataset	Method	#F	GPFC			ClusPSOFC	
			B-KNN	A±Std-KNN	T	B-KNN	A±Std-KNN
Alizadeh	All	1095	77.00	77.00±0.00	+	**83.33**	**83.33±0.00**
	CF		86.00	77.88 ±5.53	+	**91.67**	**91.67±0.00**
	CFOrg		77.00	77.00±0.00	+	**83.33**	**83.33±0.00**
Colon	All	2000	**74.28**	**74.28 ± 0.00**	-	65.38	65.38±0.00
	CF		79.28	**71.40 ± 4.46**	-	**88.46**	65.58±0.10
	CFOrg		74.28	**74.28 ± 0.00**	-	**82.69**	63.85±0.08

(continued)

Dataset	Method	#F	GPFC			ClusPSOFC	
Yeoh	All	2526	89.97	89.97±0.00	+	**100**	**100±0.00**
	CF		99.17	97.04 ±1.01	+	**100**	**100±0.00**
	CFOrg		89.97	89.97±0.00	+	**100**	**100±0.00**
CNS	All	7129	56.67	56.67 ± 0.00	+	**87.5**	**87.5 ±0.00**
	CF		70.00	57.56 ± 5.87	+	**81.25**	**60.13±0.09**
	CFOrg		56.67	56.67 ± 0.00	+	**81.25**	**67.24±0.08**
Breast	All	24481	57.78	57.78 ± 0.00	+	**68.42**	**68.42 ±0.00**
	CF		70.78	60.59 ± 5.37	=	**78.95**	60.35±0.12
	CFOrg		57.78	57.78 ± 0.00	+	**78.95**	**62.98±0.11**
Ovarian	All	15154	91.28	91.28± 0.00	+	**100**	**100±0.00**
	CF		**99.62**	**97.86±01.22**	**-**	97.06	90.00±4.77
	CFOrg		91.28	91.28±0.00	+	**100**	**98.53±1.74**

From Table 3, the application of the feature constructed by ClusPSOFC approach resulted in similar or better performance than the application of 'All' original features. The highest improvement observed in the Alizadeh dataset was on average, 8.34 %, while the highest improvement in the Colon dataset is 23.08%. Furthermore, by combining the constructed and original features, the classification performance improved significantly. Moreover, the constructed features combined with the original features were more accurate than just the application of the original features. The Colon and Breast datasets revealed a maximal increment in the best accuracy of 17.31% and 10.53% respectively, after the constructed feature and original features were combined.

ClusPSOFC versus GPFC

In comparison to the GPFC, the ClusPSOFC method assisted the KNN to obtain better results in 5 out of 6 datasets. The highest average accuracy value of 13.79% was observed when the constructed feature was used for the Alizadeh dataset. Furthermore, the proposed method achieved 100% classification accuracy for the Yeoh dataset in all cases, and two cases for the Ovarian dataset. On the Colon data, although ClusPSOFC was 5.82% lower than GPFC on average, its best accuracy was still 9.18% higher. Therefore, in comparison to the GPFC method, the proposed ClusPSOFC method had achieve 24 successful comparisons, 5 unsuccessful comparisons, and 1 unchanged. These results suggested that by eliminating irrelevant and redundant features using the clustering algorithm, the constructed feature will improve the performance, compared to those that are constructed from whole features. Furthermore, the proposed ClusPSOFC method is able to determine the hidden information present in the original features. This is due to the changes in the method of representation of the original data and has been proven to be advantageous. Moreover, the combination of low level features can improve the classification quality, while the PSO can be employed as a search technique to construct a new feature from those combinations that would lead to better performance. The use of the clustering algorithm allows for the selection of informative features from the originals, that are then used as inputs to the ClusPSOFC method to enable the construction of the new features. Hence, a more reliable approach is produced.

To further validate the significant difference between ClusPSOFC and GPFC, a statistical test (t-test) with a 95% significance interval was used. From Table 3, it can be noted that there is a significant difference between both methods in terms of classification accuracy. Therefore, it can be concluded that the ClusPSOFC approach is better than GPFC in solving FC problems.

Performance of the ClusPSOFC for the Selected features

Figure 5 presents the number of features used by ClusPSOFC and GPFC. Although both approaches construct only one feature, the proposed algorithm used a lower number of features than GPFC. Table 4 shows the original number of features and the number of clusters produced using the RFC clustering algorithm, along with the reduction rate for every dataset. As observed in the third column of Table 4, the dimensionality of every dataset is significantly reduced after the feature clustering algorithm was applied. The number of features that were applied to the PSO decreased with the highest reduction rate of 98% on the Yeoh dataset, and 97% on the CNS and Breast datasets. Moreover, the results show the variations in the number of clusters (features) produced by the different datasets, compared to the original feature number. For example, the Yeoh dataset produced a lower number of clusters compared to the Alizadeh dataset, despite its dimensionality being two times larger than that in the Alizadeh dataset. Although the Yeoh dataset generated a smaller number of clusters, its classification accuracy (presented in Table 3) suggests that with the help of the clustering algorithm, the constructed feature performed better than the feature constructed by GPFC that used all the original features.

Based on the results in Figure 5 and Table 3, ClusPSOFC approach used a smaller number of features and achieved better classification performance than GPFC, on almost all datasets. The significant enhancement achieved

Table 4

Reduction rate achieved using ClusPSOFC

Dataset	#Features	#Clusters	Reduction rate
Alizadeh	1095	126	0.88
Colon	2000	103	0.95
Yeoh	2526	49	0.98
CNS	7129	230	0.97
Breast	24481	657	0.97
Ovarian	15154	1228	0.92

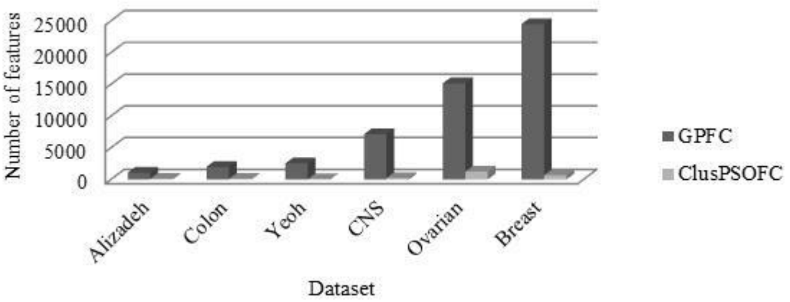


Figure 5. The size of Selected Features.

by ClusPSOFC over GPFC in selecting a small number of features with high accuracy is attributed to the use of the RFC clustering algorithm. The RFC was able to select the highest effectual features (based on the SU and CC evaluation) when initialising PSO individuals. The results of this study conforms with previous literature that suggests integrating a filter (SU and CC) with wrapper (PSO) into a singular method which produced good results and better performance, compared to the use of the wrapper method alone. Additionally, as observed in Table 4, Figure 3 and Figure 5, the RFC algorithm reduced the dimensionality of all 12 datasets to smaller sizes, by removing the redundancy between features. This was observed within the Madelon data that showed extreme reduction due to the large amounts of redundant features in the data (Yang, He, & Shao, 2013). The removal of the redundant features were presumed to have been performed by the RFC algorithm. Subsequently, the number of original features in this data was large compared to the other low dimensional datasets, which made it difficult to compare the two different sets. This rational relates to the breast cancer data; although the gene expression data has a huge number of features, most of the features were irrelevant and redundant, and only a small number of features were relevant to the problem. As shown in Table 4, all the datasets obtained at least 84% of the reduction rate after employing the clustering algorithm, with the largest reduction rate at 98%. On the other hand, in the study of low dimensional data, the smallest reduction rate is 14% while the largest is 97%. Thus, a dataset that contains a large number of redundant features will result to a lower number of clusters formed (the reduction rate will be higher), and vice versa (Tran, Xue, & Zhang, 2016a). Therefore, this paper concludes that high dimensional data can be exploited through the application of the RFC algorithm than low dimensional data, due to the large number of redundancies in the data.

Computational time

Figure 6 summarises the average computing time of both ClusPSOFC and GPFC methods over 30 independent runs, where time is expressed in minutes. According to Figure 6, the ClusPSOFC approach preformed faster than the GPFC method, in 3 out of 4 datasets. The computational time became faster due to the small number of features that were selected by ClusPSOFC. Furthermore, the results asserted the importance of the clustering algorithm in selecting smaller feature subsets and enabling the PSOFC to attain better results within an acceptable time. The RFC clustering algorithm is comprised of two filter measures; SU and CC, which are computationally less expensive and accelerates the process to construct new features. The two datasets; Yeoh and Alizadeh were not compared since their computational costs were not recorded in Tran, Xue, and Zhang (2016a), and Tran, Xue, and Zhang (2017).

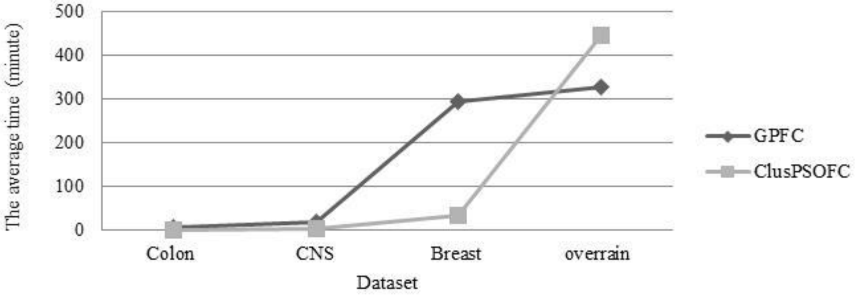


Figure 6. The computational time.

Further analysis of RFC algorithm

To further validate the efficiency and importance of the ClusPSOFC method, the features that were selected by the RFC clustering algorithm for feature construction are visually presented. Three different examples, i.e. a small-sized dataset consisting of 1095 genes (Alizadeh), a medium-sized dataset consisting of 7129 genes (CNS), and a larger dataset consisting of 15154 genes (Ovarian) were chosen. The scatter plots of the features selected by the RFC algorithm on the three datasets are shown in Figure 7. Based on the plots in Figure 7, it is clear that the selected features by the RFC algorithm are acceptable as the instances related to them are grouped in separate clouds. Each of these separated clouds represent one class, while the two classes are discriminated from each other. This could be attributed to the two effective measures (symmetrical uncertainty and correlation coefficient) that were applied to detect the most informative features in each dataset. From the results of the plots, the CNS dataset shows overlapping between the instances as features from different clusters can be highly correlated, but with a threshold value that is lesser than 0.9. However, the proposed method that utilized the clustering algorithm have shown better performance, than GPFC that used all the features of the dataset.

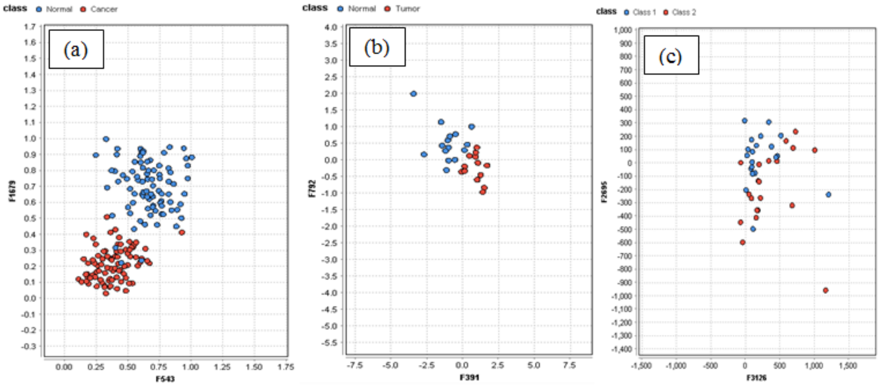


Figure 7. Distribution of two representative features selected from the features clustered by RFC method for ovarian dataset (a), Alizadeh dataset (b), and CNS dataset (c).

CONCLUSION AND FUTURE WORK

This work is the first study that utilized feature clustering with PSO for feature construction in the classification of high dimensional data. The method was designed to implement a feature clustering algorithm that clusters redundant features according to the correlation or redundancy threshold. Once the cluster algorithm obtains the best feature from each cluster, it is applied as an input for PSOFC to construct a new feature. The applicability of the proposed method was reinforced as it reduced the number of features, improved the accuracy of the approach, and lowered the computational cost. The results from this work emphasized the importance of a clustering algorithm as a type of pre-processing technique that can help obtain better results without using the whole feature set. The experiments demonstrated that the proposed method had shown improvements in lowering the dimension of features and enhancing the classification performance, compared to other PSO-based FC methods and standard GP based FC method. As the PSO was used to create a single new feature, future works can be conducted on a multiple features study that applies the PSO method to examine further improvements in the classification performance.

ACKNOWLEDGEMENT

The authors would like to thank the Faculty of Information Science and Technology - Universiti Kebangsaan Malaysia for supporting this research under Grant number FRGS/1/2016/ICT02/UKM/01/2.

REFERENCES

Ahmed, S., Zhang, M., Peng, L., & Xue, B. (2014). Multiple feature construction for effective biomarker identification and classification using genetic programming. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 249-256. doi:10.1145/2576768.2598292

Ahmed, S., Zhang, M., Peng, L., & Xue, B. (2016). A Multi-objective genetic programming biomarker detection approach in mass spectrometry data. *In European Conference on the Applications of Evolutionary Computation*, 106-122. doi.org/10.1007/978-3-319-31204-0_8

Banka, H., & Dara, S. (2015). A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation. *Pattern Recognition Letters*, *52*, 94-100. doi.org/10.1016/j.patrec.2014.10.007

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16-28. doi.org/10.1016/j.compeleceng.2013.11.024

Chen, Q., Zhang, M., & Xue, B. (2017). Genetic programming with embedded feature construction for high-dimensional symbolic regression. *Intelligent and Evolutionary Systems*, 87-102. doi.org/10.1007/978-3-319-49049-6_7

Dai, Y., Xue, B., & Zhang, M. (2014). New representations in PSO for feature construction in classification. In *European Conference on the Applications of Evolutionary Computation*, 476-488. doi.org/10.1007/978-3-662-45523-4_39

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, *1*(3), 131-156. doi.org/10.3233/IDA-1997-1302

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, *3*(02), 185-205. doi.org/10.1142/S0219720005001004

Elola, A., Del Ser, J., Bilbao, M. N., Perfecto, C., Alexandre, E., & Salcedo-Sanz, S. (2017). Hybridizing cartesian genetic programming and harmony search for adaptive feature construction in supervised learning problems. *Applied Soft Computing*, *52*, 760-770. doi.org/10.1016/j.asoc.2016.09.049

Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *13th International Joint Conference on Artificial Intelligence*, 2, 1022-1027. doi.org/hdl.handle.net/2014/35171

Frank, A., & Asuncion, A. (2010). UCI machine learning repository.

García, D., González, A., & Pérez, R. (2011). A two-step approach of feature construction for a genetic learning algorithm. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, 1255-1262. doi**:** 10.1109/FUZZY.2011.6007576

Gunasundari, S., Janakiraman, S., & Meenambal, S. (2016). Velocity bounded boolean particle swarm optimization for improved feature selection in liver and kidney disease diagnosis. *Expert Systems with Applications*, *56*, 28-47. doi.org/10.1016/j.eswa.2016.02.042

Gupta, A., Gupta, A., & Sharma, K. (2016, March). Clustering based feature selection methods from fMRI data for classification of cognitive states of the human brain. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 3581-3584.

Hall, M. A., & Smith, L. A. (1998). Practical feature subset selection for machine learning. In *Computer science'98 proceedings of the 21st Australasian computer science conference ACSC*, 98, 181-191. doi.org/hdl.handle.net/10289/1512

Hsu, H. H., & Hsieh, C. W. (2010). Feature Selection via Correlation Coefficient Clustering. *JSW*, *5*(12), 1371-1377. doi.org/10.4304/jsw.5.12.1371-1377

Hu, L., Gao, W., Zhao, K., Zhang, P., & Wang, F. (2018). Feature selection considering two types of feature relevancy and feature interdependency. *Expert Systems with Applications*, *93*, 423-434. doi.org/10.1016/j.eswa.2017.10.016

Jabbar, A. M., Ku-Mahamud, K. R., & Sagban, R. (2018). Ant-based sorting and ACO-based clustering approaches: A review. In *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 217-223. dio:10.1109/ISCAIE.2018.8405473

Jain, I., Jain, V. K., & Jain, R. (2018). Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. *Applied Soft Computing*, *62*, 203-215. doi.org/10.1016/j.asoc.2017.09.038

Jaskowiak, P. A., & Campello, R. J. (2015). A cluster-based hybrid feature selection approach. In *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, 43-48. doi.org/ 10.1109/BRACIS.2015.14

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks, 4*, 1942–1948. doi.org/10.1007/978-0-387-30164-8_630

Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, *5*, 4104-4108. doi.org/10.1109/ICSMC.1997.637339

Kumari, P., Rajeswari, K., & Vaithiyanathan, V. (2015). Correlation and clustering based efficient feature subset selection. *Journal of Engineering Technology*, *3*, 135-144.

Mahanipour, A., & Nezamabadi-pour, H. (2017). Improved PSO-based feature construction algorithm using feature selection methods. In *2017 2nd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*,1-5. dio : 10.1109/CSIEC.2017.7940173

Mahanipour, A., Nezamabadi-pour, H., & Nikpour, B. (2018, March). Using fuzzy-rough set feature selection for feature construction based on genetic programming. In *2018 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, 1-6. dio: 10.1109/CSIEC.2018.8405407

Moradi, P., & Rostami, M. (2015). Integration of graph clustering with ant colony optimization for feature selection. *Knowledge-Based Systems*, *84*, 144-161. doi.org/10.1016/j.knosys.2015.04.007

Muharram, M. A., & Smith, G. D. (2004). Evolutionary feature construction using information gain and gini index. In *European Conference on Genetic Programming* (pp. 379-388). Springer, Berlin, Heidelberg. doi.org/10.1007/978-3-540-24650-3_36

Neshatian, K., Zhang, M., & Andreae, P. (2012). A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. *IEEE Transactions on Evolutionary Computation*, *16*(5), 645-661. doi.org/10.1109/TEVC.2011.2166158

Nguyen, H. B., Xue, B., Liu, I., & Zhang, M. (2014). PSO and statistical clustering for feature selection: A new representation. In *Asia-Pacific Conference on Simulated Evolution and Learning* (pp. 569-581). Springer, Cham. doi.org/10.1007/978-3-319-13563-2_48

Roth, V., & Lange, T. (2004). Feature selection in clustering problems. In *Advances in neural information processing systems* (pp. 473-480).

Rutkowski, L. (2008). Computational intelligence: Methods and techniques. *Springer Science & Business Media*. dio: 10.1007/978-3-540-76288-1

Sahu, B., & Mishra, D. (2012). A novel feature selection algorithm using particle swarm optimization for cancer microarray data. *Procedia Engineering*, *38*, 27-31. doi.org/10.1016/j.proeng.2012.06.005

Sardana, M., Agrawal, R. K., & Kaur, B. (2016). A hybrid of clustering and quantum genetic algorithm for relevant genes selection for cancer microarray data. *International Journal of Knowledge-based and Intelligent Engineering Systems*, *20*(3), 161-173. dio**:** 10.3233/KES-160341

Setiono, R., & Liu, H. (1998). Fragmentation problem and automated feature construction. *Proceedings of 10th International Conference on Tools with Artificial Intelligence (ICTA*), 208-215. IEEE. doi.org/ 10.1109/TAI.1998.744845

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. doi.org/ 10.1109/TKDE.2011.181

Swesi, I. M., & Bakar, A. A. (2017). An enhanced binary particle swarm optimization (EBPSO) algorithm based a v-shaped transfer function for feature selection in high dimensional data. *International Journal of Advances in Soft Computing & Its Applications*, *9*(3), 217-238.

Tariq, H., Eldridge, E., & Welch, I. (2018). An efficient approach for feature construction of high-dimensional microarray data by random projections. *PloS one*, *13*(4), e0196385. doi.org/10.1371/journal.pone.0196385

Tran, B., Xue, B., & Zhang, M. (2016). Genetic programming for feature construction and selection in classification on high-dimensional data. *Memetic Computing*, *8*(1), 3-15. doi.org/10.1007/s12293-015-0173-y

Tran, B., Xue, B., & Zhang, M. (2017). Using feature clustering for GP-based feature construction on high-dimensional data. In *European Conference on Genetic Programming*, 210-226. doi.org/10.1007/978-3-319-55696-3_14

Tran, B., Zhang, M., & Xue, B. (2016). Multiple feature construction in classification on high-dimensional data using GP. In *SSCI*, 1-8. doi.org/10.1109/SSCI.2016.7850130

Vafaie, H., & De Jong, K. (1998). Feature space transformation using genetic algorithms. *IEEE Intelligent Systems and their Applications*, *13*(2), 57-65. doi.org/10.1109/5254.671093

Wong, K. C. (2015). A short survey on data clustering algorithms. In *2015 Second International Conference on Soft Computing and Machine Intelligence (ISCMI)*, 64-68. doi.org/10.1109/ISCMI.2015.10

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, *2*(2), 165-193. doi.org/10.1007/s40745-015-0040-1

Xue, B., Zhang, M., Dai, Y., & Browne, W. N. (2013). PSO for feature construction and binary classification. *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, 137-144. ACM. doi:10.1145/2463372.2463376

Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, *18*, 261-276. doi.org/10.1016/j.asoc.2013.09.018

Yang, Z. M., He, J. Y., & Shao, Y. H. (2013). Feature selection based on linear twin support vector machines. *Procedia Computer Science*, *17*, 1039-1046.

Yazdani, S., Shanbehzadeh, J., & Hadavandi, E. (2017). MBCGP-FE: A modified balanced cartesian genetic programming feature extractor. *Knowledge-Based Systems*, *135*, 89-98. doi.org/10.1016/j.knosys.2017.08.005

Zhang, Y., Gong, D. W., Sun, X. Y., & Guo, Y. N. (2017). A PSO-based multi-objective multi-label feature selection method in classification. *Scientific reports*, *7*(1), 376. doi.org/10.1038/s41598-017-00416-0