IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 16, 2018, accepted November 29, 2018, date of publication January 10, 2019, date of current version April 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891956

A Generalized Enhanced Quantum Fuzzy Approach for Efficient Data Clustering

NEHA BHARILL[1], OM PRAKASH PATEL[2], ARUNA TIWARI[3], LIFENG MU[4], DONG-LIN LI[5], MANORANJAN MOHANTY[6], OMPRAKASH KAIWARTYA[7], AND MUKESH PRASAD[8]

[1]Department of Computer Science and Engineering, Indian Institute of Information Technology Dharwad, Hubli 580029, India
[2]Department of Computer Science and Engineering, KLE Technological University, Hubballi 580031, India
[3]Department of Computer Science and Engineering, Indian Institute of Information Technology Indore, Indore 453552, India
[4]SHU-UTS SILC Business School, Shanghai University, Shanghai, China
[5]Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan
[6]Department of Computer Science, The University of Auckland, Auckland, New Zealand
[7]School of Science and Technology, Nottingham Trent University at Clifton, Nottingham NG11 8NS, U.K.
[8]Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, School of Software, University of Technology Sydney, Ultimo, NSW, Australia

Corresponding author: Omprakash Kaiwartya (omprakash.kaiwartya@ntu.ac.uk)

This work was supported in part by the School of Science and Technology, Nottingham Trent University, U.K., in part by the NSFC under Grant 71871133, in part by the University of Auckland Faculty Research Development Fund (FRDF) under Project 3716217, and in part by the Ministry of Science and Technology, China, under Grant MOST 107-2634-F-009-009.

ABSTRACT Data clustering is a challenging task to gain insights into data in various fields. In this paper, an Enhanced Quantum-Inspired Evolutionary Fuzzy C-Means (EQIE-FCM) algorithm is proposed for data clustering. In the EQIE-FCM, quantum computing concept is utilized in combination with the FCM algorithm to improve the clustering process by evolving the clustering parameters. The improvement in the clustering process leads to improvement in the quality of clustering results. To validate the quality of clustering results achieved by the proposed EQIE-FCM approach, its performance is compared with the other quantum-based fuzzy clustering approaches and also with other evolutionary clustering approaches. To evaluate the performance of these approaches, extensive experiments are being carried out on various benchmark datasets and on the protein database that comprises of four superfamilies. The results indicate that the proposed EQIE-FCM approach finds the optimal value of fitness function and the fuzzifier parameter for the reported datasets. In addition to this, the proposed EQIE-FCM approach also finds the optimal number of clusters and more accurate location of initial cluster centers for these benchmark datasets. Thus, it can be regarded as a more efficient approach for data clustering.

INDEX TERMS Clustering, quantum computing, evolutionary algorithm, fuzzy set theory, bioinformatics.

I. INTRODUCTION

Unsupervised learning is an important approach for exploratory data analysis. Clustering is one of the widely used unsupervised learning approaches. It partitions the data into various groups such that the data point belonging to the same group exhibit more similarity with each other in comparison with data points belonging to other groups. Clustering is used in many application domains such as medical imaging, disease diagnosis, and bioinformatics where data are generated at a tremendous pace. Like in the Bioinformatics area, with the progress of experimental technologies in molecular biology, the biological data in terms of protein sequences are gathering at an accelerating rate in public databases. Clustering is gaining importance in all these areas to gain insight into the accumulated data [3]–[8].

One of the most widely used clustering algorithms was initially presented by Dunn [9] and completed by Bezdek [10] known as the Fuzzy C-Means (FCM) algorithm. The FCM algorithm partition the collection of n data points $X = [x_1, \ldots, x_n]$ into C fuzzy clusters, and finds a cluster center of each group such that an objective function of a dissimilarity measure is minimized. It allows the data points to belong to the multiple clusters with a membership degree μ_{il} varies between 0 and 1. Although the fuzzy clustering can increase the accuracy of the cluster representation

The associate editor coordinating the review of this manuscript and approving it for publication was Weiping Ding.

there are several fundamental sources of ambiguity in clustering.

One of the problems is that the number of clusters C has to be known in advance for the dataset. However, different datasets require a different number of clusters, so it is tough to know beforehand. The second problem is the selection of initial cluster centers because the FCM algorithm is initially started with random assignment of the cluster centers. The behavior of the FCM algorithm is highly sensitive to the selection of the initial centers of C number of clusters. Therefore, selection of inappropriate values of the cluster centers may lead the algorithm to converge at local optima. Moreover, when the multiple runs of the same data are executed with different initial cluster centers, then the FCM algorithm does not converge to the same set of final cluster centers. Due to this, the clustering results which are generated by the FCM algorithm suffers from the drawback of producing inconsistent clustering results and will make the algorithm to converge at different local optima. Another critical problem in applying the FCM algorithm for clustering of data is the selection of fuzzifier parameter m because the appropriate value of fuzzifier parameter m widely varies from one dataset to another. The choice of the inappropriate value of m may also lead the FCM algorithm to the local optima problem. Therefore, it is essential to determine an adequate value of fuzzifier parameter m.

In the literature, many clustering methods [11]–[26] are designed which automatically identify the number of clusters but they depend on the selection of the objective function. These clustering algorithms are executed with the different values of C and the best value of C is selected on the basis of predefined criteria. Figueiredo and Jain [11] uses the minimum message length (MML) [12], [13] as the criterion function in conjunction with the Gaussian Mixture Model (GMM) to estimate the C. This approach starts with a large number of clusters and gradually starts merging the clusters if this leads to a decrease in the MML criterion. In addition to this, several validity indices are proposed by the researchers [27]–[29] to identify the number of clusters C. Also, a wide variety of clustering methods for molecular sequences based on sequence similarities are computed using homologous search programs like BLAST [30] and FASTA [31]. In molecular biology, using clustering for the identification of protein superfamilies based on sequence similarity is a traditional problem. Due to the numerous advancement and development of improved clustering algorithms, they are widely applied to the biological researches [32], [33]. One of the successful clustering algorithms is the Markov Cluster algorithm [34]. It is used to detect the protein families in the protein-protein interactions networks based on graph theory. Go fuzzy clustering algorithm was proposed by Tari *et al.* [35], which enables the simultaneous use of biological knowledge and gene expression data in a probabilistic clustering algorithm. Recently, spectral clustering [8] is applied to the clustering of protein sequences. This algorithm suffers from a problem that the number of clusters has to be specified manually, and it requires a long runtime. Furthermore, some literature available for the choice of fuzzifier parameter [10], [36]. Dembélé and Kastner [37] proposed a method to compute the upper bound value of m for clustering of microarray data using the FCM algorithm. Patel *et al.* [38] proposed a Quantum-Inspired Evolutionary Fuzzy C-Means (QIE-FCM) algorithm which aims to utilize the concept of quantum computing to find the appropriate value of the fuzzifier parameter along with the number of clusters. Despite this, the QIE-FCM approach is sensitive to the selection of the location of initial cluster centers because it is done randomly. Thus, it does not guarantee the optimal global solution for the QIE-FCM algorithm. In spite of the above-stated methods proposed by the researchers, it is still very challenging to decide which value of m, C, and location of initial cluster centers which leads to the meaningful clusters with best fuzzy partitions.

In this paper, we attempt to utilize quantum computing concept [39] to alleviate the above-discussed drawbacks of the QIE-FCM algorithm [10]. An Enhanced Quantum-Inspired Evolutionary Fuzzy C-Means (EQIE-FCM) algorithm is proposed. In EQIE-FCM, the value of fuzzifier parameter m is represented in terms of quantum bits in generation g (user defined parameter). For each value of m obtained in generation g, the number of clusters C is initialized in the range of $C = 2, 3, \ldots, c_{\max}$ where $c_{\max} = \sqrt{n}$ (n is the number of instances) [40]. As discussed above in the QIE-FCM algorithms, the location of cluster centers for each C is initialized randomly, which does not guarantee the optimal global solution and thus converge to the optimal local solution. To mitigate this problem, in the EQIE-FCM for each value of C, the set of the cluster centers V_C is initialized in terms of quantum bits. For each value of C on the initialized set of the cluster centers V_C, the several runs of the FCM algorithm are executed and corresponding fuzzy partitions are obtained. To evaluate the fitness of produced fuzzy partitions on different values of C obtained in generation g, VI_{DSO} [41] index is used as the objective function. Then a fuzzy partition with a minimum value of the VI_{DSO} index is selected, which indicate the best fuzzy partition representing the local best fitness value for generation g. After this, the fuzzifier parameter m and set of the cluster centers V_C corresponding to each C are evolved in generation g using the quantum rotation gate [42]. To achieve the optimal global value of m and C, the proposed algorithm is executed for several generations. However, to guarantee that the best fuzzy partition in any of the generation would not be lost in the evolutionary process [39], the global fitness function is evaluated. This will store a fuzzy partition which attains the minimum value of the fitness function among all the generations. Through this, the EQIE-FCM algorithm can find the optimal value of fuzzifier parameter m and number of clusters C with the best location of initial cluster centers. The same is being empirically tested on various benchmark datasets and four

baseline protein superfamilies obtained from the International Protein Sequence Database [43].

The rest of this paper is organized as follows. Section II, briefly present the preliminaries. The detailed description and the implementation perspective of the proposed EQIE-FCM algorithm are given in Section III. Section IV presents the experimental results and analysis based on these results. Finally, we conclude this paper and indicate future research perspective in Section V.

II. PRELIMINARIES

Before describing the overall concept of the proposed EQIE-FCM algorithm, we briefly present the concept of quantum computing. The main idea of quantum computing concept is to represent data in terms of quantum bits which is made up of qubits. The qubit is the smallest unit of information representation. It differs from classical computer bits that are "1" and "0" in terms of representation. As a classical bit can represent only two possibilities of an event at one time by bit "1" or "0". Meanwhile, a qubit can exist in both states simultaneously using the probability concept proposed by Han and Kim [39], [42]. Qubit represents the linear superposition of "1" and "0" bits probabilistically, which is denoted as follows:

$$q = \alpha \mid 0\rangle + \beta \mid 1\rangle \tag{1}$$

where α and β are the complex numbers specifying the probability that a qubit may appear in "0" state and "1" state. The probabilities that the qubit will be found in "0" state and "1" state are represented by $|\alpha|^2$ and $|\beta|^2$, respectively and defined as follows:

$$\alpha^2 + \beta^2 = 1; 0 \leq \alpha \leq 1, 0 \leq \beta \leq 1 \tag{2}$$

An individual quantum bit Q in generation g contains the k number of qubits q which is represented as $Q_g = [q_1^g, ..., q_k^g]$ where each qubit q is composed of complex numbers α and β as shown in Equation (1). The individual quantum bit in terms of qubits is represented as follows:

$$Q_g = \begin{bmatrix} \alpha_1^g & \alpha_2^g \text{ ..... } \alpha_k^g \\ \beta_1^g & \beta_2^g \text{ ..... } \beta_k^g \end{bmatrix} \tag{3}$$

As discussed above, a qubit can be represented in two states (2^1), i.e., "0" state and "1" state. Similarly, an individual quantum bit with two qubits can represent the linear superposition with four states, i.e., "00", "01", "10", and "11". Thus, if there is a string of k qubits in an individual quantum bit, then the string can represent 2^k states at the same time. Let us take an example of an individual quantum bit with two qubits are represented as follows:

$$Q = \left\langle \begin{matrix} 1/\sqrt{2}|1/\sqrt{2} \\ 1/\sqrt{2}|1/\sqrt{2} \end{matrix} \right\rangle \tag{4}$$

$$\begin{aligned} Q = {} & (1/\sqrt{2} \times 1/\sqrt{2})\langle 00\rangle + 1/\sqrt{2} \times 1/\sqrt{2})\langle 01\rangle \\ & +1/\sqrt{2} \times 1/\sqrt{2})\langle 10\rangle + 1/\sqrt{2} \times 1/\sqrt{2})\langle 11\rangle \end{aligned} \tag{5}$$

As shown in Equation (5) only one quantum bit is enough to represent four states. Thus, the quantum bit representation provides better characteristics of population diversity in comparison with other representations and also enables us to find the optimal global solution in a large search space which is made up of many subspaces. Han and Kim [39] used a genetic algorithm in a combination of the quantum computing concept for evolving the optimal solution of the knapsack problem in several generations. Based on the idea mentioned above, we proposed an EQIE-FCM algorithm, which uses the quantum computing concept. In this algorithm, the clustering of data is done by finding the global best value of the fuzzifier parameter m and the number of clusters C along with the best location of initial cluster centers from subspaces in several generations. The EQIE-FCM algorithm is designed with the novel quantum bit representation of the fuzzifier parameter m and set of cluster centers V_C. During this evolutionary process, the quantum rotational gate [42] is used in this algorithm for updating the qubits of m and V_C.

III. PROPOSED APPROACH

In this paper, we proposed an Enhanced Quantum-Inspired Evolutionary Fuzzy C-Means (EQIE-FCM) algorithm. As pointed out by Pal and Bezdek [29], the fuzzifier parameter m and the number of clusters C plays an important role in validating the fitness of partitions produced by fuzzy based clustering algorithms. To judge the reliability of obtained fuzzy partitions, the identified number of clusters C must be insensitive to changes in the fuzzifier parameter m. In order to investigate these measures in the EQIE-FCM algorithm, the quantum concept is utilized to evolve the different values of m in each generation g from subspaces where m is represented in terms of quantum bits. Then, the transformation process is applied in each generation to obtain the real coded value of m. In the EQIE-FCM algorithm, for the real coded value of m obtained in generation g, the number of clusters C is initialized over the range of $C = 2, 3, ..., c_{\max}$. For each value of C, the set of cluster centers V_C^g is initialized in terms of quantum bits and then the real coded value of V_C^g is obtained through the transformation process. These parameters are then passed to the conventional FCM algorithm and several runs of FCM algorithm is executed for each value of C in generation g and correspondingly fuzzy partitions are generated. Next, the VI_{DSO} [41] index is used as the objective function to compute the fitness of obtained fuzzy partitions. After this, a fuzzy partition is selected with the minimum value of the VI_{DSO} index. The optimal value of VI_{DSO} represents the local best fitness function for generation g. Then, a quantum rotational gate is used to update the quantum bits of fuzzifier parameter and set of the cluster centers for each value of C so that the distinct values of these parameters can be evolved in each generation from subspaces. Subsequently, the rest of the process is repeated similarly for several generations, and the global fitness function is evaluated, which store the fuzzy partition with a minimum value of the fitness function among all the generations.

Through this, the EQIE-FCM algorithm guarantees to achieve the optimal global value of m and C as well as it finds the best location of initial cluster centers for the datasets. The overall procedure of the EQIE-FCM algorithm is partitioned into five major steps: quantum bit representation of fuzzifier parameter m, representation of the cluster centers in quantum bits, transformation process, formulation of fitness functions, and quantum bit update process. The detailed description of these steps is presented in the subsequent sections.

A. REPRESENTATION OF A FUZZIFIER PARAMETER IN QUANTUM BIT

As discussed earlier, the fuzzifier parameter m plays a crucial role in applying fuzzy based approaches for clustering of data. The selection of an inappropriate value of m may lead the algorithm to converge at local optima. To overcome this issue, in the proposed algorithm the fuzzifier parameter m is decided using the quantum concept inspired by the above-mentioned idea of Han and Kim [39]. This concept is utilized in the proposed algorithm to evolve the different values of fuzzifier parameter m in several generations so that we can find the optimal global value of m from subspaces. To achieve this, it is required that a fuzzifier parameter m must be represented in terms of a quantum bit. Let us denote the fuzzifier parameter m in terms of a quantum bit for generation g as M'_g, which is defined as follows:

$$M'_g = Q^g_m \tag{6}$$

where Quantum bit Q^g_m consists of k qubits represented as $Q^g_m = [\alpha^g_{1m}|\alpha^g_{2m}|\alpha^g_{3m}|.....|\alpha^g_{km}]$. The purpose of representing Q^g_m in terms of k qubits implies that the best value of fuzzifier parameter m will be searched from 2^k subspaces.

To get exploration and real coded value of the fuzzifier parameter M'_g, the transformation process is used which is discussed in the further section.

B. REPRESENTATION OF CLUSTER CENTERS IN QUANTUM BITS

As mentioned earlier, the initialization of cluster centers is a critical issue for the execution of fuzzy based clustering algorithm. In the case of the FCM algorithm, the cluster centers have been initialized randomly due to which it gets trapped in the problem of local optima. To handle this problem, in the proposed (EQIE-FCM) algorithm the cluster centers have been initialized in the form of quantum bits. Then, to evolve the different locations of the cluster centers in each generation, the proposed algorithm is executed for many generations. In each generation, different location of the cluster centers is produced using a quantum update function. In this way, we can find the best location of the initial cluster centers for the appropriate value of the number of clusters. In generalized form, the set of the cluster centers V_C in generation g is represented as follows:

$$V^g_C = [(V_1)^g_C, (V_2)^g_C, ..., (V_t)^g_C] \tag{7}$$

where V^g_C consist of t number of cluster centers such that $t = 1, 2, ...C$ and each cluster center $(V_i)^g_C$ is represented as follows:

$$(V_i)^g_C = [(V_{1i})^g_C, (V_{2i})^g_C, ..., (V_{di})^g_C]^T \in \mathbb{R}^d \tag{8}$$

where $(V_{ji})^g_C$ represents the jth dimension of ith cluster center such that $j = 1, 2, .., d$ and C is the number of clusters.

As stated above, the fuzzifier parameter in generation g is represented in terms of a single quantum bit as given in Equation (6) but for a cluster number C, each cluster center in generation g consists of d-dimensions. Therefore, corresponding to each cluster number C, the jth dimension of the ith cluster center in generation g is also represented in terms of a single quantum bit which is given as follows:

$$(V'_{ji})^g_C = (Q_{ji})^g_C \tag{9}$$

where $(Q_{ji})^g_C$ contains the k qubits and represented as $(Q_{ji})^g_C = [(\alpha_{ji})^g_{1C}|(\alpha_{ji})^g_{2C}|(\alpha_{ji})^g_{3C}|.....|(\alpha_{ji})^g_{kC}]$.

To get exploration and real coded value for the cluster centers $(V'_{ji})^g_C$, we use the transformation process which is discussed next.

In general, the real coded value for an individual quantum bit Q and a qubit q is represented as Q' and q'. So in the subsequent section, we have given the general representation of the transformation process showing the conversion of a single qubit q into real coded value q' which is also applicable for the conversion of fuzzifier parameter and cluster centers.

C. TRANSFORMATION PROCESS

As discussed above to achieve exploration and to get the real coded value of qubit q' we use transformation process. The transformation process starts with a random number matrix $R^g = [r^g_1 r^g_2 r^g_3 .....r^g_k]$, corresponding to $Q^g = [\alpha^g_1|\alpha^g_2|\alpha^g_3|.....|\alpha^g_k]$ where k represents the number of qubits. The value of r^g_p is selected with the help of a random function (rand) which generates random numbers between 0 and 1. Then a further mapping is done using binary matrix S^g where $S^g = [s^g_1, ..., s^g_k]$. The value of matrix S^g is generated as follows:

$$\text{if } (r^g_p \leq (\alpha^g_p)^2) \text{ then } s^g_p = 1 \text{ else } s^g_p = 0.$$

where $p = 1, 2, ..., k$ and k denote the number of qubits. To get the real coded value of the required parameter from binary matrix S^g, we have used a uniform random number generator (urg). As shown in Equation (3) and Equation (4), the individual quantum bit with k qubits can represent the linear superposition of 2^k states. Thus, to get an optimal value of the real coded value of the qubit q' corresponding to q from 2^k subspaces, we use a formula $bin2dec(S^g)+1$. This formula helps to get a real coded value from a particular subspace from the available 2^k subspaces. The whole transformation process has been described in terms of pseudo-code as follows:

Transformation process()

begin
Step-1: Initialize q^g, random number matrix R^g, and $link = 0$.
  for $p = 1 : k$ **do**
    $q^g = \alpha_p^g; \quad 0 \le \alpha_p^g \le 1$
    $r_p^g = rand;$
  end for
Step-2: **for** $p = 1 : k$ **do**
    if $r_p^g \le (\alpha_p^g)^2$
      $s_p^g = 1;$
    else
      $s_p^g = 0;$
    end if
  end for
Step-3: $link = bin2dec(S^g) + 1$
    $(q')^g = urg();$
end
return $(q')^g$

The transformation process can be understood with the help of an example. Let a quantum bit consist of two qubits is represented as $Q = \langle 0.707|.707\rangle$, therefore a random number matrix is generated using a random number $R = [0.850.02]$. The quantum bit consists of two qubits will provide four subspaces. Now using the formulation ($r_p^g \le (\alpha_p^g)^2$), the binary matrix is generated as $S = [01]$. Once the binary matrix is achieved, then the formula (bin2dec(S)+1) used to convert a binary number to a decimal value. This gives in result a number, between 1 to 4. We have assumed four subspaces using a uniform random generator. This formula (bin2dec(S)+1) gives output 2 as a decimal value according to binary bits $S = [01]$ therefore, the real coded value corresponds to the quantum bit $Q = \langle 0.707|.707\rangle$, is selected from second subspaces.

Once, the transformation process is completed the real coded value for the fuzzifier parameter m_g is represented as follows:

$$m_g = (Q')_m^g \tag{10}$$

Similarly, the real coded value of the cluster center is represented as follows:

$$(v'_{ji})_C^g = (Q')_C^g \tag{11}$$

such that

$$(v'_i)_C^g = [(v'_{1i})_C^g, (v'_{2i})_C^g, ..., (v'_{di})_C^g]^T \in \mathbb{R}^d \tag{12}$$
$$(v')_C^g = [(v'_1)_C^g, (v'_2)_C^g, ..., (v'_t)_C^g] \tag{13}$$

where $(v'_{ji})_C^g$ is the real coded value of the jth dimension of ith cluster center and $(v')_C^g$ denotes the real coded value of set of cluster centers such that $j = 1, 2, ..., d$, $t = 1, 2, ...C$ and C is the number of clusters.

D. FORMULATION OF LOCAL AND GLOBAL FITNESS FUNCTION

As mentioned earlier, the EQIE-FCM algorithm uses a VI_{DSO} index [41] as the objective function to evaluate the fitness of produced fuzzy partitions. The main motivation behind using the VI_{DSO} index as the objective function is that it evaluates the fitness of obtained fuzzy partitions on the basis of three measures, i.e., intra-cluster compactness, inter-cluster separation, and inter-cluster overlap. This implies that the obtained fuzzy partitions are good enough if the data points belong to the same cluster are tightly coupled with each other, and data points belong to the distinct clusters are well separated from each other with the minimum overlap between the clusters. The smallest value of the VI_{DSO} index represents the better fuzzy partitions. In the EQIE-FCM algorithm, we normalized the VI_{DSO} index for each value of C over the range $C = 2, 3, ..., c_{\max}$ which is evaluated as follows:

$$VI_{DSO}^{sum}(U^g, m_g) = \sum_{C=2}^{c_{\max}} VI_{DSO}(C, U^g, m_g) \tag{14}$$

$$VI_{DSO}^{Normalized}(C, U^g, m_g) = \frac{VI_{DSO}(C, U^g, m_g)}{VI_{DSO}^{sum}(U^g, m_g)} \tag{15}$$

where U^g represents the fuzzy partition matrix for generation g. Next, we store the fitness of C number of clusters corresponding to generation g in F_C^g, which is defined as follows:

$$F_C^g = VI_{DSO}^{Normalized}(C, U^g, m_g) \tag{16}$$

In addition to this, the best fuzzy partition in generation g is selected by evaluating the local fitness function denoted by $F_{Lbest}^g(m_g, C)$ and defined as follows:

$$F_{Lbest}^g(m_g, C) = \min_{2 \le C \le c_{\max}} [VI_{DSO}^{Normalized}(C, U^g, m_g)] \tag{17}$$

where $F_{Lbest}^g(m_g, C)$ contain the minimum value of the fitness function evaluated over the range $C = 2, 3, ..., c_{\max}$ corresponding to m_g obtained in a generation g.

Furthermore, to determine the best fitness corresponding to each cluster number C, one more parameter is computed which store the global best fitness of each cluster number C from all the generations denoted by F_C^{gbest} and defined as follows:

$$F_C^{gbest} = \min(F_C^{gbest}, F_C^g) \tag{18}$$

In addition to this, to ensure the effective clustering of data the best value of m_g from subspaces is determined by finding the global best value of the fitness function from all the generations denoted by $F_{Gbest}(m_{best}, C_{best})$ and defined as follows:

$$F_{Gbest}(m_{best}, C_{best}) = \min(F_{Gbest}(m_{best}, C_{best}), F_{Lbest}^g(m_g, C)) \tag{19}$$

where m_{best} and C_{best} denote the best value of fuzzifier parameter and the number of clusters found among all the generations.

E. QUANTUM BIT UPDATE PROCESS

In quantum inspired algorithms, the quantum rotational gate is an important operator to generate a new value of qubits. As EQIE-FCM is an evolutionary algorithm, which evolves the optimal value of the fuzzifier parameter and the cluster centers in several generations.

To get the appropriate value of both the parameters, the quantum rotational gate [42] is used which update the qubits of the fuzzifier parameter and cluster centers in each generation. The quantum gate requires a proper angle to rotate the qubit. The new qubit is generated using the quantum rotational gate and the previous value of a qubit which is defined as follows:

$$\alpha_p^{g+1} = [\alpha_p^g * cos\Delta\theta - \sqrt{1-(\alpha_p^g)^2} * sin\Delta\theta] \tag{20}$$

In the above equation, the rotational angle ($\Delta\theta$) is an important variable to get the appropriate value of new qubit. The selection of an appropriate angle is done on the basis of local fitness function (F_C^g and $F_{Lbest}^g(m_g, C)$) obtained at generation g and global fitness function (F_C^{gbest} and $F_{Gbest}(m_{best}, C_{best})$) find till $g-1$ generations so that a new value of $\Delta\theta$ help to produce better qubits. As shown in the transformation process that each qubit α_p^g is associated with the binary value s_p^g. Therefore, a mapping is done between fitness function and qubit with the help of binary value. The binary value corresponding to local fitness function (F_C^g and $F_{Lbest}^g(m_g, C)$) and the global best fitness function (F_C^{gbest} and $F_{Gbest}(m_{best}, C_{best})$) is s_p^g and s_p^{global}, respectively. As suggested by [45], if the value of local fitness function (F_C^g and $F_{Lbest}^g(m_g, C)$) obtained in the current generation is better than the value of global fitness function (F_C^{gbest} and $F_{Gbest}(m_{best}, C_{best})$) obtained in the previous generation and status of the current binary value s_p^g is zero and best binary value s_p^{global} is one, then change in the qubit α_p^g towards one to zero may produce the worst result. Therefore, to increase the value of qubit α_p^g, $\Delta\theta$ must be negative, according to the quantum rotational gate. Conversely, if the value of local fitness function obtained in the current generation is worse than the value of global fitness function obtained in the previous generation and status of current binary value s_p^g is one and best binary value s_p^{global} is zero, then change in the qubit α_p^g from zero to one may produce the worst result. Therefore, to decrease the value of qubit α_p^g, $\Delta\theta$ must be positive, according to the quantum rotational gate. For the rest of the cases, $\Delta\theta$ must be zero. The value of $\Delta\theta$ must be selected in such a way that it can take the less number of iterations to cover a maximum number of values of α_p^g in the range of (0, 1). Hence, according to [42], $\Delta\theta$ must be initialized between $[0.01\times\pi, 0.05\times\pi]$. Table 1, summarizes all the possible cases for selecting the value of $\Delta\theta$.

From preventing the qubit α_p^g from attaining values 0 or 1, following constraints are applied.

$$\alpha_p^g = \begin{cases} \sqrt{\epsilon}, & if \quad \alpha_p^g < \sqrt{\epsilon} \\ \alpha_p^g & if \quad \sqrt{\epsilon} \le \alpha_p^g \le \sqrt{1-\epsilon} \\ \sqrt{1-\epsilon} & if \quad \alpha_p^g > \sqrt{1-\epsilon} \end{cases} \tag{21}$$

TABLE 1. Parameters for Qubits update.

s_p^g	s_p^{global}	$F_{Gbest}(m_{best}, C_{best}) > F_{Lbest}^g(m_g, C)$ or $F_C^{gbest} > F_C^g$	$\Delta\theta$
0	*0*	*false*	*0*
0	*0*	*true*	*0*
0	*1*	*false*	$-0.03 * \Pi$
0	*1*	*true*	*0*
1	*0*	*false*	*0*
1	*0*	*true*	$0.03 * \Pi$
1	*1*	*false*	*0*
1	*1*	*true*	*0*

where the limiting parameter ϵ is assigned a very small value (approximately approaching to zero), so that it can cover maximum value in the range of (0, 1).

F. WORKING OF THE EQIE-FCM ALGORITHM

As pointed out by Pal and Bezdek [29], the appropriate value of m lies in the interval of [1.5, 2.5]. Therefore, the EQIE-FCM algorithm is executed for $g_{\max}$ generations to find the appropriate value of m in the above-mentioned interval. The maximum number of generations $g_{\max}$ is set as the stopping criteria for EQIE-FCM algorithm because the appropriate value of m in the defined interval can be easily found within the $g_{\max}$ generations. Conversely, if the proposed algorithm is executed for more than $g_{\max}$ generations, then it will generate the repeated values of m with a significant increment in the computational overhead. The graphical representation of the overall procedure of EQIE-FCM algorithm is given in Fig. 1. However, the step-wise procedure of the EQIE-FCM algorithm is summarized in Algorithm1 as follows:

IV. EXPERIMENTAL RESULTS

In the experiments, we compare the performance of the proposed EQIE-FCM approach in comparison with the QIE-FCM approach [38] and other approaches [46]–[51].

A. EXPERIMENTAL ENVIRONMENT

The experiments were performed on the Intel(R) Xeon(R) E5-1607 Workstation PC with 64 GB of memory and running on the Windows 7 Professional operating system with a processing speed of 3.0 GHz. The proposed EQIE-FCM approach and compared QIE-FCM approach is implemented in MATLAB computing environment and executed on MATLAB version R2014. Furthermore, we have obtained the source code of all the other comparative approaches [46]–[51] from the respective sources and perform the experiments on the same partition of the datasets.

B. DATASETS

The performance of the proposed EQIE-FCM approach is evaluated in comparison with other approaches [38], [46]–[51] on various benchmark datasets. These benchmark datasets are taken from the University of California at Irvine (UCI) Machine Learning Repository [52]. Table 2 presents the details of these benchmark datasets. Also,

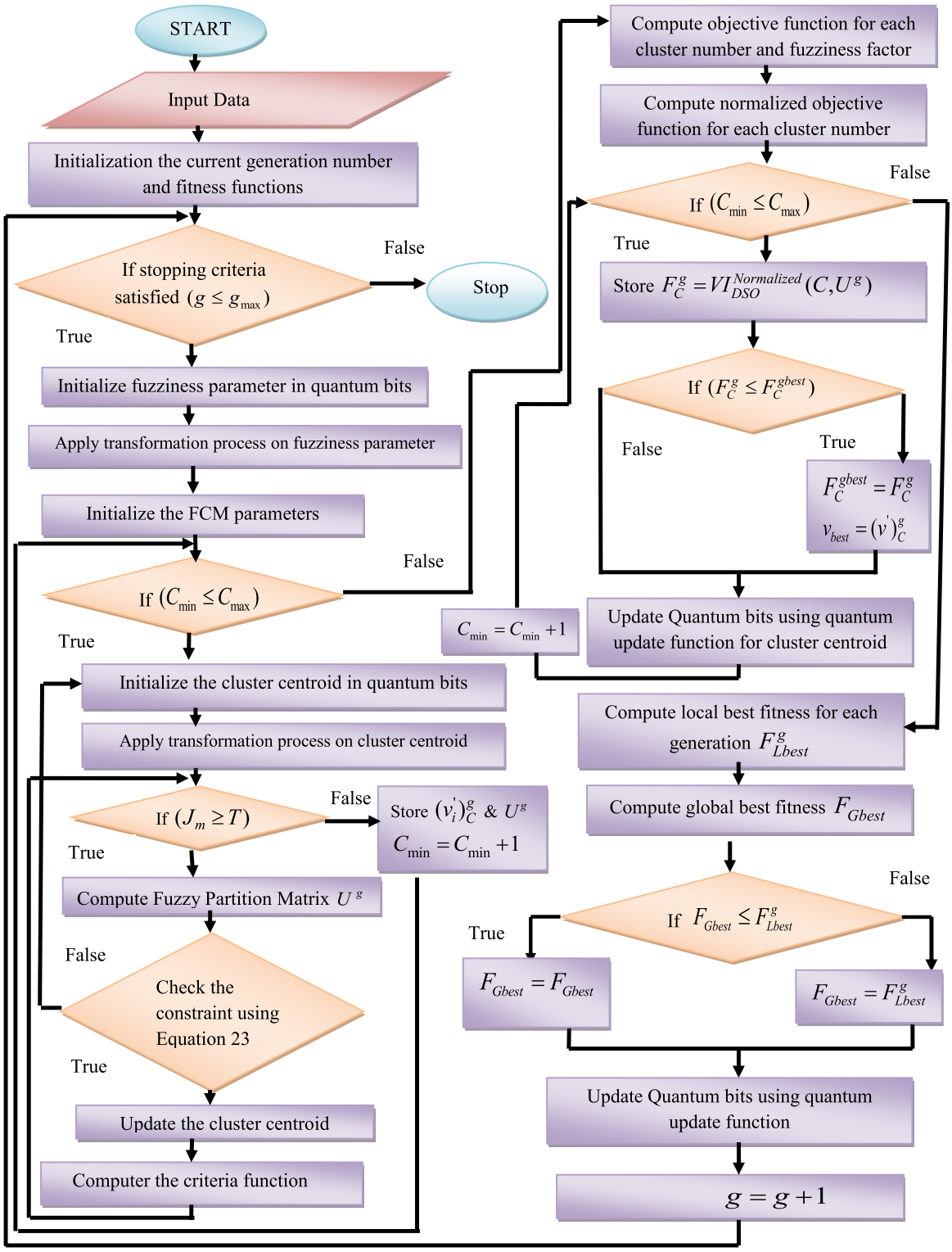

FIGURE 1. The flowchart describing the complete procedure of EQIE-FCM algorithm where the appropriate value of m, C and V_C is evolved in each generation.

the performance of the EQIE-FCM approach is evaluated on a protein database in comparison with other approaches. This protein database used for the experimental study is obtained from the International Protein Sequence Database [43] maintained by the National Biomedical Research Foundation (NBREPIR) at the Georgetown University Medical Center. The protein database is comprised of four superfamilies are RAS, Globin, Trypsin, and Kinase. The detailed

Algorithm 1 EQIE-FCM Algorithm

Input: The dataset $X = [x_1, x_2, ..., x_n]$ consist of n data points. Each data point is represented by a d-dimensional feature vector $x_z = (x_{1z}, x_{2z}, ..., x_{dz})^T \in \mathbb{R}^d$. The best location of set of cluster centers v_{best} is initialized as ϕ. The other parameters F_C^{gbest} and $F_{Gbest}(m_{best}, C_{best})$ is initialized as ∞.

Process:

1: The current generation g is initialized as 1 and set the maximum number of generation g_{max} to 100.

2: **while** $g \leq g_{max}$ **do**

(A) The fuzzifier parameter m for generation g initialized in terms of quantum bit by using Equation (6).

(B) **Call transformation process**(M'_g): Generate the real coded value m_g corresponding to the quantum value M'_g using the transformation process and Equation (10).

(C) Set termination criteria $T = 0.001$, $c_{\max} = \sqrt{n}$, σ, $\Delta\theta$, and ϵ is taken as 0.6, $0.03 \times \pi$, and 0.01, respectively.

(D) **for** $C = 2 : c_{\max}$ **do**

(I) Initialize criteria function $J_{m_g}((v')_C^g : X, m_g, C, U^g) = \infty$ and $(V_{ji})_C^g$ is initialized in terms of quantum bit using Equation (9).

(II) **Call transformation process**($V'_{ij})_C^g$: Generate the real coded value $(v'_{ji})_C^g$ corresponding to the quantum value $(V'_{ij})_C^g$ using the transformation process and Equations (11).

(III) **repeat**

(a) Compute the fuzzy partition matrix $U^g = [\mu_{il}^g]$ for $1 \leq i \leq C$ and $1 \leq l \leq n$.

$$\mu_{il}^g = \frac{\| x_l - (v'_i)_C^g \|^{\frac{-2}{m_g - 1}}}{\sum_{i=1}^{C} \| x_l - (v'_i)_C^g \|^{\frac{-2}{m_g - 1}}} \tag{22}$$

(b) Update the cluster centers $(v'_i)_C^g$ for $1 \leq i \leq C$.

$$(v'_i)_C^g = \frac{\sum_{i=1}^{C} [(\mu_{il}^g)^{m_g}] x_l}{\sum_{i=1}^{C} (\mu_{il}^g)^{m_g}} \tag{23}$$

(c) Compute the criteria function $J_{m_g}((v')_C^g : X, m_g, C, U^g)$ to evaluate the fitness of obtained fuzzy partition.

$$J_{m_g}((v')_C^g : X, m_g, C, U^g) = \sum_{l=1}^{n} \sum_{i=1}^{C} (\mu_{il}^g)^{m_g} \| x_l - (v'_i)_C^g \|^2 \tag{24}$$

until $(J_{m_g}((v')_C^g : X, m_g, C, U^g) \geq T)$

end for

(E) Compute the objective function $VI_{DSO}(C, U^g, m_g)$ [41] to evaluate the fitness of obtained partitions for all the values of C corresponding to m_g.

Algorithm 1 *(Continued.)* EQIE-FCM Algorithm

(F) Compute the summation of $VI_{DSO}(C, U^g, m_g)$ objective function corresponding to all values of C using Equation (14).

(G) Compute the normalized value of objective function $VI_{DSO}(C, U^g, m_g)$ for all values of C over the range $C = 2, 3, ..., c_{\max}$ using Equation (15).

for $C = 2 : c_{\max}$ **do**

i) Store the fitness of fuzzy partition corresponding to each cluster number C in F_C^g using Equation (16).

ii) **if** $(F_C^g \leq F_C^{gbest})$ **then**

$F_C^{gbest} = F_C^g$

$v_{best} = (v')_C^g$

Update the quantum bit of $(V'_{ji})_C^g$ by using Table 1 and Equations (20) and (21).

else

Update the quantum bits of $(V'_{ji})_C^g$ by using Table 1 and Equations (20) and (21).

end if

end for

(H) Compute the local best fitness, i.e., $F_{Lbest}^g(m_g, C)$ by using Equation (17) that determines the best fitness value in generation (g).

(I) Compute the global best fitness denoted by $F_{Gbest}(m_{best}, C_{best})$ using Equation (19) that identify the best value of fuzzifier parameter and the number of clusters from the overall generations.

(J) Update the quantum bit of M'_g by using Table 1 and Equations (20) and (21).

3: Update $g = g + 1$.

4: **end while**

5: **return** m_{best}, C_{best} and best location of set of initial cluster centers v_{best}.

6: **End**

description of these superfamilies used in a protein database for the experimental study is presented in Table 5. The protein sequences present in these superfamilies always contain the characters from the 20-letter amino acid alphabet $\Sigma = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$. These protein sequences vary in the length and may contain a combination of these amino acids in any order. The most important issue while applying an algorithm for the clustering

TABLE 2. Description of the datasets.

Datasets	No. of features	No. of classes	No. of samples
IRIS	4	3	150
Wine	13	3	178
Glass	9	6	214
Vehicle	18	4	946
Pima Indian Diabetes	8	2	768

TABLE 3. Detailed information of superfamilies present in protein database.

Name of superfamilies	Number of sequences	Minimum length of sequence	Maximum length of sequence
RAS	500	171	296
Globin	500	128	339
Trypsin	500	142	485
Kinase	500	215	860

TABLE 4. Parameters specification of EQIE-FCM and QIE-FCM for benchmark datasets and protein database.

Parameters	Description	Values
T [54]	Termination Criteria	0.001
m [29]	fuzzifier Parameter	[1.5, 2.5]
C [40], [41]	Number of clusters	2, 3,..., $c_{\max}$
$c_{\max}$ [27], [41]	Maximum number of clusters	$\sqrt{(n)}$
n [40]	Number of Instances	Size of the datasets
$g_{\max}$	Maximum number of generations	100
σ [42]	Standard deviation	0.6
$\Delta\theta$ [42]	Rotational angle	0.03 ÃŮ ÏĂ
ϵ [42]	Limiting Parameter	0.01

TABLE 5. Detailed information of superfamilies present in protein database.

Name of superfamilies	Number of sequences	Minimum length of sequence	Maximum length of sequence
RAS	500	171	296
Globin	500	128	339
Trypsin	500	142	485
Kinase	500	215	860

of protein sequences is the encoding of the protein sequences. This implies that how these protein sequences can be represented in terms of feature vectors so that these feature vectors can be applied as an input to the clustering algorithm. For this purpose, we have used the encoding technique presented in [53] that entails the extraction of six features corresponding to each protein sequence.

C. PARAMETER SPECIFICATION

In all of our experiments, we ensure that the algorithms [46]–[51] used for comparative analysis starts with the same parameters setting as stated by the researchers. The parameter values used for both the proposed EQIE-FCM algorithm and the QIE-FCM algorithm [38] is kept the same, and the details about it are presented in Table 6. The value of $c_{\max}$ will be different for the different datasets because it depends on the number of instances (n). Thus, the value of $c_{\max}$ for IRIS, Wine, Glass, Vehicle, Pima Indian Diabetes datasets, and Protein Database are $c_{\max} = \sqrt{n} \approx 12$, $c_{\max} = \sqrt{n} \approx 13$, $c_{\max} = \sqrt{n} \approx 14$, $c_{\max} = \sqrt{n} \approx 29$, $c_{\max} = \sqrt{n} \approx 28$, and $c_{\max} = \sqrt{n} \approx 44$, respectively.

D. RESULTS AND DISCUSSION

This section discusses the results of experiments conducted in order to demonstrate the effectiveness of the proposed EQIE-FCM algorithm in comparison with the QIE-FCM algorithm [38] on benchmark datasets and also on the protein database. The efficacy of the EQIE-FCM is measured in terms of the following parameters.

- Determination of best fitness value and fuzzifier parameter.
- Comparison of the best location of cluster centers.
- Computational performance comparison in terms of iterations counts per cluster.

TABLE 6. Parameters specification of eqie-fcm and qie-fcm for benchmark datasets and protein database.

Parameters	Description	Values
T [54]	Termination Criteria	0.001
m [29]	fuzzifier Parameter	[1.5, 2.5]
C [40], [41]	Number of clusters	2, 3,..., $c_{\max}$
$c_{\max}$ [27], [41]	Maximum number of clusters	$\sqrt{(n)}$
n [40]	Number of Instances	Size of the datasets
$g_{\max}$	Maximum number of generations	100
σ [42]	Standard deviation	0.6
$\Delta\theta$ [42]	Rotational angle	0.03 ÃŮ ÏĂ
ϵ [42]	Limiting Parameter	0.01

For each dataset, we compare the performance of EQIE-FCM with QIE-FCM using the parameters as described in Table 6. In Fig. 2 and Fig. 3, we have reported the results of both the algorithms on benchmark datasets and also on protein database. The detailed description and analysis of the results of the parameters discussed above are presented next.

As shown in Fig. 2, the best value of the fuzzifier parameter and the fitness function achieved by the EQIE-FCM algorithm is comparable with the QIE-FCM algorithm for all the benchmark datasets and the protein database. The comparative results are reported on different values of the fuzzifier parameter obtained in 100 generations. As discussed earlier, the VI_{DSO} index [41] is used as the objective function in both the algorithms and the fitness functions formulated in these algorithms are using the VI_{DSO} index which is discussed in section III-D. The fitness functions formulated in these algorithms are used to evaluate the fitness of obtained fuzzy partitions. The small value of the VI_{DSO} index implies the minimum value of fitness function and thus represents the better fuzzy partitions. In this figure for IRIS dataset, the minimum value of the fitness function achieved by the EQIE-FCM algorithm is at $m_{best} = 1.523$ which is in comparison 1.106 times lesser than the value of the fitness function achieved by the QIE-FCM algorithm at $m_{best} = 1.523$. For WINE dataset, the minimum value of the fitness function achieved by the EQIE-FCM algorithm at $m_{best} = 1.6605$ is comparatively 1.157 times smaller than the fitness function value attained by the QIE-FCM at $m_{best} = 1.6605$. Similarly, the minimum value of the fitness function achieved by the EQIE-FCM on GLASS, VEHICLE, and Pima Indian Diabetes datasets at $m_{best} = 1.5154$ is comparatively 1.266 times, 1.183 times, and 1.158 times lesser than the fitness function value attained by the QIE-FCM at $m_{best} = 1.5154$, respectively. Furthermore, on the protein database,

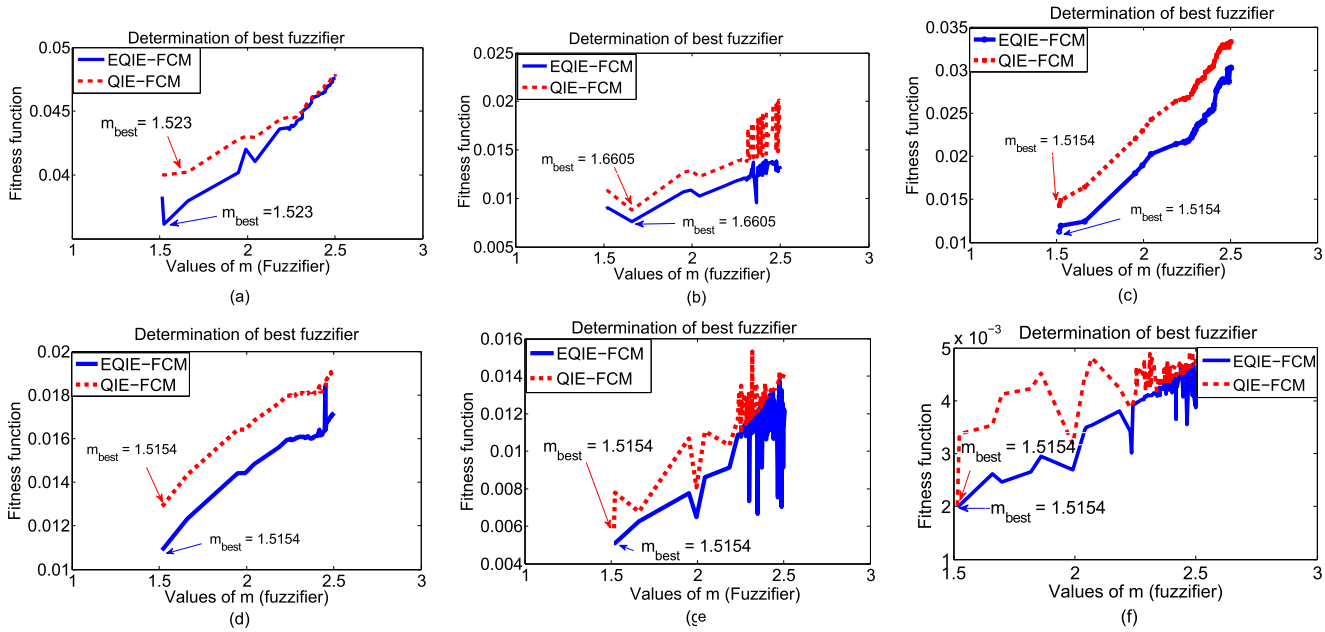

FIGURE 2. Comparison of the EQIE-FCM algorithm with QIE-FCM algorithm indicating the best value of fuzzifier parameter denoted by m_{best} for different datasets. (a) IRIS Dataset. (b) WINE dataset. (c) Glass Dataset. (d) VEHICLE Dataset. (g) Pima Indian Diabetes Dataset. (f) Protein Database.

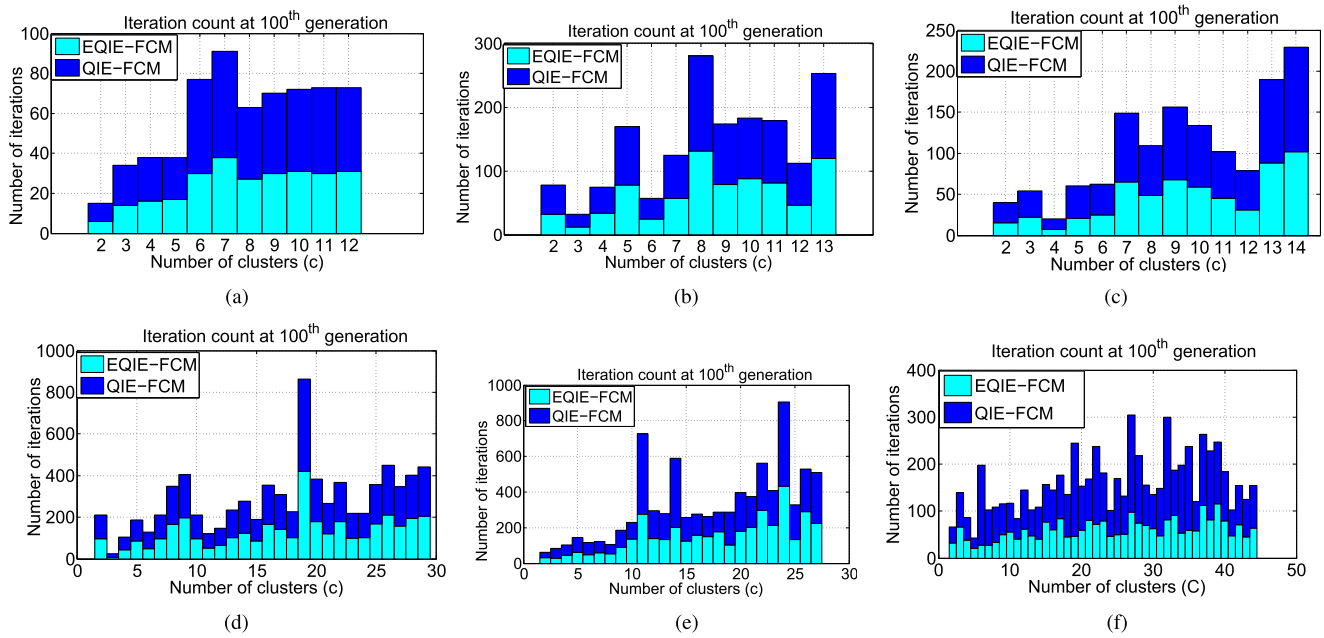

FIGURE 3. Performance comparison of the number of iterations taken by the EQIE-FCM algorithm and the QIE-FCM algorithm. (a) IRIS Dataset. (b) WINE Dataset. (c) GLASS Dataset. (d) VEHICLE Dataset. (e) Pima Indian Diabetes Dataset. (f) Protein Database.

the minimum value of the fitness function achieved by EQIE-FCM is 0.0019985 at $m = 1.5154$ which is comparatively 1.01 times lesser than the fitness function value attained by the QIE-FCM at $m = 1.5154$. Although, both the algorithms for these datasets identifies the same best value of fuzzifier but the EQIE-FCM algorithm achieved a much lesser value of the fitness function in comparison with the QIE-FCM algorithm. Hence, the above-stated results justify the superiority of the EQIE-FCM algorithm over the QIE-FCM algorithm in terms of fitness value.

Table 7 presents the initial cluster center locations predicted by the EQIE-FCM algorithm in comparison with the randomly initialized location of cluster centers taken by the QIE-FCM algorithm. In this table, we have reported the results on only two datasets, i.e., the Protein database and the Pima Indian Diabetes dataset due to space considerations.

TABLE 7. Comparison of initial cluster center locations of the eqie-fcm algorithm with the qie-fcm algorithm.

Dataset	Algorithm	Cluster	Features							
			Feature1	Feature2	Feature3	Feature4	Feature5	Feature6		
Protein Database	EQIE-FCM	1	58.6069	23.2042	24.1565	162.3314	114.3746	11.771		
		2	68.8016	39.8935	18.9785	95.2665	108.8868	17.899		
		3	48.9744	72.5804	1.3634	44.8728	49.1715	28.9451		
		4	50.4323	97.1549	3.5754	176.0261	68.3165	32.546		
		5	61.1354	104.2058	17.8276	161.0221	51.0391	40.9178		
	QIE-FCM	1	30.9005	43.2779	5.512	73.0434	56.4738	18.2194		
		2	31.6776	44.5392	5.6616	74.528	57.7031	18.5033		
		3	31.0665	43.5571	5.501	72.5854	56.782	18.2142		
		4	31.416	43.8388	5.5672	73.18	56.9985	18.2898		
		5	30.9907	43.4089	5.5091	72.8014	56.5984	18.1766		
			NTP	PGC	DBP	TSFT	SI	BMI	DPF	YOA
Pima Indian Diabetes	EQIE-FCM	1	16.9558	169.3641	6.2682	1.2567	347.1789	37.5227	1.0896	43.28969
		2	11.88	108.8	10.44	36.7589	136.7	45.45	0.8345	55.14
		3	4.0778	181.87	1.0892	49.766	197.54	44.936	2.2782	32.147
	QIE-FCM	1	3.38842	121.1809	69.009	20.9514	81.5108	31.6374	0.481	33.3568
		2	3.9476	120.9351	68.7671	20.1197	75.6281	32.2555	0.4686	33.5969
		3	3.7266	121.1631	69.5013	20.5423	81.5276	32.1018	0.4683	32.8297

The results are reported in terms of the location of the cluster centers find by both the algorithms. It is observed that the best location of cluster centers predicted by the EQIE-FCM algorithm is reasonably good because each predicted location of the centers is almost in the middle of the clusters of the dataset. However, the cluster center locations were chosen by the QIE-FCM algorithm for each cluster is almost colliding with each other. Due to the random selection of initial cluster center locations by the QIE-FCM algorithm, the clustering results achieved by this algorithm may trap into the local optima. Conversely, in the EQIE-FCM algorithm, the locations of cluster centers are initially represented in terms of a quantum bit and after several generations of evolution, the EQIE-FCM algorithm comes out with the best location of cluster centers. As we can see in Table 7, cluster center locations predicted by the EQIE-FCM algorithm is more accurate than the randomly chosen location by the QIE-FCM algorithm.

Fig. 3 shows the number of iterations required to find a stable location of cluster centers by the EQIE-FCM algorithm in comparison with the QIE-FCM algorithm for the 100th generation. The results show that the proposed EQIE-FCM algorithm always takes the lesser number of iterations in comparison with the QIE-FCM algorithm for finding a stable location of cluster centers on each cluster number. The reported results show that the QIE-FCM algorithm is much more computationally intensive than the EQIE-FCM algorithm. The reason behind the better computational performance of the EQIE-FCM algorithm in comparison with the QIE-FCM algorithm is that in the QIE-FCM algorithm, the location of initial cluster centers is decided randomly. So, if the data points are located far away from the actual location of initial cluster centers, then the algorithm will converge slowly by taking many iterations to find the proper location of cluster centers. However, in the case of the EQIE-FCM algorithm, due to the selection procedure of location of initial cluster centers, it takes, the less number of iterations in finding the proper location of cluster centers, and hence, results in faster convergence.

E. COMPARISON WITH OTHER EVOLUTIONARY APPROACHES

In order to properly examine the performance of the proposed EQIE-FCM algorithm, we compared it with other clustering algorithms [46]–[51] on the same datasets. The source codes of these compared algorithms are obtained from the respective authors and we perform the experiments on these methods. Table 8 shows the comparative results in terms of four parameters are c_{best}, fitness value, Standard Deviation (SD), and Runtime. The reported result shows that the proposed approach is found to be significantly better than compared approaches in terms of the best value of fitness function and the optimal number of clusters. The optimal number of clusters identified by the proposed EQIE-FCM algorithm for different datasets is similar to the number of clusters as per the geometrical distribution of these datasets in two-dimensional space. Moreover, the best value of the fitness function achieved by the EQIE-FCM algorithm is comparatively much lesser than the fitness value attained by the other compared approaches. Also, it can be seen from the reported results that the runtime of the proposed EQIE-FCM approach is comparatively lesser than the QIE-FCM [25], FCMVGA [35], QM-FCM [33], RQECA [34], KMQGA [36] approaches. Thus, the proposed EQIE-FCM approach is much faster in terms of runtime than the QIE-FCM [25], FCMVGA [35], QM-FCM [33], RQECA [34], KMQGA [36] approaches but it is comparatively little slower than the K-Means [37] and Affinity Propagation [38]. Hence,

TABLE 8. Performance comparison of the eqie-fcm algorithm with other approaches.

Approaches	Parameters	Datasets					
		IRIS	WINE	GLASS	VEHICLE	PIMA	PROTEIN
EQIE-FCM	c_{best}	2	2	4	3	3	5
	fitness value	0.03614	0.00761	0.01127	0.01089	0.00508	0.00199
	SD	0.05291	0.02883	0.01145	0.02463	0.09821	0.06541
	Runtime (Min:Sec)	2:09	2:58	4:10	9:39	6:48	9:13
QIE-FCM [38]	c_{best}	2	2	4	3	3	5
	fitness value	0.0401	0.0108	0.0143	0.0129	0.0067	0.0021
	SD	0.0985	0.056	0.0231	0.0456	1.098	0.9881
	Runtime (Min:Sec)	2:27	3:19	5:18	10:37	7:41	10:21
FCMVGA [48]	c_{best}	6	12	2	5	5	7
	fitness value	4.8024	4.03309	36.0576	48.098	0.01102	0.06487
	SD	2.0938	1.9739	2.0973	2.3473	0.0382	0.0115
	Runtime (Min:Sec)	4:12	6:24	8:02	14:51	11:50	14:31
QM-FCM [46]	c_{best}	4	5	3	5	16	9
	fitness value	0.3866	0.0724	0.4201	1.4289	0.0045	0.0089
	SD	1.0283	1.0982	1.9822	1.7362	0.2919	1.8909
	Runtime (Min:Sec)	2:55	4:20	5:48	12:10	9:12	11:28
RQECA [47]	c_{best}	5	7	14	25	22	11
	fitness value	0.7587	0.4865	1.8975	4.561	0.0011	0.0045
	SD	1.0922	1.0083	1.0022	1.0237	0.0181	0.1982
	Runtime (Min:Sec)	3:48	5:42	6:31	13:58	10:32	12:49
KMQGA [49]	c_{best}	3	3	2	4	2	4
	fitness value	14.131	11.186	64.1.06	14.812	13.866	22.805
	SD	3.549	2.932	2.682	2.448	2.217	3.918
	Runtime (Min:Sec)	5:24	7:46	9:37	15:39	12:48	13:46
K-Means [50]	c_{best}	6	3	6	4	2	6
	fitness value	102.57	69.6	241.03	226	182.39	48.786
	SD	11.34	1.0834	14.41	4.067	10.807	2.1989
	Runtime (Min:Sec)	1:08	1:59	2:32	6:41	4:08	7:38
Affinity Propagation [51]	c_{best}	12	12	14	11	8	9
	fitness value	0.3989	0.3881	0.4988	0.5289	0.3642	0.1097
	SD	0.01023	0.02102	0.0301	0.0191	0.0208	0.0191
	Runtime (Min:Sec)	1:51	2:15	3:45	8:21	5:50	8:25

the discussed results quantify the effectiveness of proposed EQIE-FCM algorithm over compared approaches.

V. CONCLUSION

In this article, we proposed an Enhanced Quantum-Inspired Evolutionary Fuzzy C-Means (EQIE-FCM) algorithm for clustering of data. In the fuzzy based clustering approach, the fuzzifier parameters m, number of clusters C and the selection of the initial cluster centers are the most important parameters for the effective clustering of datasets. The EQIE-FCM algorithm performs the clustering of datasets by evolving these parameters in several generations using the quantum computing concept. These parameters are evolved in each generation using five major operations: Representation of the fuzzifier parameter and the cluster centers in the quantum bit, transformation process, formulation of the local and the global fitness function and quantum update function. After several generations of evolution, we get an optimal value of these parameters from subspaces. To investigate its effectiveness, we tested it on various benchmark datasets and a real-life protein database which consist of four superfamilies. The proposed algorithm outperformed the QIE-FCM and other evolutionary clustering algorithms and acquired promising results.

We wish to investigate the possibility of extending the current EQIE-FCM algorithm by making it a scalable algorithm to be implemented in Apache Spark on Hadoop cluster so that it can efficiently handle the clustering of big datasets. Thus, it provides an important direction for our future work.

REFERENCES

[1] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1370–1386, Nov. 2004.

[2] P. D'haeseleer, "How does gene expression clustering work?" *Nature Biotechnol.*, vol. 23, no. 12, pp. 1499–1501, 2005.

[3] H. Kawaji, Y. Yamaguchi, H. Matsuda, and A. Hashimoto, "A graph-based clustering method for a large set of sequences using a graph partitioning algorithm," *Genome Inform.*, vol. 12, no. 1, pp. 93–102, 2001.

[4] D. K. Slonim, "From patterns to pathways: Gene expression data analysis comes of age," *Nature Genet.*, vol. 32, pp. 502–508, Dec. 2002.

[5] J. Quackenbush, "Computational analysis of microarray data," *Nature Rev. Genet.*, vol. 2, no. 6, pp. 418–427, 2001.

[6] N. Kaplan, M. Friedlich, M. Fromer, and M. Linial, "A functional hierarchical organization of the protein sequence space," *BMC Bioinf.*, vol. 5, no. 1, p. 196, 2004.

[7] N. Krasnogor and D. A. Pelta, "Measuring the similarity of protein structures by means of the universal similarity metric," *Bioinformatics*, vol. 20, no. 7, pp. 1015–1021, 2004.

[8] A. Paccanaro, J. A. Casbon, and M. A. S. Saqi, "Spectral clustering of protein sequences," *Nucleic Acids Res.*, vol. 34, no. 5, pp. 1571–1580, 2006.

[9] J. C. Dunn, *A Fuzzy Relative of the ISODATA Process and its use in Detecting Compact Well-Separated Clusters*. London, U.K.: Taylor & Francis, 1973.

[10] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic, 1981.

[11] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

[12] C. S. Wallace and D. M. Boulton, "An information measure for classification," *Comput. J.*, vol. 11, no. 2, pp. 185–194, 1968.

[13] C. S. Wallace and P. Freeman, "Estimation and inference by compact coding," *J. Roy. Stat. Soc., B Methodol.*, vol. 49, no. 3, pp. 240–265, 1987.

[14] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "Hybrid clustering analysis using improved krill herd algorithm," *Appl. Intell.*, vol. 48, no. 11, pp. 4047–4071, 2018.

[15] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A combination of objective functions and hybrid Krill herd algorithm for text document clustering analysis," *Eng. Appl. Artif. Intell.*, vol. 73, pp. 111–125, Aug. 2018.

[16] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A new feature selection method to improve the document clustering using particle swarm optimization algorithm," *J. Comput. Sci.*, vol. 25, pp. 456–466, Mar. 2018.

[17] L. M. Q. Abualigah and E. S. Hanandeh, "Applying genetic algorithms to information retrieval using vector space model," *Int. J. Comput. Sci., Eng. Appl.*, vol. 5, no. 1, p. 19. 2015.

[18] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A hybrid strategy for krill herd algorithm with harmony search algorithm to improve the data clustering," *Intell. Decis. Technol.*, vol. 12, no. 1, pp. 3–14, 2018.

[19] L. M. Abualigah, A. T. Khader, E. S. Hanandeh, and A. H. Gandomi, "A novel hybridization strategy for krill herd algorithm applied to clustering techniques," *Appl. Soft Comput.*, vol. 60, pp. 423–435, Nov. 2017.

[20] L. M. Abualigah and A. T. Khader, "Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering," *J. Supercomput.*, vol. 73, no. 11, pp. 4773–4795, 2017.

[21] M. Mouring, K. Dhou, and M. Hadzikadic, "A novel algorithm for bi-level image coding and lossless compression based on virtual ant colonies," in *Proc. COMPLEXIS*, 2018, pp. 72–78.

[22] K. Dhou, "A Novel agent-based modeling approach for image coding and lossless compression based on the wolf-sheep predation model," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2018, pp. 117–128.

[23] M. Prasad, C.-T. Lin, D.-L. Li, C. T. Hong, W.-P. Ding, and J. Y. Chang, "Soft-boosted self-constructing neural fuzzy inference network," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 584–588, Mar. 2017.

[24] A. Saxena *et al.*, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, Dec. 2017.

[25] M. Prasad, Y. Y. Lin, C.-T. Lin, M. J. Er, and O. K. Prasad, "A new data-driven neural fuzzy system with collaborative fuzzy clustering mechanism," *Neurocomputing*, vol. 167, pp. 558–568, Nov. 2015.

[26] C.-T. Lin, M. Prasad, and A. Saxena, "An improved polynomial neural network classifier using real-coded genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 11, pp. 1389–1401, Nov. 2015.

[27] D.-W. Kim, K. H. Lee, and D. Lee, "On cluster validity index for estimation of the optimal number of fuzzy clusters," *Pattern Recognit.*, vol. 37, no. 10, pp. 2009–2025, 2004.

[28] M. Ramze Rezaee, B. P. F. Lelieveldt, and J. H. C. Reiber, "A new cluster validity index for the fuzzy c-mean," *Pattern Recognit. Lett.*, vol. 19, nos. 3–4, pp. 237–246, 1998.

[29] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.

[30] S. F. Altschul *et al.*, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, no. 17, pp. 3389–3402, 1997.

[31] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," *Proc. Nat. Acad. Sci. USA*, vol. 85, no. 8, pp. 2444–2448, 1988.

[32] H. Pirim, B. Ekşioğlu, A. D. Perkins, and C. C. Yüceer, "Clustering of high throughput gene expression data," *Comput. Oper. Res.*, vol. 39, no. 12, pp. 3046–3061, 2012.

[33] W. Ding, J. Xie, D. Dai, H. Zhang, H. Xie, and W. Zhang, "CNNcon: Improved protein contact maps prediction using cascaded neural networks," *PLoS ONE*, vol. 8, no. 4, 2013, Art. no. e61533.

[34] A. J. Enright, S. V. Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucl. Acids Res.*, vol. 30, no. 7, pp. 1575–1584, 2002.

[35] L. Tari, C. Baral, and S. Kim, "Fuzzy c-means clustering with prior biological knowledge," *J. Biomed. Inform.*, vol. 42, no. 1, pp. 74–81, 2009.

[36] A. B. McBratney and A. W. Moore, "Application of fuzzy sets to climatic classification," *Agricult. Forest Meteorol.*, vol. 35, no. 1, pp. 165–185, 1985.

[37] D. Dembélé and P. Kastner, "Fuzzy C-means method for clustering microarray data," *Bioinformatics*, vol. 19, no. 8, pp. 973–980, 2003.

[38] O. P. Patel, N. Bharill, and A. Tiwari, "A quantum-inspired fuzzy based evolutionary algorithm for data clustering," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Aug. 2015, pp. 1–8.

[39] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.

[40] F. Höppner, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Hoboken, NJ, USA: Wiley, 1999.

[41] N. Bharill and A. Tiwari, "Enhanced cluster validity index for the evaluation of optimal number of clusters for fuzzy C-means algorithm," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Jul. 2014, pp. 1526–1533.

[42] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, H_ϵ gate, and two-phase scheme," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 156–169, Apr. 2004.

[43] W. C. Barker *et al.*, "The protein information resource (PIR)," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 41–44, 2000.

[44] S. M. Ross, *Introduction to Probability Models*. New York, NY, USA: Academic, 2014.

[45] T.-C. Lu, G.-R. Yu, and J.-C. Juang, "Quantum-based algorithm for optimizing artificial neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1266–1278, Aug. 2013.

[46] C.-C. Hung *et al.*, "A quantum-modeled fuzzy C-means clustering algorithm for remotely sensed multi-band image segmentation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2013, pp. 2501–2504.

[47] H. Wang, W. Zhu, J. Liu, L. Li, and Z. Yin, "Multidistribution center location based on real-parameter quantum evolutionary clustering algorithm," *Math. Problems Eng.*, vol. 2014, Nov. 2014, Art. no. 714657.

[48] S. Bandyopadhyay and U. Maulik, "Nonparametric genetic clustering: Comparison of validity indices," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 31, no. 1, pp. 120–125, Feb. 2001.

[49] J. Xiao, Y. Yan, J. Zhang, and Y. Tang, "A quantum-inspired genetic algorithm for k-means clustering," *Expert Syst. Appl.*, vol. 37, no. 7, pp. 4966–4973, 2010.

[50] H. Ismkhan, "I-k-means−+: An iterative clustering algorithm based on an enhanced version of the k-means," *Pattern Recognit.*, vol. 79, pp. 402–413, Jul. 2018.

[51] L. Wang, Q. Ji, and X. Han, "Adaptive semi-supervised affinity propagation clustering algorithm based on structural similarity," *Tehnicki Vjesnik/Tech. Gazette*, vol. 23, no. 2, pp. 425–435, 2016.

[52] C. Blake and C. J. Merz, "UCI repository of machine learning databases," Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep., vol. 55, no. 1, 1998. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[53] N. Bharill, A. Tiwari, and A. Rawat, "A novel technique of feature extraction with dual similarity measures for protein sequence classification," *Procedia Comput. Sci.*, vol. 48, no. 1, pp. 796–802, 2015.
[54] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, "Fuzzy c-means algorithms for very large data," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 6, pp. 1130–1146, Dec. 2012.

NEHA BHARILL received the B.E. degree in information technology from the Bansal Institute of Science and Technology, Bhopal, India, in 2008, the M.E. degree in computer engineering from the Shri Govindram Seksaria Institute of Technology and Science, Indore, in 2011, and the Ph.D. degree in computer science and engineering from IIT Indore, India, in 2018. She has been an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Information Technology Dharwad, India, since 2017. She has published various refereed journals and conferences of international repute. Her current research interests include fuzzy sets and systems, big data, pattern recognition, data mining, machine learning, and scalable machine learning approaches for genome identification. She is also a member of various research societies, such as the IEEE Computational Intelligence Society and the Soft Computing research Society. She is also a Reviewer of the IEEE Transactions on Cybernetics, IEEE Transactions on Fuzzy Systems, IEEE Transactions on Big Data, *Swarm and Evolutionary Computation* (Elsevier), and *Complex & Intelligent Systems* (Springer).

OM PRAKASH PATEL received the M.E. degree from the Department of Computer Science and Engineering, Shri Govindram Seksaria Institute of Technology and Science, Indore, India, in 2011, and the Ph.D. degree in computer science and engineering from the IIT Indore, India, in 2018. He has been an Assistant Professor with the Department of Computer Science and Engineering, KLE Technological University, India, since 2018. His current research interests include quantum-based neural network learning algorithm, pattern recognition, data mining, and fuzzy-based soft computing algorithm. He is a reviewer of *Neural Processing Letters* (Springer).

ARUNA TIWARI received the B.E. and M.E. degrees in computer engineering and the Ph.D. degree in computer science and engineering. She has been an Associate Professor of computer science and engineering with IIT Indore, since 2012. She has research collaboration with the CSIR CEERI Pilani and the Indian Institute of Soybean Research (Indian Council of Agricultural Research). She has many publications in peer-reviewed journals (of the IEEE Transactions, Elsevier, and Springer), international conferences, and book chapters. Her research interests include soft computing, neural networks, fuzzy clustering, evolutionary computation, genome analysis, and health-care applications. She has been a Reviewer for many reputed journals and conferences.

LIFENG MU received the Ph.D. degree from the College of Information Science and Engineering, Northeastern University, China, in 2010. Prior to joining the SHU-UTS SILC Business School, he had held variant research positions in word leading research laboratories and universities, including an Optimization Expert at the IBM China Development Laboratory and a Research Associate at The Hong Kong Polytechnic University. He is currently with the SILC Business School, Shanghai University. He has been very actively involved in business analytics and optimization consulting activities for major industry groups, including IBM, the Ministry of Railways, China, DB Schenker, German, the Central Iron & Steel Research Institute, China, the China Academy of Railway Sciences, and Chongqing CISDI Engineering Consulting, China. He also investigates the combinatorial optimization problems with applications in railway industry. He has produced more than ten scientific papers. His works have been published in leading international journals and conferences, such as the *European Journal of Operational Research*, *Computers & Industrial Engineering*, *Computers & Mathematics with Applications*, and IJUFKS. His main research interests include operations research, optimization theory, optimization problems in industries, and artificial intelligence.

DONG-LIN LI received the master's degree in electrical engineering from National Chung Hsing University, Taichung, Taiwan, in 2006, and the Ph.D. degree in electrical engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2013, where he is currently an Associate Research Fellow with National Chiao Tung University. He has published papers in international journal and conferences, including the IEEE Transactions, ACM, Elsevier, and Springer. His current research interests include machine learning, fuzzy systems, neural networks and computer vision, and smart living technology.

MANORANJAN MOHANTY received the B.Sc. degree in computer science from the Ravenshaw College, Cuttack, India, the master's degree in computer applications from Jawaharlal Nehru University, New Delhi, India, and the Ph.D. degree in computer science from the School of Computing, National University of Singapore, Singapore. He was a Postdoctoral Researcher with the Center for Cyber Security, New York University Abu Dhabi, United Arab Emirates. He is currently a Lecturer with the Department of Computer Science, University of Auckland, New Zealand. His research interests include encrypted domain processing, digital forensics, multimedia security, and privacy-preserving biometrics.

OMPRAKASH KAIWARTYA received the Ph.D. degree in computer science from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India, in 2015.

He was a Research Associate with Northumbria University, Newcastle, U.K., and a Postdoctoral Research Fellow with the Universiti Teknologi Malaysia (UTM). He was a Postdoctoral Research Fellow with the Faculty of Computing, UTM, Malaysia, until 2017. He is currently a Lecturer with the School of Science and Technology, Nottingham Trent University (NTU), U.K. He is also a research-focused Lecturer with the School of Science and Technology, NTU. His recent scientific contributions are in then Internet of connected Vehicles (IoV), electronic vehicles charging management, the Internet of Healthcare Things (IoHT), and smart use case implementations of sensor networks. His research interests include the IoT centric future technologies for diverse domain areas, including transport, healthcare, and industrial production, the Internet of Vehicles (IoV), electronic vehicles (EVs), Vehicular Ad-hoc Networks (VANETs), and wireless sensor networks (WSNs). He is also an Associate Editor of reputed SCI journals, including the *IET Intelligent Transport Systems*, the IEEE Internet of Things Journal, the *EURASIP Journal on Wireless Communications and Networking*, and *Ad Hoc & Sensor Wireless Networks*. He is also the Guest Editor of many recent special issues in reputed journals, including the IEEE Internet of Things Journal, the IEEE Access, *Sensors* MDPI, and *Electronics* (MDPI).

MUKESH PRASAD received the master's degree in computer application from Jawaharlal Nehru University, New Delhi, India, in 2009, and the Ph.D. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2015. He is currently a Lecturer with the School of Software, University of Technology Sydney, Australia. He has published papers in international journal and conferences, including the IEEE Transactions, ACM, Elsevier, and Springer. His current research interests include machine learning, pattern recognition, fuzzy systems, neural networks, artificial intelligence, and brain–computer interface.