




# An improved rough set approach for optimal trust measure parameter selection in cloud environments

Somu Nivethitha<sup>1</sup> · M. R. Gauthama Raman<sup>2</sup> · Obulaporam Gireesha<sup>3</sup> · Krithivasan Kannan<sup>4</sup> · V. S. Shankar Sriram<sup>3</sup> 

Published online: 31 January 2019  
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

## Abstract

The existence of a multitude of cloud service providers (CSPs) for each service type increases the difficulty in the identification of appropriate and trustworthy service providers based on their abilities and cloud users' unique functional and non-functional quality-of-service (QoS) requirements. Further, the dynamic nature of the cloud ecosystem in terms of performance and new services increases the complexity of the cloud service selection problem. Trust-based service selection mechanisms which involve the intrinsic relations among the QoS parameters or trust measure parameters (TMPs) to evaluate the quality of the CSPs are the most preferred solution for the problem of cloud service selection. However, the accuracy of the trust-based service selection models and the CSP's trust value relies on the optimality of the TMP subset obtained with respect to the service type. Hence, this work presents an efficient rough set theory-based hypergraph-binary fruit fly optimization (RST-HGBFFO), a cooperative bio-inspired technique to identify the optimal service-specific TMPs. Experiments on QWS dataset, Cloud Armor, and CISH—SASTRA trust feedback dataset reveal the predominance of RST-HGBFFO over the state-of-the-art feature selection techniques. The performance of RST-HGBFFO feature selection technique was validated using hypergraph-based computational model and WEKA tool in terms of reduct size, service ranking, classification accuracy, and time complexity.

**Keywords** Trust measure parameters (TMPs) · Rough set theory (RST) · Hypergraph · Binary fruit fly optimization (BFFO) · Hypergraph-based computational model (HGCM) · Cloud service ranking

---

Communicated by V. Loia.

---

✉ V. S. Shankar Sriram  
sriram@it.sastra.edu

- <sup>1</sup> Smart Energy Informatics Lab (SEIL), Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, India
- <sup>2</sup> iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design (SUTD), Singapore, Singapore
- <sup>3</sup> Centre for Information Super Highway (CISH), School of Computing, SASTRA Deemed to be University, Thanjavur, Tamil Nadu, India
- <sup>4</sup> Discrete Mathematics Research Laboratory (DMRL), School of Humanities and Sciences, SASTRA Deemed to be University, Thanjavur, Tamil Nadu, India

## 1 Introduction

Cloud computing has been gaining much attention by a wide variety of business and academic organizations due to its overwhelming benefits such as broad network access, measured service, resource pooling, on-demand service, and rapid elasticity (Sosinsky 2010). Being an open standard model, cloud provides 'XaaS'—anything as a service (software, platform, infrastructure, storage, security, data, database, etc.) to its users through various deployment models (private cloud, community cloud, public cloud, and hybrid cloud) (Sosinsky 2010; Mell and Grance 2011). The giants of IT (Amazon, IBM, Google, Rackspace, etc.) have become the pioneers of cloud service offerings in almost all cloud service delivery models (XaaS), while the rest have begun to migrate their business applications to cloud, in order to bring down the infrastructure and operational costs (Ding et al. 2014a).

This scenario has resulted in two major research challenges with respect to the selection of appropriate and trustworthy cloud service providers which can be stated as,

### 1.1 Problem 1 How to identify the trustworthy cloud service providers?

The phenomenal increase in the number of CSPs offering functionally similar cloud services at different feature, cost, and quality makes the selection of trustworthy CSPs, a challenging task (Thampi and Bhargava 2013). The application of trust assessment mechanism to the cloud service selection problem helps the user to find the trustworthy service provider (Somu et al. 2017b). Trustworthiness is a most important comprehensive quality metric of the cloud service, which can be assessed either in an objective or subjective manner with respect to the several trust measure parameters (TMPs) such as usability, assurance, security, availability, performance, and cost (Marudhadevi et al. 2014; Qu 2016; Tang et al. 2017). The importance of trust metric and their relation to the quality of the cloud services has resulted in the development of several trust-based service evaluation and service selection models for the identification of suitable and trustworthy CSPs (Liao et al. 2007; Ding et al. 2014a, b, Sun et al. 2014, 2016; Wang et al. 2015; Qu 2016; Ma et al. 2016). However, these models evaluate the trustworthiness of the CSPs based on a pre-defined set of TMPs which affects the accuracy and complexity of the service selection model.

### 1.2 Problem 2 What are the major requirements to build an efficient trust-based cloud service selection model?

In general, a trust-based cloud service selection model identifies the trustworthy CSPs using various trust evaluation mechanisms based on objective (QoS values) and subjective (user feedbacks) assessment data. An efficient cloud service selection model should be adaptive to the dynamic nature of cloud services which further complicates the process of trust-based service evaluation. For example, it is hard to evaluate the trustworthiness of the existing cloud service and a new cloud service due to the context (time and location)-based performance variation and lack of assessment data, respectively (Qu et al. 2015). Trust prediction, a classification, and a multi-criteria decision-making (MCDM) problem is one plausible solution for the above-said challenge through the accurate trust evaluation of cloud service providers, especially when the trustworthiness of new CSPs needs to be evaluated (Akshya Kaveri et al. 2017; Obulaporam et al. 2018). The significance of trust prediction models in service-oriented environments has resulted in several research contributions in web service selection, cloud service selection, recommender systems, pervasive environments, and social networks (Ding et al. 2017b; Mao et al. 2017; Su et al. 2017).

According to Vimal Kumar et al. 'A good TMP would be easily measurable from the service provider's side and would be contextually related to the service the end user wants to use on the cloud' (Thampi and Bhargava 2013). For example, let us consider a cloud ecosystem where 'n' cloud service users need to find a suitable and trustworthy cloud service provider from 'm' functionally similar cloud service providers. In a real-world scenario, the severity of the cloud service selection gets complicated with the ever-growing nature of the cloud user requests and the cloud service providers. In general, the performance assessment of each cloud service provider is based on the historical QoS and feedback information provided by the historical users across the QoS attributes. Further, the importance of the QoS attributes varies with respect to the cloud service type and user preferences. Hence, a generic solution to the above-said challenges (Problems 1 and 2) can be presented through the design of an efficient feature selection technique which can identify the optimal TMP reduct with respect to the service type requested by the users or their own preferences (Somu et al. 2017b). It also addresses 'Lack of an advanced multi-criteria-based measurement of user preferences,' an open issue in the present cloud service selection models and approaches (Sun et al. 2014). To summarize, the design of an efficient feature selection technique enhances the accuracy of the cloud service selection model in terms of reduced complexity (minimizes the search space) and better service selection (recommends the service providers that have maximum adherence with the users' QoS requirements).

Feature selection is found to be one of the most challenging problems in the field of engineering science which aims at the identification of the minimal subset of features that have same information content as that of the original information system (Chen et al. 2015, 2017). In specific, feature selection techniques have their significance in cloud service selection problem through the identification of optimal service-specific TMPs, thereby reducing the complexity and enhancing the accuracy of the cloud service selection model (Somu et al. 2017b). Among the various computational intelligent models like principle component analysis (PCA), independent component analysis (ICA), instance-based feature selection, correlation-based feature selection, information gain-based feature selection, Bayesian feature selection, etc. rough set theory (RST) is found to be the major choice of researchers due to its capability of handling imprecise, uncertain and vague data (Pawlak 1982b; Chen et al. 2015; Raza and Qamar 2018; Zouache and Ben Abdelaziz 2018). In general, RST for feature selection operates over the positive region based on attribute dependency to find the subset of original features (reducts) from the given dataset. However, the dependency computation of RST is found to be computationally extensive since it involves the generation of equivalence class using both

conditional and decision attribute followed by the generation of the positive region (Raza and Qamar 2018). One simple solution toward this problem is the generation of all possible reducts and choosing the one which has a higher dependency on the decision attribute and minimal cardinality. However, this approach is not applicable for the dataset with a large number of features and samples since the time incurred for the generation of all possible subset is in the order of exponential powers (Jiang et al. 2015). Several research works toward the integration of RST with the greedy approaches like hill climbing, forward selection, backward elimination, conditional entropy-based significant estimation, discernibility matrix-based methods, etc., failed to guarantee an optimal solution within a reasonable time (Chen et al. 2015; Das et al. 2018).

In general, finding all possible subset is an NP-hard problem (Gheyas and Smith 2010). Recent research contributions in the development of an efficient feature selection technique have revealed the significance of meta-heuristic techniques like genetic algorithm, particle swarm optimization, cuckoo search optimization, gray wolf optimization, firefly optimization, etc., in solving NP-hard problems (Table 1) (Gheyas and Smith 2010; Raman et al. 2016; Somu et al. 2016; Gauthama Raman et al. 2017a). Even though these meta-heuristic techniques guarantee the global optimal solution, they suffer from high time and computational complexity issues due to the distribution of

computing and memory ability of various degrees to each individual in the generations (Chen et al. 2015).

Fruit fly optimization algorithm (FFOA) is a new, simple, computationally effective swarm intelligent evolutionary algorithm proposed by Pan 2012 which is inspired by the food foraging behavior of fruit flies. Due to the inherent ability of FFOA in providing faster convergence and guaranteeing global optimal solution, it has been successfully applied for parameter optimization, combinatorial optimization and function estimation in various domains like image processing, bio-informatics, stock market analysis, etc. (Pan 2012; Dai et al. 2014; Shen et al. 2016). Hence, in this work, we have introduced a binary variant of FFOA (BFFO) for identifying the minimal rough set reducts. The major contribution in this work is as follows:

1. RST-HGBFFO exploits the hyperclique property of hypergraph to enhance the performance of BFFO through minimal time complexity and improved convergence rate.
2. HGBFFO integrated with RST (supervised quick reduct (SQR) and quick relative reduct (QRR)) to enhance the performance of RST for the identification of the optimal reduct in minimal time. This work is the extension of our previous work, where we have proposed a novel trust assessment model (TrustCom) and an efficient feature selection technique based on RST and hypergraph properties (Somu et al. 2017b).

**Table 1** Related works

Authors	Technique proposed	Optimization technique hybridized with rough sets	Applications
Inbarani et al. (2015)	Rough set improved harmony search quick reduct algorithm	Improved harmony search	Medical Applications
Jothi and Inbarani (2016 and Ganesan et al. (2017))	Tolerance rough set firefly-based quick reduct	Fire fly optimization algorithm	
Udhaya Kumar and Hannah Inbarani (2017)	PSO-based rough set feature selection technique and neighborhood rough set classifier method	Particle swarm optimization	
Chen et al. (2010)	Rough set approach for feature selection based on ant colony optimization	Ant colony optimization	Generic Applications
Chen et al. (2015)	Rough set reduct using fish swarm algorithm	Fish swarm algorithm	
Luan et al. (2016)	Attribute reduction algorithm based on rough sets and improved artificial fish swarm algorithm	Improved artificial fish swarm algorithm	
Chen et al. (2017)	Neighborhood rough set reduct with fish swarm	Fish swarm algorithm	
Su and Guo (2017)	Minimum attribute reduction based on rough set theory and artificial fish swarm	Artificial fish swarm algorithm	
Aziz and Hassanien (2018)	Rough sets based on modified cuckoo algorithm	Modified cuckoo search algorithm	
Das et al. (2018)	Incremental feature selection using rough set based on genetic algorithm	Genetic algorithm	
Zouache and Ben Abdelaziz (2018)	Swarm intelligence algorithm based on quantum-inspired and rough sets	Firefly optimization and particle swarm optimization	
Hassanien et al. (2017)	Improved moth flame optimization algorithm for rough sets	Improved moth flame optimization algorithm	Agricultural applications

3. This work uses Cloud Service Measurement Index Consortium—Service Measurement Index (CSMIC—SMI) metric as a standard for TMP (Valley 2011; Garg et al. 2013; Somu et al. 2017a). Further, chi-square test was applied to the considered datasets to assess the independent nature of the TMPs (Moore 1976).
4. Experiments were conducted using three different datasets, namely quality web service (QWS) dataset, Cloud Armor trust feedback dataset, and Centre for Information Super Highway (CISH)—SASTRA synthetic trust feedback dataset. The performance of RST-HGBFFO was validated using hypergraph-based computational model (HGCM) and WEKA tool in terms of reduct size, service ranking, classification accuracy, and time complexity (Øhrn 2001; Somu et al. 2017a).

The paper is structured as follows: Sect. 2 gives a brief introduction to the rough set theory, hypergraph, and fruit fly optimization. Section 3 discusses the proposed RST-HGBFFO feature selection technique. Section 4 highlights the experimentation and performance analysis of RST-HGBFFO over the state-of-the-art techniques based on the reduct size, service ranking, classification accuracy, and runtime analysis. Section 5 concludes the paper.

## 2 Materials and methods

### 2.1 Rough set theory

Rough set theory is an intelligent mathematical tool, proposed by Pawlak (1982b) for handling imprecise, uncertain and vague dataset. Unlike traditional soft computing techniques like fuzzy sets, Dempster-Shafer theory, etc., RST does not require any additional or prior information about the dataset. Therefore, RST has been extensively used in a wide range of applications for knowledge and data discovery. The fundamental assumptions in RST are

as follows: (i) Objects in the universe of discourse are represented by the attributes with some information associated with them, and (ii) objects that consist of similar information are indiscernible. This section discusses a few basic concepts and definition of RST which are usually employed for the identification of the optimal reduct (Pawlak 1982a, b; Abraham 2009).

The data analysis process in the rough set theory (RST) initiates from the Information System ( $I_S$ ) represented in the form of a table where rows correspond to the samples and columns correspond to the attributes (Table 2a). Mathematically, an information system can be viewed as four-tuple vector  $I_S = (U, A, V, f)$ , where  $U$  is the universe with non-empty set of objects,  $A$  is the non-empty set of attributes,  $V$  is the union of attributes such that  $V = \bigcup_{a \in A} V_a$  and  $f : U \times A \rightarrow V_a$  is the information function such that  $a \in A$  and  $x \in U, f(x, a) \in V_a$ . For every  $M$  subset of  $A$ , there exists an associated equivalence relation called indiscernibility relation as defined in Eq. (1) (Min and Liu 2009; Pawlak 1982a; Yao 2004).

$$IND_M = \{(x, y) \in U^2 | \forall a \in M, f(x, a) = f(y, a)\} \tag{1}$$

The family of all equivalent classes of  $IND_M$  is denoted either by  $U/IND_M$  or  $U/M$  and an object  $x$  contained in the equivalent class of  $IND_M$  is represented as  $[x]_M$ . Basically, the class of attributes in  $I_S$  can be distinguished into conditional ( $C$ ) and decisional ( $D$ ) attributes ( $A = C \cup D$ ), where such case is referred as decision table ( $D_S$ ) (Table 2b). For any  $M \subseteq C \cup D$  and  $X \in U$ , the  $M$ -lower approximation and  $M$ -upper approximation are defined in Eqs. (2) and (3) (Pawlak 1982a).

$$Y_M = \leftarrow \{[x]_M \in U/IND(M) : [x]_M \subseteq X\} \tag{2}$$

$$Y^M = \bigcup \{[x]_M \in U/IND(M) : [x]_M \cap X \neq \emptyset\} \tag{3}$$

The  $M$ - boundary region of set  $Y$  is defined as  $Bnd_M = Y_M - Y^M$ . Generally, the roughness of the given decision table ( $D_T$ ) of set  $Y$  is computed using Eq. (4).

**Table 2** Information system ( $I_s$ ) and decision system ( $D_s$ )

Information system ( $I_s$ )				Decision system ( $D_s$ )				
U	Conditional attributes (A)			U	Conditional attributes (A)			Decisional attributes (D)
	Education	Subject	State		Education	Subject	State	Job
(a)				(b)				
1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	1
3	2	1	1	3	2	1	1	3
4	3	3	1	4	3	3	1	2
5	3	2	2	5	3	2	2	3
6	1	4	3	6	1	4	3	1
7	3	2	3	7	3	2	3	3
8	1	4	1	8	1	4	1	1

$$Y_\delta = \frac{|\text{Bnd}_M|}{|Y^M|} \tag{4}$$

The measure of roughness is found to be an important metric in RST since it reflects the degree of uncertainty of an approximation set. The value of  $Y_\delta$  lies in the range of  $[0,1]$ , where 0 implies every information in the underlying set  $Y$  is known and 1 implies lack of information regarding the underlying set  $Y$ . The  $M$ -positive region  $\text{PoS}(D)$  of  $U/\text{IND}(D)$  with respect to  $Y$  is defined as a set of all objects in  $U$  that can be certainly classified to the classes of  $U/\text{IND}(D)$  by the means of  $M$  (Eq. 5)(Pawlak 1982a).

$$\text{PoS}_M(D) = \bigcup_{Y \in U/\text{IND}(D)} Y_M \tag{5}$$

The degree of dependency among the set of attributes ( $M$ ) and the decisional attribute ( $D$ ) can be computed through the positive region using Eq. (6).

$$\gamma_M(D) = |\text{PoS}_B(D)|/|U| \tag{6}$$

Obviously,  $\gamma_M(D) \in \{0, 1\}$ , where 0 represents  $M$  is independent of  $D$ ; 1 represents  $M$  is completely dependent on  $D$ ; and  $0 < \gamma_M(D) < 1$  represents the partial dependency between  $M$  and  $D$ . The main objective of RST is the removal of redundant and irrelevant features, and therefore, the reduced set of attributes (reducts) consists of the same information content as that of the original dataset. Mathematically, the set of all reducts can be denoted as in Eq. (7).

$$R(C) = \{R \subseteq C | \gamma_R(D)\} = \gamma_C(D) \forall M \subseteq R, \gamma_R(D) \neq \gamma_C(D) \tag{7}$$

Since the dataset consists of multiple reducts, the optimal reduct is defined as in Eq. (8).

$$R_{\text{Opt}} = \{R' \in R | \forall R'' \in R, |R'| < |R''|\} \tag{8}$$

### 2.2 Hypergraph

Hypergraph is a generalization of the traditional graph theory in which the edges (hyperedges) correspond to the collection of two or more vertices. Mathematically, hypergraph ( $H_G$ ) is represented as an order 2-tuple  $H_G = \{V, E\}$ , where  $V = \{V_1, V_2, \dots, V_n\}$  is the finite set of vertices and  $E = \{e_1, e_2, \dots, e_m\}$  corresponds to the hyperedges which is a non-empty subset of  $V$ , such that  $E_i \neq \emptyset$  and  $\bigcup E_i = V; \forall i = (1, 2, \dots, n)$ . In recent years, the integration of hypergraph with various computational models has proven its efficiency in many applications in terms of minimal complexity since they are modeled based on the algebraic and duality concepts, i.e., geometry and topology structure (Raman et al. 2016, 2017, Somu et al. 2016, 2017a, b, 2018a, b; Gauthama Raman et al. 2017a, b). In this subsection, we discuss few basic concepts

and definitions of hypergraph and clique property to improve the performance of binary fruit fly optimization algorithm (Berge 1973).

**Definition 1 Hypergraph** Let  $H_G = \{V, E\}$ , be a hypergraph, then the representative graph ( $R_G = \{V_r, E_r\}$ ) of hypergraph is defined as follows (Fig. 1):

- (a)  $V_r = n$  or  $V_r = E$ , when hypergraph  $H$  has no repeated hyperedges
- (b)  $\{a, b\} \in E | a \neq b$ , if and only if  $e_a \cap e_b \neq \emptyset$

**Definition 2 Neighborhood hypergraph** Let  $R_G = \{V_r, E_r\}$  be a graph and  $m \subseteq V_r$ . Then, the closed neighbor  $L$  in  $R_G$  is defined in Eq. (9).

$$C_L(m) = \{L \in V_r | L \text{ is adjacent of } m \text{ in } R_G\} \cup \{m\} \tag{9}$$

**Definition 3 Complete graph:** A graph ( $R_G = \{V_r, E_r\}$ ) is said to be complete, if there exists edge  $e \in E_r$  between the vertices  $(V_i, V_j) \in V_r \forall i, j = \{1, 2, \dots, n'\} | i \neq j$  where  $n'$  is the number of vertices in  $R_G$ .

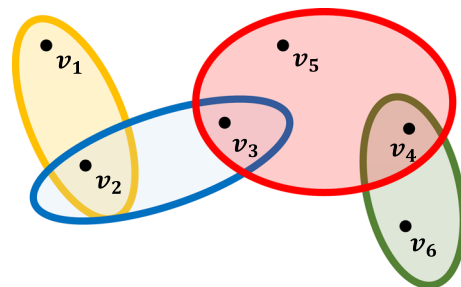
**Definition 4 K-clique** Given a graph  $R_G = \{V_r, E_r\}$ ,  $K$ -clique of  $R_G$  is a subset  $D \subset V_r$  and  $K = |D|$ , such that  $D$  obeys definition (Fig. 2).

**Proposition 1** Consider a  $R_G = \{V_r, E_r\}$  and clique  $D \subset V_r$ , then  $D$  is the clique of  $R_G$ , if and only if  $\bigcap_{i=1}^m D_L(V_i)$ .

**Proof** Let  $D$  be a clique of graph  $R_G$  and  $q \subseteq V_r$ , then  $q \in D_L(V_i), \forall i = (1, 2, \dots, m) \& D \subset \bigcap_{i=1}^m D_L(V_i)$ . Conversely, consider  $D \subset \bigcap_{i=1}^m D_L(V_i)$ . Let  $g$  and  $h$  be two distinct elements of  $D$ , then from the hypothesis  $q \in D_L(V_i), \forall i = (1, 2, \dots, m)$ ,  $g$  and  $h$  are adjacent. Hence,  $D$  is the clique of  $R_G$  (Fig. 3).

### 2.3 Fruit fly optimization algorithm

Fruit fly optimization algorithm (FFOA) is a novel meta-heuristic technique based on the food foraging behavior of



**Fig. 1**  $H = \{V, E\}; V = \{v_1, v_2, v_3, v_4, v_5, v_6\}; e = \{e_1, e_2, e_3, e_4\}; e_1 = \{v_1, v_2\}; e_2 = \{v_2, v_3\}; e_3 = \{v_3, v_4, v_5\}; e_4 = \{v_4, v_5\}$

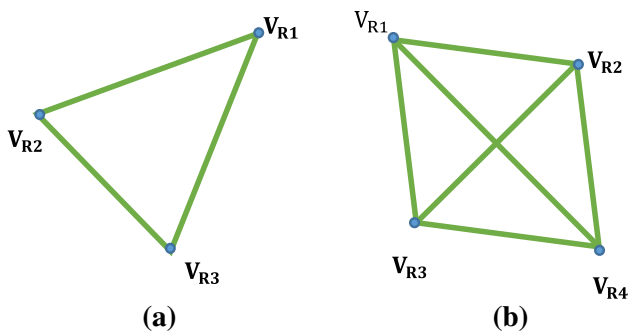


Fig. 2 a 3-clique, b 4-clique

fruit flies to identify the global best solution for optimization problems (Pan 2012; Dai et al. 2014; Shen et al. 2016). The fruit flies have a superior sense of smell (osphresis) and perception (vision) compared to other species. The osphresis organ of a fruit fly can collect all kinds of smell in the environment. It can smell the food source at a distance of 40 km. Once a fly reaches a location nearer to the food source, it makes use of its sensitive vision and the company’s flocking location to locate the food. Figure 4 depicts the food foraging behavior of the fruit flies (particles) in the search space. Initially, the particles begin to traverse the search space in search for an optimum solution in a randomized manner. During each iteration, the fitness of each particle was assessed using the fitness evaluation function in terms of particle’s current location. The smell concentration of each particle represents their fitness value. On to the subsequent iterations, the

particles fly based on the location of the particle with maximum fitness value. Based on the food foraging behavior of the fruit flies, the steps in FFOA can be categorized as follows:

*Step 1* The major parameters of the FFOA, such as maximum number of generations ( $Max_{Gen}$ ), population size ( $Pop_{Size}$ ), number of population ( $Pop_{Num}$ ), fitness value ( $Fitness$ ), local best smell concentration ( $Best_{Smell}$ ), global best smell concentration ( $G_{Best_{Smell}}$ ), best position ( $Best_{Pos}$ ), and random position of the particles were initialized with appropriate values as defined in Eqs. (10) and (11).

$$X_{Axis} = Rand() \tag{10}$$

$$Y_{Axis} = Rand() \tag{11}$$

*Step 2* The particles were stimulated to construct the population and traverse the search space using the osphresis gland for its food foraging process using Eqs. (12) and (13).

$$X_i = X_{Axis} + Random\ value \tag{12}$$

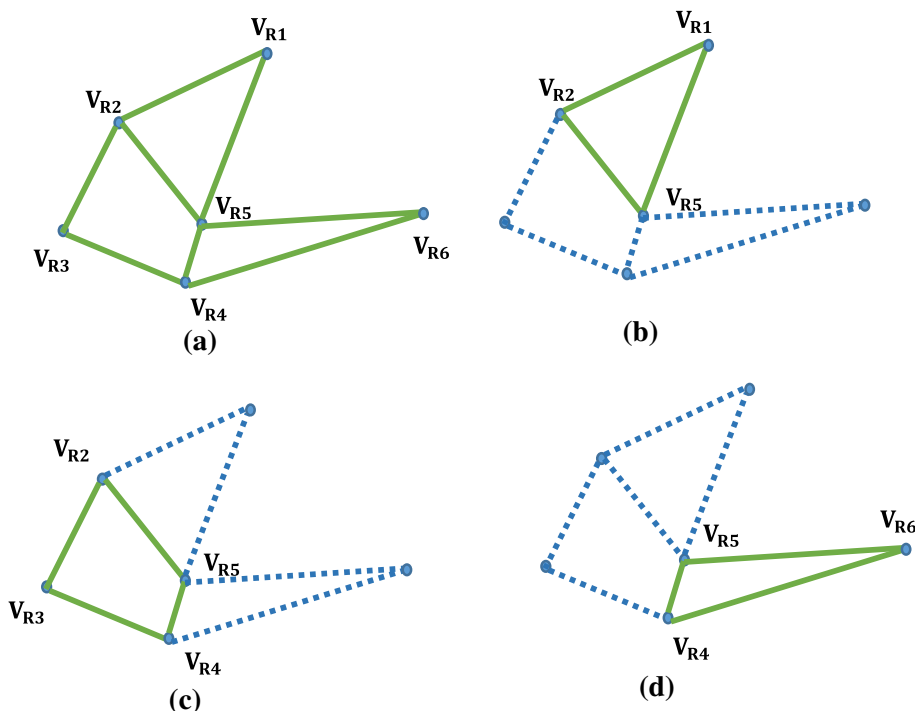
$$Y_i = Y_{Axis} + Random\ Value \tag{13}$$

*Step 3* For each iteration, the fitness of each particle was assessed by substituting its current position in the fitness evaluation function (Eqs. (14) and (15)).

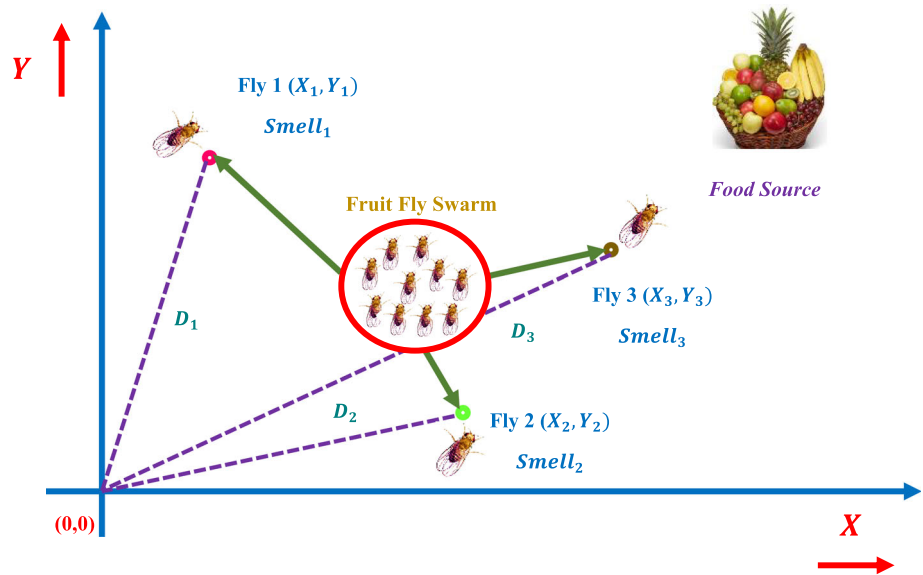
$$S_i = \frac{1}{D_i}; D_i = \sqrt{X_i^2 + Y_i^2} \tag{14}$$

$$Smell_i = Fitness(S_i) \tag{15}$$

Fig. 3 a A graph ( $G_{Rep}$ ) with 6 vertices and 8 edges; b, c, and d possible cliques of  $G_{Rep}$  is  $D_1=\{V_{R1}, V_{R2}, V_{R5}\}$ ,  $D_2=\{V_{R2}, V_{R3}, V_{R4}, V_{R5}\}$ , and  $D_3=\{V_{R6}, V_{R5}, V_{R4}\}$



**Fig. 4** Fruit fly's food foraging behavior



*Step 4* The smell concentration and position coordinates of the fittest particle (a particle with maximum fitness value) were retained. For subsequent iterations, the particles traverse the search space with respect to the position of the fittest particle (Eqs. 16–19).

$$Best_{Smell} = Max(Smell_i) \tag{16}$$

$$(G_{Best_{Smell}}) = Best_{Smell} \tag{17}$$

$$X_{Axis} = X(Best\ index) \tag{18}$$

$$Y_{Axis} = Y(Best\ index) \tag{19}$$

*Step 5* Steps 2 and 4 were repeated in an iterative manner for a maximum number of generations or until an optimal solution is attained.

### 3 RST-HGBFFO: the proposed trust measure parameter selection technique

Feature selection or dimensionality reduction is a primary and indispensable phase in many data analytic applications. It is the process of finding the minimal set of attributes or

reducts that reveals the maximum information content of the original dataset (Deogun et al. 1998). The major objective behind the design of any feature selection technique is to increase the accuracy and reduce the computation burden of the learning model. In general, feature selection techniques evolve from the decision system ( $D_S$ ) which consists of ' $S$ ' observations  $O = \{O_1, O_2, \dots, O_S\}$ , ' $A$ ' conditional attributes  $A = \{A_1, A_2, \dots, A_C\}$ , and a decisional attribute ( $D$ ) (Table 3) (Jia-Yuarn 2003; Hu 2012). Since all the conditional attributes in the decision table may not be important at all times, the importance of these attributes can be determined by the measure of relevancy and redundancy (Nina 2007). An attribute or feature is relevant, when it is highly predictive of the decisional attributes ( $D$ ), and is redundant, when it is highly correlated with the neighboring conditional attributes ( $D$ ). Hence, an optimal reduct must consist of features that are independent on each other ( $A$ ) and closely related to the decisional attribute ( $D$ ). Further, the performance of any learning model can be improved in terms of enhanced accuracy and reduced complexity due to minimized search space through the identification of optimal reducts (Jensen and Shen 2003).

**Table 3** Decision system

Observations ( $O$ )	Conditional attributes ( $A$ )					Decisional attribute ( $D$ )
	$A_1$	$A_2$	$A_3$	...	$A_C$	
$O_1$	$V_{11}$	$V_{12}$	$V_{13}$	...	$V_{1C}$	$D_1$
$O_2$	$V_{21}$	$V_{22}$	$V_{23}$	...	$V_{2C}$	$D_2$
$O_3$	$V_{31}$	$V_{32}$	$V_{33}$	...	$V_{3C}$	$D_3$
.	.	.	.	...	.	.
.	.	.	.	...	.	.
.	.	.	.	...	.	.
$O_S$	$V_{S1}$	$V_{S2}$	$V_{S3}$	...	$V_{SC}$	$D_D$

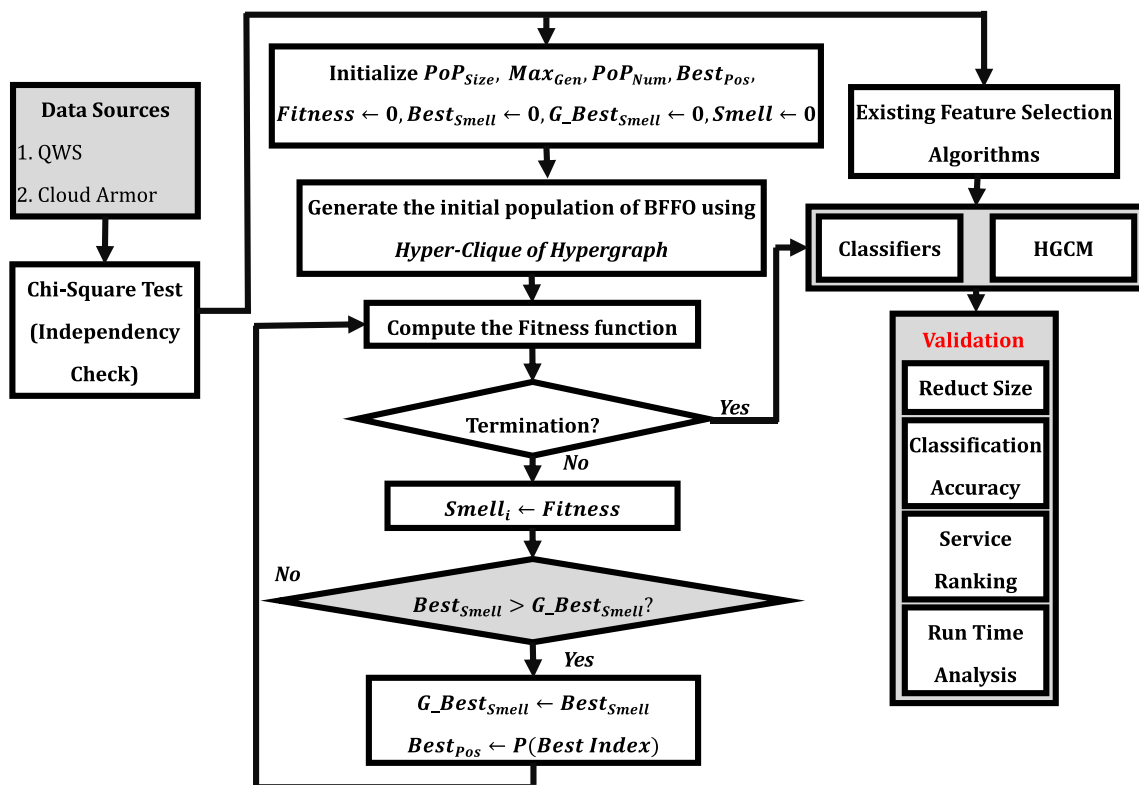
The prominence of RST over other statistical and machine learning techniques has motivated the researchers and academicians toward the exploitation of RST for various research problems in medical diagnosis (Somu et al. 2016), intrusion detection systems (Raman et al. 2016; Gauthama Raman et al. 2017a), cloud service selection (Liu et al. 2016; Somu et al. 2017a), etc. A more generic way to obtain the reducts using RST is to generate all possible combinations of the feature subsets and identify the subset that maximizes the dependency function (Zhong et al. 2001; Jensen and Shen 2004; Nina 2007; Wang et al. 2007). However, the above-said approach holds good for small volume datasets, while in the case of massive and high-dimensional datasets, the complexity of the learning model increases exponentially (Somu et al. 2016). Therefore, for large-scale datasets, SQR and QRR were used to obtain the reducts (Velayutham and Thangavel 2011; Chen

et al. 2015). The execution of SQR begins with an empty set, followed by the iterative addition of substantial features that maximize the dependency function. The major problem with this approach is the lack of global optima (traps at local minima) due to the instability in the dependency metric experienced by the iterative inclusion and exclusion of features. To avoid this, QRR replaces the dependency metric of RST by relative dependency function (Inbarani et al. 2014).

This work presents RST-HGBFFO, a cooperative bio-inspired algorithm which hybridizes the benefits of hypergraph and binary fruit fly optimization technique with RST (SQR and QRR) for the identification of optimal service-specific trust measure parameters or conditional attributes with minimum time complexity (Algorithm 1). The main objective of the proposed approach is to identify the optimal reducts, i.e., service-specific trust measure parameters thereby minimizing the computational and time complexity of the cloud service selection model. RST-HGBFFO operates over a  $m$ -dimensional binary search space with  $PoP_{Num}$  fruit flies or particles which are represented by a ' $m$ ' tuple vector (particle's location). Each tuple vector varies in its length with respect to the number of features ( $A$ ). For example, a four-tuple vector  $\langle 0011 \rangle$  corresponds to the presence of  $A_3, A_4$  and the absence of  $A_1, A_2$  (Table 4). Figure 5 presents a generic data flow of RST-HGBFFO.

**Table 4**  $n$ -tuple vector and feature subset

Conditional features				Feature subset
$A_1$	$A_2$	$A_3$	$A_4$	
1	1	0	1	$(A_1, A_2, A_4)$
0	0	1	1	$(A_3, A_4)$
1	1	1	0	$(A_1, A_2, A_3)$
0	0	1	0	$(A_3)$



**Fig. 5** Rough set theory-based hypergraph-binary fruit fly optimization trust measure parameter selection technique



**Algorithm 1: RST-HGBFFO: Rough Set Theory – Hypergraph based Binary Fruit Fly Optimization****Input**

- A ← Set of Conditional Attributes  
D ← Set of Decisional Attributes

**Output**

- $O_{TMP}$  ← Optimal TMP

**RST-HGBFFO ( )****\*\*\* Initialization Phase \*\*\***

1. Initialize the value of Population size ( $PoP_{Size}$ ), Number of Population ( $PoP_{Num}$ ), Maximum Generation ( $Max_{Gen}$ ), Fitness ( $Fitness \leftarrow 0$ ), Local Best smell ( $Best_{Smell} \leftarrow 0$ ), Gobal best smell ( $G_{Best_{Smell}} \leftarrow 0$ )

**\*\*\* Generation of the Initial population using Hyper clique property \*\*\***

2. Generate a matrix  $[M]_{PoP_{Num} \times PoP_{Size}}$  with randomly generated binary string
3. Construct the hypergraph ( $H_G$ ) using  $[M]$  and apply hyper clique property for refined population

**\*\*\* Identification of Optimal TMPs \*\*\***

4. **for each**  $i \leftarrow 1$  **to**  $PoP_{Size}$  **begin**
5. Obtain the position of each fruit fly using Line 3
6.  $TMP \leftarrow$  Feature Subset  $F_f$
7.  $\forall x \in (A - TMP)$

**\*\*\* SQR Algorithm \*\*\***

8.  $\gamma_{TMP \cup \{x\}} \leftarrow \frac{|PoS_{TMP \cup \{x\}}(D)|}{|U|}$
9.  $Fitness_{SQR} \leftarrow \gamma_{R \cup \{x\}}(D) \forall x \subset R, \gamma_{\{x\}}(D) = \gamma_{\{A\}}(D)$

**\*\*\* QRR Algorithm \*\*\***

10.  $\gamma_{TMP \cup \{x\}} \leftarrow \frac{|U/IND(TMP)|}{|U/IND(TMP \cup D)|}$
11.  $Fitness_{QRR} \leftarrow \gamma_{TMP - \{x\}}(D) \forall x \subset R, \gamma_{\{x\}}(D) = \gamma_{\{A\}}(D)$
12.  $Smell_i \leftarrow Fitness$
13. **end**
14.  $Best_{Smell} \leftarrow Max(Smell)$
15.  $G_{Best_{Smell}} \leftarrow Best_{Smell}$

```

16.  $Best_{pos} \leftarrow P(BestIndex)$ 
17. While(!(Fitness == 1)) begin
18.   for each  $g \leftarrow 1$  to  $Max_{Gen}$  begin
19.     for each  $i \leftarrow 1$  to  $PoP_{Size}$  begin
20.       Obtain the position of each fruit fly using  $Best_{pos}$ 
21.        $TMP \leftarrow Feature\ Subset\ F_f$ 
22.        $\forall x \in (A - TMP)$ 
                *** SQR Algorithm ***
23.        $\gamma_{TMP \cup \{x\}} \leftarrow \frac{|PoS_{TMP \cup \{x\}}(D)|}{|U|}$ 
24.        $Fitness_{SQR} \leftarrow \gamma_{R \cup \{x\}}(D) \forall x \subset R, \gamma_{\{x\}}(D) = \gamma_{\{A\}}(D)$ 
                *** QRR Algorithm ***
25.        $\gamma_{TMP \cup \{x\}} \leftarrow \frac{|U/IND(TMP)|}{|U/IND(TMP \cup D)|}$ 
26.        $Fitness_{QRR} \leftarrow \gamma_{TMP - \{x\}}(D) \forall x \subset R, \gamma_{\{x\}}(D) = \gamma_{\{A\}}(D)$ 
27.        $Smell_i \leftarrow Fitness$ 
28.     end
29.   end
30.   if ( $Best_{Smell} > G_{Best_{Smell}}$ ) begin
31.      $G_{Best_{Smell}} \leftarrow Best_{Smell}$ 
32.      $Best_{pos} \leftarrow P(Best\ Index)$ 
33.   end
34.    $O_{TMP} \leftarrow P(Best\ Index)$ 
35. end
36. return  $TMP$ 
37. end

```

The detailed explanation of the working of RST-HGBFFO is given as follows:

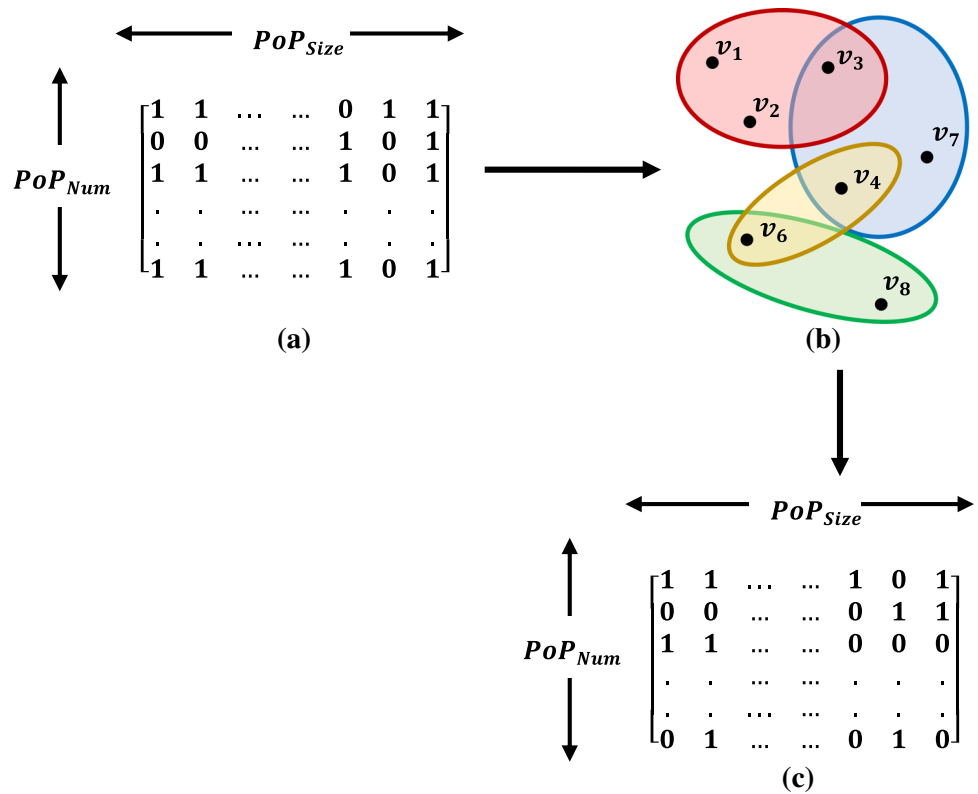
*Phase 1—Initialization phase* It initializes the population size ( $PoP_{Size}$ ), maximum generation ( $Max_{Gen}$ ), fitness function ( $Fitness$ ), local best smell concentration ( $Best_{Smell}$ ), global best smell concentration ( $G_{Best_{Smell}}$ ), best position ( $Best_{pos}$ ), and number of particles ( $PoP_{Num}$ ).

*Phase 2—Generation of initial population* In general, the position of each particle in the traditional BFFO was

initialized in a random fashion. Similarly, the initial population of RST-HGBFFO was obtained through a step-by-step procedure: (i) construct a two-dimensional matrix  $[M]_{PoP_{Num} \times PoP_{Size}}$  with randomly generated binary string (Fig. 6(a)), (ii) construct a hypergraph ( $H_G$ ) using  $[M]_{PoP_{Num} \times PoP_{Size}}$  (Fig. 6b), and (iii) application of hyperclique property on  $H_G$  (Definition 4) to obtain the refined population (Gauthama Raman et al. 2017b) (Fig. 6c).

*Phase 3—Fitness evaluation* Each initial set of particles or population obtained from phase 1 was evaluated using

**Fig. 6** Generation of initial population: **a** random population, **b** construction of hypergraph and application of hyperclique property, **c** initial population



the fitness function defined based on the attribute dependency measure of RST. The fitness evaluation function of RST-HGBFFO was designed with respect to the dependency metric ( $Fitness_{SQR}$ ) and relative dependency metric ( $Fitness_{QRR}$ ) of RST given in Eqs. (20) and (21), respectively.

$$Fitness_{SQR} = \gamma_R(D) = \frac{|PoS_R(D)|}{|U|} \tag{20}$$

$$Fitness_{SRR} = \gamma_R(D) = \frac{|U/IND(R)|}{|U/IND(R \cup D)|} \tag{21}$$

**Phase 4—Updation** Once the fitness of each population was computed using the fitness evaluation function [Eqs. (20) and (21)], the value of the *Smell* function ( $Best_{Smell}$ ) was updated with the fitness value of the fittest particle, i.e., the population with high fitness value (dependency metric). For each iteration, the global best smell value ( $G_{Best_{Smell}}$ ) was updated based on the local best smell ( $Best_{Smell}$ ), i.e., the smell concentration value of the ( $G_{Best_{Smell}}$ ) and  $Best_{Smell}$  was compared against each other. Further, the  $G_{Best_{Smell}}$  is updated with the high smell concentration value ( $Best_{Smell} > G_{Best_{Smell}}; G_{Best_{Smell}} \leftarrow Best_{Smell}$ ). The position of the fittest particle is stored in  $Best_{PoS}$  and set as a seed for the subsequent iteration.

**Phase 5—Generation of new population** Check for the termination condition (optimal solution ( $Fitness \cong 1$ ) or

$Max_{Gen}$ ). If the optimal solution was found, return the  $Best_{PoS}$ , i.e., the optimal TMP. Else, repeat phase 3 and 4.

### 3.1 Example

Let us consider a decision table (Table 5) with 20 observations represented in the form of seven conditional attributes ( $A_1, A_2, \dots, A_7$ ) and a decisional attribute ( $D$ ). The execution of RST-HGBFFO TMP selection technique begins with the initialization of  $Max_{Gen} = 3; PoP_{Size} = 7; PoP_{Num} = 20, Fitness = 0; Best_{Smell} = 0; G_{Best_{Smell}} = 0$  and  $Best_{PoS} = \langle 0, 0, 0, 0, 0, 0, 0 \rangle$  (Table 6). The hyperclique property of hypergraph was exploited for the generation of initial population of BFFO (Fig. 7). The particles fly over the search space from their initial random position in search for the optimal reduct based on their sense of smell and vision. Over a certain period, the particles get displaced to a new position ( $P_1 = 0, 1, 0, 1, 1, 1, 0; P_2 = 1, 0, 1, 0, 1, 1, 1; P_3 = 1, 0, 1, 1, 0, 1, 1; P_4 = \langle 0, 0, 1, 1, 1, 1 \rangle$ ), where 0 and 1 represent the absence and presence of the feature, respectively. The fitness of each particle was evaluated based on the particle’s current location using the fitness evaluation function of SQR and QRR given in Algorithm 1 ( $P_1 = 0.34; P_2 = 0.60; P_3 = 0.21; P_4 = 0.53$ ) (Table 7).

The smell concentration value of the particle with high fitness value was stored in  $Best_{Smell}$  and compared against

**Table 5** Decision table: an example

Observations ( <i>O</i> )	Conditional attributes ( <i>A</i> )							Decisional attributes ( <i>D</i> )
	<i>A</i> <sub>1</sub>	<i>A</i> <sub>2</sub>	<i>A</i> <sub>3</sub>	<i>A</i> <sub>4</sub>	<i>A</i> <sub>5</sub>	<i>A</i> <sub>6</sub>	<i>A</i> <sub>7</sub>	
1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	2	2	1
·	·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·	·
18	2	1	2	2	2	1	2	1
19	1	2	3	3	1	2	1	1
20	1	2	2	2	1	1	1	1

**Table 6** Initialization phase

S. no	Parameters	Value
1	<i>Fitness</i>	0
2	<i>BestSmell</i>	0
3	( <i>G</i> <sub><i>BestSmell</i></sub> )	0
4	<i>Smell</i>	0
5	<i>BestPos</i>	⟨0,0,0,0,0,0,0⟩

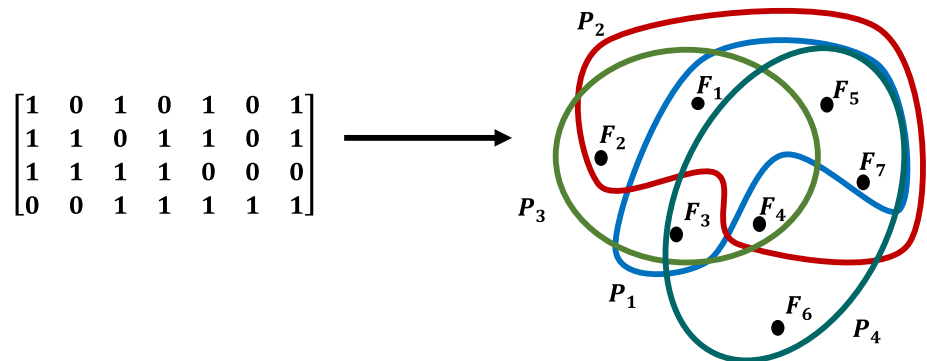
**Table 8** Fitness and position update

S. no	Parameters	Value
1	<i>BestSmell</i>	0.60
2	<i>G</i> <sub><i>BestSmell</i></sub>	0.60
3	<i>BestPos</i>	⟨1,0,1,0,1,1,1⟩

$G_{BestSmell} (G_{BestSmell} = 0)$ . The comparison resulted in the replacement of the smell concentration of  $G_{BestSmell}$  with higher smell concentration value ( $BestSmell = 0.60$ ;  $G_{BestSmell} = 0$ ;  $BestSmell > G_{BestSmell}$ ;  $G_{BestSmell} = 0.60$ ). The position of the particle with high fitness value was stored in  $BestPos (BestPos = 1, 0, 1, 0, 1, 1, 1)$  (Table 8).

In the subsequent iteration, the random position of the particles was initialized based on  $BestPos = 1, 0, 1, 0, 1, 1, 1$ . The displaced position of the particles  $P_{11}, P_{12}, P_{13}$  and  $P_{14}$  after time interval  $t$  was  $1, 1, 1, 0, 1, 1, 1$ ,  $1, 0, 0, 1, 1, 1, 0$ ,  $\langle 1, 1, 1, 1, 0, 0, 1 \rangle$ , and  $0, 1, 1, 0, 1, 1, 1$  (Table 9). The fitness value of each particle based on their current position was computed as  $P_{11} =$

**Fig. 7** Construction of hypergraph and application of hyperclique property



**Table 7** Iteration 1—optimal fitness value and reduct

Particle ( <i>P<sub>f</sub></i> )	Conditional attributes							Feature subset	Fitness value
	<i>A</i> <sub>1</sub>	<i>A</i> <sub>2</sub>	<i>A</i> <sub>3</sub>	<i>A</i> <sub>4</sub>	<i>A</i> <sub>5</sub>	<i>A</i> <sub>6</sub>	<i>A</i> <sub>7</sub>		
<i>P</i> <sub>1</sub>	0	1	0	1	1	1	0	{ <i>A</i> <sub>2</sub> , <i>A</i> <sub>4</sub> , <i>A</i> <sub>5</sub> , <i>A</i> <sub>6</sub> }	0.34
<i>P</i> <sub>2</sub>	1	0	1	0	1	1	1	{ <i>A</i> <sub>1</sub> , <i>A</i> <sub>3</sub> , <i>A</i> <sub>5</sub> , <i>A</i> <sub>6</sub> , <i>A</i> <sub>7</sub> }	0.60
<i>P</i> <sub>3</sub>	1	0	1	1	0	1	1	{ <i>A</i> <sub>1</sub> , <i>A</i> <sub>3</sub> , <i>A</i> <sub>4</sub> , <i>A</i> <sub>6</sub> , <i>A</i> <sub>7</sub> }	0.21
<i>P</i> <sub>4</sub>	0	0	1	1	1	1	1	{ <i>A</i> <sub>3</sub> , <i>A</i> <sub>4</sub> , <i>A</i> <sub>5</sub> , <i>A</i> <sub>6</sub> , <i>A</i> <sub>7</sub> }	0.53

**Table 9** Iteration 1—optimal fitness value and reduct

Particle ( $P_f$ )	Conditional attributes							Feature subset	Fitness value
	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$		
$P_{11}$	1	1	1	0	1	1	1	$\{A_1, A_2, A_3, A_5, A_6, A_7\}$	0.17
$P_{12}$	1	0	0	1	1	1	0	$\{A_1, A_4, A_5, A_6\}$	1
$P_{13}$	1	1	1	1	0	0	1	$\{A_1, A_2, A_3, A_4, A_7\}$	0.21
$P_{14}$	0	1	1	0	1	1	1	$\{A_2, A_3, A_5, A_6, A_7\}$	0.52

0.17,  $P_{12} = 1$ ,  $P_{13} = 0.21$ , and  $P_{14} = 0.52$  (Table 9). An optimality check on the fitness value resulted in the attainment of the optimal solution ( $P_{13} = 1, 0, 0, 1, 1, 1, 0 = 1$ ). The smell concentration value of  $G_{BestSmell}$  was updated with the optimal fitness value ( $BestSmell = 1; G_{BestSmell} = 0.60; BestSmell > G_{BestSmell}; G_{BestSmell} = 1$ ). The position of the fittest particle stored in  $BestPos$  ( $BestPos = 1, 0, 0, 1, 1, 1, 0$ ) represents the optimal reduct ( $TMP$ ), which can be decoded as  $A_1, A_4, A_5, A_6$ .

### 4 Experimental results and discussion

The implementation of RST-HGBFFO was carried out using Python 3.6 (Scikit-rough sets package) on an Intel® Core™ i5 processor @ 2.40 GHz system running Windows 7 OS with 16 GB RAM. The performance evaluation of RST-HGBFFO was validated using two benchmark datasets, namely QWS dataset, and Cloud Armor trust feedback dataset & a synthetic trust feedback dataset obtained from the Centre for Information Super Highway (CISH), SASTRA Deemed University (Sheng; Al-Masri and Mahmoud 2007).

#### 4.1 Dataset description

- (i) *QWS dataset* QWS is a public QoS dataset owned and maintained by E. Al-Masri and Q.H Mahmoud, University of Guelph, Canada (Al-Masri and Mahmoud 2007). This QoS dataset is widely used in several research works and evaluation studies on QoS in service-oriented environments. It contains QoS records of 365 real-world web services which were collected using Web Service Crawler Engine (WSCE) developed by Al-Masri and Mahmoud. These services were tested over a period of 10 min for three continuous days. The trust rate or service classification of each service was assessed using the web service relevancy function (WsRF) with respect to various quality metrics like availability, response time, security, etc. (Table 10). The web services were classified into four categories, namely platinum (high quality), gold, silver, and bronze (low quality) based

**Table 10** Trust measure parameters in QWS dataset ( $TMP_1 - TMP_{12}$  : conditional attributes; *service classification*: decisional attribute)

Trust measure parameters (QoS attributes)			
$TMP_1$	Response time	$TMP_6$	Compliance
$TMP_2$	Availability	$TMP_7$	Best practices
$TMP_3$	Throughput	$TMP_8$	Latency
$TMP_4$	Successability	$TMP_9$	Documentation
$TMP_5$	Reliability	$TMP_{10}$	WSRF

on the rating provided by WsRF. The service offering qualities of the web services were represented in numerical form (1 to 4).

- (ii) *Cloud Armor trust feedback dataset* It was obtained from Cloud Armor, a research project at University of Adelaide which deals with the development of a robust and scalable trust management model for cloud environments. The Cloud Armor trust feedback dataset comprises 10,080 QoS feedbacks given by approximately 7,000 customers over various time stamps (9 years) for 114 real-world cloud services (Table 11).
- (iii) *Synthetic trust feedback dataset at CISH, SASTRA University* It comprises the trust feedbacks obtained from the students for a diverse set of cloud services (academic and student) provided by the private cloud at CISH, SASTRA University. This dataset consists of users' feedbacks in the range of [0, 5] for 12 trust measure parameters

**Table 11** Trust measure parameters in Cloud Armor and synthetic trust feedback dataset ( $TMP_1 - TMP_{10}$  : conditional attributes; *trust result*: decisional attribute)

Trust measure parameters (QoS attributes)			
$TMP_1$	Availability	$TMP_7$	Storage space
$TMP_2$	Security	$TMP_8$	Features
$TMP_3$	Response time	$TMP_9$	Ease of use
$TMP_4$	Accessibility	$TMP_{10}$	Technical support
$TMP_5$	Price	$TMP_{11}$	Customer service
$TMP_6$	Speed	$TMP_{12}$	Level of expertise

(availability, security, response time, etc.) and corresponding trust result (Table 11). Since the significance of the TMPs varies with respect to the service type requested by the users and their preferences, the cloud services in the considered datasets were grouped based on their service type. The TMPs or quality service attributes given in Tables 10 and 11 represent the KPIs in the CSMIC-SMI metrics (Valley 2011; Somu et al. 2017b).

RST-HGBFFO employs CSMIC-SMI as a standard metric for the QoS attributes or TMPs (Valley 2011). The CSMIC-SMI metric is generally viewed as a hierarchy of categories, attributes, and sub-attributes or key performance indicators (KPIs). At the first level of the CSMIC-SMI hierarchy, the entire metric space is divided into seven categories. Further, each category is partitioned into four or more attributes. At the extremity, the attributes are further divided into one or more sub-attributes or KPIs, whose actual count varies with the nature of cloud business solutions. The CSMIC-SMI hierarchy reveals the high-dimensional nature of the cloud objective (monitoring QoS value) and subjective (user feedbacks) assessment data with respect to the QoS attributes and QoS records (samples). However, due to the unavailability of appropriate benchmark datasets, we have used two benchmark datasets (QWS dataset and Cloud Armor synthetic feedback dataset) and one synthetic dataset (CISH—SASTRA trust feedback dataset) to prove the efficiency of RST-HGBFFO over the state-of-the-art rough set-based feature selection techniques. Further, the relation between the QoS attributes in the considered datasets and CSMIC-SMI metrics is given in Tables 10 and 11.

## 4.2 Results and discussions

The experiments carried out can be divided into three phases, namely data preprocessing, generation of reducts, and performance validation. At the initial phase, the considered datasets were processed using appropriate data preprocessing techniques and the training and testing datasets were generated in 80:20 ratio. In the subsequent phase, the reducts were obtained from the existing and the proposed (RST-HGBFFO) feature selection techniques. Finally, the performance of RST-HGBFFO was evaluated using Weka tool and HGCM in terms of reduct size, service ranking, classification accuracy, and runtime analysis (Øhrn 2001; Somu et al. 2017a).

### 4.2.1 Phase 1—data preprocessing

At the initial stage of the experiment, each sample in the considered dataset was transformed into a compatible format

supported by the classifiers. Further, data normalization was used to minimize the impact of the features with high value. Each feature in the sample is normalized in such a way that all the values lie in the range of [0,1] [Eq. (22)].

$$f_{ab} = \frac{f_{ab}}{\sum_{b=1}^n f_{ab}}; \forall a = (1, 2, \dots, S); b = (1, 2, \dots, n) \quad (22)$$

where  $S$  and  $n$  are total number of samples and features in the dataset, respectively.

The discrete and continuous values in the synthetic trust feedback datasets were converted into a compatible format by data mapping technique, i.e., mapping the discrete and continuous values to numerical values ranging from 1 to  $A_C$ , where  $A_C$  represents the total number of conditional attributes. Further, the considered dataset was subjected to chi-square test with  $(r - 1, c - 1)$  degrees of freedom, where  $r$  and  $c$  represent the number of rows and columns, respectively, to check the degree of dependency among the QoS attributes. During the experiments, it was noted that a minimum of 80% of the TMPs were found to be dependent on each other, even at a level of 10% significance. Hence, the inter-relation between the QoS attributes has to be exploited for identification of the optimal feature subset. Finally, the training and testing datasets for each considered dataset were generated in 80:20 ratio.

### 4.2.2 Phase 2: generation of reducts

In this phase, the optimal TMP reduct was generated for RST-HGBFFO and the state-of-the-art feature selection techniques. The reducts for the existing rough set theory-based feature selection techniques like RSARSubsetEval\_Genetic Search ( $RST_{GS}$ ), RSARSubsetEval\_PSO Search ( $RST_{PSO}$ ), RSARSubsetEval\_Cuckoo Search ( $RST_{CU}$ ), RSARSubsetEval\_ANT Search ( $RST_{AN}$ ), RSARSubsetEval\_Harmony Search ( $RST_{HS}$ ), RSARSubsetEval\_Firefly Search ( $RST_{FF}$ ), and RSARSubsetEval\_Bat Search ( $RST_{BS}$ ) were generated by the integrating attribute evaluator (RSARSubsetEval) with the search techniques (PSO Search, Cuckoo Search, Genetic Search, etc.) (Witten et al. 2016). In addition, we have used few state-of-the-art classifiers like Bayes Net, SVM, K Star, BF Tree, J48, and Random forest in WEKA tool to validate the reducts obtained from the proposed and the existing rough set-based feature selection techniques in terms of classification accuracy. As mentioned earlier, the identification of all possible subsets in a high-dimensional and massive dataset is an NP-hard problem. Further, from the extensive survey, it is evident that meta-heuristic techniques outperform the statistical and graph-based techniques in solving NP-hard problems. Hence, the comparative analysis of RST-HGBFFO (SQR and QRR) was carried out using the state-of-the-art RST-based feature selection

techniques rather than our previous work, rough set hypergraph-based feature selection technique (RSHT) (Somu et al. 2017b). The critical parameter in RST-HGBFFO was set as  $Max_{Gen} = 100$ , and  $PoP_{Num}$  varies from 10 to 50 in the scale of 5; similarly, the default parameter values were used for the existing rough set-based feature selection techniques.

### 4.2.3 Phase 3: performance validation

The performance of RST-HGBFFO over the other RST-based feature selection techniques was evaluated using Weka tool and HGCM in terms of various quality metrics described as follows:

(i) Reduct size

Figures 8 and 9 present the impact of the number of population ( $Num_{Pop}$ ) on the fitness value of RST-HGBFFO (SQR and QRR) for the considered datasets. From Figs. 8

and 9, it can be observed that RST-HGBFFO achieves better dependency metric, i.e., the optimal fitness value (0.999) at  $Num_{Pop} = 50$ , i.e.,  $Best_{Pos}$ , the position of the fittest particle at  $Num_{Pop} = 50$  is presented as optimal reducts for RST-HGBFFO in Figs. 10, 11, and 12.

The reducts obtained from the existing and proposed (RST-HGBFFO) feature selection techniques (phase 2) were evaluated based on the reduct size. Based on the experiments, the following observations were made,

1. For QWS dataset, the optimal reduct provided by RST-HGBFFO (SQR and QRR) was similar to that of  $RST_{PSO}$ ,  $RST_{AN}$ ,  $RST_{HS}$ , and  $RST_{FF}$  (Fig. 10).
2. For Cloud Armor trust feedback dataset, the optimal reduct provided by RST-HGBFFO (SQR and QRR) was similar to that of  $RST_{CU}$ ,  $RST_{HS}$ , and  $RST_{BS}$  (Fig. 11).
3. For CISH-SASTRA trust feedback dataset, the optimal reduct provided by RST-HGBFFO (SQR

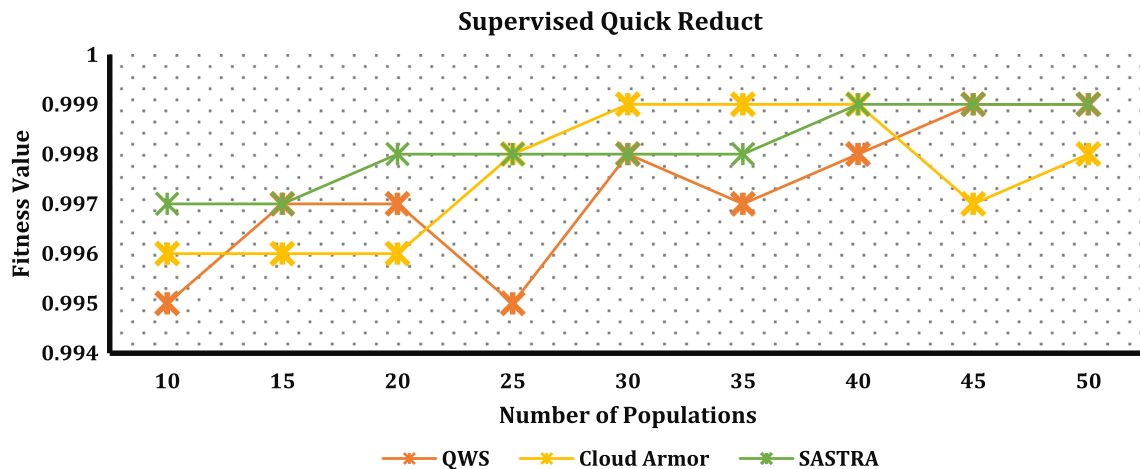


Fig. 8 Impact of population on the fitness value—SQR

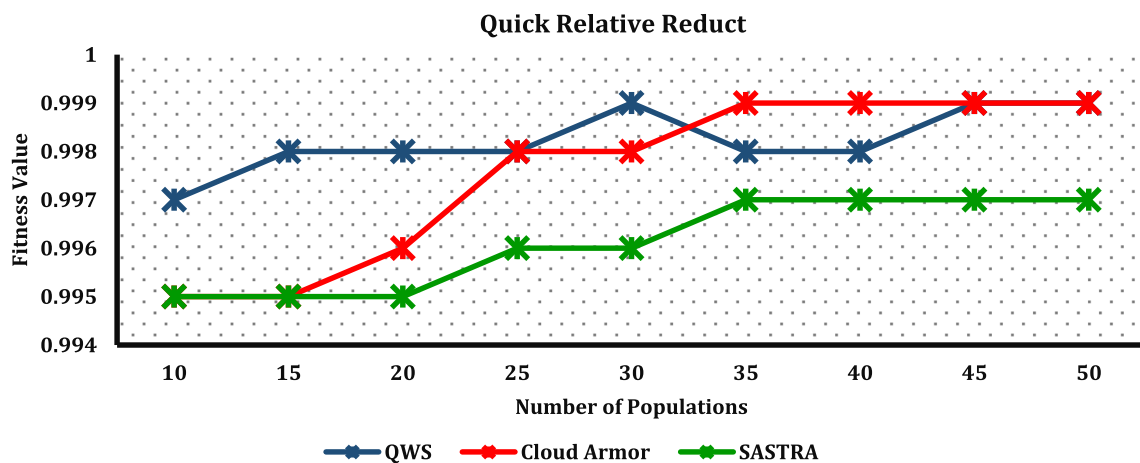


Fig. 9 Impact of population on the fitness value—QRR

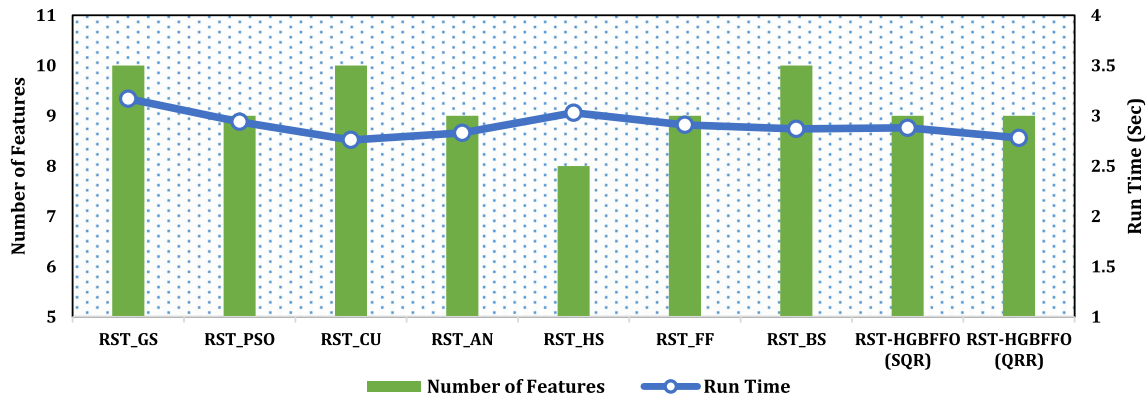


Fig. 10 Validation—reduct size (QWS dataset)

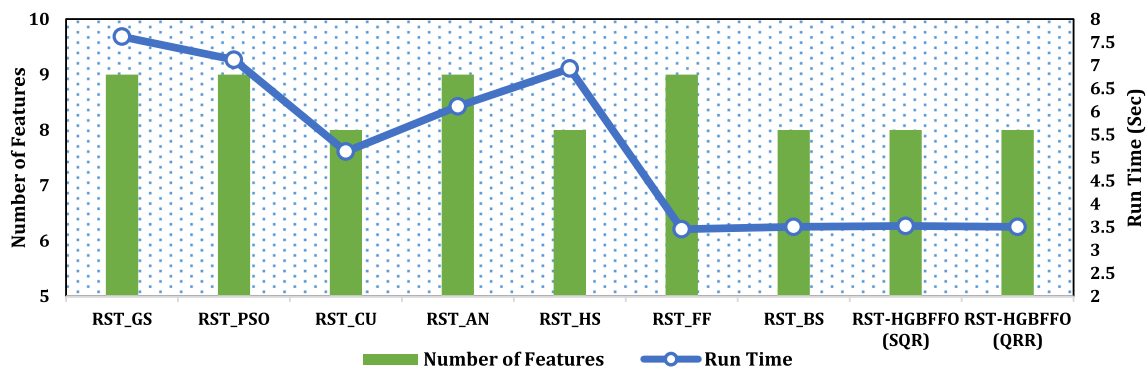


Fig. 11 Validation—reduct size (Cloud Armor trust feedback dataset)

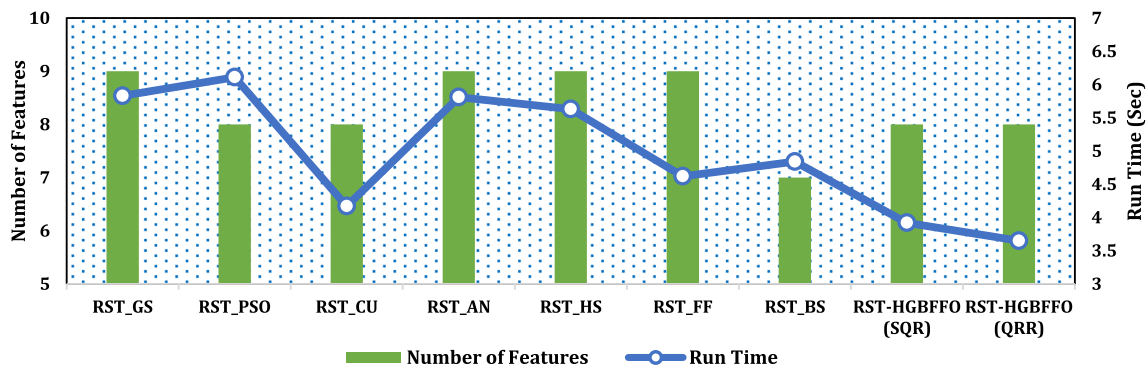


Fig. 12 Validation—reduct size (synthetic trust feedback dataset—CISH, SASTRA)

and QRR) was similar to that of  $RST_{PSO}$  and  $RST_{CU}$  (Fig. 12).

(ii) Classification accuracy

For further validation, the reducts obtained from the existing and proposed (RST-HGBFFO) feature selection techniques (phase 2) were evaluated based on the classification accuracy. Based on the experimental evaluations, it can be observed that the available classifiers (Bayes Net, SVM, K Star, BF Tree, J48, and Random Forest) in Weka

tool exhibited high classification accuracy when trained with the optimal reducts obtained from RST-HGBFFO (SQR and QRR) for the considered datasets (Figs. 13, 14, and 15). A noteworthy point is that RST-HGBFFO has similar performance to  $RST_{CU}$  when SVM was trained with the reducts obtained from QWS dataset.

The validation based on the reduct size, i.e., the cardinality of the reduct, shows that the performance of RST-HGBFFO was similar to the state-of-the-art rough set-based feature selection techniques (Figs. 10, 11, and 12).



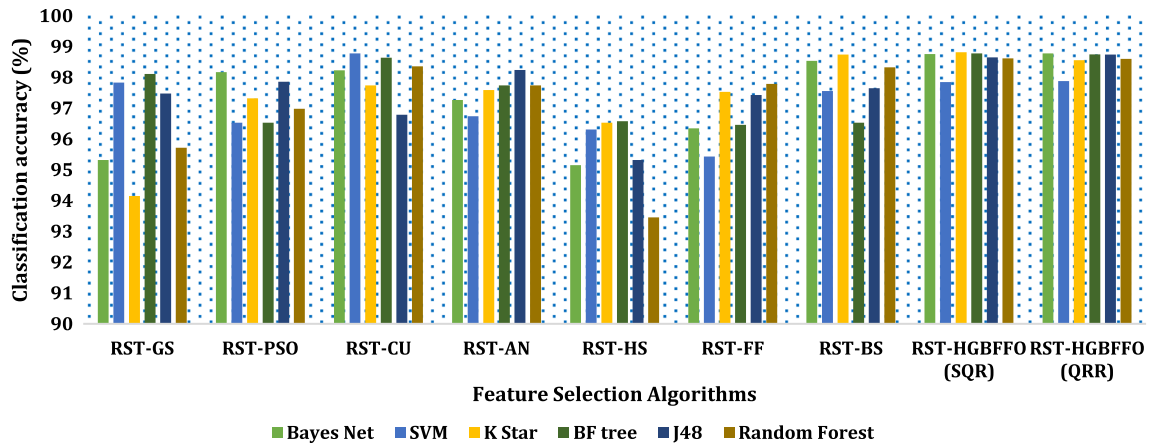


Fig. 13 Validation—classification accuracy (QWS dataset)

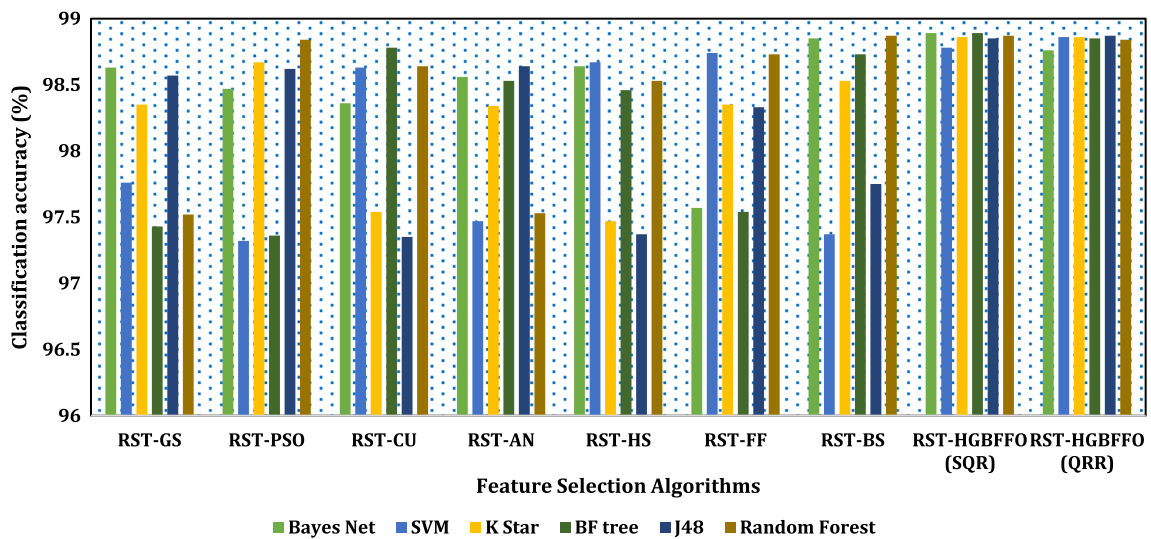


Fig. 14 Validation—classification accuracy (Cloud Armor trust feedback dataset)

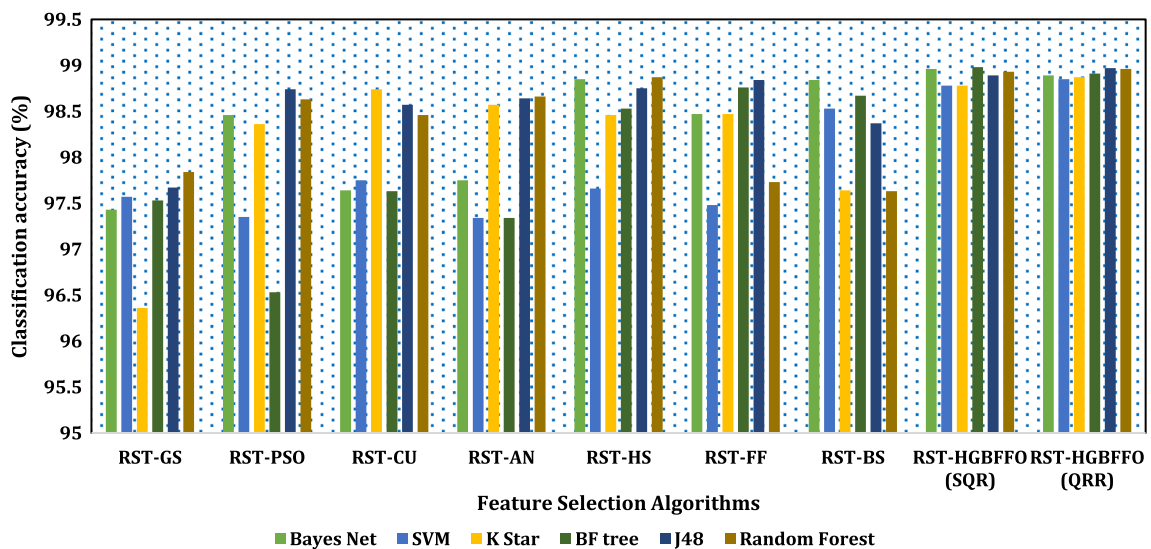


Fig. 15 Validation—classification accuracy (CISH—SASTRA trust feedback dataset)

The similarity in the cardinality of the feature subset does not imply the optimality of the features in the optimal feature subset. Even though RST-HGBFFO and few existing RST-based feature selection techniques have similar reduct size, the reducts obtained from RST-HGBFFO provide the optimal reduct that maximize the fitness function, i.e., dependency metric. Further, from Figs. 13, 14, and 15, it was evident that the classifiers exhibit high classification accuracy when trained with the optimal feature subset obtained from RST-HGBFFO.

(iii) Service ranking

The optimal TMP reduct obtained from phase 2 was fed as input to the hypergraph-based computational model (Fig. 5). HGCM ranks the CSPs based on the adherence between the CSPs’ provision and users’ unique requirements for various CSMIC-SMI KPIs. For validation purposes, we have considered case study 2, which comprises 5 CSPs provisions (Amazon EC2 ( $S_1$ ), Windows Azure ( $S_2$ ), Rackspace ( $S_3$ ), cloud services from private cloud set up at SASTRA University ( $S_4$  and  $S_5$ )) and one user requirement. HGCM processes the CSP provisions and user’s requirements for the entire set of CSMIC-SMI metric and ranks the cloud service providers as  $S_3, S_4, S_5, S_1,$  and  $S_2$ . During the validation phase, HGCM processes the QoS data of CSP provisions and CU’s requirements based on the optimal reduct obtained from phase 2 and ranks the CSPs as  $S_3, S_4, S_5, S_1,$  and  $S_2$ . From Table 12, it was more clear that the service ranking of RST-HGBFFO (SQR and QRR) was similar to  $RST_{GS}, RST_{PSO}, RST_{AN},$  and  $RST_{FF}$ . However, RST-HGBFFO outperforms  $RST_{GS}, RST_{PSO}, RST_{AN},$  and  $RST_{FF}$  in terms of minimal reduct and run time.

(iv) Run time analysis

In addition to above quality metrics, runtime analysis has its own significance in evaluating the performance of the feature selection technique. With respect to feature selection technique, time complexity can be redefined as the

total time taken by the technique to identify the optimal reduct. From the experiments carried out, the following can be observed:

1. For QWS dataset, the run time of RST-HGBFFO (SQR and QRR) was slightly higher than  $RST_{CU}, RST_{AN},$  and  $RST_{BS}$  (Fig. 8).
2. For Cloud Armor trust feedback dataset, the run time of RST-HGBFFO (SQR and QRR) was similar to  $RST_{BS}$ , but slightly higher than  $RST_{FF}$  (Fig. 9).
3. For CISH-SASTRA trust feedback dataset, the run time of RST-HGBFFO (SQR and QRR) to find the optimal subset was found to be minimal (Fig. 10).

To summarize, even though the reduct size of RST-HGBFFO (SQR and QRR) was found to be similar to some of the state-of-the-art feature selection techniques (RSARSubsetEval\_PSO Search ( $RST_{PSO}$ ), RSARSubsetEval\_Cuckoo Search ( $RST_{CU}$ ), RSARSubsetEval\_ANT Search ( $RST_{AN}$ ), RSARSubsetEval\_Firefly Search ( $RST_{FF}$ ), RSARSubsetEval\_Harmony Search ( $RST_{HS}$ ), and RSARSubsetEval\_Bat Search ( $RST_{BS}$ )), it substantiates its efficiency in terms of classification accuracy and run time analysis. Even though the run time of RSARSubsetEval\_Firefly Search ( $RST_{FF}$ ), RSARSubsetEval\_ANT Search ( $RST_{AN}$ ), RSARSubsetEval\_Cuckoo Search ( $RST_{CU}$ ), and RSARSubsetEval\_Bat Search ( $RST_{BS}$ ) was found to be minimal, RST-HGBFFO outperforms the state-of-the-art rough set-based feature selection techniques in terms of classification accuracy and reduct size (Fig. 10-12).

The overwhelming performance of RST-HGBFFO was due to the search strategy of HGBFFO which involves the application of hyperclique property of hypergraph for the optimal number of 1’s in the population of BFFO. Further, the search strategy of HGBFFO optimizes the relevance between the features (TMPs) and the target class (trust value) and reduces the redundancy among the selected

**Table 12** Validation of RST-HGBFFO TMP selection technique—service ranking

S. no	Feature selection techniques	Service ranking
1	RSARSubsetEval_Genetic Search ( $RST_{GS}$ )	$S_3 < S_4 < S_5 < S_1 < S_2^*$
2	RSARSubsetEval_PSO Search ( $RST_{PSO}$ )	$S_3 < S_4 < S_5 < S_1 < S_2^*$
3	RSARSubsetEval_Cuckoo Search ( $RST_{CU}$ )	$S_4 < S_2 < S_1 < S_5 < S_3$
4	RSARSubsetEval_ANT Search ( $RST_{AN}$ )	$S_3 < S_4 < S_5 < S_1 < S_2^*$
5	RSARSubsetEval_Harmony Search ( $RST_{HS}$ )	$S_1 < S_5 < S_3 < S_4 < S_2$
6	RSARSubsetEval_Firefly Search ( $RST_{FF}$ )	$S_3 < S_4 < S_5 < S_1 < S_2^*$
7	RSARSubsetEval_Bat Search ( $RST_{BS}$ )	$S_2 < S_4 < S_1 < S_3 < S_7$
8	Rough set theory-based hypergraph-binary fruit fly optimization-supervised quick reduct (RST-HGBFFO SQR)	$S_3 < S_4 < S_5 < S_1 < S_2^*$
9	Rough set theory-based hypergraph-binary fruit fly optimization-quick relative reduct (RST-HGBFFO-QRR)	$S_3 < S_4 < S_5 < S_1 < S_2^*$

\* Original service ranking of the 5 cloud service providers

TMPs which resulted in an optimal feature subset that maximizes the fitness function, i.e., dependency metric.

## 5 Conclusions

The immense benefits provided by the cloud computing ecosystem have enticed diverse organizations to become a part of this phenomenal technology. This scenario has led to the substantial increase in the number of cloud service providers offering a wide variety of cloud services, which makes it difficult for the users to find suitable and trustworthy CSPs who comply with their QoS requirements. Tireless efforts by the research community have yielded trust-based service selection models as a prominent solution for the above-said challenge. However, the intrinsic relation between the accuracy of the trust-based service selection models and the optimality of the service-specific TMP subset makes the identification of the optimal service-specific TMP subset an open research challenge.

Hence, this paper put forth an efficient rough set theory-based hypergraph-binary fruit fly optimization (RST-HGBFFO), a cooperative bio-inspired technique for the identification of the optimal TMPs based on the service type and user preferences. The hyperclique property of hypergraph was exploited for the generation of initial population to enhance the performance of the binary fruit fly optimization algorithm in terms of reduced time complexity and by avoiding premature convergence. Further, HGBFFO was integrated with rough set theory for the identification of optimal TMP with minimal time complexity, i.e., the fitness of each particle of HGBFFO was evaluated based on the RST-specific fitness function. The CSMIC-SMI metric was used as a standard for TMPs. The experiments were conducted using QWS dataset, Cloud Armor and synthetic trust feedback dataset obtained from the private cloud at CISH, SASTRA University. The performance of RST-HGBFFO over the state-of-the-art feature selection technique was evaluated using HGCM and Weka tool in terms of reduct size, service ranking, classification accuracy, and time complexity. Further, RST-HGBFFO was found to be computationally attractive, scalable, and applicable in the field of intrusion detection, stock market analysis, service ranking, weather forecasting, DNA sequencing, metadata validation in geographical information system (GIS), etc.

**Acknowledgements** This work was supported by The Department of Science and Technology – India, The Council for Scientific and Industrial Research – India, and TATA Realty – SASTRA Srinivasa Ramanujan Research Cell (Grant No: CSIR-SRF Fellowship/143345/2K17/1, SR/FST/MSI-107/2015, MRT/2017/000155, and SR/FST/ETI-349/2013).

## Compliance with ethical standards

**Conflict of interest** All the authors declare that they do not have any conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## Glossary

CSP	Cloud service provider(s)
CU	Cloud user(s)
QoS	Quality of service
TMP	Trust measure parameter(s)
RST	Rough set theory
BFFO	Binary fruit fly optimization
RST-HGBFFO	Rough set theory-based hypergraph-binary fruit fly optimization
HGCM	Hypergraph-based computational model
XaaS	Anything as a service
MCDM	Multi-criteria decision making
FFOA	Fruit fly optimization algorithm
CSMIC-SMI	Cloud service measurement index consortium-service measurement index
SQR	Supervised quick reduct
QRR	Quick relative reduct
RST-HGBFFO	The proposed trust measure parameter selection technique
$O = \{O_1, O_2, \dots, O_S\}$	Observations
$A = A_1, A_2, \dots, A_C$	Conditional attributes
$D$	Decisional attributes
$Max_{Gen}$	Maximum number of generations
$PoP_{Size}$	Population size
$PoP_{Num}$	Number of populations
Fitness	Fitness value
$Best_{Smell}$	Local best smell concentration
$G_{Best_{Smell}}$	Global best smell concentration
$Best_{Pos}$	Best position

## References

- Abraham A, Falc R, Bello R (2009) Rough set theory: a true landmark in data analysis. Springer, Berlin
- Akshya Kaveri B, Gireesha O, Somu N et al (2017) E-FPRO-METHEE: An Entropy Based Fuzzy Multi Criteria Decision Making Service Ranking Approach for Cloud Service Selection. In: Venkataramani G, Sankaranarayanan K, Mukherjee S, Arputharaj KSNS (eds) Smart Secure Systems—IoT and Analytics Perspective. ICHT 2017 Communications in Computer and Information Science. Springer, Singapore, pp 224–238

- Al-Masri E, Mahmoud QH (2007) QoS-based discovery and ranking of Web services. In: Proceedings—international conference on computer communications and networks, ICCCN. pp 529–534
- Berge C (1973) Graphs and hypergraphs. North-Holland Pub, Co, Amsterdam
- Chen Y, Miao D, Wang R (2010) A rough set approach to feature selection based on ant colony optimization. *Pattern Recognit Lett* 31:226–233. <https://doi.org/10.1016/J.PATREC.2009.10.013>
- Chen Y, Zhu Q, Xu H (2015) Finding rough set reducts with fish swarm algorithm. *Knowl-Based Syst* 81:22–29. <https://doi.org/10.1016/J.KNOSYS.2015.02.002>
- Chen Y, Zeng Z, Lu J (2017) Neighborhood rough set reduction with fish swarm algorithm. *Soft Comput* 21:6907–6918. <https://doi.org/10.1007/s00500-016-2393-6>
- Dai H, Zhao G, Lu J, Dai S (2014) Comment and improvement on “a new Fruit Fly Optimization Algorithm: taking the financial distress model as an example”. *Knowl-Based Syst* 59:159–160. <https://doi.org/10.1016/j.knosys.2014.01.010>
- Das AK, Sengupta S, Bhattacharyya S (2018) A group incremental feature selection for classification using rough set theory based genetic algorithm. *Appl Soft Comput* 65:400–411. <https://doi.org/10.1016/J.ASOC.2018.01.040>
- Deogun JS, Choubey SK, Raghavan VV, Sever H (1998) Feature selection and effective classifiers. *J Am Soc Inf Sci* 49:423–434
- Ding S, Xia C, Zhou K et al (2014a) Decision support for personalized cloud service selection through multi-attribute trustworthiness evaluation. *PLoS ONE* 9:1–11. <https://doi.org/10.1371/journal.pone.0097762>
- Ding S, Yang S, Zhang Y et al (2014b) Combining QoS prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems. *Knowl-Based Syst* 56:216–225
- Ding S, Wang Z, Wu D, Olson DL (2017a) Utilizing customer satisfaction in ranking prediction for personalized cloud service selection. *Decis Support Syst* 93:1–10. <https://doi.org/10.1016/j.dss.2016.09.001>
- Ding S, Xia C, Wang C et al (2017b) Multi-objective optimization based ranking prediction for cloud service recommendation. *Decis Support Syst* 101:106–114. <https://doi.org/10.1016/j.dss.2017.06.005>
- El Aziz MA, Hassanien AE (2018) Modified cuckoo search algorithm with rough sets for feature selection. *Neural Comput Appl* 29:925–934. <https://doi.org/10.1007/s00521-016-2473-7>
- Ganesan J, Inbarani HH, Azar AT, Polat K (2017) Tolerance rough set firefly-based quick reduct. *Neural Comput Appl* 28:2995–3008. <https://doi.org/10.1007/s00521-016-2514-2>
- Garg SK, Versteeg S, Buyya R (2013) A framework for ranking of cloud computing services. *Future Gener Comput Syst* 29:1012–1023. <https://doi.org/10.1016/j.future.2012.06.006>
- Gauthama Raman MR, Kirthivasan K, Shankar Sriram VS (2017a) Development of rough set—hypergraph technique for key feature identification in intrusion detection systems. *Comput Electr Eng* 59:189–200. <https://doi.org/10.1016/j.compeleceng.2017.01.006>
- Gauthama Raman MR, Somu N, Kirthivasan K et al (2017b) An efficient intrusion detection system based on hypergraph—genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowl-Based Syst* 134:1–12. <https://doi.org/10.1016/j.knosys.2017.07.005>
- Gheysa I, Smith L (2010) Feature subset selection in large dimensionality domains. *Pattern Recognit* 43:5–13
- Hassanien AE, Gaber T, Mokhtar U, Hefny H (2017) An improved moth flame optimization algorithm based on rough sets for tomato diseases detection. *Comput Electron Agric* 136:86–96. <https://doi.org/10.1016/j.compag.2017.02.026>
- Hu Z (2012) Decision rule induction for service sector using data mining—A rough set theory approach. The University of Texas at El Paso, El Paso
- Inbarani HH, Azar AT, Jothi G (2014) Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. *Comput Methods Programs Biomed* 113:175–185. <https://doi.org/10.1016/j.cmpb.2013.10.007>
- Inbarani HH, Bagyamathi M, Azar AT (2015) A novel hybrid feature selection method based on rough set and improved harmony search. *Neural Comput Appl* 26:1859–1880. <https://doi.org/10.1007/s00521-015-1840-0>
- Jensen R, Shen Q (2003) Finding rough set reducts with ant colony optimization. In: Proceedings of the 2003 UK workshop on. pp 15–22
- Jensen R, Shen Q (2004) Fuzzy—rough attribute reduction with application to web categorization. *Fuzzy Sets Syst* 141:469–485
- Jiang F, Sui Y, Zhou L (2015) A relative decision entropy-based feature selection approach. *Pattern Recognit* 48:2151–2163. <https://doi.org/10.1016/j.patcog.2015.01.023>
- Jia-Yuarn Guo (2003) Rough set-based approach to data mining. Case Western University, Cleveland
- Jothi G, Hannah Inbarani H (2016) Hybrid tolerance rough set-firefly based supervised feature selection for MRI brain tumor image classification. *Appl Soft Comput* 46:639–651. <https://doi.org/10.1016/j.asoc.2016.03.014>
- Liao Y, Vemuri VR, Pasos A (2007) Adaptive anomaly detection with evolving connectionist systems. *J Netw Comput Appl* 30:60–80. <https://doi.org/10.1016/j.jnca.2005.08.005>
- Liu Y, Esseghir M, Boulahia LM (2016) Evaluation of parameters importance in cloud service selection using rough sets. *Creat Commons Attrib Int Licens (CC BY)* 7:527–541
- Luan X-Y, Li Z-P, Liu T-Z (2016) A novel attribute reduction algorithm based on rough set and improved artificial fish swarm algorithm. *Neurocomputing* 174:522–529. <https://doi.org/10.1016/J.NEUCOM.2015.06.090>
- Ma H, Hu Z, Li KZH (2016) Toward trustworthy cloud service selection: a time-aware approach using interval neutrosophic set. *J Parallel Distrib Comput* 96:75–94. <https://doi.org/10.1016/J.JPDC.2016.05.008>
- Mao C, Lin R, Xu C, He Q (2017) Towards a trust prediction framework for cloud services based on PSO-driven neural network. *IEEE Access* 5:2187–2199. <https://doi.org/10.1109/ACCESS.2017.2654378>
- Marudhadevi D, Dhatchayani VN, Sriram VSS (2014) A trust evaluation model for cloud computing using service level agreement. *Comput J*. <https://doi.org/10.1093/comjnl/bxu129>
- Mell P, Grance T (2011) The NIST definition of cloud computing. *NIST Spec Publ* 145:7. <https://doi.org/10.1136/emj.2010.096966>
- Min F, Liu Q (2009) A hierarchical model for test-cost-sensitive decision systems. *Info Sci* 179(14):2442–2452
- Moore D (1976) Chi square tests. Purdue Univ Lafayette Ind Dept of Statistics, West Lafayette
- Nina FRC (2007) On applications of rough sets theory to knowledge discovery. University of Puerto Rico, San Juan
- Obulaparam G, Somu N, Manilyer Ramani GR, Boopathy AK, Vathula Sankaran SS (2018) GCRITICPA: A CRITIC and Grey relational analysis based service ranking approach for cloud service selection. In: Akoglu L, Ferrara E, Deivamani M, Baeza-Yates R, Yogesh P (eds) *Advances in Data Science. ICIIT 2018. Communications in Computer and Information Science*, vol 941. Springer, Singapore, pp 3–16
- Øhrn A (2001) Rosetta technical reference manual. Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway
- Pan W-T (2012) A new Fruit Fly Optimization Algorithm: taking the financial distress model as an example. *Knowl-Based Syst* 26:69–74. <https://doi.org/10.1016/j.knosys.2011.07.001>

- Pawlak Z (1982a) Rough sets. *Int J Comput Inf* 99:1–51. <https://doi.org/10.1007/978-94-011-3534-4>
- Pawlak Z (1982b) Rough sets. *Int J Parallel Program* 11:341–356. <https://doi.org/10.1007/978-94-011-3534-4>
- Qu L (2016) Credible service selection in cloud environments. Macquarie University, Sydney
- Qu L, Wang Y, Orgun MA et al (2015) CCCloud: context-aware and credible cloud service selection based on subjective assessment and objective assessment. *IEEE Trans Serv Comput* 8:369–383. <https://doi.org/10.1109/TSC.2015.2413111>
- Raman M, Kannan K, Pal S (2016) Rough set-hypergraph-based feature selection approach for intrusion detection systems. *Def Sci* 66:612
- Raman MRG, Somu N, Kirthivasan K, Sriram VSS (2017) A hypergraph and arithmetic residue-based probabilistic neural network for classification in intrusion detection systems. *Neural Netw* 92:89–97. <https://doi.org/10.1016/j.neunet.2017.01.012>
- Raza MS, Qamar U (2018) Feature selection using rough set-based direct dependency calculation by avoiding the positive region. *Int J Approx Reason* 92:175–197. <https://doi.org/10.1016/J.IJAR.2017.10.012>
- Sheng M Cloud Armor. <https://cs.adelaide.edu.au/~cloudarmor/home.html>. Accessed 12 April 2018
- Shen L, Chen H, Yu Z et al (2016) Evolving support vector machines using fruit fly optimization for medical data classification. *Knowl-Based Syst* 96:61–75. <https://doi.org/10.1016/j.knosys.2016.01.002>
- Somu N, Raman MRG, Kirthivasan K, Sriram VSS (2016) Hypergraph based feature selection technique for medical diagnosis. *J Med Syst* 40:239. <https://doi.org/10.1007/s10916-016-0600-8>
- Somu N, Kirthivasan K, Shankar SS (2017a) A computational model for ranking cloud service providers using hypergraph based techniques. *Future Gener Comput Syst* 68:14–30. <https://doi.org/10.1016/j.future.2016.08.014>
- Somu N, Kirthivasan K, Sriram VSS (2017b) A rough set-based hypergraph trust measure parameter selection technique for cloud service selection. *J Supercomput* 73:4535–4559. <https://doi.org/10.1007/s11227-017-2032-8>
- Somu N, Gauthama Raman MR, Kalpana V et al (2018a) An improved robust heteroscedastic probabilistic neural network based trust prediction approach for cloud service selection. *Neural Netw* 108:339–354
- Somu N, Gauthama Raman MR, Kirthivasan K, Shankar Sriram VS (2018b) A trust centric optimal service ranking approach for cloud service selection. *Futur Gener Comput Syst* 86:234–252. <https://doi.org/10.1016/j.future.2018.04.033>
- Sosinsky B (2010) *Cloud Computing Bible*. Wiley, Hoboken
- Su Y, Guo J (2017) A novel strategy for minimum attribute reduction based on rough set theory and fish swarm algorithm. *Comput Intell Neurosci* 2017:1–7. <https://doi.org/10.1155/2017/6573623>
- Su K, Xiao B, Liu B et al (2017) TAP: a personalized trust-aware QoS prediction approach for web service recommendation. *Knowl-Based Syst* 115:55–65. <https://doi.org/10.1016/j.knosys.2016.09.033>
- Sun L, Dong H, Hussain FK et al (2014) Cloud service selection: state-of-the-art and future research directions. *J Netw Comput Appl* 45:134–150. <https://doi.org/10.1016/j.jnca.2014.07.019>
- Sun L, Ma J, Zhang Y et al (2016) Cloud-FuSeR: fuzzy ontology and MCDM based cloud service selection. *Future Gener Comput Syst* 57:42–55. <https://doi.org/10.1016/j.future.2015.11.025>
- Tang M, Dai X, Liu J, Chen J (2017) Towards a trust evaluation middleware for cloud service selection. *Future Gener Comput Syst* 74:302–312. <https://doi.org/10.1016/j.future.2016.01.009>
- Thampi S, Bhargava PA (2013) *Managing trust in cyberspace*. CRC Press, Boca Raton
- Udhaya Kumar S, Hannah Inbarani H (2017) PSO-based feature selection and neighborhood rough set-based classification for BCI multiclass motor imagery task. *Neural Comput Appl* 28:3239–3258. <https://doi.org/10.1007/s00521-016-2236-5>
- Valley CC (2011) *Service Measurement Index Version 1.0*. Silicon Valley, Moffett Field
- Velayutham C, Thangavel K (2011) Unsupervised quick reduct algorithm using rough set theory. *J Electron Sci Technol* 9:193–201
- Wang X, Yang J, Teng X, Weijun Xia RJ (2007) Feature selection based on rough sets and particle swarm optimization. *Pattern Recognit Lett* 28:459–471. <https://doi.org/10.1016/j.patrec.2006.09.003>
- Wang X, Cao J, Xiang Y (2015) Dynamic cloud service selection using an adaptive learning mechanism in multi-cloud computing. *J Syst Softw* 100:195–210. <https://doi.org/10.1016/j.jss.2014.10.047>
- Witten IH, Frank E, Hall MA, Pal CJ (2016) *Data mining: practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, Burlington
- Yao Y (2004) A Partition Model of Granular Computing. In: Peters JF, Skowron A, Grzymała-Busse JW, Kostek B, Świniarski RWSMS (eds) *Transactions on rough sets I*. Springer, Berlin, pp 232–253
- Zhong N, Dong J, Ohsuga S (2001) Using rough sets with heuristics for feature selection. *J Intell Inf Syst* 16:199–214
- Zouache D, Ben Abdelaziz F (2018) A cooperative swarm intelligence algorithm based on quantum-inspired and rough sets for feature selection. *Comput Ind Eng* 115:26–36. <https://doi.org/10.1016/J.CIE.2017.10.025>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.