Neural Computing and Applications (2019) 31:4287–4303
https://doi.org/10.1007/s00521-017-3317-9

ORIGINAL ARTICLE

Efficient feature selection and classification algorithm based on PSO and rough sets

Ramesh Kumar Huda[1] · Haider Banka[1]

Received: 6 March 2017 / Accepted: 28 December 2017 / Published online: 17 January 2018

Abstract
The high-dimensional data are often characterized by more number of features with less number of instances. Many of the features are irrelevant and redundant. These features may be especially harmful in case of extreme number of features carries the problem of memory usage in order to represent the datasets. On the other hand relatively small training set, where this irrelevancy and redundancy makes harder to evaluate. Hence, in this paper we propose an efficient feature selection and classification method based on Particle Swarm Optimization (PSO) and rough sets. In this study, we propose the inconsistency handler algorithm for handling inconsistency in dataset, new quick reduct algorithm for handling irrelevant/noisy features and fitness function with three parameters, the classification quality of feature subset, remaining features and the accuracy of approximation. The proposed method is compared with two traditional and three fusion of PSO and rough set-based feature selection methods. In this study, Decision Tree and Naive Bayes classifiers are used to calculate the classification accuracy of the selected feature subset on nine benchmark datasets. The result shows that the proposed method can automatically selects small feature subset with better classification accuracy than using all features. The proposed method also outperforms the two traditional and three existing PSO and rough set-based feature selection methods in terms of the classification accuracy, cardinality of feature and stability indices. It is also observed that with increased weight on the classification quality of feature subset of the fitness function, there is a significant reduction in the cardinality of features and also achieve better classification accuracy as well.

Keywords Feature selection · Rough sets · New quick reduct · Inconsistency handler · Classification · Fitness function · Particle Swarm Optimization

1 Introduction

Feature selection and classification are two important tasks in the fields of machine learning, pattern recognition and data mining [1]. Feature selection is the process of choosing a feature subset by eliminating irrelevant and redundant features from the given original feature set to form the pattern in a dataset. The selected subset should be sufficient to describe the target class with higher accuracy [2]. To handle imprecise and inconsistent information (i.e., noisy, irrelevant and relevant) in the real-world task, there is a need of feature selection [3]. Rough sets [4–6] can handle uncertainty and vagueness, discovering patterns in inconsistent data. It is a useful feature selection method in pattern recognition [7], in which selected feature subset can be predict the target concepts and the original feature set as well. The aim for feature selection based on rough set is to find minimal reduct with high classification accuracy according to the selected feature subset [7–9]. The advantages of feature selection are to reduce the dimensionality, computational complexity and search space for classification algorithms, and improve the classification performance [10].

There are two important part of feature selection methods as evolution criteria and search strategy. Based on evaluation criteria, the feature selection methods are classified into wrapper approaches, filter approaches and

✉ Ramesh Kumar Huda
ramesh.hudda@cse.ism.ac.in

Haider Banka
banka.h.cse@ismdhanbad.ac.in

[1] Indian Institute of Technology (ISM), Dhanbad, India

hybrid approaches [11]. Wrapper approach incorporates a learning algorithm as a part of the evaluation procedure, while filter approach does not. Therefore, wrappers approach can regularly accomplish the preferred outcomes over filter approach [12], but its computational cost is high. Filter approach is computationally less expensive and more general than wrappers approach, but appropriate evaluation criteria is needed for filter approach. Hybrid approach uses the independent measure to decide the best subsets for a given cardinality and uses the mining algorithm to select the final best subset among the best subsets [12]. Based on search strategy, feature selection is a difficult problem mainly due to the large search space, which increases exponentially with respect to available number of features [13]. Therefore, an exhaustive search is practically impossible in most of the situations. Several heuristic search techniques have been applied for feature selection as greedy search-based techniques [2]. However, most of the existing algorithms are still stuck at local optima or being computationally costly [14]. So, an efficient global search technique like Evolutionary Computing (EC) is introduced for better addressing the feature selection problem. EC technique has been applied for feature selection problems such as Genetic Algorithms (GAs) [15, 16], Genetic Programming (GP) [17], Ant Colony Optimization (ACO) [18], Particle Swarm Optimization (PSO) [19]. PSO is a relatively recent EC technique, which is computationally less costly and gives better result than some other EC algorithms [20, 21].

In this paper, an efficient feature selection method is proposed that explores how PSO and rough set techniques can be viable to discover optimal reduct. Kennedy and Eberhart [22, 23] proposed evolutionary computation technique like PSO. It mimics the behavior of flying birds and their means of information exchange to solve optimization problems [24]. It is especially alluring for feature selection, in which particle swarms will discover the best feature subset as they fly within the problem space. PSO with rough set has been successfully applied to find the reduced feature subset from original feature set, and results demonstrate that it beats some conventional and EC-based existing feature selection methods as far as features cardinality, classification accuracy and computational cost [25]. The performance of the proposed method is computed on six datasets. It can be observed that PSO and rough set have strong search capability in problem space and can discover quick reduct.

1.1 Objective and contribution

This paper aims to develop a feature selection and classification algorithm with the expectation of selecting a small feature subset while achieving higher classification accuracy. This has been achieved with rough set and PSO as described in Algorithm 3 (EPSORSNA). The proposed method is examined on nine benchmark datasets with different numbers of features, classes and instances. Specifically, with the following considerations:

(a) To develop a new quick reduct algorithm using rough set theory for handling redundant and noisy features (i.e., Algorithm 1).
(b) To develop an inconsistency handler algorithm for handling the inconsistency in datasets using rough set theory (i.e., Algorithm 2).
(c) To develop a fitness function by considering three parameters such as classification quality of feature subset, remaining features and accuracy of approximation (i.e., Eq. 16).
(d) To develop an efficient feature selection and classification method based on rough sets and PSO in high-dimensional data (EPSORSNA) (i.e., Proposed Algorithm 3).
(e) To investigate whether this proposed algorithm can perform better on reduced feature subsets than the whole feature sets and existing algorithm as well.
(f) To investigate whether there is an significant effect on results for tunable parameters of the algorithm using different weights for parameter of fitness function, so that the performance may increase to some extent (i.e., cardinality of feature subsets and/or classification accuracy).
(g) To investigate the effect on the performance of stability indices for assigning different weights as mentioned above.

The rest of this paper is structured as follows. Section 2 describes the fundamentals of rough set theory, PSO, stability indices and related works on feature selection. Existing feature selection methods based on PSO and rough set and proposed approaches are introduced in Sect. 3. The effectiveness of the proposed method and comparison with other existing methods are demonstrated in Sect. 4. Finally, Sect. 5 shows the conclusions and future work.

2 Preliminaries

2.1 Rough set theory

Rough set theory (RST) [4] is a mathematics based approach used to handle imprecision, vagueness and riskiness. Every instance (object) of universe has some information in an information system. Objects characterized by the similar information are indiscernible according to the present information about them. Any union of elementary sets (any set of indiscernible objects) are known as crisp

set, and other than crisp set is rough (imprecise, vague). Vague concepts cannot be categorized according to information they are having. A rough set is the approximation of a vague concept based on two basic concepts, known as lower and upper approximation of set.

The fundamental favorable position of rough set is that it need not bother with any earlier information about the data. Feature selection is performed in rough set using only the granularity structure of the data [5].

Let Z = (U, l, M, N) be an information system, where universe (U) is a non-empty finite set of objects, M, N ⊂ L, and they are known as condition and decision attributes and L is a non-empty finite set of attributes, respectively [6]. For $\forall_a \in$ L determines a function $F_a : U \longrightarrow V_a$. If T⊆ L, there is an associated equivalence relation:

$$IND(T) = \{(g,h) \in U * U | \forall_a \in T, F_a(g) = F_a(h)\}. \quad (1)$$

If two objects in U satisfy IND(T), they are indiscernible with respect to T. The IND(T) equivalence relation originates a partition of U denoted by U/T, which originates the concept of the equivalence classes. The equivalence class of U/T containing g is given by $[g]_P = [g]_L =$ h ∈ U | (g;h) ∈ IND(T). The equivalence classes are the basic blocks to construct rough set approximations. For D ⊂ U, a lower approximation(TD) and an upper approximation ($\bar{T}$D) of D with respect to IND(T) are defined as follows [10, 25].

$$\underline{T}D = \{g \in U \mid [g]_T \subseteq D\}, \quad (2)$$

$$\bar{T}D = \{g \in U \mid [g]_T \cap D \neq \varnothing\}, \quad (3)$$

where TD and $\bar{T}$D represent those objects which are surely belong to the target set D, and the objects which are surely or probably belong to the target set D, respectively. The C-positive region of N is the set of all objects from the universe U which can be classified with certainty to classes of U/N employing attributes from M, i.e., Let M,N ⊆ L be equivalence relation over U, and then the positive, negative and boundary region can be defined as:

$$POS_M(N) = \bigcup_{D \in U/N} \underline{M}D.$$

$$NEG_M(N) = U - \bigcup_{D \in U/N} \bar{M}D$$

$$BND_M(N) = \bigcup_{D \in U/N} \bar{M}D - \bigcup_{D \in U/N} \underline{M}D.$$

The M-positive region of N is the set of all objects from the universe U which can be classified with certainty to classes of U/N employing attributes from M, i.e., where $\underline{M}D$ denotes the lower approximation of the set D with respect to M, i.e., the set of all objects from U that can be with certainty classified as elements of D based on attributes from N. Rough set reduct can be found by using the degree of dependency [25]. The dependency function calculates the approximating power of a feature set (i.e., Eq. 4).

$$\gamma_M(N) = \frac{|POS_M N|}{|U|}. \quad (4)$$

Dispensable and indispensable features: Let m ∈ M, if $POS_{M-m}(N) = POS_M(N)$, then a feature m is dispensable in Z; otherwise, feature m is indispensable in Z. If m is an indispensable feature, deleting it from Z will cause Z to be inconsistent. Z is independent if all m ∈ M are indispensable.

Reduct: A set of features R ⊆′ M is called a reduct of M, if $Z^{'}$ = (U, L, M, N) is independent and $POS_R(N) = POS_M(N)$. In other words, a reduct is the minimal feature subset that follows the above condition.

Core: The set of all the features indispensable in M is denoted by Core(M), in which Core(M) = ⋂ Red(M) where Red(M) is the set of all reduct of M.

An ordered pair (TD, $\bar{T}$D) is known as an rough set. A reduct is the imperative piece of Z = (U, L) (rough set), which can able to achieve same approximation power of classification like as original feature set L. There could be a wide range of reduct, but the goal of feature selection using RST is to eliminate redundant and irrelevant attributes to search for the reduced reduct. Therefore, researchers explore the probabilistic RST to relax the definitions of the lower and upper approximation [25]. The lower estimate is re-imagined as Eq. (5), where μ_T[d] demonstrates is characterized as an approach to compute the fitness of a given instance d∈D shown in Eq. (6).

$$\underline{\text{apr}}_T D = \{d \mid \mu_T[d] \geq \alpha\}, \quad (5)$$

where

$$\mu_T[d] = \frac{|\,[D]_T \cap D\,|}{|[D]_T|}, \quad (6)$$

where α can be fixed to restrict or loosen up the lower approximation. In the event that most number of objects (D) are in the goal set yet a little number are not in a given equivalence class, it can incorporate them in the lower approximation. $\underline{\text{apr}}_T[D]$ at the point when α = 1.

According to the theoretical perspective, Yao and Zhao [25] suggest that RST can be a better approach for feature selection tasks. However, it has not proven experimentally.

2.2 Particle swarm optimization

For D-dimensional search space and N particles, let X be the particle of the population, pbest is the personal information or self-best solution obtained so far, gbest is the

best solution obtained by the particle population so far, and V be the velocities of the particles. X and V are represented by NxD matrix, and pbest and gbest are represented by 1xD vector.

For the initialization of the particle population, Eq. (7) is used.

$$X_{i,j}(0) = L_j^{\min} + r_i^j * (U_j^{\max} - L_j^{\min}), \tag{7}$$

where $i = 1, 2, \ldots, N$; $j = 1, 2 \ldots, D$, $L_j^{\min}$ is the lower bound of the search space for j^th dimension, $U_j^{\max}$ is the upper bound of the search space for jth dimension and r_i^j is a random number produced for each particle for each dimension, in range of [0,1].

$$V_{i,j}(0) = [L_j^{\min} + r_i^j * (U_j^{\max} - L_j^{\min}) - X_{i,j}(0)]/2, \tag{8}$$

where $i = 1, 2, \ldots, N$ and $j = 1, 2 \ldots, D$. The initialization of velocities of the particles depends on both the upper and lower bounds of search space and the current particle positions (Eq. (8) [26]). In the initialized stage, the current particle positions are assigned as self-best solution (pbest) of the particles.

The best solution of the population in the initialized phase is determined using Eq. (9).

$$gbest = Bestf(anypbest), \quad \text{where} \quad i = 1, 2, \ldots, N. \tag{9}$$

PSO is an iterative algorithm. So, Eqs. (9–12) are executed repeatedly until a pre-determined termination is met.

$$X_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} i = 1, 2, \ldots, N \quad \text{and} \quad j = 1, 2, \ldots, D \tag{10}$$

$$V_{i,j}^{t+1} = w * x_{i,j}^t + c_1 * r_{i,j}^1 * (pbest_{i,j}^{(t)} - x_{i,j}^t) + c_2 * r_{i,j}^2 * (gbest_j^{(t)} - x_{i,j}^t) \tag{11}$$

$$pbest_{t+1}^i = \begin{cases} X_i^{t+1} & \text{if } f(X_i^{(t+1)} \text{better than } pbest_i^{(t)}) \\ pbest_i^{(t)} & \text{otherwise} \end{cases} \tag{12}$$

In Eq. (10), w is inertia weight [21] and it is not in the basic PSO algorithm but it is used in all contemporary versions of PSO algorithm.

2.3 Stability indices

The stability indices are also important characteristics for feature selection methods, in which the relevant features should not change for different samples of data, when the target concept of datum is fixed. There are several stability calculation methods for the purpose of calculating stability indices for feature selection methods, and these methods are categorized according to index based, rank based and weight based [8].

So, here we are going to discuss some common stability indices methods like as Dice, Tanimoto and Jaccards indices:

(a) ***Dices coefficient***: Dice index is used to evaluate the overlap value between two feature sets, and it takes the value between 0 and 1, where 1 means both feature sets are identical and 0 means no overlapping. The dice index between two feature sets L_1 and L_2 is given by:
Dice(L_1',L_2')= $(2|L_1' \cap L_2'|)/(|L_1' \cup L_2'|)$.

(b) ***Tanimoto index and Jaccards index***: They calculate the value of overlap between two feature sets the range of overlap is same as dice index: Jaccard (L_1',L_2') = $(2|L_1' \cap L_2'|)/(|L_1' \cup L_2'|)$, Tanimoto($L_1$',$L_2$') = $(|L_1'|+|L_2'|-2|L_1' \cap L_2'|)/(|L_1'|+|L_2'|-2|L_1' \cup L_2'|$. In general, all three stability measurement methods behave same in all cases. But, it notice that dice index gives result slightly better than other with respect to the intersection between two feature subsets and set of different feature subsets. All three takes number of features into account rather than dimensionality d into account [27].

2.4 Related work on feature selection

2.4.1 Traditional feature selection methods

Hall [28] proposes the method based on correlation between attributes and class labels for feature selection (Cfs). Filter algorithm, FOCUS [29] is based on the concept of exhaustive search, which means it examines all possible feature subsets and then selects smaller feature subset, which is very costly. Two generally used methods as Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) are based on the concept of greedy search. In SFS [30], once a feature is added, it cannot be deleted later on, and in SBS [31], once a feature is deleted, it cannot be added later on. However, this causes the problem of the so-called nesting effect. Stearns [32] proposed a method "plus-l-take away-r" for overcoming the problem of nesting effect in which there were l times forward selection and r times backward selection. However, finding the optimal value for (l,r) is a difficult problem.

2.4.2 Evolutionary computing methods for feature selection

EC techniques have been used to address feature selection tasks. Based on fuzzy sets, GA, PSO, ACO and GP, Chakraborty proposed a GA [33] and PSO [34]-based

feature selection methods. The comparison shows that PSO outperformed GA method. Based on GP, the multi-objective filter feature selection problem in binary classification was proposed by Kourosh and Zhang [35]. Based on ACO and fuzzy-rough theory, feature selection method for web content classification and complex system monitoring were proposed by Jensen [36]. Mohamed Abd El Azizet et al. [37] proposed the feature selection method based on modified cuckoo search and rough set.

Wang et al. [38] proposed an improved binary PSO and rough set methods for feature selection. Chen et al. [39] proposed an ACO-based rough set approach for feature selection. Cervante et al. [40] proposed dimensionality reduction approach based on PSO and rough set theory. Rman et al. [41] proposed feature selection and recognition using rough set methods. Yang et al. [38] proposed a feature selection based on PSO and rough sets. Xue et al. [11, 42] proposed a multi-objective feature selection method using BPSO and RST. However, the RST for feature selection has not been fully explored for feature selection in terms of classification accuracy, cardinality of feature (reduct) and computational cost. Therefore, the development of efficient method based on PSO and RST for feature selection is still an open issue.

3 Existing and proposed methods

We have proposed an efficient feature selection method using PSO and probabilistic RST (EPSORSNA). To compare the performance of proposed method, three existing feature selection methods are briefly described (PSOPRS, PSOPRSN and PSOPRSE), which provide an idea about proposal method. When using RS for feature selection, a dataset should be represented as an information system $Z = (N, E)$, where E represents a number of attributes or features. According to the equivalence relation described by E, N can be partitioned as N_1,N_2,N_3, . . . ,N_n, where n represents the number of classes in datasets. After performing feature selection, the result is achieved as feature subset P∈E. Therefore, the fitness of P can be calculated by how well P represents each target set in N, i.e., a class in the datasets.

3.1 Existing feature selection methods based on PSO and rough set

- PSOPRS: As discussed in Sect. 2.1, the definition of lower and upper approximation limits the application of standard RST. Therefore, a feature selection method (PSOPRS) based on PSO and probabilistic rough set (PRS) was proposed [40]. In PSOPRS, for target set N_1 in PRS defined by Eq. (6) where $\mu_P[x]$ quantifies the proportion of $[x]_P$ is in N_1; $\underline{\text{apr}}_P N_1$ in Eq. (5) defines the lower approximation of P according to N_1. $[x]_P$ only contains instances in N_1. α can be adjusted to restrict or relax $\underline{\text{apr}}_P N_1$. Therefore, how well P depicted the target class in N can be assessed by Eq. (13), which is the objective function in PSOPRS. Equation (13) essentially measures the number of instances that P correctly makes indistinguishable from others of the same classification.

$$\text{Fitness}(P) = \frac{\sum_{i=1}^{n} |\underline{\text{apr}}_P N_i|}{|N|}. \tag{13}$$

- PSOPRSN: PSOPRS based on probabilistic RST, in which the cardinality of the feature is not considered as a part of fitness function, if two feature subsets have same fitness values, PSOPRS does not necessarily select the smaller one. Therefore, cardinality of feature is added into the fitness function to introduce the new fitness function (new algorithm PSOPRSN)[40], which aims to maximize the representation power of the feature subset and with same time minimize the size of the feature subset, according to Eq. (14).

$$\text{Fitness}(P) = \gamma * \frac{\sum_{i=1}^{n} |\underline{\text{apr}}_p X_i|}{|N|} + (1-\gamma) * \left(1 - \frac{M}{T}\right). \tag{14}$$

where M is the number of selected attributes, T is the total number of attributes, and $\gamma \in [0,1]$ indicates how much importance is given to representation power of the attributes subset, while (1-γ) indicates how much importance is given to the cardinality of feature. When γ = 1.0, PSOPRSN regenerates PSOPRS.

- PSOPRSE: Improving the performance of PSOPRSN, Cervante et al. [11] uses probabilistic RST to develop a new function to minimize the size of reduct, which aims to minimize the number of equivalence classes and maximize the number of instances in each equivalence class. According to these parameters, PSOPRSE is introduced [42], in which Eq. (15) uses as a objective function.

$$\text{Fitness}(P) = \frac{\sum_{i=1}^{n} |\underline{\text{apr}}_p N_i|}{|N|} + \frac{\Sigma_{x \in \text{equivlence classes}} \frac{|x|}{|N|}}{\text{no. of equivalence classes}}. \tag{15}$$

3.2 Supportive proposed algorithms

We have proposed two new algorithms, first algorithm is used for evaluation of quick reduct (feature subset) from

given datasets. Second algorithm is used for handling inconsistency in datasets and it is fruitful for small dataset.

3.2.1 New quick reduct algorithm (NQR)

Nowadays, finding feature reduct of a decision table has been got much more intention in research point of view. The proposed algorithm (i.e., Algorithm 1) calculates the minimal reduct without examine all exhaustive generated subsets. It starts with empty set and then add l features and delete r feature at a time, according to dependency degree, until this procedure its maximum possible value for dataset.

According to the new quick reduct algorithm, first take the subset of l features and then calculate the dependency degree of {R ∪ l}, if dependency degree is greater than dependency degree of previous features of R, then add l with R. Second, take subset of r features from R and then calculate the dependency degree of {R-{r}}, if dependency degree is greater than dependency degree of only R and then delete r from R. However, it is not guaranteed to find minimal feature subset as its to greedy. Using dependency degree to discriminate the features most of the time gives the optimal feature subset.

The proposed New Quick reduct (NQR) algorithm also has been compared with existing feature reduct methods, Supervised Quick Reduct (SQR) [43] and Supervised Relative Reduct [43] in terms of size of reduct. The experimental results and comparison can be seen in Sect. 4.2.2.

Algorithm 1: Find Speedy Reduct(NQR(P,D))

Input : P: Represent the set of all condition feature;
D: Represent the set of decision feature;
l,r: Represents the set of feature for add and delete purpose;
Output: Return $\gamma_R(D)$. where, R is the Reduct

R← {}
Do
M← R
$\forall l \subset (P - R)$
$\gamma_{R\cup\{l\}}(D)=\frac{|Pos_{R\cup\{l\}}(D)|}{|U|}$
if $\gamma_{R\cup\{l\}}(D) > \gamma_M(D)$ **then**
  $R \leftarrow R \cup \{l\}$
end
R← M
T=R
r ⊆ l
R=R-{r}
$\gamma_{R\cup}(D)=\frac{|Pos_R(D)|}{|U|}$
if $\gamma_{R\cup}(D) > \gamma_T(D))$ **then**
  $R \leftarrow R - \{r\}$
end
R=T
While($\gamma_R(D)= \gamma_P(D)$)

3.2.2 Inconsistency handler algorithm

When decisions are inconsistent because of not clear information present in decision table. Therefore, decision makers hesitate to take the clear decision because of inconsistency. These inconsistencies are not taking as simple error or noise. They can create problem at the time of constructing decision makers preference model. The rough set varies good to deal with inconsistency.

According to Algorithm 2, first separate the conflicting instances from tables and than remove the instances with less support according to quality measure of lower and upper approximation.

Algorithm 2: Inconsistency Handler Procedure

Input : $X \subseteq U$;
U=Number of Objects;
D= Decision Variable and $D \in \{0,1\}$;
Output: Return X^1;
X= $X_{D=0}$+ $X_{D=1}$;
Calculate upper and lower approximation $\overline{B}X_{D=1}$, $\overline{B}X_{D=0}$, $\underline{B}X_{D=1}$ and $\underline{B}X_{D=0}$;
Calculate the accuracy of lower and upper approximations $\underline{\gamma_{D=1}}=\frac{|\underline{B}X_{D=1}|}{|X|}$, $\underline{\gamma_{D=0}}=\frac{|\underline{B}X_{D=0}|}{|X|}$, $\overline{\gamma_{D=1}}=\frac{|\overline{B}X_{D=1}|}{|X|}$ and $\overline{\gamma_{D=1}}=\frac{|\overline{B}X_{D=1}|}{|X|}$
if $\underline{\gamma_{D=1}} < \underline{\gamma_{D=0}}$ **then**
  $X^1 = X_{D=0} + \{X_{D=1}$ - $\underline{B}X_{D=1}\}$
end
else
  $X^1 = X_{D=1} + \{X_{D=0}$ - $\underline{B}X_{D=1}\}$
end
OR
if$\overline{\gamma_{D=1}} < \overline{\gamma_{D=0}}$) $X^1 = X_{D=0} + \{X_{D=1}$ - $\overline{B}X_{D=1}\}$
else
  $X^1 = X_{D=1} + \{X_{D=0}$ - $\overline{B}X_{D=1}\}$
end

3.3 Proposed method (EPSORSNA)

In PSOPRSN, cardinality of feature is directly considered in the objective function. By setting the value of γ, it is anticipated that it would choose smaller feature subset with better (equal) or slightly reduced classification accuracy in PSOPRSN. Be that as it may, in PSOPRSN it may be not accomplished as a result of probabilistic nature of RST. In order to solve this problem, PSOPRSE was presented in which the size of equivalence classes and representation power of feature subset into fitness function were considered and a new method was proposed to select reduced reduct. The aim of this fitness function is to minimize the number of equivalence classes and maximize the number of instances in each equivalence class. PSOPRSE can obtain a small reduct with average good classification accuracy, but it not performs well on unseen test dataset every time and takes more computational time.

In order to solve this issue, we proposed EPSORSNA method in which the classification quality of feature subset, the number of features and accuracy of approximation are directly considered in fitness function. By adjusting the values of α, β and γ, we expected to find a smaller feature subset with high classification accuracy. However, this might be achieved by EPSORSNA (i.e., Algorithm 3).

3.3.1 Proposed fitness function

We define the fitness function in Eq. (16):

$$\text{Fitness}(P) = \alpha \times NQR(P,D) + \gamma \times \alpha_P(X) + \beta \times \left(1 - \frac{|R|}{|T|}\right). \tag{16}$$

where $|R|$ is the number of selected features, $|T|$ is the total number of features, NQR(P,D) is the classification quality of conditional attribute set P relative to decision D, which is evaluated according to Algorithm 1 and α_p(X) is the accuracy of approximation and its calculated according to Eq. (17). The α ,β, and γ are three parameters crossroading to the importance of the classification quality of feature subset, number of features and accuracy of approximation, α, β, $\gamma \in \{0,1\}$. where, $(\alpha + \beta + \gamma = 1)$, β = rand(0, $1 - \alpha$) and $\gamma = (1 - \alpha - \beta)$

Accuracy of approximation:

$$\alpha_P(N_i) = \frac{|\underline{P}N_i|}{|\overline{P}N_i|}. \tag{17}$$

where $|\underline{P}X|$ and $|\overline{P}X|$ are lower and upper approximation, respectively. If $\alpha_P(X) = 1$, then it is called crisp set; otherwise, it is called non-crisp set.

Algorithm 3: EPSORSNA Algorithm

Input : M: The swarm size, D: The dimensional
c_1 and c_2,: Positive acceleration constants ;
W: Inertia weight;
V_{max}: Represent max velocity of particles;
Gen_{max}: Represent max generation;
Fit_{max}: Represent highest fitness value;
Output: Compute the performance of the selected feature subset on Test set;
Return best feature subset(gbest);
Return the classification accuracy of Test set;
Divide data into training and test set;
Handle the inconsistency of training set with set of attributes according to Algorithm(2);
Swarm$\{x_{id}, v_{id}\}$= Generate(M); /*Initialization is done by Eq. (6) and (7) on D dimensional*/
pbest(i)=0; i=1.....M, D=1.....S;
gbest=0, Iter=0;
while *Iter*< Gen_{max} *and gbest*<Fit_{max} **do**
  Compute the fitness value of every particle on Training set (According to Eq. (16));
  for *i=1 to M* **do**
    The pbest of particle i is updated by Eq. (12);
    The gbest of particle i is updated by Eq. (9);
  end
  for *i=1 to M* **do**
    for *D=1 to S* **do**
      Velocity of the particle i updated by using Eq. (11);
      Position of the particle i updated by using Eq. (10);
    end
  end
end

In this fitness function, the classification quality of feature subset, cardinality of feature subset and accuracy of approximation have different importance for feature selection task. In our experiment, we take different values of α , β, and γ for finding reduced feature subset with high classification accuracy. The goodness of each position is evaluated by this fitness function. The criteria are to maximize fitness value.

3.3.2 Procedure

In this experiment, we used PSO for selection of optimal feature subset in which the number of feature subsets are there in feature space. Every feature subset represents position in feature space. If N is the number of features, then the total 2^N possible number of feature subsets, and these all are different from each and other with respect to size or number of features contained by feature subset. The optimal position (feature subset) is having a small number of features with high classification accuracy in the given feature space. The procedure for selecting optimal feature subset using proposed Algorithm 3, it finds the reduct set without generating all possible subset. It starts with dividing the dataset into two parts, training set and testing set. Apply step 2 handling the inconsistency in training set, if the training set has less number of features; otherwise, directly go to step 3. In next steps, we consider particle swarm into this feature space, in which every particle is occupied by one position. The particles fly in this feature space and try to find best position. For every iteration, all the particles change their position, communicate with each other and try to find local best and global best according to fitness function (Eq. 16). After that, eventually they should converge on good, possibly optimal position. Therefore, we can say that PSO with rough set has the exploration ability of particle swarms to converge on global optima for discover optimal feature subset.

To apply PSO for optimal feature subset, the below given subsection is important.

3.3.3 Encoding

For applying the proposed method, each particle position is represented as binary string of length N, where N is the total number of attributes. Every bit represents an attribute, the value '1' means the corresponding attributes is selected, while '0' means not selected. For example, if x, y and z are attributes and if the selected random particle is (1, 1, 0), then the attribute subset is (x, y)

3.3.4 Representation of velocity and updating position

Each particle of the PSO is associated with positive velocity within range 1 and Vmax. It indicates the number of particles bits (i.e., feature) change as global best position in particular moment of time. So, velocity of the particle is flying according to the best position of the particle. The difference between two positions of the particle is same as the different bits lie between two particles. An example illustrates as follows:

Let $P_i = [0\,1\,0\,0\,1\,0\,1\,1\,1\,0]$ and $P_{gbest} = [1\,1\,0\,0\,1\,0\,1\,0\,1\,1]$, and then the difference between the current position of the particle and gbest is P_{gbest}-P_i = [1 0 0 0 0 0 0 −1 0 1]. '1' means that, this bit compared with gbest, this feature (bit) should be selected but it is not, which will decrease classification performance. On the other hand, '-1' means that , this bit compared with gbest, this feature (bit) should not be selected but it is. Both the cases lead to a lower fitness value.

3.3.5 Position updating strategies

After updation of velocity, the position of particle updated according to the new velocity. if V is the new velocity and x_g is the number of different bits between P_{gbest} and P_i (i.e., $x_g = P_{gbest}$-P_i). Then position updation done according two situation [43]:

1. $V \leq x_g$. In this situation, if the velocity of particle less than or equal to the number of different bits between current particle and gbest. V bits of particle are randomly changed, which is different from that of gbest. The particle then moves toward the global best while keeping its exploration ability.
2. $V > x_g$. In this situation, in addition to changing all the different bits to be same as that of gbest, a further (V-x_g) bits should be randomly changed. So, after the particle reaches the global best position, it keeps on moving some distance toward other directions, enabling further search.

3.3.6 Velocity limitation

In experiment, the velocity of particle was initially limited to the range [1,N]. However, it was observed that after several iteration the swarm converges to be the best solution but not guarantees optimal solution, and during these iteration, the gbest remained stationary. Hence, this shows that the velocity varies high and particles often 'fly past' the optimal solution. This problem is overcome by limiting the range of velocity with in [1, (1/3)*N] and setting the Vmax = (1/3)*N. After limiting the Vmax, the particles

cannot fly too far away from the optimal solution. Once finding a global best position, other particles will adjust their velocities and positions, searching around the best position [43].

4 Experimental results and discussion

4.1 Experimental setup

To assess the performance of the proposed method, a set of experiments have been led on given datasets (i.e., Table 1) [44]. These nine datasets have diverse numbers of instances, features and classes that are utilized as representative example of the issue where the proposed method will be examined. RST only works on discrete or categorical data. In Table 1, first six datasets are categorical, where RST can work easily. All the discrete datasets have a small number of features. To further test the performance of proposed method, we added three more datasets (i.e., Musk 1, Semeion and Madelon), which are continuous dataset. To keep RST in mind, we applied filter discretization techniques in WEKA [45] tool to pre-process continuous data to discrete data. In each dataset, 70% and 30% instances are picked as the training and testing sets. The proposed method first runs on training set and selects set of features, and this process is also independent on any classification algorithm. The performance of test set is evaluated by classification algorithm according to the selected training attributes. Decision Trees (DT) and Naive Bayes (NB) are used in the experiment as classifier (Table 2).

The values of α ought to be greater than 0.5 in light of the fact that the lower approximation in RST characterizes that 50% instances in each equivalence class have a place with target set. In view of existing work [40], the value of $\alpha$is 0.8 (for Eq. 5) for all methods in this experiment. In each method, every particle is represented by binary string, whose length is the aggregate number of attributes in the dataset, which likewise represents the dimension of the solution space. Binary strings '1' and '0' represent that corresponding feature is chosen and that corresponding is not chosen, respectively. The swarm size is 30, the fully connected topology, w = 0.7298, v_{max} = 6.0, c_1 = c_2 = 1.49618 [21], and the maximum iteration is 200 uses for all methods. In EPSORSNA, γ is set as 0.9, 0.8, 0.7 and 0.5 to demonstrate the distinctive significance of the classification accuracy and the cardinality of features. Every algorithm is conducted for 50 independent runs on every dataset.

To additionally analyze the accuracy of the proposed method, two existing conventional methods (CfsF and CfsB) in WEKA [45] are utilized as performance comparison. Hall proposed CfsF and CfsB [28] in view of idea of correlation measure, which measure the correlation between class label and attributes. The search technique used in WEKA for forward selection (CfsF) and backward selection (CfsB) methods is greedy search, and DT uses as a classifier for computing the performance.

Table 1 Datasets

Dataset	#Features	#Classes	#Instances
Lymphography	18	4	148
Waveform	40	3	5000
Dermatology	33	6	366
Soybean large	37	19	307
Chess	36	2	3196
Statlog	36	6	6435
Musk Version 1(Musk 1)	166	2	476
Semeion	256	2	1593
Madelon	500	2	4400

4.2 Experimental results and comparisons

Tables 3, 4 and 5 show the results of existing methods (i.e., PSOPRS, PSOPRSN and PSOPRSE) and the proposed method (i.e., EPSORSNA). In Tables 3, 4 and 5, the result of PSOPRSN is tested with values of γ = 0.9 and 0.5, respectively (i.e., PSOPRSN 0.9 and PSOPRSN 0.5).

The DT and NB are two classifiers used for computing the performance of the selected attributes set on test set of every dataset. In Tables 3, 4 and 5, "All" represents, original feature set used for classification. "Size" represents average cardinality of feature subset selected in 50 independent runs. "Best" (i.e., Best accuracy) and "Ave" (i.e., Average accuracy) accuracy represent the best values and the average values of the testing classification performance achieved by every method throughout 50 independent runs, respectively.

Table 6 shows the result of existing methods (i.e., SRR and SQR) and proposed new quick reduct method (i.e., NQR) in terms of selected feature subset (i.e., reduct). Table 7 shows the comparison result between proposed and existing methods in terms of stability indices. Table 8 shows the experimental results of the proposed method (i.e., EPSORSNA) for different weights in fitness functions. In the Table 8, "Size", "Ave" (i.e., Average accuracy) and "Best-Acc" (i.e., Best accuracy) have the same meaning as in Tables 3, 4 and 5. Figure 1 shows the comparison between existing methods and the proposed method on nine given datasets in which, Fig. 1a shows the comparison between existing methods and EPSORSNA in terms of number of feature, where "X-Axis" represents particular dataset and "Y-Axis" represents the size of

Table 2 Result of traditional algorithm with DT as classifier

Dataset	Waveform	Dermatology	Soybean large	Chess	Statlog	Lympho graphy
Methods	Size (Acc)	Size (Acc)	Size (Acc)	Size (Acc)	Size (Acc)	Size (Acc)
CfsF	32 (72)	16 (87.63)	13 (83.79)	5 (77.43)	4 (71.71)	4 (86.37)
CfsB	32 (72)	19 (87.71)	14 (86.61)	6 (79.21)	4 (71.71)	3 (85.43)

Table 3 Result of Chess, dermatology and lymphography datasets

Dataset	Method	Size	DT Best (Ave Acc)%	NB Best (Ave Acc)%
Chess	All	36	98.5	87.89
	PSOPRS	30.49	98.57 (98.37)	91.11 (88.47)
	PSOPRSN 0.9	17.03	98.5 (98.01)	93.45 (91.39)
	PSOPRSN 0.5	8.4	97.63 (94.98)	93.08 (92.03)
	PSOPRSE	28.9	98.6 (98.43)	92.03 (89.31)
	EPSORSNA 0.5	7.9	98.6 (97.99)	92.13 (91.71)
Dermatology	All	34	82.79	95.79
	PSOPRS	20.92	97.51 (86.03)	98.21 (92.76)
	PSOPRSN 0.9	8.91	95.98 (93.31)	94.91 (80.10)
	PSOPRSN 0.5	7.79	89.91 (83.91)	90.16 (82.44)
	PSOPRSE	11.4	97.51 (92.03)	96.72 (92.67)
	EPSORSNA 0.5	6.98	97.51 (91.92)	95.79 (91.07)
Lymphography	All	18	75.51	87.76
	PSOPRS	11.39	80.41 (73.38)	92.07 (84.83)
	PSOPRSN 0.9	5.19	72.42 (65.78)	83.67 (78.16)
	PSOPRSN 0.5	4.96	68.39 (63.79)	89.78 (81.46)
	PSOPRSE	6.56	75.51 (70.12)	85.71 (81.70)
	EPSORSNA 0.5	4.6	75.71 (72.79)	92.01 (89.31)

Table 4 Result of waveform, Statlog and Soybean large datasets

Dataset	Method	Size	DT Best (Ave Acc)%	NB Best (Ave Acc)%
Waveform	All	40	74.79	79.71
	PSOPRS	24.47	77.37 (74.81)	81.27 (77.72)
	PSOPRSN 0.9	8.36	77.17 (73.91)	75.75 (69.86)
	PSOPRSN 0.5	7.7	72.92 (69.71)	75.01 (79.59)
	PSOPRSE	18.3	77.17 (72.5)	81.27 (74.87)
	EPSORSNA 0.5	7.1	75.91 (71.92)	77.0 (71.97)
Statlog	All	36	86.39	82.61
	PSOPRS	24.97	87.57 (85.47)	82.61 (82.06)
	PSOPRSN 0.9	13.6	86.37 (83.92)	82.03 (81.39)
	PSOPRSN 0.5	9.8	85.98 (84.32)	81.79 (80.21)
	PSOPRSE	20.13	87.13 (85.91)	83.14 (81.91)
	EPSORSNA 0.5	8.8	86.39 (85.19)	82.04 (80.91)
Soybean large	All	35	81.94	90.31
	PSOPRS	21.3	87.77 (80.01)	92.94 (85.39)
	PSOPRSN 0.9	10.34	81.18 (72.36)	81.94 (76.9)
	PSOPRSN 0.5	9.01	78.79 (72.98)	85.9 (76.04)
	PSOPRSE	19.12	85.46 (80.9)	85.46 (81.22)
	EPSORSNA 0.5	8.13	79.99 (73.13)	80.91 (72.79)

Table 5 Result of Musk 1, Semeion and Madelon datasets

Dataset	Method	Size	DT Best (Ave Acc)%	NB Best (Ave Acc)%
Musk 1	All	166	70.25	71.87
	PSOPRS	101.1	77.85 (72.22)	78.09 (73.27)
	PSOPRSN 0.9	44.77	77.22 (71.14)	81.14 (73.48)
	PSOPRSN 0.5	44.77	77.22 (71.14)	81.14 (73.48)
	PSOPRSE	81.13	76.58 (70.34)	80.81 (72.13)
	EPSORSNA 0.5	37.09	79.01 (77.08)	82.52 (79.32)
Semeion	All	256	94.35	87.31
	PSOPRS	159.67	94.35 (92.52)	88.53 (84.05)
	PSOPRSN 0.9	84.07	94.92 (92.35)	89.77 (86.92)
	PSOPRSN 0.5	84.07	94.92 (92.35)	89.77 (86.92)
	PSOPRSE	143.07	94.35 (92.27)	90.11 (88.21)
	EPSORSNA 0.5	62.17	95.19 (93.37)	91.14 (89.77)
Madelon	All	500	62.36	71.11
	PSOPRS	301.97	82.91 (76.52)	83.27 (78.11)
	PSOPRSN 0.9	183.43	82.68 (66.73)	84.29 (78.81)
	PSOPRSN 0.5	183.43	82.68 (66.73)	84.29 (78.81)
	PSOPRSE	301.97	82.91 (76.52)	84.72 (80.1)
	EPSORSNA 0.5	160.13	84.09 (81.71)	85.72 (82.33)

Table 6 Results of RST-based proposed (i.e., NQR) and existing methods (i.e., SRR and SQR)

Dataset	SRR	SQR	NQR	Min_Reduct
Lymphography	21	19	15	15
Waveform	19	22	7	7
Dermatology	9	8	6	6
Soyabean	29	27	13	13
Chess	22	20	24	20
Statlog	14	15	10	10
Musk 1	93	101	52	52
Semeion	253	250	74	74
Madelon	251	244	159	159

selected feature subset. Figure 1b, c shows the comparison between existing methods and EPSORSNA in terms of classification performance, where "X-Axis" represents particular dataset and "Y-Axis" represents the classification accuracy (DT or NB as classifier) of selected feature subset. And, color bar represents the particular method.

Figure 2 shows the comparison of the proposed method with different α values on nine given datasets, in which Fig. 1a shows the selected number of features by EPSORSNA with different α values, where "X-Axis" represents particular dataset and "Y-Axis" represents the size of selected feature subset. Figure 1b, c shows classification performances of EPSORSNA with different α values, where "X-Axis" represents particular dataset and "Y-Axis" represents the classification accuracy (DT or NB

Table 7 Result of stability indices

Dataset	PSOPRS	PSOPRSN	PSOPRSE	EPSOR -SNA-0.9	EPSOR -SNA-0.8	EPSOR -SNA-0.7
Lymphography	0.99	1	1	1	1	1
Waveform	1	1	0.99	1	1	1
Dermatology	1	1	1	1	1	1
Soyabean	0.97	1	1	1	1	1
Chess	0.99	1	1	1	1	1
Statlog	0.96	0.98	0.97	1	1	1
Musk 1	0.96	1	1	1	1	1
Semeion	0.96	1	0.97	1	1	1
Madelon	0.96	1	1	1	1	1

Table 8 Result of EPSORSNA algorithm with different α values

Dataset	Method	Size	DT Best (Avg Acc)%	NB Best (Avg Acc)%
Chess	EPSORSNA 0.5	7.9	98.6 (97.99)	92.13 (91.71)
	EPSORSNA 0.7	8.2	98.6 (98.01)	93.47 (91.81)
	EPSORSNA 0.8	11.3	97.7 (95.69)	93.47 (91.81)
	EPSORSNA 0.9	13.2	98.9 (98.10)	93.98 (91.82)
Dermatology	EPSORSNA 0.5	6.98	97.51 (91.92)	95.79 (91.07)
	EPSORSNA 0.7	6.99	97.51 (91.13)	95.84 (91.31)
	EPSORSNA 0.8	7.17	97.98 (91.90)	95.97 (89.78)
	EPSORSNA 0.9	7.37	98.07 (92.93)	98.31 (93.32)
Lymphography	EPSORSNA 0.5	4.6	75.71 (72.79)	92.01 (89.31)
	EPSORSNA 0.7	4.7	79.0 (73.73)	91.78 (84.79)
	EPSORSNA 0.8	4.7	79.0 (72.73)	92.78 (89.07)
	EPSORSNA 0.9	4.9	85.27 (79.16)	92.88 (89.14)
Waveform	EPSORSNA 0.5	7.1	75.91 (71.92)	77.0 (71.97)
	EPSORSNA 0.7	7.2	79.92 (77.12)	80.17 (72.79)
	EPSORSNA 0.8	7.4	79.98 (76.63)	83.0 (72.4)
	EPSORSNA 0.9	7.9	83.98 (78.11)	86.85 (82.69)
Statlog	EPSORSNA 0.5	8.8	86.23 (85.19)	82.04 (80.91)
	EPSORSNA 0.7	9.07	86.71 (86.01)	81.31 (80.17)
	EPSORSNA 0.8	10.17	86.71 (84.21)	81.79 (79.33)
	EPSORSNA 0.9	11.3	87.37 (85.79)	82.50 (79.98)
Soybean large	EPSORSNA 0.5	8.13	79.79 (73.13)	84.91 (73.79)
	EPSORSNA 0.7	8.13	81.98 (80.11)	87.81 (79.81)
	EPSORSNA 0.8	8.27	87.16 (83.71)	89.37 (86.77)
	EPSORSNA 0.9	8.39	88.92 (86.63)	94.79 (91.03)
Musk 1	EPSORSNA 0.5	37.09	79.01 (77.08)	82.52 (79.32)
	EPSORSNA 0.7	38.17	79.07 (77.41)	83.02 (80.22)
	EPSORSNA 0.8	39.31	79.87 (78.31)	83.22 (80.31)
	EPSORSNA 0.9	39.97	80.02 (78.38)	83.87 (81.33)
Semeion	EPSORSNA 0.5	62.17	95.19 (93.37)	91.14 (89.77)
	EPSORSNA 0.7	64.11	95.88 (93.11)	92.16 (89.91)
	EPSORSNA 0.8	68.05	96.01 (94.17)	92.16 (89.91)
	EPSORSNA 0.9	68.05	96.01 (94.21)	92.79 (90.03)
Madelon	EPSORSNA 0.5	160.13	84.09 (81.71)	85.72 (82.33)
	EPSORSNA 0.7	162.37	84.09 (81.13)	85.72 (82.7)
	EPSORSNA 0.8	167.42	86.93 (82.92)	84.83 (82.24)
	EPSORSNA 0.9	167.42	86.93 (82.92)	86.95 (83.02)

as classifier) of selected feature subset. And, color bar represents the particular method. Figure 3 shows the reduction in feature set, where "X-Axis" represents particular dataset and "Y-Axis" represents the size of selected feature subset (i.e., reduct). And, color bar represents the particular method.

4.2.1 Result of existing and proposed methods

PSOPRS: According to Tables 3, 4 and 5, PSOPRS selects subset with around 75% features of available total features in dataset and gets equal or higher classification accuracy than using original feature set in almost all cases. Best classification accuracy always better than using all feature in all cases, but average classification accuracy not better than using all features in some of cases. The outcomes suggest that PSOPRS can be effectively selects reduced feature subset with better or equal classification accuracy.

PSOPRSN: According to Tables 3, 4 and 5, PSOPRSN further reduces the feature subset and improves the classification performance. PSOPRSN with small γ selects less number of feature unless large γ because PSOPRSN with

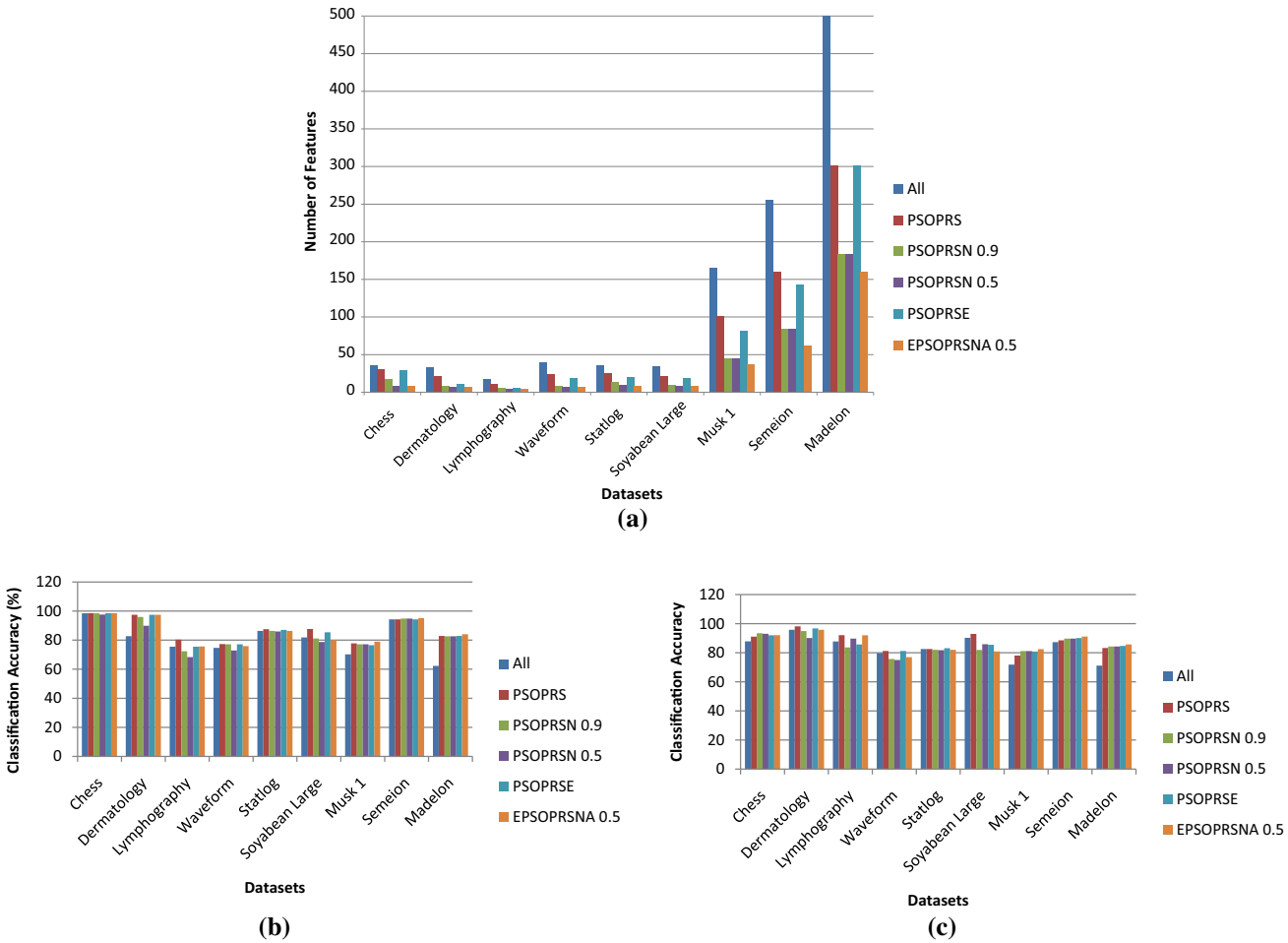

Fig. 1 Comparison between existing methods and the proposed method on nine given datasets. **a** Comparison between existing methods and EPSORSNA in terms of number of features. **b** Comparison between existing methods and EPSORSNA in terms of classification performances, where DT is a classifier. **c** Comparison between existing and EPSORSNA in terms of classification performances, where NB is a classifier

small γ means given more importance to the cardinality of feature and less importance to the classification performance, vise versa for PSOPRSN with large γ. Therefore, the objective function will play important role in PSOPRSN to search for resulted space with reduced feature subset. The results show when the cardinality of features reduces, the classification accuracy also decreases in most of the situations. In PSOPRSN 0.9 and PSOPRSN 0.5, classification accuracy is always better than using all features in all cases, but average classification accuracy is not better than using all features in some of cases. This suggests that the parameter γprovides balance to the classification accuracy and the cardinality of features.

PSOPRSE: According to Tables 3, 4 and 5, PSOPRSE selects subset with around 50% features of available total features in dataset and gets equal or higher classification accuracy than using original feature set in almost all cases. Classification performance always better than using all features in all cases, but average classification accuracy is not better than using all features in some of cases. The results recommend the PSOPRSE considering the representation power of the selected features and the number of equivalence classes; both are part of fitness function, which can successfully select reduced subset with better classification performance than using original feature set.

EPSORSNA: According to Tables 3, 4 and 5, in most cases, EPSORSNA 0.5 selects less than 25 % of the available features and in terms of accuracy obtained better accuracy than using all features (except classification accuracy for soybean large dataset). Although, in some cases, the average classification accuracy of the selected features is little worse than using original feature set. The results show that EPSORSNA 0.5 considering all three parameter, representation power (classification performance) of the selected attributes, cardinality of feature and accuracy of approximation can successfully select reduced feature subset with higher classification accuracy than using original feature set.

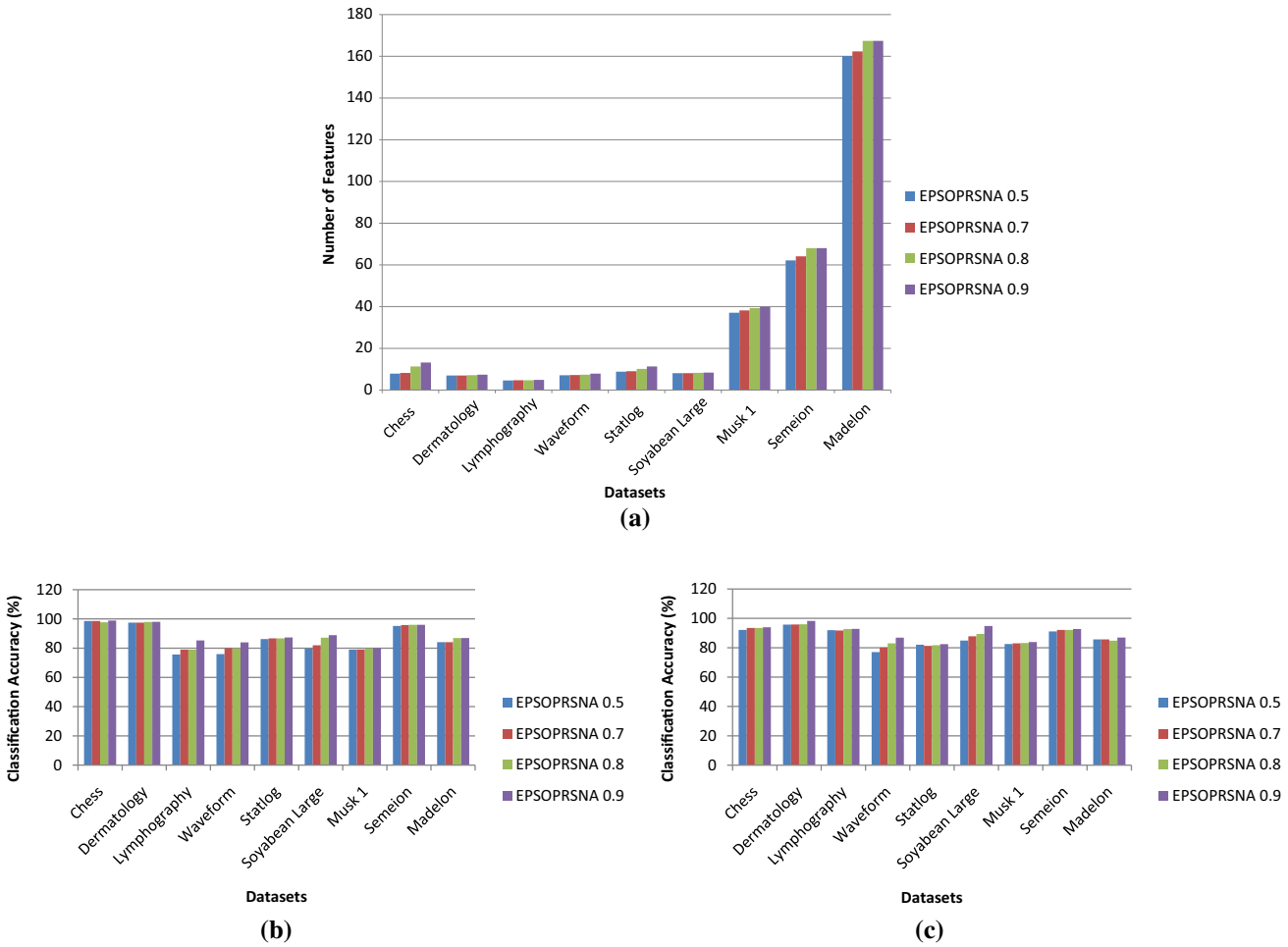

Fig. 2 Comparison of the proposed method with different α values on nine given datasets. **a** Selected number of features by EPSORSNA with different α values. **b** Classification performances of EPSORSNA with different α values, where DT as a classifier. **c** Classification performances of EPSORSNA with different α values, where DT is a classifier

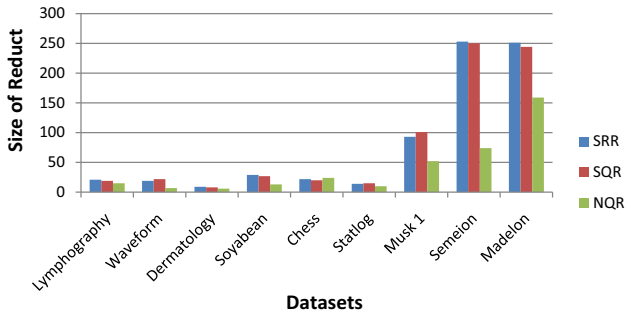

Fig. 3 Reduction in feature set

4.2.2 Results and comparison of proposed algorithm (i.e., NQR) with SQR and SRR

The performance of the proposed RST-based new quick reduct method is studied and compared with some existing methods, SQR and SRR. The proposed method is applied for feature selection and reduces the dimension of the dataset. The experimental results are recapitulated in Table 6. The first column consists of dataset names, and the second and third columns consist of the result of proposed and existing algorithms. The last column (i.e., Min_Reduct) consists of the minimum size of reduct among three methods (i.e., SRR, SQR and NQR). Figure 3 shows the comparisons of the proposed algorithm with existing methods, where "X-Axis" represents the size of feature subset (i.e., reduct) and "Y-Axis" represents the datasets, and color bar represents the particular method.

According to Table 6, the proposed method always selects smaller feature subset (reduct) in all cases (except Chess dataset). For example in waveform dataset, SQR and SRR selected around 22 and 19 from the 40 original feature set. The proposed method further reduces almost 68% and 63% feature of SQR and SRR, respectively. According to Fig. 3, the proposed method outperformed the existing methods in terms size of reduct in almost all cases.

4.2.3 Comparison of proposed algorithm (i.e., EPSORSNA) with traditional algorithms

Experiments have been performed on traditional (CfsF and CfsB) methods for dimensional reduction using WEKA as a tool. Table 2 shows the experimental results of CfsF and CfsB methods, where DT was used as a classifier. Tables 3, 4 and 5 show the experimental results of PSOPRS, PSOPRSN, PSOPRSE and EPSORSNA methods, where DT and NB were used as a classifier. Comparing the result of Tables 3, 4 and 5 with experiment results of CfsF and CfsB methods, we can see in all cases that PSOPRS, PSOPRSN, PSOPRSE and EPSORSNA achieved much more better classification accuracy than traditional methods, but CfsF and CfsB selected a equal number of features.

4.2.4 Comparison of proposed algorithm (i.e., EPSORSNA) with PSOPRS, PSOPRSN and PSOPRSE

According to Table 3, 4 and 5, comparing the results of EPSORSNA with PSOPRS, EPSORSNA obtained better or equal classification accuracy and the cardinality of selected feature subset in EPSORSNA is always much smaller than in PSOPRS. For example, PSOPRS selected around 31 features from the 36 original feature sets and its best classification accuracy is 98.57% for Chess dataset when DT is used as classification algorithm. EPSORSNA further reduced almost 75% of the features and obtained the best classification accuracy to 98.67%. Because the proposed method considers accuracy of approximation, the number of features and the classification quality of feature subset are the part of fitness function, which can further reduce the cardinality of selected feature subset with better accuracy.

Both PSOPRSN and PSOPRSE are considered the classification power of the features represented by reduct and the cardinality of features, which are seen as the number of features and the number of equivalence classes in PSOPRSN and PSOPRSE, respectively. Compared PSOPRSN with EPSORSNA , the EPSORSNA includes two new parameters in fitness function, which is the classification quality of the feature subset and the accuracy of approximation. Whenever, increasing the value of α PSOPRSN always selects the reduced feature subset and achieve good classification performance. But, it may loss the generality and not able to achieve better performance on unseen dataset. According to Tables 3, 4 and 5, comparing the results of EPSORSNA with PSOPRSN, EPSORSNA obtained better or equal classification performance and the cardinality of selected feature subset in EPSORSNA is always smaller than in PSOPRSN. For example, PSOPRSN 0.9 selected around 84 features from the 256 original feature set and its best classification accuracy is 77.22% for Semeion dataset when DT used as classification algorithm. EPSORSNA 0.5 further reduced almost 26% of the features and obtained the best classification accuracy to 79.01%. Therefore, EPSORSNA outperformed the PSOPRSN in terms of number of feature and classification accuracy.

According to Tables 3, 4 and 5 and 8, comparing the results of proposed method (EPSORSNA) with PSOPRSE, the proposed method achieved better or equal classification accuracy than PSOPRSE. But, cardinality of selected feature subset in EPSORSNA is always smaller than PSOPRSE. Initially, PSOPRSE performed better than EPSORSNA in terms of classification accuracy for small value of α in EPSORSNA. But, after increasing the value of α, the EPSORSNA outperformed the PSOPRSN in terms of number of feature, stability indices and classification performance. For example, in Chess dataset DT as classifier, PSOPRSE selected almost 29 features out of 36 original feature set and its best classification accuracy is 98.6%. EPSORSNA-0.9 further minimizes around 60% of the features and improves the best classification accuracy to 98.9%.

Table 7 shows the comparison result of proposed and existing methods in terms of stability indices. The outcomes suggest that EPSORSNA 0.9, EPSORSNA 0.8 and EPSORSNA 0.7 methods achieved stability indices value 1 in all cases. Hence, the proposed methods (i.e., EPSORSNA 0.9, EPSORSNA 0.8 and EPSORSNA 0.7) are more stable compared to other existing methods.

Figure 1 shows the comparison between existing methods and the proposed method on nine datasets. According to Fig. 1a, it can be clearly observed that EPSORSNA, PSOPRSN, PSOPRSE and PSOPRS obtained ranks first, second, third and fourth in terms of selected number of feature. According to Fig. 1b, c, it can be clearly observed that EPSORSNA outperforms the existing methods in terms of classification accuracy for almost all cases.

4.3 Results of proposed method (i.e., EPSORSNA) with different values of α, β and γ

According to Table 8, the EPSORSNA with any value of α can select the reduced feature set while achieving higher or equal classification accuracy than using all features. The proposed algorithm (i.e., EPSORSNA) is tested on four values of α 0.5, 0.7, 0.8 and 0.8, in which increasing the value of α, gives more importance to the classification quality of feature subset and less importance to the number of features and accuracy of approximation.

Outcomes suggest that for increasing values of α, EPSORSNA may select equal or few more number of feature while achieving higher classification accuracy. For example, in the Chess dataset DT and NB are used as the

classification algorithms, EPSORSNA selects average 8.2 features from the 36 original feature set and its best accuracy is 98.6% for $\alpha = 0.7$ and EPSORSNA selected around 13.2 features from the 36 available features and its best classification accuracy is 98.9% for $\alpha = 0.9$.

Figure 2 shows the comparison of proposed method with different α values on nine datasets. According to Fig. 1a, it can be clearly observed that EPSORSNA 0.5 obtained fist rank, EPSORSNA 0.7 the second, EPSORSNA 0.8 third, and EPSORSNA 0.9 gets the fourth rank in terms of selected number of features. According to Fig. 1b, c, it can be clearly observed that the EPSORSNA 0.9 obtained the fist rank, EPSORSNA 0.8 the second, EPSORSNA 0.7 third, and EPSORSNA 0.5 gets the fourth rank in terms of classification accuracy for almost all cases.

Therefore, EPSORSNA outperformed the three existing single objective methods and two traditional feature selection methods, in terms of number of feature, stability indices and classification performance.

5 Conclusion and future work

The goal of this work to propose the feature selection and classification method to select reduced feature subset with higher classification accuracy than original feature set. The goal was fulfilled by proposed an efficient feature selection method using PSO and rough sets (i.e., EPSORSNA). The aim of the proposed method to improve the classification accuracy and reduce the size of feature subset, which depends on fitness function, which includes three parameters, the classification quality of feature subset, number of feature and accuracy of approximation. The performance of EPSORSNA was observed and compared with five feature selection methods (including three existing and two traditional). Experiments were conducted on nine datasets with different number of instances, number of attributes and number of classes. DT and NB are two learning algorithms used to test the generality of the proposed method. The result shows that EPSORSNA outperformed PSOPRS, PSOPRSN, PSOPRSE and traditional algorithms in terms of the number of features, stability indices and the classification performance. Although it is also observed that with increased weight on the classification quality of feature subset of the fitness function, there is a significant reduction in the number of features while achieving higher classification accuracy.

In the future, we will investigate the ways to further reduce the feature subset with maximize classification accuracy and also explore multi-objective PSO and rough set for feature selection and classification for more than one objective.

Compliance with ethical standards

Conflict of interest We have no conflict of interest.

References

1. Settouti N, Bechar MEA, Chikh MA (2016) Statistical comparisons of the top 10 algorithms in data mining for classification task. Int J Interact Multimed Artif Intell Spec Issue Artif Intell 4:46–51 (Underpinning)
2. Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1(3):131–156
3. Pujari JD, Yakkundimath R, Byadgi A et al. (2016) SVM and ANN based classification of plant diseases using feature reduction technique. Int J Interact Multimed Artif Intell 3(7):1–9
4. Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11(5):341–356
5. Pawlak Z (2012) Rough sets: theoretical aspects of reasoning about data, vol 9. Springer, New York
6. Pawlak Z (1997) Rough set approach to knowledge-based decision support. Eur J Oper Res 99(1):48–57
7. Chouchoulas A, Shen Q (2001) Rough set-aided keyword reduction for text categorization. Appl Artif Intell 15(9):843–873
8. Cervante L, Xue B, Shang L, Zhang M (2013) Binary particle swarm optimisation and rough set theory for dimension reduction in classification. In: 2013 IEEE congress on evolutionary computation. IEEE, pp 2428–2435
9. Bae C, Yeh W-C, Chung YY, Liu S-L (2010) Feature selection with intelligent dynamic swarm and rough set. Expert Syst Appl 37(10):7026–7032
10. Kudo M, Sklansky J (2000) Comparison of algorithms that select features for pattern classifiers. Pattern Recogn 33(1):25–41
11. Cervante L, Xue B, Shang L, Zhang M (2013) A multi-objective feature selection approach based on binary pso and rough set theory. In: European conference on evolutionary computation in combinatorial optimization. Springer, pp 25–36
12. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1):273–324
13. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(Mar):1157–1182
14. Whitney AW (1971) A direct method of nonparametric measurement selection. IEEE Trans Comput 100(9):1100–1103
15. Huang C-L, Wang C-J (2006) A GA-based feature selection and parameters optimization for support vector machines. Expert Syst Appl 31(2):231–240
16. Stein G, Chen B, Wu AS, Hua KA (2005) Decision tree classifier for network intrusion detection with GA-based feature selection. In: Proceedings of the 43rd annual Southeast regional conference, vol 2. ACM, pp 136–141
17. Muni DP, Pal NR, Das J (2006) Genetic programming for simultaneous feature selection and classifier design. IEEE Trans Syst Man Cybern Part B Cybern 36(1):106–117
18. Al-Ani A (2005) Feature subset selection using ant colony optimization. Int J Comput Intell 2(1):53–58
19. Unler A, Murat A (2010) A discrete particle swarm optimization method for feature selection in binary classification problems. Eur J Oper Res 206(3):528–539
20. Meza J, Espitia H, Montenegro C, Giménez E, González-Crespo R (2017) MOVPSO: vortex multi-objective particle swarm optimization. Appl Soft Comput 52:1042–1057
21. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: The 1998 IEEE international conference on evolutionary computation proceedings, 1998. IEEE world congress on computational intelligence. IEEE, pp 69–73

22. Kennedy J (2011) Particle swarm optimization. Encyclopedia of machine learning. Springer, Berlin, pp 760–766
23. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. Swarm Intell 1(1):33–57
24. Meza J, Espitia H, Montenegro C, Crespo RG (2016) Statistical analysis of a multi-objective optimization algorithm based on a model of particles with vorticity behavior. Soft Comput 20(9):3521–3536
25. Yao Y, Zhao Y (2008) Attribute reduction in decision-theoretic rough set models. Inf Sci 178(17):3356–3373
26. Clerc M (2012) Standard particle swarm optimisation. https://hal.archives-ouvertes.fr/hal-00764996
27. Banka H, Dara S (2015) A hamming distance based binary particle swarm optimization (hdbpso) algorithm for high dimensional feature selection, classification and validation. Pattern Recogn Lett 52:94–100
28. Hall MA (1999) Correlation-based feature selection for machine learning. PhD thesis, The University of Waikato
29. Almuallim H, Dietterich TG (1994) Learning boolean concepts in the presence of many irrelevant features. Artif Intell 69(1–2):279–305
30. Whitney AW (1971) A direct method of nonparametric measurement selection. IEEE Trans Comput 100(9):1100–1103
31. Marill T, Green D (1963) On the effectiveness of receptors in recognition systems. IEEE Trans Inf Theory 9(1):11–17
32. Stearns, Stephen D (1976) On selecting features for pattern classifiers. In: Proceedings of the 3rd international joint conference on pattern recognition, pp 71–75
33. Chakraborty B (2002) Genetic algorithm with fuzzy fitness function for feature selection. In: IEEE international symposium on industrial electronics (ISIE02), vol 1, pp 315–319
34. Chakraborty B (2008) Feature subset selection by particle swarm optimization with fuzzy fitness function. In: 3rd international conference on intelligent system and knowledge engineering, 2008. ISKE 2008, vol 1. IEEE, pp 1038–1042
35. Neshatian K, Zhang M (2009) Pareto front feature selection: using genetic programming to explore feature space. In: Proceedings of the 11th annual conference on genetic and evolutionary computation. ACM, pp 1027–1034
36. Jensen R (2006) Performing feature selection with ACO. Swarm intelligence in data mining. Springer, Berlin, pp 45–73
37. El Aziz MA, Hassanien AE (2016) Modified cuckoo search algorithm with rough sets for feature selection. Neural Comput Appl. https://doi.org/10.1007/s00521-016-2473-7
38. Wang X, Yang J, Teng X, Xia W, Jensen R (2007) Feature selection based on rough sets and particle swarm optimization. Pattern Recogn Lett 28(4):459–471
39. Chen Y, Miao D, Wang R (2010) A rough set approach to feature selection based on ant colony optimization. Pattern Recogn Lett 31(3):226–233
40. Cervante L, Xue B, Shang L, Zhang M (2012) A dimension reduction approach to classification based on particle swarm optimisation and rough set theory. In: Australasian joint conference on artificial intelligence. Springer, pp 313–325
41. Swiniarski RW, Skowron A (2003) Rough set methods in feature selection and recognition. Pattern Recogn Lett 24(6):833–849
42. Xue B, Cervante L, Shang L, Browne WN, Zhang M (2014) Binary pso and rough set theory for feature selection: a multi-objective filter based approach. Int J Comput Intell Appl 13(02):1450009
43. Inbarani HH, Azar AT, Jothi G (2014) Supervised hybrid feature selection based on pso and rough sets for medical diagnosis. Comput Methods Programs Biomed 113(1):175–185
44. Frank A, Asuncion A (2010) UCI machine learning repository. School of information and computer science, vol 213. http://archive.ics.uci.edu/ml
45. Witten Ian H, Eibe F (2005) Data mining: practical machine learning tools and techniques. Morgan Kaufmann, Los Altos