Shared Nearest-neighbor Quantum Game-based Attribute Reduction with Hierarchical Coevolutionary Spark and Its Application in Consistent Segmentation of Neonatal Cerebral Cortical Surfaces

Weiping Ding*, *Member*, *IEEE*, Chin-Teng Lin, *Fellow, IEEE*, and Zehong Cao *Member*, *IEEE*

***Abstract*—**The unprecedented increase in data volume has become a severe challenge for conventional patterns of data mining and knowledge discovery tasked with handling big data. The recently introduced Spark platform is a new processing method for big data analysis that has attracted increasing attention from both the scientific community and industry. In this paper, we propose a shared nearest-neighbor quantum game-based attribute reduction algorithm (SNNQGAR) that incorporates the hierarchical coevolutionary Spark model. We first present a shared coevolutionary nearest-neighbor hierarchy with self-evolving compensation that considers the features of nearest-neighborhood attribute subsets and calculates the similarity between attribute subsets according to the shared neighbor information of attribute sample points. We then present a novel weight tensor attribute model to generate ranking vectors of attributes and apply them to balance the relative contributions of different neighborhood attribute subsets. To optimize the model, we propose an embedded quantum equilibrium game paradigm to can ensure that noisy attributes do not degrade the big data reduction results. A combination of the hierarchical coevolutionary Spark model and an improved MapReduce framework is then constructed that it can better parallelize SNNQGAR to efficiently determine the preferred reduction solutions of the distributed attribute subsets. Experimental comparisons demonstrate the superior performance of SNNQGAR, which outperforms most of the state-of-the-art attribute reduction algorithms. Moreover, the results indicate that SNNQGAR can be successfully applied to segment overlapping and interdependent fuzzy cerebral tissues, and that it exhibits a stable and consistent segmentation performance for neonatal cerebral cortical surfaces.

***Index Terms*—** Shared nearest-neighbor hierarchy, attribute reduction, quantum equilibrium game paradigm, hierarchical coevolutionary Spark, neonatal cortical surface segmentation.

I. INTRODUCTION

Big data has become a hot topic in many aspects of research and industry and is currently experiencing explosive growth characterized by the "Five Vs" (high volume, variety, velocity, veracity and value). These "Five Vs" are the key features of big data and the cause of inherent uncertainties in representing, processing, and analyzing big data [1][2][3]. As research interest in artificial intelligence (AI) and big data application fields has increased over the past decade, their scientific, technological and application prospects have been dramatically improved [4][5][6]. Nevertheless, efficient big data analytics requires significant innovations in existing AI techniques. Due to the uncertain and intricate nature of big data, existing AI techniques must be modified so that they are capable of producing quality analytics while remaining practical for real-world deployment under modest computational resources. Although big data can provide a large candidate attribute set, most of these attributes are irrelevant or redundant, which degrades the learning performance of AI algorithms [7][8]. This situation requires adjusting the details of the solution space to transform the original problem into one with a higher level of abstraction to account for imprecision, uncertainty and inaccuracy during the decision-making process. Attribute reduction for knowledge acquisition of big data is a crucial preprocessing step that reduces the modeling complexity. To winnow out superfluous attributes, we must design better potential attribute reduction algorithms that provide more efficient solutions to reduce large candidate attribute sets in which redundancy, uncertainty and imprecision exist.

Rough set theory (RST) was introduced by Pawlak as a tool to handle uncertainty caused by indiscernibility and incompleteness [9][10][11][12]. RST offers a theoretical framework that supports attribute reduction, which is a crucial preprocessing step in data mining [13][14][15][16]. The uncertainty in big data results from biased domain knowledge, imperfect measurements, faulty sensors, operator errors and other factors that reflect real-world conditions. Currently, attribute reduction based on RST has become an effective approach for using information granules to build efficient models for complex applications, especially for those encountered in big data contexts [17][18][19]. Developing attribute reduction algorithms can be helpful for big data analyses that present imprecision, uncertainty and vagueness.

In recent years, significant advances in RST have been made in the scientific and engineering domains, and numerous attribute reduction algorithms have been proposed and discussed. In the classical RST model, the classification quality, information entropy, positive regions and lower approximation bounds under each decision class vary consistently and monotonically during the attribute reduction procedure. For example, Wang [20] demonstrated the equivalence of the reductions obtained using the positive region and information entropy. Hu et al. [21] derived a few attribute significance measures based on a fuzzy rough model and constructed a novel forward greedy attribute reduction algorithm. Yao et al. [22] proposed a discernibility matrix-based attribute reduction method by simplifying the matrix to reduce the computational

* CIBCI, Centre for Artificial Intelligence, FEIT, University of Technology Sydney, Australia. E-mail: Weiping.Ding@uts.edu.au

cost. Chen et al. [23] presented a sample pair selection procedure to complete the attribute reduction procedure. Li et al. [24] developed a decision-theoretic RST model based on neighborhoods to analyze the positive regions related to attribute reduction. Yeung et al. [25] proposed different classes of generalized upper and lower approximation operators of fuzzy rough sets. Chen et al. [26] explored an integrated classification method to simultaneously select useful attributes and extract fuzzy rules. Because the classical RST models are sensitive to noisy information, An et al. [27] presented a robust fuzzy RST model based on a soft minimum enclosing sphere with a fuzzy dependency function; however, this model suffered from a large search space and resulted in a large number of inferior solutions. Zhang et al. [28] proposed an attribute reduction and approximate reasoning framework via an α-dominance-based quantitative RST for large-scale set-valued information tables. Yang et al. [29] defined the relative discernibility relation of condition attributes and developed two algorithms to identify reductions in variable-precision rough sets.

Although attribute reduction algorithms based on RST have shown promising performances in classification rule acquisition, recommender systems, regression learning and intrusion detection, these algorithms unavoidably generate an exponential number of irrelevant attribute subsets that are flooded with interesting interrelated features. Thus, when a large number of new objects are simultaneously generated in a single database, a considerable waste of executed space and computational time occurs when traditional attribute reduction algorithms are employed.

Therefore, to effectively address this problem of attribute reduction for big data, new approaches must be explored via distributed computational strategies. The MapReduce model devised by Google in 2003 is intended to operate on large-scale datasets in a distributed environment, and it has become a well-known parallel framework [30][31][32]. MapReduce supplies a distributed file system that can store massive datasets and provides a suitable method for parallelizing big data analyses. Benefiting from its parallelization framework, MapReduce continues to attract growing research interest because of its applicability in big data analysis, where enormous amounts of data are partitioned and scattered across numerous computing nodes. Several researchers have endeavored to exploit parallel attribute reduction algorithms based on MapReduce. For example, Qian et al. [33] proposed parallel attribute reduction strategies based on MapReduce to enhance computational efficiency. Zhang et al. [34] investigated parallel algorithms for knowledge acquisition on MapReduce and presented three different methods based on parallel matrixes to operate on big data. Zhang et al. [35] proposed a parallel implementation for computing composite rough set approximations on Multiple GPUs. Chen et al. [36] studied a parallel attribute reduction algorithm using dominance-based neighborhood rough sets, which mainly considered the partial orders among the numerical and categorical attribute values. El-Alfy et al. [37] adopted the MapReduce model to determine the minimum rough set reduction by employing a parallel genetic algorithm implementation. These extensive research efforts have directly impacted attribute reduction in many big data applications. However, although they benefit from the parallelization framework, these parallel attribute reduction algorithms must output many intermediate results to the Hadoop Distributed File System (HDFS), which causes large amounts of disk and network I/O. Consequently, these algorithms are still time-consuming when processing big data, and their performances might be unreliable.

Literature surveys reveal that although certain attribute reduction algorithms have been used for big data processing, most of them may be inefficient at performing complex attribute reduction and at extracting useful knowledge from these dynamically changing massive datasets. This situation arises not only because of the algorithms' scalability but also because of the uncertain and intricate nature of big data. Thus, attribute reduction processes that address big data can be time-consuming, and the available algorithms suffer from the following limitations and challenges.

As previously indicated, the high-dimensional number of attributes with complex structures and ever-greater volumes lead attribute reduction algorithms to become either inapplicable or ineffective in the attribute space. These algorithms must be modified so that they are capable of producing quality analytics while remaining practical for real-time deployment under modest computational resources. Designing an efficient algorithm to rapidly solve the attribute reduction problem of big data first requires determining an attribute reduction model that prefers to select the nearest-neighborhood attribute subsets. Although efforts have been made to define and characterize attribute reduction methods with MapReduce, we must still address the limitations of existing MapReduce structures and interactions through dynamically adapting MapReduce with the reorganization model. Notwithstanding the advantages of the MapReduce technique, determining a method of addressing speckle noise is one of the most difficult problems because speckle noise is basically multiplicative. Another challenge in big data research pertains to the uncertainty issue. Uncertainty in big data results from biased domain knowledge, imperfect measurements, and other factors that reflect real-world situations. This challenge requires adjusting the details in the solution space to transform the original problem into one with a higher level of abstraction that can account for imprecision, uncertainty and inaccuracy in both the decision-making process and the knowledge sources. Further research on these issues calls for an exploration of the critical challenges, and the development of a systematic and effective attribute reduction model and algorithm for big data to improve the quality of solutions and decrease the computational complexity.

Attribute reduction in big data relies on distributed computational strategies because the data cannot be stored and processed in a single node. Apache Spark (hereafter, Spark) has become a well-recognized tool for sophisticated big data analysis [38][39][40]. Moreover, Spark performs better than MapReduce for iterative algorithms and interactive data analysis, allowing an enormous amount of data to be partitioned and scattered into a number of computing nodes [41][42]. Spark has two main components: a Master and Workers. One master node assigns jobs to the Worker nodes [43]. Although Spark continues to attract growing research interest in the realm of big data, attribute reduction with Spark remains almost uncharted research territory. Moreover, the scalability of attribute reduction presents additional challenges when addressing large amounts of big data.

Motivated by the above observations, we aim to address complex big data attribute reduction from high-dimensional attribute space, and we propose a shared nearest-neighbor

quantum game-based attribute reduction algorithm (SNNQGAR) using hierarchical coevolutionary Spark. This algorithm avoids the limitations of traditional algorithms based on RST and expands the attribute weight tensor into the shared nearest-neighbor relation to partition complex attribute sets with unstructured, uncertain and imprecise data. The proposed SNNQGAR algorithm can be parallelized to improve the processing efficiency of hierarchical coevolutionary Spark. Moreover, SNNQGAR shows additional benefits as the attribute-noise ratio increases. The major contributions of this paper are fourfold.

- First, we present a shared coevolutionary nearest-neighbor hierarchy (SCNNH) with self-evolving compensation that ensures the similarity between attribute subsets is calculated according to the shared neighbor information of sample attributes. The proposed coevolutionary procedure converges quickly, which greatly improves the classification accuracy.
- Second, we construct a novel attribute weight tensor to generate ranking vectors for the nearest-neighbor attributes. This approach balances the relative contributions of different neighbor attribute subsets. Then, we propose a quantum equilibrium game paradigm (QEGP) to ensure that uncertain and imprecise attributes cannot degrade the final attribute reduction results. Hence, all the useful candidate attribute subsets are well preserved in the attribute space.
- Third, we propose a new hierarchical coevolutionary Spark model combined with an improved MapReduce. This model allows better parallelization of the proposed SNNQGAR algorithm while also providing efficient attribute reduction solutions for big data analytics.
- Finally, the proposed SNNQGAR algorithm is successfully applied to complex neonatal brain regions to perform consistent segmentations of cerebral cortical surfaces. The experimental results show that SNNQGAR can segment overlapping and interdependent fuzzy cerebral tissues, and its results are consistent with those of expert manual segmentations.

The remainder of this paper is organized as follows. Section II presents pertinent preliminary data on attribute reduction based on RST. Section III describes the hierarchical coevolutionary Spark model in detail. Section IV proposes a shared coevolutionary nearest-neighbor hierarchy with a self-evolving compensator. Section V establishes a novel weight tensor-based quantum equilibrium game paradigm. Section VI details the primary steps of SNNQGAR. Section VII presents the experimental results for the datasets and their corresponding analyses. Section VIII presents the performances of the application to consistently segment neonatal cerebral cortical surfaces, and Section IV presents conclusions.

II. Preliminaries

This section provides the relevant definitions for attribute reduction based on RST.

Definition 1. In RST, an information system is characterized as $S=(U,A,V,f)$, where

$\begin{cases} U \text{ is a nonempty finite set of objects;} \\ A \text{ is a nonempty finite set of attributes (features);} \\ V \text{ equals } \bigcup_{a\in A} V_a, V_a \text{ and is a domain of the attribute;} \end{cases}$

f is an information function $U\times A\to V$, such that $f(x,a)\in V_a$ for every $x\in U, a\in A$.

More specifically, a composite information system is also called a composite decision table when conditions and decision attributes are included in the information system. This table is denoted by $CDT=(U,C \quad D,V,f)$, where C is a finite set of condition attributes, and D is a finite set of decision attributes.

Definition 2. Each nonempty subset $B\in A$ determines the indiscernibility relation, which is denoted as follows:

$$Re_B=\{(x,y)\in U\times U \mid f(x,a)=f(y,a),\forall a\in B\}. \quad (1)$$

The relation R_B partitions U into equivalence classes via

$$U/R_B=\{[x]_B \mid x\in U\}, \quad (2)$$

where $[x]_B$ represents the equivalence class, which is determined by x with respect to B. U/R_B is often abbreviated as U/B.

Definition 3. A partial relation on the family $U/B=\{B\subseteq A\}$ is denoted as $U/P \quad U/Q$. $Q_j\in U/Q$ is observed for each $P_i\in U/P$ and $P_i\subseteq Q_j$, where $U/P=\{P_1,P_2,...,P_m\}$ and $U/Q=\{Q_1,Q_2,...,Q_n\}$ are partitions induced by $P,Q\in A$. Thus, P is more finely defined than is Q.

Definition 4. For an object $x\in U$, its membership in the fuzzy positive region is defined as

$$POS_B(D)(x)=\bigcup_{i=1}^{r} \underline{B}D_i(x), \quad (3)$$

where D is a set of decision attributes.

Definition 5. This fuzzy rough dependency is used to evaluate the significance of a subset of features in the feature space, and it is defined as the ratio of the size of the positive regions over all samples and expressed as follows:

$$\gamma_B(D)=\frac{\sum_{x\in U} POS_B(D)(x)}{|U|}, \quad (4)$$

where $0\le\gamma_B(D)\le 1$ denotes the union operation and $|\cdot|$ denotes the set cardinality. The closer $\gamma_B(D)$ is to 1, the more D depends on B.

Definition 6. If an attribute a can be removed from a set of attributes P without changing the partitioning of U into equivalence classes with respect to P, then it is a dispensable or superfluous attribute in P; otherwise, it is an indispensable attribute.

Definition 7. Attribute reduction aims to remove redundant attributes so that the reduced set provides the same qualities for classification as does the original. A reduction is defined as a subset Re of the conditional attribute set C, such as $\gamma_{Re}(D)=\gamma_C(D)$. A decision table can have many attribute reductions, and the set of all reductions is defined as

$$RED=\{Re\subseteq C \mid \gamma_{Re}(D)=\gamma_C(D),\forall B\subset Re,\gamma_B(D)=\gamma_C(D)\}. \quad (5)$$

In general, any element from the set of all reductions can be thought of as a sufficient subset of attributes, such as $\gamma_{Re}(D) = \gamma_C(D)$.

Definition 8. For attribute reduction based on a rough set, a reduction with minimal cardinality is identified. An attempt is made to locate a single element of the minimal reduction set. If we use to refer to the set of all reductions that have the same dependency value as $\gamma_C(D)$, then a minimal reduction is defined as follows:

$$Re^* \in \{Re \in \quad | \forall S \in \quad , | Re | \leq | S | \}. \tag{6}$$

III. Shared Coevolutionary Nearest-Neighbor Hierarchy (SCNNH) with Self-evolving Compensation

In this section, we present an SCNNH with self-evolving compensation, which ensures that similarity between attribute subsets will be calculated according to the shared neighbor information of sample attributes. The sample attributes include both condition attributes and decision attributes of selected samples in the shared nearest neighbor [18]. An explanation of this SCNNH structure is provided in Fig. 1. First, we define the similarity between shared coevolutionary nearest neighbors. Then, we introduce an indirect distance measurement method that considers the effects of attribute neighbors and draws on the concept of shared neighbors to generate interaction neighborhood vectors with self-evolving compensation that characterize the distance between coevolutionary nearest neighbors. The basic idea of shared nearest neighbors is that two sample attributes are considered to be more similar when they have more common neighbors. The coevolutionary algorithm decomposes a problem into several subcomponents and then evolves these subcomponents cooperatively for a predefined number of cycles to achieve a common goal [44][45][46]. Hence, the coevolutionary procedure of SCNNH converges rapidly and greatly improves the classification accuracy.

To clarify the explanation of SCNNH, the corresponding layer that contains the neighbor radius is denoted as $l^i (i = 1, 2, ..., n)$. As shown in the membership matrix $\mathbf{M}$ of Fig. 1, four types of neighbor radiuses can be observed: R_p^i, R_t^i, R_n^i, and R_m^i, where i is the row number and p, l, n, and m are the column numbers. A hierarchy is employed to handle the condition attribute jobs with the underlying interdependent structure. It consists of n related neighborhood radiuses using the seed set $W = \{d_1, d_2, ..., d_i, ..., d_n\}$.

A nearest-neighbor set K consists of m shared nearest neighbors from $K = \{k_1, k_2, ..., k_i, ..., k_m\}$. To determine the useful features from the SCNNH, the neighborhood radius is initially calculated using the i^{th} layer and then propagated to other higher layers via the membership matrix $\mathbf{M}$ [9] with self-evolving compensation [46][47]. Each layer corresponds to a neighborhood radius with different solutions. The overall flowchart of the SCNNH is shown in Algorithm 1.

Algorithm 1 Shared coevolutionary nearest-neighbor hierarchy (SCNNH)

1. Represent each hierarchy d_i by a shared nearest-neighbor vector as follows:

$$d_i = \{v_{i1}, v_{i2}, ..., v_{ij}, ..., v_{im}\}, \tag{7a}$$

$$v_{ij} = (1 + \log f_t(R_i^j)) * \log(1 + n / f_d(R_i^j)), \tag{7b}$$

where $f_t(R_i^j)$ is the term frequency of the shared nearest-neighbor radius R_i^j in d_i, and $f_d(R_i^j)$ is the hierarchy frequency of the shared nearest-neighbor radius R_i^j in W.

2. Obtain the $N^i \times N^i$ matrix $\mathbf{C}^i$, where N^i is the number of neighborhood radiuses in d_i. $\mathbf{C}^i(i, j)$ corresponding to the shared weight between R_i^i and R_i^j in d_i, which is defined as

$$\mathbf{C}^i(i, j) = corr(\mathbf{f}_i, \mathbf{f}_j), \tag{8}$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ are the feature vectors R_i^i and R_i^j, respectively.

3. Input the condition attributes $A_i \in C$ into the i^{th} hierarchy, and generate four interaction neighborhood vectors, $\mathbf{R}_p^i$, $\mathbf{R}_t^i$, $\mathbf{R}_n^i$, and $\mathbf{R}_m^i$. Then, decompose them into four vector subsets:

$$\mathbf{R}_n^i = [R_n^1, R_n^2, ..., R_n^i]^{\mathrm{T}}, \quad \mathbf{R}_m^i = [R_m^1, R_m^2, ..., R_m^i]^{\mathrm{T}},$$
$$\mathbf{R}_p^i = [R_p^1, R_p^2, ..., R_p^i]^{\mathrm{T}}, \quad \mathbf{R}_t^i = [R_t^1, R_t^2, ..., R_t^i]^{\mathrm{T}}. \tag{9}$$

4. The nearest neighborhood based on a similarity measurement can be used to calculate the similarity between points according to the shared neighbor information. For any point i, its nearest neighbor ρ_i is expressed with the kernel distance method:

$$\rho_i = \sum_{i \neq j} \exp\left[-\left(\frac{\sigma_{ij}}{\sigma_c} \right)^2 \right], \tag{10}$$

where σ_{ij} is the Euclidean distance between the data points i and j. The cutoff distance, $\sigma_c > 0$, is the neighborhood radius of a point. Thus, the nearest neighbor ρ_i is positively correlated to the number of points whose distance from i is less than σ_c. The set of nearest neighbors of point i is $\Gamma(i)$. Similarly, the set of nearest neighbors for j is $\Gamma(j)$.

5. The shared nearest neighborhood of point i and point j is defined as the intersection set of their common neighbor sets, expressed as

$$SNN(i, j) = \Gamma(i) \cap \Gamma(j). \tag{11a}$$

Accordingly, the shared neighbor sets for R_m^i and R_n^i are computed by

$$SNN(m, n) = \Gamma(R_m^i) \quad \Gamma(R_n^i), \tag{11b}$$

where $\Gamma(R_m^i)$ and $\Gamma(R_n^i)$ are the sets of K-nearest neighbors of R_m^i and R_n^i, respectively.

6. Determine the similarity of shared nearest-neighborhood vectors by a rigorous math formulation as follows:

$$Sim(m,n)=\begin{cases}\dfrac{|SNN(m,n)|^2}{\sum_{k\in SNN(m,n)}(d_{mk}+d_{nk})}, & if\ m,n\in SNN(m,n),\\ 0 & otherwise.\end{cases} \quad (11)$$

This approach can generate interaction neighborhood vectors with self-evolving compensation that characterizes the distance between coevolutionary nearest neighbors.

7. The nearest-neighbor hierarchy in W can be represented by the shared nearest-neighbor vector in K, which can be expressed as follows:

$$\zeta(e)=\begin{pmatrix} Sim_{11} & Sim_{1m} \\ & \\ Sim_{n1} & Sim_{nm}\end{pmatrix}, \quad (12)$$

where Sim_{ij} indicates the similarity of R_i^j with R_i^i.

8. Calculate the weight f_i^j of the self-evolving compensation between R_i^i and R_i^j using

$$f_i^j=df(k_ik_j)\big/df(R_i^j), \quad (13)$$

where $df(k_ik_j)$ denotes the number of nearest neighborhood ranking vectors in W that contain both R_i^i and R_i^j.

9. Construct the shared coevolutionary nearest-neighbor vectors $\mathbf{f}_m,\mathbf{f}_n,\mathbf{f}_p,\mathbf{f}_t$ by

$$\mathbf{f}_m=\{\xi_1 f_m^1,...,\xi_i f_m^i\},\ \mathbf{f}_n=\{\xi_1 f_n^1,...,\xi_i f_n^i\},$$
$$\mathbf{f}_p=\{\xi_1 f_p^1,...,\xi_i f_p^i\},\ \mathbf{f}_t=\{\xi_1 f_t^1,...,\xi_i f_t^i\}, \quad (14)$$

where ξ_i is the number of different neighborhood radii that belong to the same decision attribute.

Using this SCNNH, the coevolutionary nearest-neighbor radius will share the self-evolving compensation to better optimize the condition attributes of jobs on the hierarchical coevolutionary Spark model. Using SCNNH, we try to solve the attribute reduction of multiple overlapping and interdependent attribute subsets in interconnected big datasets. The SCNNH approach allows a better approximation of the contribution of various neighborhood radiuses so that the underlying interdependent structure of the condition attribute jobs can be revealed. Furthermore, the attribute reduction stability is measured by comparing the similarities of shared nearest neighborhood vectors; thus, the attribute reductions for big data are guaranteed to be equivalent to those observed using the SCNNH.

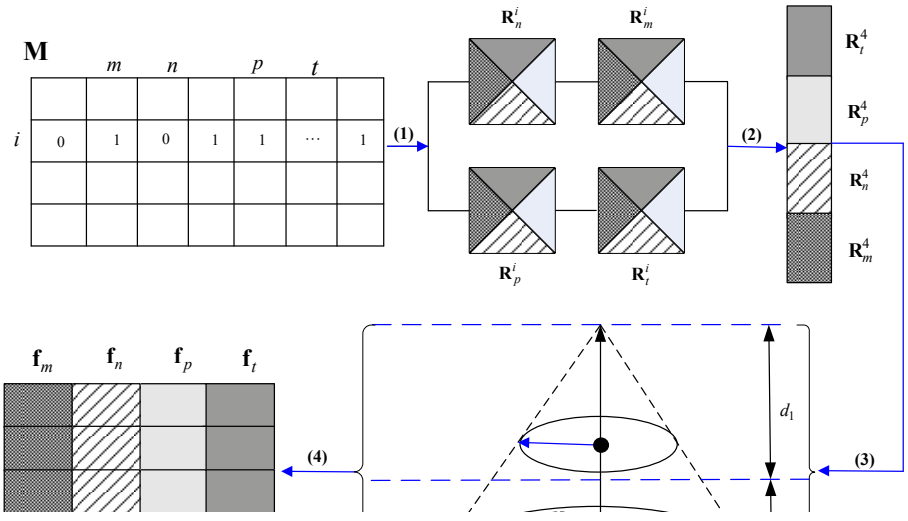
IV. Attribute Weight-Tensor-based Quantum Equilibrium Game

In this section, we construct an attribute weight tensor to generate the ranking attribute vectors in each SCNNH, and we balance the relative contributions of different neighborhoods. Then, we propose a quantum equilibrium game paradigm to exploit the special structure of the attribute weight tensor to ensure that uncertain and imprecise attributes do not degrade the reduction results of SCNNH. Hence, all the useful candidate attribute subsets are well preserved in the feature space.

A. Attribute weight tensor model construction

In this section, we construct an attribute weight tensor model (AWTM) to instruct allocations by observing the sets of shared coevolutionary nearest neighbors to which more neighbors belong. The weights of different attribute combinations can be calculated. Then, we present an attribute weight ranking approach to generate the ranking attribute vectors in each SCNNH. Finally, the attribute weight tensor with weight ranking vectors is designed.

We represent each shared coevolutionary nearest-neighbor vectors as a *k*-order tensor $I_{f_1},I_{f_2},...,I_{f_k}$, which corresponds to k feature spaces described by different attributes. By counting the nonzero elements that occur in all feature spaces for each specific coordinate, an association tensor $\mathrm{T}_a\in\Re^{I_{f1}\times I_{f2}\times,...,\times I_{fk}}$ can be obtained. The element $t_{i_1i_2...i_k}$ in an association tensor with a nonnegative integer value denotes the number of co-occurrences from the i_1^{th} to the i_k^{th} attribute. The overall flowchart of AWTM is shown in Algorithm 2.



Algorithm 2 Attribute weight tensor model (AWTM)

1. Determine the importance of the attributes of each feature space using a higher-order power method to calculate the ranking vector $w_1,w_2,...,w_k$, where $w_l(l=1,2,...,k)$ is calculated by

$$w_l=\alpha\mathrm{T}_{ir}^{(l)}\times_1 w_l\ \ \times_{l-1}w_l\times_{l+1}w_l\ \ \times_k w_l+(1-\alpha)\mu, \quad (15)$$

and w_l is the eigenvector that corresponds to the dominant eigenvalue l of the first *k*-order tensor $\mathrm{T}_{tr}^{(l)}$. Here, μ is a stochastic vector, and α is a probability $(\alpha < 1)$.

2. Generate the *k*-order tensor by first transforming T_a to $\mathrm{T}_{tr}^{(l)}$ using

$$t_{i_1\ i_l\ \ i_k}^{tr(l)} = \frac{t_{i_1\ i_l\ \ i_k}^{a}}{\sum_{i_l=1}^{z} t_{i_1\ i_l\ \ i_k}^{a}}, \tag{16}$$

where z is the maximum dimension of all orders of T_a.

3. Design the ranking vector $w_1, w_2, ..., w_k$ using the higher-order power method as follows:

(*i*) Set a probability $0 \le \alpha \le 1$, and select a threshold $\varepsilon \in [0.5, 1]$.

(*ii*) Select an initial vector w_0, where $\sum_{i=1}^{m} w_0 = 1$.

(*iii*) Set a stochastic vector $\sum_{i=1}^{m} \mu = 1$, and set $j = 0$.

(*iv*) Do { $j = j + 1$,

$$w_j = \alpha \mathrm{T}_{tr}^{(l)} \times_1 w_{j-1} \quad \times_{l-1} w_{j-1} \times_{l+1} w_{j-1} \times_k w_{j-1} + (1-\alpha)\mu. \tag{17}$$

} while $\left\| w_j - w_{j-1} \right\| > \varepsilon$.

(*v*) Represent the first-order tensor I_{f_1} of w_j as the ranking vector w_j', and set $w_j = w_j'$.

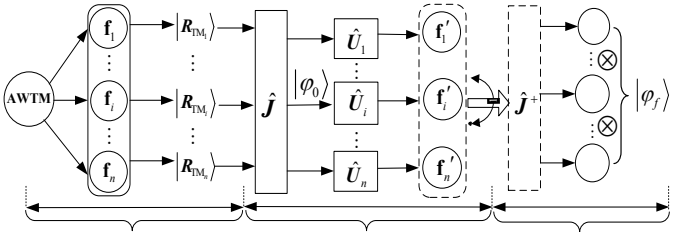
4. Return the attribute weight ranking vector as

$$w_1 \in \Re^{I_{f_1}}, w_2 \in \Re^{I_{f_2}}, \quad , w_k \in \Re^{I_{f_k}}. \tag{18}$$

5. Construct the product of the equation weight tensor $\mathrm{T}_w \in \Re^{I_{f_1} \times I_{f_2} \times ,...,\times I_{f_k}}$ by

$$\mathrm{T}_w = w_1 \times w_2 \times \quad w_k. \tag{19}$$

The weight tensor T_w effectively balances the relative contributions of the coevolutionary nearest-neighbor radii.



B. attribute weight tensor

The quantum game is a combination of game theory and quantum computation that differs from classical game theory, and has emerged as a hot topic in certain aspects of various research fields [47][48][49]. To bridge the gap between the shared nearest-neighbor hierarchy and the attribute weight tensor for the attribute reduction problem of big data, a novel quantum equilibrium game paradigm (QEGP) based on an attribute weight tensor is proposed to better achieve the Pareto front of nondominated solutions in attribute reduction for big data. The QEGP can prepare a superposition of quantum bit states with the distance among nearest-neighbor attribute weight tensors using a suitable quantum subroutine that encodes the distances in the quantum amplitudes.

We use a new quantum bit representation, which is defined as a pair of complex numbers (α, β). The process of implementing the QEGP is illustrated in Fig. 2 and consists of three aspects: 1) normalize the shared nearest neighbors, 2) update the weights of the basis using gradient descent, and 3) conduct entanglement among the neighboring weight tensors. The overall steps of the QEGP are shown in Algorithm 3.

Algorithm 3. Quantum equilibrium game paradigm (QEGP)

1. Represent the shared coevolutionary nearest-neighbor vector $\mathbf{f}_i$ as $\left| \boldsymbol{R}_{\mathrm{TM}_i} \right\rangle$, which is the superposition of states and the collapse of states of quantum bits [50][51] and conduct the normalization of all shared nearest-neighbor vectors as $\left| \boldsymbol{R}_{\mathrm{TM}_1}, \boldsymbol{R}_{\mathrm{TM}_2}, ..., \boldsymbol{R}_{\mathrm{TM}_m} \right\rangle$.

2. Entangle the nearest-neighbor status using the quantum gate $\hat{\boldsymbol{J}}$, whose initial status is formed as follows:

$$\left| \varphi_0 \right\rangle = \hat{\boldsymbol{J}} \left| \boldsymbol{R}_{\mathrm{TM}_1}, \boldsymbol{R}_{\mathrm{TM}_2}, ..., \boldsymbol{R}_{\mathrm{TM}_m} \right\rangle, \tag{20}$$

where $\hat{\boldsymbol{J}} = \exp\left[\gamma \cdot \hat{\sigma}_i^{\otimes m} \right]$ $(\gamma \in [0, \frac{\pi}{2}])$,

$\hat{\sigma}_i^{\otimes m} = \underbrace{\hat{\sigma}_1 \otimes \hat{\sigma}_2 \cdots \otimes \hat{\sigma}_m}_{m}$, and $\hat{\sigma}_i = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{(-1)^i \mathrm{T}}$.

3. Initialize the basis H=3, $B \in \Re^{3\times 2}$, and calculate the projection of X on *B* as follows:

$$\hat{x}^i = proj_b^x = \sum_{h=1}^{H} \left((b_h^i) T_x \right) b_h^i. \tag{21}$$

h is

$$c = \arg\min \| x - x \|. \tag{22}$$

5. Update the weights of the basis using the gradient descent method as a cost function

$$\boldsymbol{E} = \sum_i G_C^i \sum \left\| x - \hat{x}^i \right\|^2, \tag{23a}$$

$$b_h^i(t-1) = b_h^i(t) - \eta \frac{\partial E}{\partial b_h^i}(t). \tag{23b}$$

6. Assign the unitary operator $\hat{U}_i$ to f_i by

$$\hat{U}_i(\theta,\phi)=\begin{bmatrix} e^{i\phi}\cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & e^{-i\phi}\cos(\theta/2) \end{bmatrix} \quad (0\le\theta\le\pi, 0\le\phi\le\pi/2). \tag{24}$$

Then, a new Pareto optimal Nash equilibrium is obtained, labeled as $\hat{Q}\otimes\hat{Q}$.

7. Generalize Eisert et al.'s scheme [44] by performing the entanglement operation

$$|\varphi_i\rangle=\hat{J}|CC\rangle=\cos(\gamma/2)|CC\rangle+i\sin(\gamma/2)|DD\rangle, \tag{25}$$

where $\gamma\in[0,\pi/2]$ measures the entanglement of the initial state.

8. Update the f_i status to f_i' after completing the quantum equilibrium game among the shared coevolutionary nearest-neighbor vectors.

9. Decode the entanglement operator using the quantum gate $\hat{J}^+$, and combine the updated f_i' into the ensemble status:

$$|\varphi_f\rangle=\hat{J}^+\hat{U}_1\otimes\hat{U}_2\otimes\cdots\otimes\hat{U}_n\hat{J}|R_{TM_1}\cdots R_{TM_n}\rangle. \tag{26}$$

Using the QEGP based on the attribute weight tensor, the dilemma in the classical game is resolved, and uncertain and imprecise attributes do not degrade the reduction results. The entire trend achieves dynamic balance from disequilibrium to equilibrium. Hence, all the useful candidate attribute subsets can be well preserved in the feature space, which fosters high-quality reduction.

V. Hierarchical Coevolutionary Spark Model

MapReduce allo
amounts of data by
cluster of machines
scheduling are the responsibility of MapReduce. Despite its popularity, MapReduce is not appropriate for programs that continuously read data and must retain the data in memory.

In this section, HDFS is adopted to support the distributed runtime environment, and Spark is adopted to support distributed data storage. We construct a hierarchical coevolutionary Spark model combined with an improved MapReduce to provide a coevolutionary platform for attribute reduction of big data. The main steps in processing big data are as follows:

- Split the full big datasets into n Jobs (J_1,J_2,…J_n) using HDFS, which includes (m-1) condition attributes of jobs of incomplete big datasets (J_1,J_2, …J_{m1}) and (n-m+1) decision attributes of jobs of incomplete big datasets (J_m, J_{m+1}, …J_n).
- Design a task-coevolution structure [52] based on the improved MapReduce (TC-MapReduce) to enhance the Spark performance. Fully exploit TC-MapReduce on Spark to parallelize jobs (J_1,J_2, …J_{m-1}) within assignments across nodes, in which J_1 generates the condition attributes of the job sequence (J_2, …J_{m-1}) and J_m generates the decision attributes of the job sequence (J_{m+1}, …J_n). Then, establish the index of conditional attributes with missing data.
- Construct the condition-decision attributes of job pairs as $\{J_{1m}, J_{2(m+1)},\ldots,J_{(m-2)(n-1)}, J_{(m-2)n}\}$ to analyze the missing condition attribute and remove the record with missing decision attributes. Then, write the reduction sets (R_1,R_2, …R_m…R_n) to HDFS.

As depicted in Fig. 3, a hierarchy is an arrangement of different jobs according their condition-decision attributes. Each hierarchy tries to handle the attribute reduction jobs with multiple overlapping and interdependent datasets. The proposed SCNNH algorithm can be parallelized within the different hierarchies using Spark to allow for a better approximation of the contribution of various neighborhood radii. Thus, the underlying interdependent structure of the condition attribute jobs can be revealed. This model allows data to be preserved in memory and read rapidly. The master node retrieves the dataset from both the distributed cloud service provider and cloud servers in HDFS, enabling each client to read those data allocated to the local disk. Then, each client starts to process more Map tasks. The set of $<Key_m, Value_m>$ pairs is stored in the Combine Vector. After all the Map tasks have completed, the master

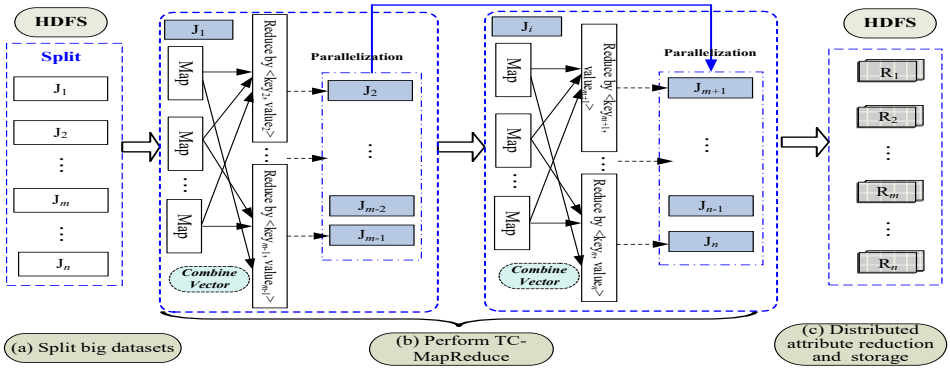


Fig. 3. Parallel architecture of hierarchical co-evolutionary Spark model.

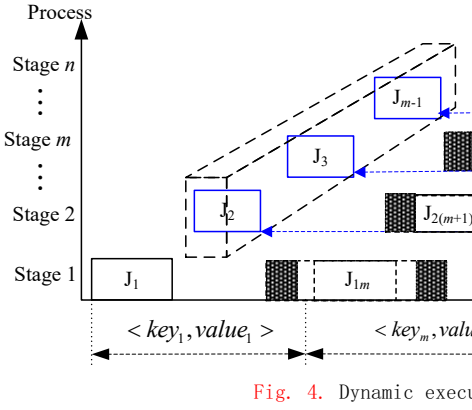


Fig. 4. Dynamic execu

node starts the Reduce tasks by distributing the pairs with matching keys to the same node. Each Reduce task combines those matching pairs to yield the final output in HDFS.

In the proposed hierarchical coevolutionary Spark model, TC-MapReduce is expressed via two functions, *Map* and *Reduce*, denoted as follows:

$$Map: < in\_key, in\_value > \rightarrow \{< key_m, value_m > | m = 1,2,...,n\};$$
$$Reduce: (key, [value_1, value_2, ..., value_n]) \rightarrow$$
$$< final\_key, final\_value >. \quad (27)$$

The dynamic execution process of jobs based on $< key_m, value_m >$ is depicted in Fig. 4.

This model is suitable for big data attribute reduction implementations, and it significantly improves the execution efficiency of the proposed attribute reduction algorithm.

VI. Proposed SNNQGAR Algorithm for Big Data Attribute Reduction

Based on the abovementioned hierarchical coevolutionary Spark model and the shared nearest-neighbor hierarchy-based quantum equilibrium game, we propose the SNNQGAR algorithm, which expands the attribute weight tensor to the shared nearest-neighbor relation to better partition complex attribute sets on Spark. This algorithm can explicitly identify the interdependent variables in such a way that the complexity and nonseparability of interdependent variables can be minimized among different attribute subsets. Thus, it achieves superior attribute reduction performance. The basic ideas are presented as follows.

First, we construct the minimum attribute reduction model as the optimization object. SCNNH is designed to consider the features of the nearest-neighborhood attribute subsets and calculate the similarity between the attribute subsets.

Second, we construct the attribute weight tensor to generate ranking vectors for the attributes. The neighborhood attribute subsets can be derived from the separability of the attribute weight ranking vectors. This process generates an optimal list of candidate attribute subsets and facilitates achieving the Pareto front of nondominated reduction solutions.

Third, we adopt a QEGP based on the attribute weight tensor to guarantee that uncertain and imprecise attributes do not degrade the reduction results.

Fourth, we construct a hierarchical coevolutionary Spark model combined with an improved MapReduce framework to parallelize SNNQGAR and provide improved attribute reduction solutions for big data analytics.

A graphical representation of the flowchart of the proposed SNNQGAR is presented in Fig. An in the supplementary file, and its main steps are provided in Algorithm 4.

Algorithm 4 Shared nearest-neighbor quantum game-based attribute reduction (SNNQGAR)

1. Initialize the search space of the attribute set and construct the attribute reduction model:
$$F(x) = min(S(x)) \ (x \in \{0,1\}^m, \gamma_{\xi(x)} = \gamma_C, \forall q \in \xi(x), \gamma_{\xi(x)\setminus\{q\}} = \gamma_{\xi(x)}). \quad (28)$$
2. Calculate the upper $\overline{\gamma}_{A_i}(D)$ and lower $\underline{\gamma}_{A_i}(D)$ related to each condition attribute $A_i \in C$. Then, select the most relevant attribute subset that has the highest upper relevance value, $\overline{\gamma}_{A_i}(D)$.
3. Obtain the equivalence class of the attribute set using the hierarchical coevolutionary Spark model:
{ (*i*) Convert S_i on Spark to $S \leftarrow spark.textfile(S_i)$.
(*ii*) $EC \leftarrow S.Map(key \leftarrow EC_i; value \leftarrow id_x).groupByKey().values$.
(*iii*) $S' \leftarrow EC.Map(result \leftarrow EC_i.getOneMaxDecisionArr())$.
(*iv*) Merge the same equivalence subclasses $\{EC_1, EC_2, ...EC_i...EC_n\}$ to obtain the total equivalence class set.
}
4. Let $Red \leftarrow \varnothing$. For each attribute $a \in C - Red$, do
{ (*i*) Calculate equivalence classes for the candidate attribute subset $Red \ \{a\}$ using Algorithm 1 and calculate the attribute significance $sig_{Red(a)}$ and $POS_{Red \ \{a\}}(D)$ using Algorithm 2.
(*ii*) $A_m \leftarrow select(\{sig_{Red}(A_1), ..., sig_{Red}(A_n)\})$.
(*iii*) Select the best candidate attribute set A_m and $Red = Red \ \{A_m\}$ using Algorithm 3.
(*iv*) $EC_{Red \ \{a\}} \leftarrow S'.Map(key \leftarrow EC_i', value \leftarrow id_x').groupByKey().values$.
(*v*) $U' \leftarrow U' - POS_{Red}(D)$.
(*vi*) Output the i^{th} attribute reduction subset AR_i.
} while (the termination criterion is not met).
5. Calculate the fitness $Fit(AR_i)$ of the i^{th} attribute reduction subset AR_i, and achieve the best reduction solution, AR_i^{best}.

6. Evaluate whether the accuracy of the reduction resolution satisfies the predefined accuracy.
If it does, output the optimal reduction set $\mathbf{FR}_{Opt} = \bigcup_{i=1}^{n} AR_i^{best}$; otherwise, go to Step 4.

VII. Experimental Evaluation and Discussion

In this section, we implement a series of experiments to illustrate the performance of the proposed SNNQGAR algorithm compared with those of four representative algorithms: S3 [34], Reducer [37], PACCA [54], and PAHAR-S [58]. Specifically, we describe the experimental setup in Section VII-A and assess the comparisons of the attribute reduction and classification using different classifiers and different big datasets in Section VII-B. The stability of the SNNQGAR algorithm is further evaluated in Section VII-C. Finally, a related discussion is provided in Section VII-D.

A. Experimental setup

We executed our experiments on the Hadoop platform, which is a software framework and programming model for big data analytics. We developed both Spark and MapReduce applications based on this platform. All the algorithms were implemented in Java. The public computing service platform provided by our University of Technology Sydney (UTS) consists of a virtual machine with 12 CPUs and 256 GB of memory. It is a High-performance Computing Linux Cluster with 8 nodes, which is strictly reserved for very large parallel and multithreaded computations. The machine configuration listed in Table I shows the slave node configurations of different users' computers. We use the personal computer only as a member of the power users group to perform Spark and MapReduce applications based on this platform. The computers were connected via Ethernet (100 Mbps). Detailed information about the experimental platform is presented in Table I.

TABLE I
SUMMARY OF EXPERIMENTAL PLATFORM

Platform	Version	Programming language	CPU (M)	Memory (GB)	Hard disk	Node
Spark	Hadoop-2.6.0 spark-1.5.1	Scale, scale-2.11.7	i5-2410	6	480 GB	5
MapReduce	Hadoop-2.6.0	Java, jdk 1.7.0_55	i5-2410	6	2TB	2

We selected five publicly available big datasets from the UCI repository [53] with different statistical characteristics and a large number of samples. In addition, we employed the well-known WEKA data generator from the WEKA data mining software (http://www.cs.waikato.ac.nz/ml/weka) to generate three synthetic large-scale datasets (Weka-1.8G, Weka-3.2G and Weka-6.4G). Descriptive information about the attributes of these datasets is shown in Table II. We first adopted the hierarchical coevolutionary Spark model to partition large number of attributes into different attribute subsets. Then, we employed the proposed SCNNH with self-evolving compensation to ensure calculation of the similarity between attribute subsets with multiple overlapping and interdependent attributes. Finally, the proposed SNNQGAR algorithm was parallelized to perform attribute reduction for the attribute subsets to achieve the best reduction solutions.

TABLE II
DESCRIPTION OF EXPERIMENTAL DATASETS

No.	Datasets	Samples	Attributes
1	PokerHand	1,025,010	10
2	KddCup 1999	4,856,151	41
3	Susy	5,000,000	18
4	RLCP	5,749,132	4
5	Higgs	11,000,000	28
6	Weka-1.8G	32,000,000	10
7	Weka-3.2G	40,000,000	15
8	Weka-6.4G	80,000,000	15

B. Attribute reduction comparison on big datasets

Fig. 5 presents the average comparison results for the attribute reduction accuracy and running time with different sample-to-noise ratios. We added some random numbers to each attribute value to form incremental sample-to-noise ratios. The random numbers satisfy the normal distribution with a mean of zero and a standard deviation of i%, where i=1.5, 2.5, 5.0, 7.5, and 10. For the attribute sets characterized by the sample-to-noise ratios, the experiment showed significant results for real-world attribute reduction problems with different sample-noise ratios. We computed the reduction accuracy (R_{Acc}) on these noisy attribute values as follows:

$$R_{Acc} = \frac{1}{n}\sum_{i=1}^{n}\frac{AR_i}{A_{total}}, \tag{29}$$

where AR_i is the number of correct attribute reductions yielded by each algorithm, A_{total} is the total number of attributes in each big dataset, and n is the number of independent runs.

In Fig. 5, the x-axis denotes different levels of incremental sample-noise ratios, the left y-axis indicates variations in the attribute reduction accuracy, and the right y-axis indicates the CPU running time. The experimental results show that the reduction accuracy decreases as the sample-to-noise ratio increases. As shown in Fig. 5 (a) and (b), although PACCA and Reducer achieve relatively similar performances regarding reduction accuracy as does SNNQGAR, they have much longer running times—at least 40% higher than that of SNNQGAR. Compared with PACCA and Reducer, SNNQGAR based on Spark significantly improves the reduction accuracy. As an example, in the Susy dataset, when the level of the sample-to-noise ratio increases from 2.5% to 5.0%, the variation in the reduction accuracy of SNNQGAR is 1.8%. When the level of the attribute-to-noise ratio increases from 5.0% to 7.5%, the variation in the reduction accuracy is 2.3%. Similar results can be observed in Fig. 5 (c) and (d). Thus, as the sample-to-noise ratio levels dynamically increase, SNNQGAR's advantage becomes considerably more obvious, and it can achieve satisfactory results.

The average running time increases with the incremental sample-to-noise ratio, but SNNQGAR is significantly more stable than are the compared algorithms because as the sample-noise ratio in big data increases, the Spark performance of increases substantially compared with that of MapReduce. Fig. 5 shows that the parallel attribute reduction time is reduced to half of the original time using Spark. Spark's in-memory-based calculations accelerate the parallel processing of SNNQGAR and greatly reduce its attribute reduction overhead. Thus, these experimental results show that SNNQGAR's effectiveness and efficiency of increase when using the proposed hierarchical coevolutionary Spark model, and its sensitivity to noise is reduced to some extent.

The experimental results in Fig. 5 show that the computational complexity of SNNQGAR is $O(nm\ log^m)$, where m and n are the number of samples and features, respectively. Regarding the compared algorithms, the overall computational complexity of PACCA is $O(m^2n)$, while that of Reducer is $O(mn^2)$. Hence, if we were to evaluate the algorithms using even larger datasets, the computational times of the compared algorithms would be unacceptable, but SNNQGAR requires less training time to obtain the optimal solutions and does not impose any serious burden on runtime complexity. SNNQGAR's success occurs because it deletes many more unnecessary attribute sets by using the hierarchical coevolutionary Spark model, and it needs to search a smaller region to obtain an optimal solution. Consequently, SNNQGAR's running time is considerably less than its standard counterpart in most cases.

C. Classification and stability comparison using big data

Classifier accuracy is used as a metric to assess the quality of the attribute reduction algorithms. In the following experiment, the features selected by the different algorithms are fed into three classifiers: linear SVM [55], C4.5 [56] and 1-Nearest Neighbor (1NN) [57]. We conducted 10 trials on each dataset, which were randomly split into training and testing subsets at a ratio of 6:4. To evaluate the classification accuracy more objectively and reasonably, we simulated a case in which anomalies were created by adding zero-mean Gaussian noise to normal observations. More specifically, after splitting the normal dataset into training and test sets, we further split the test set into two equally sized datasets. One of the newly split test sets was kept as is and represented normal observations, whereas in the remaining set, we randomly selected K% attributes from the entire data space and added zero-mean Gaussian noise to the projected subspace to represent anomalies. In this way, we obtained an incomplete dataset with different missing data rates.

Table III details the classification accuracy means and standard deviations for 10 trials. The symbols "†" and "††" denote that the performance of the corresponding algorithm is worse than or better than that of SNNQGAR, respectively. The best mean value is highlighted in boldface with a gray background. No single algorithm is consistently better than the others on all the tested datasets. For the SVM classifier on the KddCup 1999 dataset, with variances of K=20% attributes, SNNQGAR fails to obtain the optimal average classification accuracy but is close to the best performance achieved by PAHAR-S. This result is identified by the symbol "††" and primarily caused by the occasionally aggressive reduction behavior of the SCNNH in SNNQGAR, which can degenerate and result in a local-minimal value. Then, the solution distribution is not accurately reflected, and a small distance from the attained solution is observed. Furthermore, involving additional zero-mean Gaussian noise deteriorates the convergence ability of SNNQGAR.

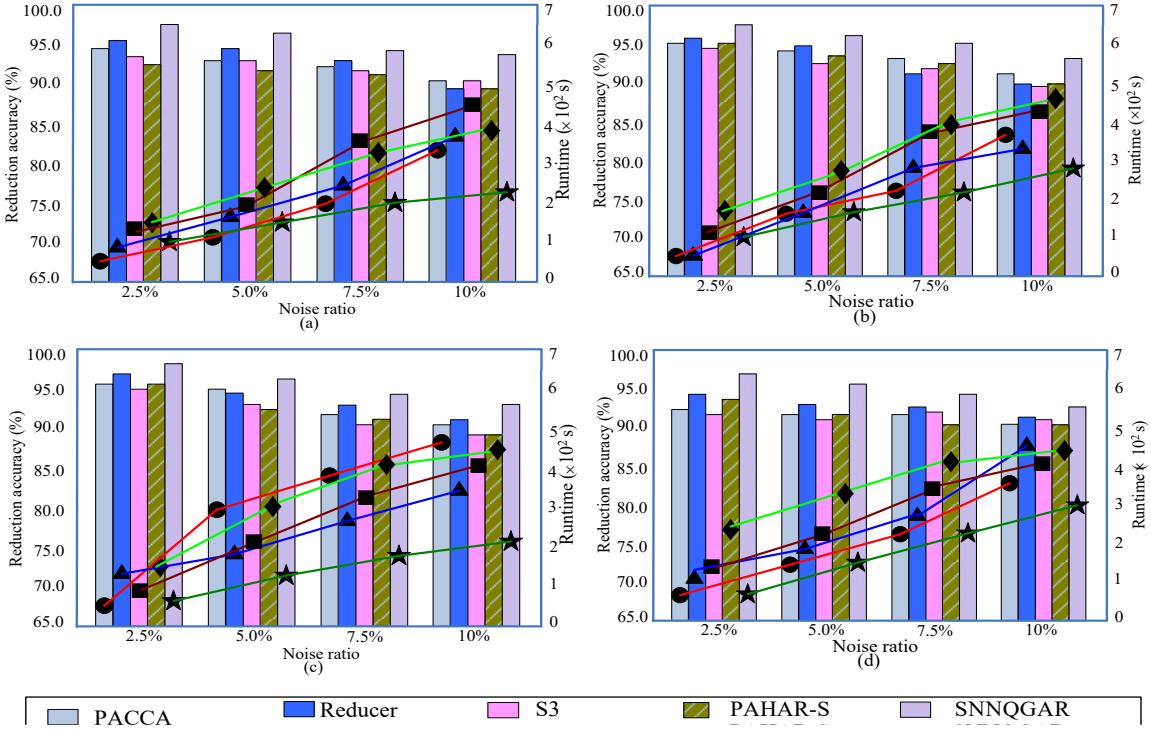


Fig. 5. Comparisons of the attribute reduction accuracy and runtime of five algorithms on four big datasets: (a) PokerHand, (b) KddCup 1999, (c) Susy, (d) RLCP.

viously outperforms the compared algorithms in terms of classification accuracy on most of the larger datasets, achieving an overall 2.5%~8.0% improvement. Specifically, SNNQGAR has a significant better average classification accuracy with the SVM classifier at variances of K=30% attributes. The classification accuracies presented in Table III show that SNNQGAR is more effective and efficient than the compared algorithms.

Reducer and S3 are unable to produce scalable solutions for larger datasets from the classification performance perspective. In addition, although PAHAR-S can determine some scalable solutions for big datasets, its performance also suffers under the different attribute variances.

Next, to verify the stability of the SNNQGAR algorithm, we evaluate the variation trends in classification accuracy as the percentage of perturbed attributes increases. Fig. 6 in the supplementary file shows the variation trends in the

classification accuracies of S3 [34], Reducer [37], PACCA [54],
PAHAR-S[58], and SNNQGAR as a function of the percentage of attributes synthetically perturbed by additive zero-mean Gaussian noise in two of the larger datasets. The *x*-axis indicates the percentage of perturbed attributes out of the total number of attributes, and the *y*-axis shows the average classification accuracy values. As shown in Fig. 6, SNNQGAR significantly outperforms the other algorithms when the percentage of perturbed attributes is below 40%. When the percentage of perturbed attributes exceeds 40%, SNNQGAR's performance remains more stable and achieves lower variance in its classification accuracy values. The experimental results indicate that SNNQGAR is suitable for addressing attribute reduction in large-scale datasets with different perturbed attributes, thereby overcoming the limitations of the representative parallel attribute reduction algorithms. Nevertheless, PAHAR-S is sensitive to perturbed attributes in the big datasets because it does not consider a certain percentage of these attributes.

TABLE III
AVERAGE CLASSIFICATION ACCURACY COMPARISON OF 10 TRIALS FOR CLASSIFIERS BY DIFFERENT ALGORITHMS WITH VARIANCES OF K=20% AND 30% ATTRIBUTES (TEST±STD /%)

Different classifiers	Different algorithms	Variances of attributes (K=20%)				Variances of attributes (K=30%)			
		PokerHand	KddCup 1999	Susy	RLCP	Higgs	Weka-1.8G	Weka-3.2G	Weka-6.4G
SVM	Reducer	91.23±0.23†	91.07±0.34†	90.67±0.86†	93.25±0.56†	92.12±0.24†	91.16±1.25†	90.07±0.29†	91.87±0.53†
	PAHAR-S	91.08±0.17†	93.16±0.86††	91.28±1.08†	94.17±0.26†	93.12±0.46†	92.02±0.11†	91.32±0.34†	91.90±0.38†
	PACCA	89.23±0.35†	89.29±0.68†	92.17±0.36†	91.45±0.36†	92.45±0.22†	92.21±0.32†	89.78±0.34†	92.54±0.65†
	S3	88.21±0.17†	88.98±0.25†	90.12±0.23†	92.05±0.67†	91.09±0.32†	91.99±0.76†	86.28±0.54†	87.49±0.21†
	SNNQGAR	**93.28±0.65**	**92.31±0.65**	**94.89±0.15**	**94.21±0.25**	**93.51±0.39**	**93.22±0.65**	**92.19±0.23**	**93.49±0.65**
C4.5	Reducer	90.21±0.78†	87.34±1.23†	89.23±0.65†	91.35±0.59†	90.37±0.42†	90.87±0.31†	86.15±0.35†	90.89±0.78†
	PAHAR-S	88.23±0.74†	91.78±1.06††	87.37±0.45†	90.24±0.57†	92.19±0.39†	92.19±0.43††	88.10±0.87†	89.69±0.53†
	PACCA	87.12±1.09†	87.33±1.28†	89.63±0.67†	89.66±0.79†	90.22±0.39†	91.67±0.59†	87.11±0.69†	91.39±0.67†
	S3	87.18±0.63†	88.57±1.08†	87.15±1.08†	90.24±0.78†	89.31±0.35†	89.28±0.17†	85.21±0.89†	87.07±0.78†
	SNNQGAR	**92.06±0.54**	**91.18±0.56**	**92.80±0.87**	**92.67±0.28**	**93.39±0.42**	**92.09±0.17**	**91.03±0.20**	**92.09±0.31**
K-NN	Reducer	89.18±0.27†	86.18±1.26†	87.34±1.23†	90.56±0.71†	90.11±0.54†	88.18±0.89†	87.43±0.52†	88.56±0.59†
	PAHAR-S	90.23±0.65†	89.21±0.72†	88.19±0.32†	87.90±1.23†	89.45±0.67†	89.21±0.69†	88.09±0.68†	87.94±0.78†
	PACCA	88.24±1.52†	91.35±1.09†	90.29±0.56†					
	S3	88.79±1.32†	86.98±0.73†	89.76±0.34†					
	SNNQGAR	**93.18±0.59**	**91.78±0.68**	**92.89±0.21**					

Our
dete
grap
perf
attri
those of the other algorithms because more attribute weight tensors in the coevolutionary nearest neighbors are available for classification as the percentage of perturbed attributes increases. Hence, the proposed SNNQGAR algorithm provides a better tradeoff between accuracy and robustness.

D. Discussion

In summary, our experimental study has indicated that the SNNQGAR algorithm outperforms four representative algorithms on most big datasets. We conclude that SNNQGAR is suitable for handling big data of various shapes and sizes and that it achieves satisfactory performance on multiscale datasets with cross winding and that have significant differences in density or high dimensionality. Compared with the representative attribute reduction algorithms S3 [34], Reducer [37], PACCA [54], and PAHAR-S [58], the proposed SNNQGAR algorithm achieved better attribute reduction performance by a large margin on most of the datasets. Moreover, the classification systems that employed SNNQGAR as the attribute reduction algorithm usually achieved the highest classification accuracy values. In the few cases in which the performance of SNNQGAR was not optimal, it still outperformed almost all the compared algorithms. Despite the appealing performance of the representative algorithms in accuracy-oriented classification systems, SNNQGAR was not affected by the increasing sample sizes and variations in noise levels in most of the cases throughout our experiments.

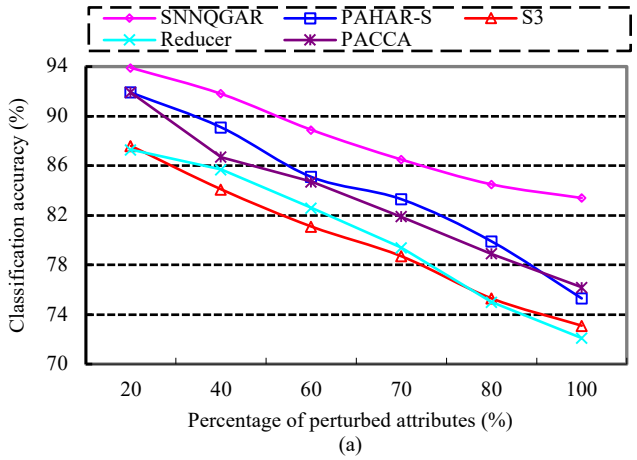


Fig. 6. Stability comparison results for the classification accuracy of three algorithms zero-mean Gaussian noise: (a) Weka-3.2G, (b) Weka-6.4G.

Although PACCA and PAHAR-S are considered efficient and flexible and can guarantee fast convergence results, they select only a few redundant attributes for certain iteration steps, which increases their computational complexity. Hence, this approach might limit their applications in classification problems involving heterogeneous big data. In addition, PACCA and PAHAR-S are less robust to outliers or noisy attributes than our proposed SNNQGAR algorithm. This finding again confirms that the shared nearest-neighbor quantum game enhances the efficiency of SNNQGAR. As a dataset becomes larger, SNNQGAR's efficiency gain increases.

The main contribution of this paper is the novel shared nearest-neighbor quantum game-based attribute reduction approach using hierarchical coevolutionary Spark. The metrics of the attribute reduction and the classification results of SNNQGAR are remarkably superior to those of the compared algorithms. The significant advantages are summarized as follows.

(1) Reasonability: In SNNQGAR, the SCNNH adopts an indirect distance measurement method that accounts for the effects of attribute neighbors, and it draws on the concept of shared neighbors to generate interaction neighborhood vectors with self-evolving compensation that characterize the distance between coevolutionary nearest neighbors. This approach greatly accelerates the big data attribute reduction process by removing the relatively dispensable and redundant objects and retaining the decisive attributes. Thus, this approach has the ability to remove redundancies in the collected reductions and allow for rapid updates of the final reduction sets. Consequently, our proposed SNNQGAR algorithm is more advantageous than are the existing attribute reduction algorithms.

(2) Efficiency: We construct a novel attribute weight tensor to generate the ranking vectors for the nearest-neighbor attributes that balances the relative contributions of different neighbor attribute subsets. On large-scale datasets, we utilize distributed hierarchical coevolutionary Spark to accelerate the loadable big datasets and parallelize SNNQGAR to improve its processing efficiency. This approach can relieve the huge data volume anxiety when processing big data. SNNQGAR shows a significant improvement in running time, especially when the attribute variances are higher. However, the compared algorithms show big-*O* time complexity in most cases. The time complexity of SNNQGAR is better than those of the compared algorithms in most cases. In our experiment, the complexity order of all the algorithms is as follows: SNNQGAR < PACCA < PAHAR-S < S3 < Reducer. Therefore, SNNQGAR achieves the highest efficiency.

(3) Robustness: SNNQGAR is robust to large sample data problems with higher perturbations. In Fig. 6, SNNQGAR shows almost the same accuracy under 70%–100% perturbations for the 3.2G and 6.4G datasets. This result occurs because the proposed quantum equilibrium game strategy based on the attribute weight tensor exploits inherent attribute structures and reduces the impact of the additional zero-mean Gaussian noise. Accordingly, the strongly related noisy features might be avoided in some data subsets after sampling, guaranteeing that higher perturbations will not obviously degrade the attribute reduction performance when dealing with big datasets that have large numbers of samples. SNNQGAR's benefits increase as the data attribute-noise ratio increases, which further indicates that SNNQGAR is a feasible and efficient big dataset attribute reduction approach.

A comparison of S3, Reducer, PACCA, PAHAR-S, and the SNNQGAR algorithm showed that the proposed algorithm greatly reduces the execution time via the quantum equilibrium game based on the attribute weight tensor, and it is significantly less sensitive to noise. Thus, SNNQGAR achieves lower variance errors, which means that it is more stable than are traditional attribute reduction algorithms. Furthermore, the classifications are highly correlated with human evaluations.

The abovementioned significant advantages are applicable to the critical challenges discussed in Section I, including the scalability, efficiency, and robustness of the attribute reduction of big data. In summary, the superiority of SNNQGAR has been clearly demonstrated. The results indicate that SNNQGAR is a promising attribute reduction algorithm for real-world big data applications.

VIII. Consistent Segmentation Application in Neonatal Cerebral Cortical Surfaces

In recent decades, the rapid development of noninvasive brain interference technologies has opened new horizons in the study of brain anatomy and function. Enormous progress has been made in exploring brain anatomy using magnetic resonance imaging (MRI)[59][60][61]. Consistent and accurate automatic segmentation of newborn brain anatomical regions is of great importance when studying longitudinal subtle changes of the cerebral cortex at neonatal ages; however, a neonatal brain MRI has a much lower contrast-to-noise ratio (CNR) and a lower signal-to-noise ratio than does an MRI of an adult brain. Moreover, the brain structure varies enormously in terms of shape and appearance during the neonatal period. In this segmentation experiment, we automatically generate neonatal brain MRIs using the Brain Extraction Tool (BET) from the FMRIB Software Library [62]. These brain data consist of the T1 channel of MRIs of the complex neonatal brain. Each dataset contains MRI data for 96 (512 × 512) 12-bit images obtained in the axial plane using a 1.5-Tesla Siemens Sonata with a standard head coil. These images are first processed to remove the air and skull pixels. The remaining pixels in each of the 96 images are unrolled into 1-D (pixel-value) feature vectors. Then, the feature vectors from each of the 96 slices are combined, creating a dataset containing approximately 4 million 1-D objects. Fig. 7 (a) and Fig. 8 (a) show examples of

two coronal slices from the MRI data before skull and air removal.

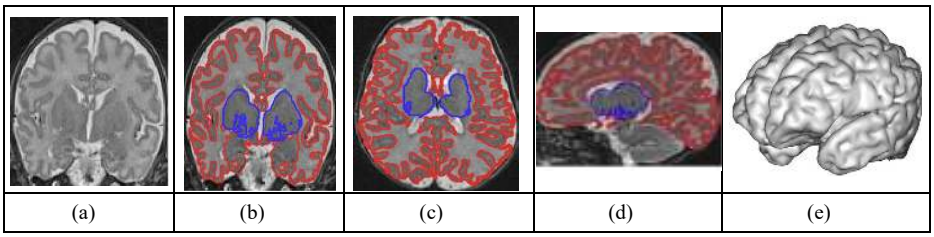


Fig. 7. Segmentation of subcortical gray matter: (a) Original subject of coronal MRI-1slice; (b) Segmented contours of subcortical gray matters; (c) Axial slice; (d) Sagittal slice; (e) 3D surface of gray matter.

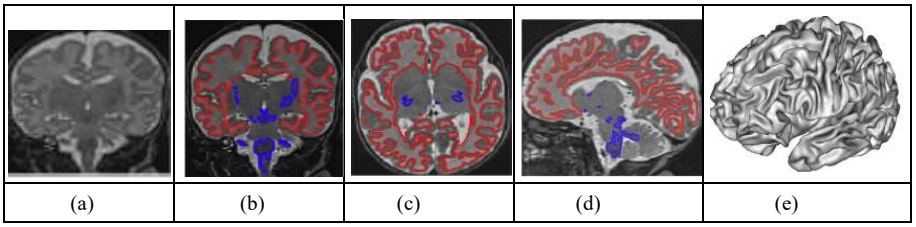


Fig. 8. Segmentation of unmyelinated white matter: (a) Original subject of coronal MRI-2 slice; (b) Segmented contours of unmyelinated and myelinated white matter; (c) Axial slice; (d) Sagittal slice; (e) 3D surface of white matter.

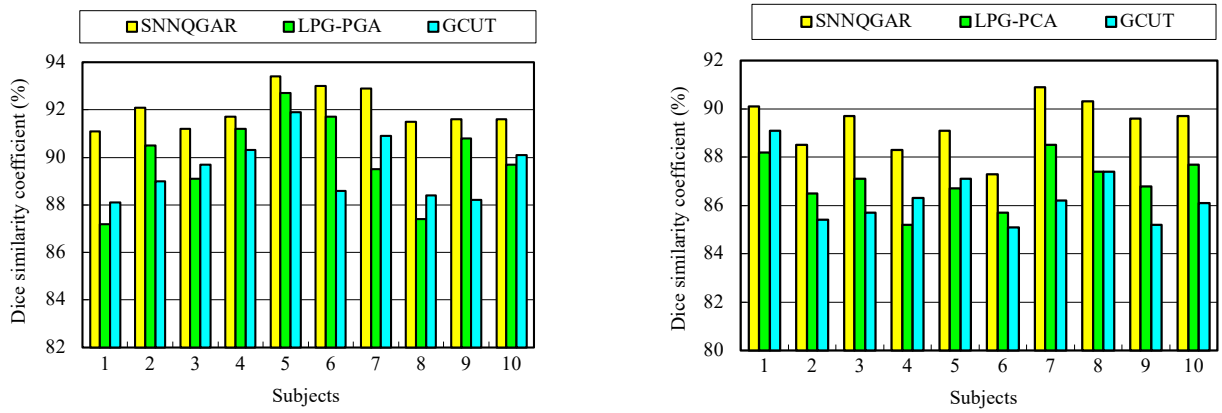


Fig. 9. Dice coefficients comparison for two quantitative evaluations. (a) Subcortical gray matter. (b) Myelinated white matter.

because it presents intensity levels similar to those of cortical gray matter. In this experiment, we employed the proposed SNNQGAR algorithm to segment the large dark regions associated with subcortical gray matter from the neonatal cerebral cortex surfaces while preserving the fine dark regions of the cortical gray matter. For the experiment, we select three views: a coronal slice, an axial slice, and a sagittal slice. In Fig. 7, all the subcortical gray matter from the three views is denoted by blue while the cortical gray matter is denoted by red. The results show that the subcortical gray matter can be accurately distinguished from the complex connected homogeneous regions.

Another challenge involves accurately discriminating unmyelinated white matter from cortical gray matter because it is posed based on the partial volume effects in the neighborhood of external cerebral spinal fluid. Moreover, many blurred interfaces occur between the unmyelinated white matter and cortical gray matter. In this experiment, we focus on segmenting the myelinated white matter regions from the newborn brain. In Fig. 8, the unmyelinated white matter is denoted by red, while the myelinated white matter is denoted by blue. The results show that SNNQGAR accurately captures the cortical gray matter region and correctly distinguishes unmyelinated white matter from other tissue. The resulting tissue surfaces show that the unmyelinated white matter is correctly identified and presents clearly distinguishable gyri and sulci.

cerebral subcortical gray matter and myelinated white matter, we quantitatively analyzed the details of the Dice similarity coefficient values averaged for ten subjects, as shown in Fig. 9. We compared SNNQGAR with two popular methods: brain surface extraction (LPG-PCA) [63] and skull stripping using graph cuts (GCUT) [64]. The Dice similarity coefficient is calculated as follows:

$$\mathrm{Dice}(A,B) = \frac{2|A \quad B|}{|A|+|B|} \times 100\%, \tag{30}$$

where A and B are the voxel sets of two different tissue segmentations.

Fig. 9 shows that compared with LPG-PCA and GCUT, SNNQGAR achieves the highest Dice coefficient, which verifies that SNNQGAR can segment distinct regions of neonatal cerebral cortex surfaces with good overall accuracy and consistency.

To further validate the algorithms, the SNNQGAR and the two compared algorithms are used to process the larger 1D-MRI dataset, which includes 50 million objects from the FMRIB software library. We tested the three algorithms at sample sizes of 0.0001%, 0.001%, 0.01% and 0.1%, which required approximately 10, 50, 100 and 200 MB of memory, respectively. We employed the runtime (RT, /s) and adjusted

rand index (ARI) [65] as two types of performance criteria. Table IV shows the comparison of three algorithms based on these performance criteria on the larger 1D-MRI dataset. As expected, SNNQGAR achieves an efficient and effective solution every time, regardless of the sample sizes. In contrast, GCUT has longest running time due to the large data accumulation. In particular, when the sample size is 0.01%, GCUT requires 100.09 s, while SNNQGAR requires only 10.23 s, achieving a $10\times$ speed improvement. Furthermore, LPG-PCA and GCUT are less consistent, as evidenced by their inferior ARI results. As the sample size increases, SNNQGAR becomes the preferred algorithm, achieving an ARI close to 1.

TABLE IV
Performance criterion comparison on the larger 1D-MRI dataset

Volume		SNNQGAR		LPG-PCA		GCUT	
		RT	ARI	RT	ARI	RT	ARI
Sample size	0.0001%	3.18	0.96	4.29	0.88	12.89	0.86
	0.001%	7.89	0.98	10.98	0.89	41.90	0.85
	0.01%	10.23	0.98	50.89	0.92	100.09	0.91
	0.1%	25.98	1.00	70.90	0.94	160.67	0.92

SNNQGAR's improvement over these two popular methods is significant, which indicates the superiority of SNNQGAR in characterizing neonatal brain structural anomalies for overlapping and interdependent fuzzy cerebral tissues. SNNQGAR consistently provides satisfactory segmentations as well as quantitative comparisons

In summary, based on the observations above, SNNQGAR running on the hierarchical coevolutionary Spark platform exhibits great potential and exciting advantages for real-world big data applications.

IX. CONCLUSIONS

The uncertain and intricate nature of big data, which includes high-dimensional attributes with complex structures and ever-increasing volumes, greatly affects the attribute reduction performance. Consequently, attribute reduction processes are time-consuming and inefficient at performing complex attribute reduction and at extracting useful knowledge from these dynamically changing massive datasets. However, the proposed shared nearest-neighbor quantum game-based attribute reduction (SNNQGAR) running on the hierarchical coevolutionary Spark platform can solve these problems for the following reasons.

We present an SCNNH with self-evolving compensation to calculate the similarity among multiple overlapping and interdependent attribute subsets in a large search space according to the shared neighbor information of the sample attributes. The proposed coevolutionary procedure converges quickly, which greatly improves the algorithm's efficiency and accuracy. Second, we construct an attribute weight tensor to generate ranking vectors for attributes that balance the relative contributions of different neighbor attribute subsets. Third, we employ a quantum equilibrium game paradigm based on an attribute weight tensor to ensure that the uncertain and imprecise attributes do not degrade the final attribute reduction results. Hence, all the useful candidate attribute subsets of massive datasets are well preserved in the attribute space. Finally, we adopt a new hierarchical coevolutionary Spark model combined with an improved MapReduce model that, together, allow better parallelization of the proposed SNNQGAR algorithm and provide efficient attribute reduction solutions for dynamically changing massive datasets.

The experimental results clearly demonstrate SNNQGAR's superior performance. SNNQGAR outperforms most of the tested state-of-the-art attribute reduction algorithms. We also evaluated the accuracy and efficiency of SNNQGAR at the task of segmenting subcortical gray matter and unmyelinated white matter from complex neonatal cerebral cortex surfaces. The results clearly show that SNNQGAR is helpful in cortical folding studies of the neonatal cerebrum.

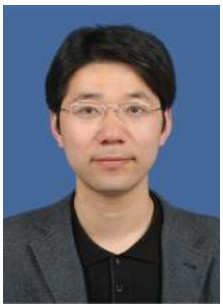
ACKNOWLEDGMENTS

The authors would like to express their sincere appreciation to the anonymous reviewers for their insightful comments, which greatly improved the quality of this paper.

REFERENCES

[1] S. Lohr, "The age of big data," *The New York Times*, Feb. 2012. http://www.nytimes.com/2012/02/12/S.
[2] Chris, U. Matzat, and U.-D. Reips, "Big data: Big gaps of knowledge in the field of internet science," *Int. J. Internet Sci.,* vol. 7, no. 1, pp. 1-5, Jan. 2012.
[3] P. Basanta-Val, N. C. Audsley, A. Wellings, I. Gray, and N. Fernandez-Garcia, "Architecting time-critic*al big-data systems," IEEE Trans. Big Data*, vol. 2, no. 4, pp. 310-324, Dec. 2016.
[4] K. Kambatla, G. Kollias, V. Kumar, and A. Grama, "Trends in big data analytics," *J. Parallel Distrib. Comput., vol. 74, n*o. 7, pp. 2561-2573, Jul. 2014.
[5] H. V. Jagadish, J.Gehrke, and A. Labrinidis, et al., "Big data and its technical challenges," *Commun. ACM*, vol. 57, no. 7, pp. 86-94, Jul. 2014.
[6] B. Wenjie, M. Cai, M. Liu, and G. Li, "A big data clustering algorithm for mitigating the risk of customer churn," *IEEE Trans. Ind. Informat*., vol. 12, no. 3, pp. 1270-1281, Jun. 2016.
[7] C. P. Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Inf. Sci.*, vol. 275, pp. 314-347, Aug. 2014.
[8] Z. H. Lv, H. B. Song, and P. Basanta-Val, et al., "Next-generation big data analytics: State of the art, challenges, and future research topics," *IEEE Trans. Ind. Informat*., vol.13, no. 4, pp. 1891-1899, Feb. 2017
[9] Z. Pawlak, "Rough set approach to knowledge-based decision support," *Eur. J. Operat. Res.*, vol. 99, no. 1, pp. 48-57, May 1997.
[10] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Inf. Sci.*, vol. 177, no. 1, pp. 3-27, Jan. 2007.
[11]Z. Pawlak and A. Skowron, "Rough sets: Some extensions," *Inf. Sci.*, vol. 177, no. 1, pp. 28-40, Jan. 2007.
[12]Z. Pawlak and A. Skowron, "Rough sets and boolean reasoning," *Inf. Sci.*, vol. 177, no. 1, pp. 41-73, Jan. 2007.
[13] Y. Qian, S. Li, J. Liang, Z. Shi, and F. Wang, "Pessimistic rough setbased decisions: A multigranulation fusion strategy," *Inf. Sci.*, vol. 264, pp. 196-210, Apr. 2014.
[14]J. Zhang, T. Li, and H. Chen, "Composite rough sets for dynamic data mining," *Inf. Sci.*, vol. 257, pp. 81-100, Feb. 2014.
[15] W. Ziarko, "Variable precision rough sets model," *J. Comput. Syst. Sci.*, vol. 43, no. 1, pp. 39-59, 1993.
[16] Y. Y. Yao, Z. Q.Wang, and C. Gan, et al, "Multi-source alert data understanding for security semantic discovery based on rough set theory," *Neurocomputing*, vol. 208, pp. 39-45, Oct. 2016.
[17]J. Wang, J. Peng, and O. Liu, "A classification approach for less popular webpages based on latent semantic analysis and rough set model," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 642-648, Jan. 2015.
[18]Y. Y. Yao, and Y. H. She, "Rough set models in multigranulation spaces," *Inf. Sci.* vol. 327, pp. 40-56, Jan. 2016.
[19]R. Susmaga, "Reducts and constructs in classic and dominance-based rough sets approach," *Inf. Sci.*, vol. 271, pp. 45-64, Jul. 2014.
[20]G. Wang, "Rough reduction in algebra view and information view," *International Journal of Intelligent Systems,* vol. 18, no. 6, pp. 679-688, Jun. 2003.
[21] Q. H. Hu, Z. X. Xie, and D. R. Yu, "Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation," *Pattern Recognit.*, vol. 40, no. 12, pp. 3509-3521, Dec. 2007.
[22]Y. Yao and Y. Zhao, "Discernibility matrix simplification for constructing attribute reducts," *Inf. Sci.*, vol. 179, no. 7, pp. 867-882, Mar. 2009.
[23] D. Chen, S. Zhao, L. Zhang, et al, "Sample pair selection for attribute reduction with rough set," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 2080-2093, Nov. 2012.
[24] W. W. Li, Z. Q. Huang, and X. Y. Jia, et al, "Neighborhood based decision-theoretic rough set models," *Int. J. Approx. Reason.*, vol. 69, pp. 1-17, Feb. 2016.
[25] D. S. Yeung, D. G. Chen, E. C. C. Tsang, J. W. T. Lee, and X. Z.Wang, "On the generalization of fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 13, no.

3, pp. 343–361, Jun. 2005.
[26] Y.-C. Chen, N. R. Pal, and I. F. Chung, "An integrated mechanism for feature selection and fuzzy rule extraction for classification," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 4, pp. 683–698, Aug. 2012.
[27] S. An, Q. H. Hu, D. R. Yu, and J. F. Liu, "Soft minimum-enclosing-ball based robust fuzzy rough sets," *Fundam. Inf.,* vol. 115, no. 2-3, pp. 1893–202, Apr. 2012.
[28] H. Y. Zhang, and S. Y. Yang, "Feature selection and approximate reasoning of large-scale set-valued decision tables based on α-dominance-based quantitative rough sets," *Inf. Sci.*, vol.378, pp. 328–347, Feb. 2017.
[29] Y. Yang, D. Chen, and Z. Dong, "Novel algorithms of attribute reduction with variable precision rough set model," *Neurocomputing*, vol. 139, pp. 336-344, Sep. 2014.
[30] J. Dean, and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
[31] M. A. Alshammari , E.-S.M. El-Alfy , "Mapreduce implementation for minimum reduct using parallel genetic algorithm," *in Proc. the 6th International Conference on Information and Communication Systems* (ICICS), 2015, pp. 13–18 .
[32] D. Keco, and A. Subasi, "Parallelization of genetic algorithms using Hadoop Map/Reduce," *Southeast Eur. J. Soft Comput*, vol. 1, no.2, pp. 56–59, Sep. 2012.
[33] J. Qian, P. Lv, and X. Yue, et al, "Hierarchical attribute reduction algorithms for big data using mapreduce," *Knowl. Based Syst.*, vol. 73, pp. 18–31, Jan. 2015.
[34] J. Zhang, J. Wong, and Y. Pan, et al. "A parallel matrix-based method for computing approximations in incomplete information systems," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 326-339, Feb. 2015
[35] J. Zhang , Y. Zhu , Y. Pan , and T. Li , "Efficient parallel Boolean matrix based algorithms for computing composite rough set approximations," *Inf. Sci.*, vol. 329, pp. 287–302, Feb. 2016.
[36] H. Chen , T. Li , Y. Cai , C. Luo, and H Fujita, "Parallel attribute reduction in dominance-based neighborhood rough set," *Inf. Sci.* vol, 373, pp. 351–368, Dec. 2016.
[37] E. S. M. El-Alfy and M. A. Alshammari, "Towards scalable rough set based attribute subset selection for intrusion detection using parallel genetic algorithm in mapreduce," *Simulation Modelling Practice and Theory*, vol. 64, pp. 18–29, May 2016.
[38] M. Hamstra, H. Karau, M. Zaharia, A. Konwinski, and P. Wendell, "Learning Spark: Lightning-fast big data analytics," O'Reilly Media, Incorporated, 2015.
[39] S. Ramírez-Gallego, S. García, and H. Mouriño-Taín, et al, "Distributed entropy minimization discretizer for big data analysis under apache Spark," *in Proc. 2015 IEEE Trustcom/BigDataSE/ISPA,* Aug. 20-22, pp. 33-40, 2015.
[40] J. Lin, "Mapreduce is good enough? " *Big Data*, vol. 1, no.1, pp. 28–37, Jan. 2013.
[41] Apache Spark: Lightning-fast cluster computing, "Apache spark," 2015. https://spark.apache.org/.
[42] S. Ramírez-Gallego, S. García, J. M. Benítez, and F. Herrera, "A distributed evolutionary multivariate discretizer for Big Data processing on Apache Spark" *Swarm Evol. Comput.*, vol. 38, pp.240-250, Feb. 2018.
[43] M. Hamstra, H. Karau, M. Zaharia, A. Konwinski, and P. Wendell, "Learning Spark: Lightning-fast big data analytics," O'Reilly Media, Incorporated, 2015
[44] T. Jansen, and R. Wiegand, "The cooperative coevolutionary (1+1) EA," *Evol. Comput.*, vol. 12, no. 4, pp. 405-434, Jan. 2004.
[45] K.C. Tan, Y. J. Yang, and C.K. Goh, "A distributed cooperative co-evolutionary algorithm for multi-objective optimization," *IEEE Trans. Evol. Comput*, vol. 10 , no. 5, pp. 527–549, Oct. 2006.
[46] X. Li, and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012.
[47] J. Eisert, M. Wilkens, and M. Lewensein, "Quantum game and quantum strategies,tr*Physical Review Letters*, vol. 83, no.15, pp. 3077-3080, Oct. 1999.
[48] N. Wiebe, A. Kapoor, and K.Svore, "Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning," arXiv:1401.2142v2, Jul. 2014.
[49] D. Meyer, "Quantum strategies," *Physical Review Letters*, vol. 82, pp. 1052-1055, Feb. 1999,
[50] M.Ying, "Quantum computation, quantum theory and AI," *Artificial Intelligence*, vol.174, no. 2, pp. 162–176, Feb. 2010.
[51] E. Ahmed, M. F. Elettreby, and A. S. Hegazi, "On quantum team games," *International Journal of Theoretical Physics*, vol. 45, no. 5, pp. 907-913. May 2006.
[52] M. A. Potter, and K. A. De Jong, "Cooperative co-evolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. 1, pp. 1-29, Mar. 2000.
[53] K. Bache, and M. Lichman, "UCI Machine Learning Repository," 2013, http://archive. ics.uci.edu/ml
[54] J. Qian, D. Q. Miao, Z. H. Zhang, and X. D. Yue, "Parallel attribute reduction algorithms using MapReduce," *Inf. Sci.* vol. 279, pp. 671-690, Sep. 2014.
[55] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.
[56] R. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[57] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no.1, pp. 37-66, Jan. 1991.
[58] M. C. Chen, J. L. Yuan, and L. Li, et al. "A fast heuristic attribute reduction algorithm using spark," *in Proc. 2017 IEEE 37th International Conference on Distributed Computing Systems*, 2017, 2393-2398.
[59] K. Oishi, A. V. Faria, and S. Yoshida, et al. "Quantitative evaluation of brain development using anatomical MRI and diffusion tensor imaging," *Int. J. Devl Neuroscience*, vol. 31, pp. 512–524, Jun. 2013.
[60] C. N. Devi, A. Chandrasekharan, V. K. Sundararaman, and Z. C. Alex, "Neonatal brain MRI segmentation: A review," *Computers in Biology and Medicine*, vol. 64, pp. 163–178, Sep. 2015.
[61] I. Išgum, M. J.N.L. Benders, and B. Avants, et al, "Evaluation of automatic neonatal brain segmentation algorithms: The NeoBrainS12 challenge," *Medical Image Analysis*, vol. 20, no.1, pp. 135–151, Feb. 2015.
[62] S. M. Simth, "Fast robust automated brain extraction," *Human Brain Map.*, vol. 17, no.3, pp. 143–155, Nov. 2002.
[63] L. Zhang, W. Dong, and D. Zhang, et al, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognit.*, vol. 43, no.4, pp.1531–1549, Apr. 2010.
[64] S. Sadananthan , W. Zheng , M. Chee, and V. Zagorodnov, "Skull stripping using graph cuts," NeuroImage, vol. 49, no. 1, pp. 225-239, Jan. 2010 .
[65] L. Hubert, and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, Dec.1985.



Weiping Ding (M'16) received a Ph.D. in Computation Application from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2013. He was a visiting researcher at University of Lethbridge, Alberta, Canada, in 2011. From 2014 to 2015, he was a postdoctoral researcher at the Brain Research Center, National Chiao Tung University (NCTU), Hsinchu, Taiwan. In 2016, he was a visiting scholar at the National University of Singapore (NUS), Singapore. Currently, he is a visiting scholar at the University of Technology Sydney (UTS), Australia. His current research interests include data mining, machine learning and granular computing. He has published over 50 papers in flagship journals and conference proceedings as the first author. To date, he holds 10 approved invention patents and over 18 issued patents. Dr. Ding was a recipient of the Computer Education Excellent Paper Award (First-Prize) from the National Computer Education Committee of China in 2009. He was an Excellent-Young Teacher (Qing Lan Project) of Jiangsu Province in 2014 and designated a High-Level Talent (Six Talent Peak) of Jiangsu Province in 2016. He was awarded the Best Paper of ICDMA'15, Hong Kong. Dr. Ding was awarded two Chinese Government Scholarships for Overseas Studies in 2011 and 2016. He currently serves as an associate editor of *IEEE Transaction on Fuzzy Systems*, *Information Sciences* and *Swarm and Evolutionary Computation.*(Email: dwp9988@hotmail.com)

Chin-Teng Lin (S'88–M'91–SM'99–F'05) received M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1989 and 1992, respectively. He is currently a distinguished professor in the Faculty of Engineering and Information Technology at the University of Technology Sydney, and a university chair professor of Electrical and Computer Engineering, NCTU, International Faculty of University of California at San-Diego (UCSD). He holds an honorary professorship at the University of Nottingham. Dr. Lin was elevated to an IEEE Fellow for his contributions to biologically inspired information systems in 2005 and was elevated to an International Fuzzy Systems Association (IFSA) Fellow in 2012. He was elected as the editor-in-chief of *IEEE Transactions on Fuzzy Systems* for 2011–2016. He also served on the Board of Governors at the IEEE Circuits and Systems (CAS) Society from 2005–2008, the IEEE Systems, Man, Cybernetics (SMC) Society from 2003–2005, and the IEEE Computational Intelligence Society (CIS) from 2008–2010. Dr. Lin was a distinguished lecturer for the IEEE CIS Society from 2015–2017. He served as the deputy editor-in-chief of *IEEE Transactions on Circuits and Systems-II* in 2006-2008. Dr. Lin is the coauthor of Neural Fuzzy Systems (Prentice-Hall) and the author of Neural Fuzzy Control Systems with Structure and Parameter Learning (World Scientific). He has published more than 200 journal papers (Total Citation: 19,166, H-index: 53, i10-index: 332) in the areas of fuzzy systems, neural networks and cognitive neuro-engineering, including approximately 105 IEEE journal papers. (Email: Chin-Teng.Lin @uts.edu.au)

Zehong Cao received a B.E. degree in Electronic and Information Engineering, from Northeastern

University in 2012, and an M.S. degree in Electronic Engineering from The Chinese University of Hong Kong in 2013. He holds a dual Ph.D. degree in Information Technology from UTS and Electrical and Control Engineering from the National Chiao Tung University (NCTU) (2017). He is currently a postdoc research fellow at the Center for Artificial Intelligence at the University of Technology, Sydney, Australia. He serves as an associate editor of IEEE Access and is an editorial board member of the SCI-journals Advances in Robotics & Automation, and the International Journal of Sensor Networks and Data Communications. His research interests include data science, human-machine interfaces, computational intelligence, pattern recognition, machine learning, and clinical applications. (Email: zhcaonctu@gmail.com)