IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 February 2023, accepted 4 March 2023, date of publication 16 March 2023, date of current version 12 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3257875

RESEARCH ARTICLE

Handling Big Microarray Data: A Novel Approach to Design Accurate Fuzzy-Based Medical Expert System

GANESHKUMAR PUGALENDHI[1], (Senior Member, IEEE), M. MAZHAR RATHORE[2], (Member, IEEE), DHIRENDRA SHUKLA[2], AND ANAND PAUL[3]

[1]Department of Information Technology, Anna University Regional Campus, Coimbatore 641046, India
[2]Dr. J. Herbert Smith Centre, University of New Brunswick, Fredericton, NB E3B 5A3, Canada
[3]School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding author: Anand Paul (paul.editor@gmail.com)

This work was supported by the National Research Foundation of Korea under Grant 2020R1A2C1012196.

ABSTRACT The genes data produced by microarray experiments is complex in terms of dimensions and samples. It consumes a lot of computation power and time when it is processed for a disease analysis while working with an expert system. At the same time, data can help doctors identify a patient's health condition if it is presented in a meaningful way and processed on time. Several methods have been proposed to reduce the dimensions of medical microarray data and optimize its search space with minimal accuracy loss. However, the discretization of continuous gene-values in the process of dimension reduction is failed to preserve the inherent meaning of genes. Also, ensuring high accuracy and interpretability in the reduction process may result in extra processing time, which is unfavorable for time-critical applications. To overcome these issues, in this paper, we propose a dimension reduction method in conjunction with a fuzzy expert system (FES) optimization approach, while keeping an accuracy-interpretability-speedy tradeoff in mind. To accomplish this, we use a fuzzy rough set on *f*-information to identify meaningful genes without changing their original values. We propose a conditionally guided particle swarm optimization for faster knowledge acquisition, where the velocity is adjusted based on a predefined update probability, resulting in a faster search. A big data processing architecture is designed using the Hadoop ecosystem along with a *MapReduce*-equivalent algorithm of the proposed method for speedy processing, enabling parallel processing on microarray data to reduce dimensions and perform classification through knowledge extraction. The proposed method is thoroughly tested on eleven microarray datasets by considering accuracy-interpretability-speed tradeoff. The results show that the proposed method is effective in identifying disease-causing genes while also understanding the patient's genetic profile with only a few operations and a small amount of CPU time. Statistical tests are also run to validate the proposed method's efficacy in comparison to other methods.

INDEX TERMS *f*-information, fuzzy expert system, microarray data, particle swarm optimization.

I. INTRODUCTION

Instead of clinical or morphological data, physicians have started using deoxyribonucleic acid (DNA) data for individual patients, to provide a "personalized medicine" based on each patient's unique genetic profile [1]. Next decade, doctors will routinely use our DNA to diagnose and treat our health conditions [2]. To this end, deoxyribonucleic acid microarrays (DNA-mircoarrays) [3] are a portion of an auspicious group of big medical data that is being used for reviewing and analyzing genetic material. However, DNA-mircoarrays are extremely large and complex to analyze for diagnosis and treatment. With this perspective, dimensionality, scarcity, and disparity are the three practical difficulties faced by

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

physicians when seeking a targeted therapy using a patient's gene expression [4]. An expert system can assist a physician to understand the molecular variations among the genes that are responsible for a disease. On this account, it is challenging to select the disease-causing genes from a vast genes pool in a DNA-microarray and construct an optimized expert system, to predict, diagnose, or treat a disease.

Several approaches used mutual information (MI) and *f*-information (FI) to identify relevant genes from microarray data. Our past approaches utilized mutual information (MI) [5], [6] for appropriate genes selection, guaranteeing a decent tradeoff between impartiality and discernment. But, these approaches could not completely address the issue of finding relevant genes by eliminating redundant genes and intimating an assortment of dismissed genes. On this account, *f*-information (FI)—as used by Maji [7]—is an encouraging substitute to MI for genes selection. Despite its effectiveness, the criterion function of FI divides continuous gene-expression values into numerous disconnected segments when estimating the marginal and joint probabilities. The discretization of continuous gene values during dimension reduction using MI and FI is failed to preserve the inherent meaning of genes. This misleads the original genetic meaning of the microarray data and produces more false positives.

To overcome the issues of using standalone MI or FI for gene selection, researchers combined multiple approaches. MI is combined with the rough set by Foithong et al. [8] to estimate the criterion function in crisp approximation spaces. Yet, the system was unsuccessful for ill-defined boundaries between gene-expression values. Contrarily, parzen-window and histogram-based MI [9] selects the relevant genes by estimating the density of unbroken numerical values of genes. But, this approach was only suitable for short attributed records. Recent approaches [10] and [11] combined the fuzzy and rough sets to deal with vague values and pacts with approximation spaces, to select suitable genes without discretization. As a result, in this paper, we combine the concepts of fuzzy and rough sets to modify the fundamental meaning of FI, calculating the relevance and redundancy between genes without dividing their expression values into discrete parts. The genes are ranked based on computed values of modified *f*-information (MFI).

After a successful dimension reduction of microarray data, which means the most relevant genes are selected, the selected genes are fed into Fuzzy Expert System (FES) to produce rules for disease diagnosis, prediction, or treatment. Knowledge acquisition is the key task performed by an FES whose objective is to find suitable fuzzy "if-then" rules and membership functions (MFs) from the data, to perform better decision-making. In this regard, an ideal FES can be formed by an optimization approach like genetic swarm algorithm (GSA), ant-bee algorithm (ABA), particle swarm optimization (PSO) [12], and others. The genetic swarm algorithm (GSA) [5] improved the classification accuracy of the FES, yet at the cost of interpretability. The "if-then" rules produced by the GSA are lengthy and complex, making it difficult for physicians to understand. The approach based on an ABA [6] addressed the interpretability-accuracy tradeoff using modified representation and hybrid problem-specific methods. Although a compact rules set with better comprehensiveness was produced, the use of more compound operations with more tunable control parameters consumed more CPU time.

To reduce processing time, PSO is a popular optimization algorithm with a few adjustable parameters, and it locates the optimal point more quickly than other algorithms while designing FES. A PSO-based approach [12] has been exposed to experiential and hypothetical surveys by numerous investigators [13] to progress its learning skill. In the PSO-based approach, velocity is considered a crucial feature that determines the resolution and restricts the movement size and direction of each particle. Velocity clamping [14] process controls the movement of a particle in the search space. During this process, the position of a particle is not guided, resulting in a poor exploration and exploitation capability. Even though a guided velocity modification was included in [15], it is still domain-dependent and unable to govern a particle for an exclusive search of a solution pool.

To address the aforementioned issues in establishing an optimum FES, in this paper, we designed a conditionally guided PSO (CGPSO) algorithm—with suitable modification to PSO—to formulate an optimum FES, to get the diagnostic response faster with improved accuracy and better interpretability with a big data processing platform. In addition to velocity clamping, we presented a simple and novel indicator for evaluating the velocity of each particle in PSO. The proposed CGPSO evaluates particle velocity using an updated probability, thereby eliminating randomly nominated particles and concentrating the main exploration near the global best. The proposed CGPSO frequently changes the search direction, hence producing a better exploration and exploitation capability for rapid extraction of the fuzzy "if-then" rules and MFs.

Each ruleset and MF extracted by CGPSO from the dataset is fired using a fuzzy inference procedure to find out the number of correctly classified samples that in turn used to find the fitness value. Even though the Mamdani inference system used in our previous work [5], [6] has widespread acceptance due to its intuitive nature, many studies have proven that the inference procedure followed in Tagaki, Suguno and Kang (TSK) [16] is compact and computationally efficient. The author suggests using the TSK inference procedure in this work because our goal is to make qualitative thinking faster to reach a definite conclusion. Because the TSK procedure works well with adaptive optimization techniques, it is recommended for the proposed CGPSO-based FES that is adaptive in terms of velocity updation to ensure a definite output.

Apart from dimension reduction of genes data and FES optimization, there is also an obligation of advanced tools and technology to process the big microarray data, efficiently. Existing approaches [17] lack such tools and technologies, and they did not disclose the algorithmic details for the implementation on a patient's genetic profile. On this account, we came up with an advanced Hadoop-based architecture equipped with *MapReduce*-based algorithmic details of the proposed dimension reduction method and FES design, to analyze data in a faster way for time-critical decisions.

The key contributions of this article are summarized as follows.

1) Initially, we introduced the accuracy-interpretability-speedy tradeoff that has to be considered while designing an FES. We pointed out gaps in existing research, neglecting this tradeoff.
2) We proposed a dimension reduction method to filter out the biologically meaningful genes from the microarray data. By the proposed method, we modify *f*-information by combining fuzzy with rough set theory to select relevant genes—from a pool of genes—for disease analysis.
3) We developed a solo algorithm to handle the standalone objective function with less tunable parameters. To this end, a novel conditionally guided particle swarm optimization (CGPSO) with TSK inference procedure is proposed for an ideal FES design, to find accurate as well as interpretable fuzzy ''if-then'' rules and MF in a faster manner.
4) Finally, we designed a big data processing framework with Hadoop ecosystem and *MapReduce* programming paradigm to handle a large microarray data. Deliberately, a *MapReduce*-equivalent algorithm of the proposed approach is implemented on the design framework.
5) At the end, the overall system is thoroughly investigated to meet the accuracy-interpretability-speedy tradeoff. The system is evaluated to relevant genes selection, precise, linguistic, and fast fuzzy modeling. The proposed system is compared with state-of-the-art research and statistically validated.

The rest of the article is organized as follows. Section II discusses the preliminary concepts that may help in understanding the overall proposed system. Section III highlights various components of the proposed MFI-CGPSO-based FES. Whereas, Section IV explains the architecture to implement these components in a faster way using the Hadoop framework. The simulation experiments, gene-expression datasets used in the experiments, and the outcomes are presented in Section V. Finally, the article is concluded in Section VI.

II. PRELIMINARIES

A. ACCURACY-INTERPRETABILITY-SPEEDY TRADEOFF IN FUZZY EXPERT SYSTEM

Accuracy maximization and complexity minimization are the two main goals of a fuzzy expert system. The former is concerned with the correctness of the sample classification, while the latter is concerned with the interpretability of the rules set.

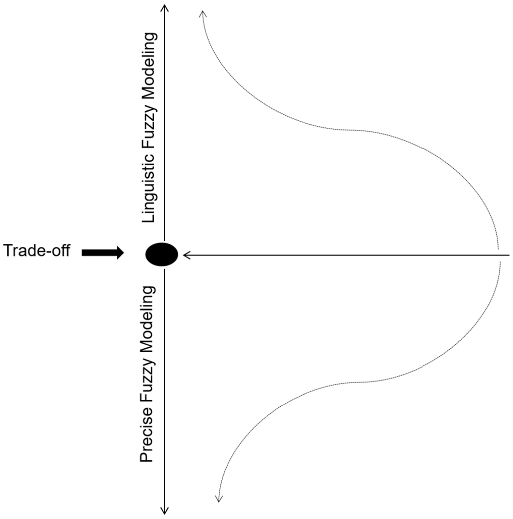


FIGURE 1. Accuracy - interpretability - speedy tradeoff.

In essence, these two objectives are opposed. Approaches to precise fuzzy modeling aim for accuracy while also attempting to improve interpretability. Meanwhile, linguistic fuzzy modeling sets interpretability as the objective and tries to improve the accuracy.

In the medical context, doctors need to process data very quickly to take decisions for gene therapy-based personalized medicine. Furthermore, in the big data arena, data must be analyzed during processing rather than after storage. As a result, the issue of fast knowledge extraction must be considered in the accuracy versus interpretability tradeoff, as illustrated in Figure 1. Fast fuzzy modeling tries to improve the accuracy and interpretability in less CPU time. The precision of an FES can be enhanced by fine-tuning the MFs.

Our previous GSA [5] focused only on the accuracy of FES with a large number of complex parameters. To this end, setting the ideal value for many adjustable control parameters was cumbersome. Another approach ABA [6] tried to improve the interpretability at the cost of accuracy with problem-specific representation and algorithms; however, the convergence was very uncertain. Furthermore, our previous approaches executed two algorithms concurrently to achieve the desired goal, which consumed more CPU time and was problematic for a time-critical analysis. As a result, in this paper, we propose a solo algorithm with a novel modification to PSO of adjusting the velocity based on update probability to acquire prime rules set and membership-function points. As a result, the proposed conditionally guided PSO (CGPSO) is very fast in locating the optimal points in the search space with improved accuracy and interpretability.

B. PARTICLE SWARM OPTIMIZATION (PSO)

PSO is a popular optimization method that preserves a swarm of particles' positions to represent candidate solutions for a problem. Each particle position in a swarm has a variable velocity that it uses to move around the exploration space. Using a fitness function, each position in the swarm

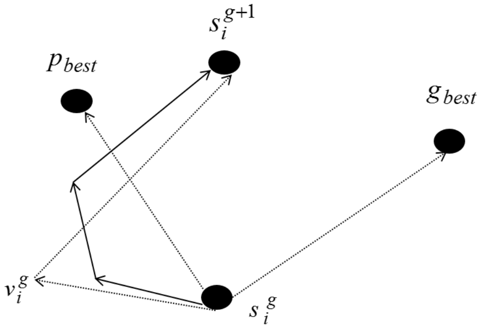


FIGURE 2. Depiction of velocity and position updates in PSO.

is evaluated for its contribution to the problem. Reiteration results in the creation of a new pool based on the following two calculations:

$$v_i^{g+1} = w_i v_i^g + c_1 rand * (p_{best} - s_i^g) + c_2 rand * (g_{best} - s_i^g) \tag{1}$$

$$s_i^{g+1} = s_i^g + v_i^{g+1} \tag{2}$$

here v_i^g represents velocity of i^{th} particle at iteration g, v_i^{g+1} is the velocity of i^{th} particle at iteration $g+1$, s_i^g is the position of i^{th} particle at iteration g, s_i^{g+1} is the position of i^{th} particle at generation $g+1$, w is the inertia weight, c_1 is the self-confidence factor and c_2 is the swarm confidence factor, p_{best} is the particle's individual best, and g_{best} is the global best. Equations (1) and (2) appraise the velocity and position for every particle iteratively, until reaching the optimal condition. Figure 2 depicts the velocity and position updates in conventional PSO. Our proposed approach adopts the conditional velocities—which are moderate based on an arbitrary threshold—while using PSO to achieve faster optimization and best accuracy. We named this version of PSO as 'CGPSO'.

III. DESIGN OF MFI-CGPSO-BASED FUZZY EXPERT SYSTEM

A schematic depiction of the major tasks involved in the proposed MFI-CGPSO-FES to combat the accuracy-interpretability-speedy tradeoff is shown in Figure 3. Initially, the preprocessing is performed to handle missing values and noise in the data. The dataset is then normalized to remove outliers. MFI ranks each gene in the microarray data and chooses the best ones for FES construction. The top genes are then used by the proposed CGPSO algorithm to construct an optimized FES. Finally, one of the existing inference systems is used to compact the FES, further reducing processing time. Each of the tasks is thoroughly discussed in the following subsections.

A. PREPROCESSING AND NORMALIZATION

The dataset should be preprocessed to remove missing values, and noise [18]. Missing values are filled in this process using the attribute mean of all samples from the same class. The noisy data is smoothed out using binning. During gene selection, higher-valued genes (in terms of expression value) may tend to suppress the influence of lower-valued genes. To minimize the effects of magnitudes among gene expressions, the dataset is normalized [18] to set each gene-expression value in the range of 0 to 1 using (3), Where x_n is the normalized value, x is the actual value, x_{min} and x_{max} are the minimum and maximum expression values of a gene, respectively. *StartingValue* is the new minimum value (0), considered in the range (0,1).

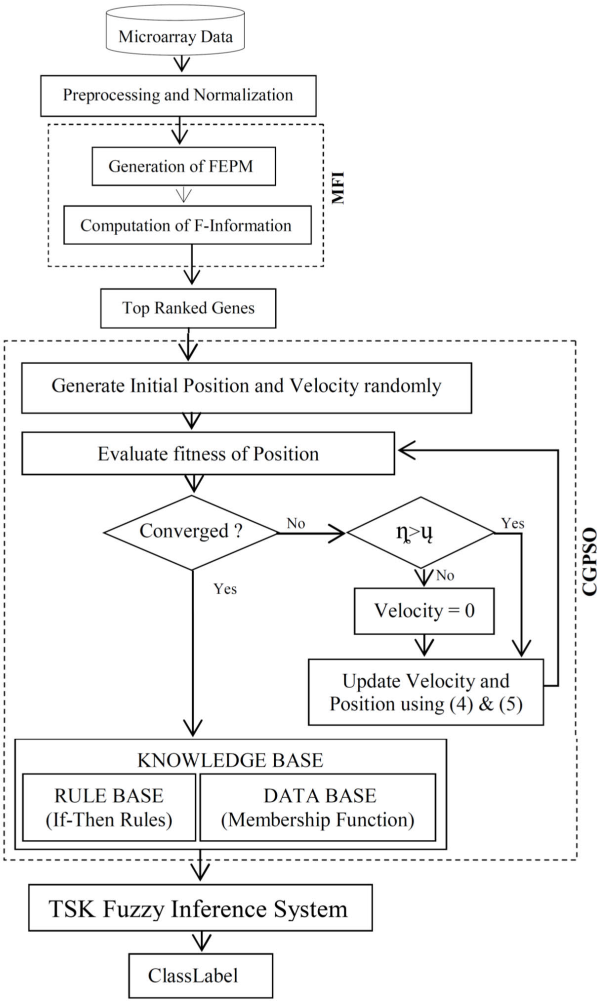

Microarray Data
Preprocessing and Normalization
Generation of FEPM
Computation of F-Information
MFI
Top Ranked Genes
Generate Initial Position and Velocity randomly
Evaluate fitness of Position
Converged ?
No
η>ų
Yes
No
Yes
Velocity = 0
Update Velocity and Position using (4) & (5)
CGPSO
KNOWLEDGE BASE
RULE BASE (If-Then Rules)
DATA BASE (Membership Function)
TSK Fuzzy Inference System
ClassLabel

FIGURE 3. Flowchart to design CGPSO-based fuzzy expert system.

$$x_n = \frac{(x - x_{min}) * range}{x_{max} - x_{min}} + StartingValue \tag{3}$$

The datasets considered in this work are already processed to free from outliers for making them available in the public repositories.

B. MODIFIED f-INFORMATION

In general, *f*-information [7] calculates the relevant and redundant genes through discretization of the continuous gene-expression values. During analysis, the original meaning of genes is lost due to discretization. Although the rough set constructs characteristics relationships to describe the uncertainty in gene-expression values, it considers two equal continuous gene-expression values to be different due to the crispness in the lower and upper approximation spaces. To overcome these problems, this paper proposes a modified way of computing *f*-information by combining the concept of fuzzy with rough set theory [10], [11]. The fuzzy set replaces a rough equivalence relation with a flexible fuzzy similarity relation, resulting in fuzzy approximation spaces that address the vagueness and coarseness nature of uncertainties found in continuous gene-expression values. The procedure for the selection of the top-ranked genes using modified *f*-information (MFI) with the help of hybrid fuzzy rough set theory is formulated as follows.

Procedure: genes selection using MFI

Input:

$G =$ Gene Expression Value of Microarray Data

$m =$ no. of samples, $n =$ no. of genes, $c =$ class label

Compute:

$\pi =$ membership value, $P =$ positional value, $FM =$ FEPM matrix

Output:

$G_{sig} =$ significant genes, $G_{sev} =$ severance genes

$G_{ran} =$ rank for genes, $G_{mng} =$ meaningful genes

Steps:

1) *Read* G, and partition into three gene groups Low (L) and Medium (M) and High (H), using the mean (μ) such that Gene Low is $G_L \subseteq G < \mu$, Gene Medium is $G_M \subseteq G = \mu$ and Gene High is $G_H \subseteq G > \mu$.
2) *Calculate* Mean (μ_L, μ_M, μ_H) and Standard Deviation ($\sigma_L, \sigma_M, \sigma_H$) for each group G_L, G_M and G_H.
3) *Calculate* for each genes-group (L, M, H)

$$\begin{aligned} & 2(1 - ||G - \mu_{L|M|H}||)^2, \\ & \quad \frac{\sigma_L}{2} \leq ||G - \mu_{L|M|H}|| \leq \sigma_{L|M|H} \\ \pi_{L|M|H} &= 2(1 - ||G - \mu_{L|M|H}||)^2, \\ & \qquad 0 \leq ||G - \mu_{L|M|H}|| \leq \sigma_{L|M|H} \\ & \qquad 0, \quad otherwise \end{aligned}$$

4) Calculate P for each genes-group ($L, M\ andH$)

$$P_{L|M|H} = \frac{\pi_{L|M|H}}{\pi_L + \pi_M + \pi_H}$$

5) Calculate

$$FM = \begin{bmatrix} P_L \\ P_M \\ P_H \end{bmatrix}$$

6) Significant genes are found using

$$\begin{aligned} G_{sig} = & \left| \frac{1}{n_l} \sum_{j=1}^{n_l} (P_L^G \cap P_L^C) - \frac{1}{(n_l)^2} \sum_{j=1}^{n_l} P_L^G \cdot \sum_{j=1}^{n_l} P_L^C \right| \\ & + \left| \frac{1}{n_h} \sum_{j=1}^{n_h} (P_H^G \cap P_H^C) - \frac{1}{(n_h)^2} \sum_{j=1}^{n_h} P_H^G \cdot \sum_{j=1}^{n_h} P_H^C \right| \\ & + \left| \frac{1}{n_m} \sum_{j=1}^{n_m} (P_M^G \cap P_M^C) - \frac{1}{(n_m)^2} \sum_{j=1}^{n_m} P_M^G \cdot \sum_{j=1}^{n_m} P_M^C \right| \end{aligned}$$

here $n_{l|h|m}$ is the number of genes in the corresponding group, G is the expression value and c is the class label.

7) Severance genes are found using

$$\begin{aligned} & G_{sev} \\ & = \left| \frac{1}{n_l} \sum_{j=1}^{n_l} (P_L^{rel} \cap P_L^{rem}) - \frac{1}{(n_l)^2} \sum_{j=1}^{n_l} P_L^{rel} \cdot \sum_{j=1}^{n_l} P_L^{rem} \right| \\ & \quad + \left| \frac{1}{n_h} \sum_{j=1}^{n_h} (P_H^{rel} \cap P_H^{rem}) - \frac{1}{(n_h)^2} \sum_{j=1}^{n_h} P_H^{rel} \cdot \sum_{j=1}^{n_h} P_H^{rem} \right| \\ & \quad + \left| \frac{1}{n_m} \sum_{j=1}^{n_m} (P_M^{rel} \cap P_M^{rem}) - \frac{1}{(n_m)^2} \sum_{j=1}^{n_m} P_M^{rel} \cdot \sum_{j=1}^{n_m} P_M^{rem} \right| \end{aligned}$$

here $n_{l|h|m}$ is the number of genes in the corresponding group, *rel* is relevant genes and *rem* is remaining genes.

8) Genes are ranked based on a high significance (relevancy) with a low severance (redundancy) value to select only the top-scored non-redundant and relevant genes to reduce the computation cost in the big data arena. For that MFI is calculated as a magnitude difference between G_{sig} and G_{sev}. The *MFI* values are ranked (G_{ran}), and the top genes are selected as meaningful genes (G_{mng}).

$$\begin{aligned} MFI &= |G_{sig} - G_{sev}| \\ G_{ran} &= min(MFI) \\ G_{mng} &= Top(G_{ran}) \end{aligned}$$

C. CONDITIONALLY GUIDED PARTICLE SWARM OPTIMIZATION (CGPSO)

In the traditional PSO, the amount and direction of the velocity usually determine the space and track of a particle's motion. If the velocity of a particle exceeds the determined allowable speed boundary during its update, it will be set to a higher rate of velocity. In general, the effect of momentum on particle movement allows for faster convergence and greater diversity in the search space. On the one hand, allowing velocities to increase increases the probability of a particle leaving the prominent search space, which may degrade performance. Smaller velocities, on the other hand, fail to fully investigate the feasible solutions for additional improvements. These considerations lead us to a conditionally guided PSO (CGPSO), where moderate velocities are produced (not too higher, not too smaller) using an arbitrary threshold value ŋ. This ŋ is completely random and being compared with predefined updated-probability ɥ to control the velocity updating process consistently, throughout the optimization process to bring out faster exploration and exploitation ability in PSO. Figure 4 depicts the scenario of selecting the velocity conditionally in CGPSO.

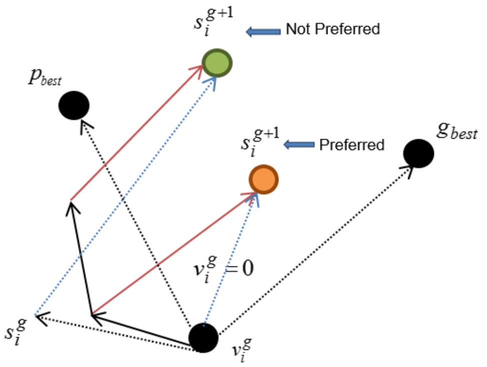


FIGURE 4. Conditional adaptive velocity updation in CGPSO.

The CGPSO runs as a standard PSO up to the updating of p_{best} and g_{best}. Thereafter, for every particle position and velocity, an arbitrary floating number is created between zero and one, and compared with the defined updated-probability ɥ . If the floating number is greater, the previous velocity of the particle is modified using the velocity updating equation (1), to obtain a new velocity. Otherwise, the previous velocity of the particle is set to zero and then modified using (1). As elaborated in Figure 4, there is two-position; one in green color, calculated using velocity updated from the previous velocity, whereas the other one in orange color, calculated using the velocity updated from zero. A kind of adaptive selection mechanism [19], [20] is followed where a position is preferred which is closer to the g_{best}, to cope up with the bound-constrained continuous search spaces. This conditional velocity updating method is similar to a mutation operator in a genetic algorithm and helps to suppress the movement of a few arbitrarily selected particles, causing a prime search around the overall best value. Furthermore, it also helps in determining the solution pool for a faster optimization process.

D. IMPLEMENTING CGPSO FOR FES

Defining the solution variables (rules set and membership function (MF)) and constructing the objective function are the two major tasks of implementing CGPSO to design the best FES. To encode the solution variables of the FES, the range of expression values of an individual gene is segregated into parts to identify the linguistics. In general, three to seven fuzzy partitions are appropriate. As shown in Figure 5, our method partitions the input gene into three regions as low 'L', medium 'M', and high 'H'. A trapezoidal MF is used to cover L & H, while a triangular MF is used for M. Three points are usually needed to plot each MF, thus, nine (3×3) points ($P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$) are required to encode the position of a particle in the swarm. P_1 and P_9 are permanent to signify the limits of the gene-expression value. The optimal values for the other points are found between the limits $[P_1, P_9]$ for P_2, $[P_2, P_9]$ for P_3, $[P_2, P_3]$ for P_4, $[P_4, P_9]$ for P_5, $[P_5, P_9]$ for P_6, $[P_5, P_9]$ for P_7, and $[P_7, P_9]$ for P_8. Suppose, if five partitions are considered, then fifteen (5 × 3) membership-function points (P_1 to P_{15}) are needed to represent the solution.

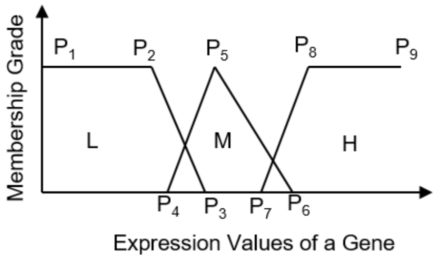


FIGURE 5. Input genes partitioning in fuzzy space.

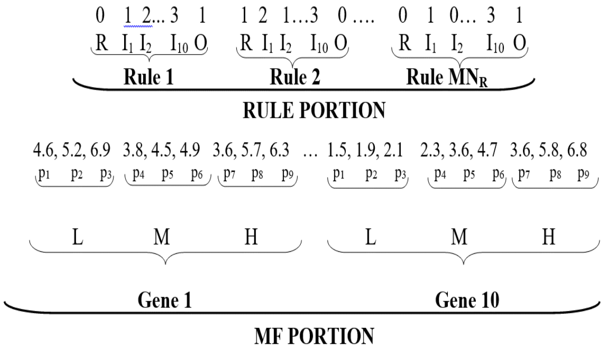


FIGURE 6. Representation of particle's position in group.

In this way, a single rule consists of three sections, represented by R, I_i, and O. R denotes the rule selection, $I_1, I_2, I_3, \ldots\ldots, I_n$ denote the linguistics of the gene expressions, and O denotes the class label. Using these indications, a particle's position in the group is given, as shown by Figure 6. The MF portion of gene 1 ($L/M/H$) flows to the I_1 (1/2/3) part of every rule ($1, 2, .., MN_R$) in rule-portion to obtain the membership grade value as per the inference procedure. For example, if the expression value read from the data for gene 1 is 4.2, then as per the Figure 6, this value is between P_4 and P_5 of medium partition (M) of gene 1. Since the I_1 of rule 2 holds '2' to represent the medium partition, it gets fired to give its corresponding membership grade.

Representing the rules set as integers and the MF as floating points avoids the hamming cliff problem, and is suitable for the amorphous expression values of genes. Moreover, using the class label as a rules-set variable keeps away the situation of having more than two rules firing for the same predicted class. During the CGPSO, each position in the group is appraised by formulating an objective function using (4).

$$f_{min} = (m - C_c) + (k * SN_R) \tag{4}$$

Here m is the total number of samples, C_c is the number of correctly classified samples, SN_R is the selected number of rules from the maximum number of rules (MN_R), and k is a constant used to amplify the small value SN_R. From (4), it is evident that the component $(m - C_c)$ calculates the error, while the CGPSO tries to minimize it to improve the accuracy of the system. The component $(k * SN_R)$ attempts to produce a compact rules set where the interpretability and production speed are more decently addressed by the proposed single conditional velocity updating operation, as compared to existing approaches [5], [6] where a higher number of complex operations and adjustable parameters were used.

E. INFERENCE PROCEDURE

After an optimized rules set by CGPSO, it is important to choose a better inference procedure to finalize the rules set, meeting the objective of speed, efficiency, and accuracy. We know that the Mamdani inference system used in state-of-the-art research [5], [6] have widespread acceptance due to its intuitive nature. But, many other studies have proven that the inference procedure produced by Sugeno and Kang (called TSK [21]) is compact and computationally efficient [16]. Since our goal is to make qualitative thinking in a faster manner to arrive definite conclusion, we suggest using the TSK inference procedure [21] with CGPSO. Furthermore, because the TSK procedure works well with adaptive optimization techniques, it is recommended that it be used in conjunction with the proposed CGPSO-based FES—as the CGPSO is adaptive in terms of velocity updation—to ensure a definite output. The TSK procedure results in the production of a lesser number of rules by replacing each fuzzy set with the function of input genes. TSK method performs faster as well as better than the Mamdani procedure in classifying complex ultra-high dimensional microarray data in the medical context.

IV. MULTILAYERED ARCHITECTURE FOR FAST PROCESSING OF MICROARRAY DATA

In addition to microarray dimension reduction and FES optimization, we need advanced hardware systems and tools to speed up the process. To accommodate this need, we came up with a multilayered architecture, as shown in Figure 7, to process large medical datasets in a faster way. Initially, the complex medical dataset—which might be in a raw form—is taken as an input source for feature reduction and classification. In the next 'preprocessing and filtration' layer, missing values are identified in the data and filled using preprocessing and normalization operation. Furthermore, at this stage, any unnecessary data that could not be normalized is filtered. The data in the form of gene-expression values is processed at the next layer to select the top-ranked genes using MFI. CGPSO is used to analyze the core (i.e., relevant) genes, resulting in a knowledge base that is then used to build an FES-based genetic model. Finally, the built model is used for testing, which includes labeling. Because model development is a large, complex, and time-consuming process that includes preprocessing and filtration, gene selection, fuzzy system modeling, and CGPSO processing, it is handled by the Hadoop ecosystem, which is equipped with the distributed file system 'HDFS' and the *MapReduce* programming paradigm. Hadoop is a very powerful tool that has a distributed file system 'HDFS' and distributed and parallel programming environment '*MapReduce*.' The data is divided into several chunks and stored on various data nodes using HDFS. The parallel processing on each of the chunks is performed using the *MapReduce* mechanism. The algorithm is implemented in the *MapReduce* environment using the *map* and *reduce* functions. The *map* function goes through each chunk line by line and returns a result for each line. The *reduce* function collects all of the results for a given dataset. There is also a *combiner* that collects all the inputs from each chunk and merges them. The division of the dataset into chunks, storing chunks into multiple data nodes using HDFS, and processing them using the *MapReduce* mechanism remarkably increases the efficiency of the system while handling big medical data.

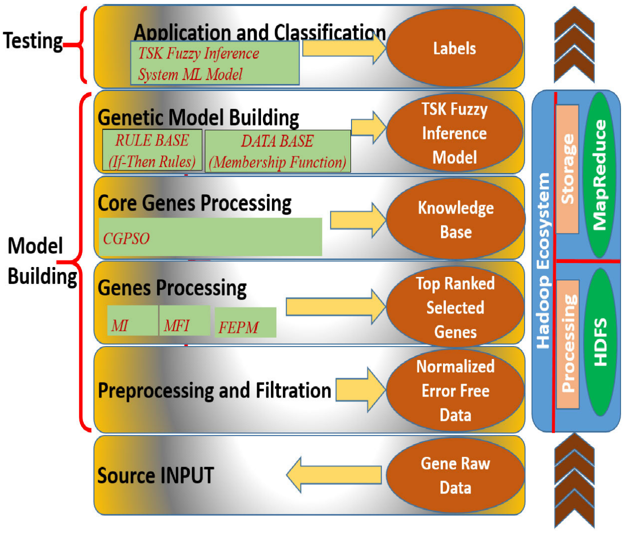

Testing
Application and Classification
TSK Fuzzy Inference System ML Model
Labels
Model Building
Genetic Model Building
RULE BASE (If-Then Rules)
DATA BASE (Membership Function)
TSK Fuzzy Inference Model
Core Genes Processing
CGPSO
Knowledge Base
Genes Processing
MI
MFI
FEPM
Top Ranked Selected Genes
Preprocessing and Filtration
Normalized Error Free Data
Source INPUT
Gene Raw Data
Hadoop Ecosystem
Storage
MapReduce
Processing
HDFS

FIGURE 7. Multilayered architecture to process big medical data.

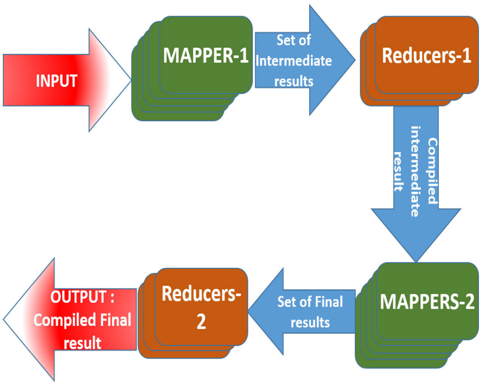

INPUT
MAPPER-1
Set of Intermediate results
Reducers-1
Compiled intermediate result
MAPPERS-2
Set of Final results
Reducers-2
OUTPUT : Compiled Final result

FIGURE 8. *MapReduce* programming model to process big medical data.

For better performance, we implemented both the MFI-based genes selection process and CGSPO-based FES optimization using multiple *map* and *reduce* functions. As previously stated, the *map* function processes the data line by line and returns a unique result for each line based on the code written in the *map* function. The results of the *map* function are passed to the corresponding *reduce* function, which compiles the results and produces an aggregated result. Depending on the system's complexity, multiple *map* and *reduce* functions may collaborate. We used the two-layered *map* and *reduce* mechanism, where the first level output by

Reducers-1 is used as input for the next level of mapper *Mappers*-2 for further processing, as shown in Figure 8. The complete *MapReduce* process for finding top-ranked genes is presented below as a pseudo-code in algorithm 1.

Algorithm 1 *MapReduce* Algorithm

```
map1(Key Gene, Value GeneValues)
    for each $g$ in Gene do
        $sum = sum + GeneValue$
        $m = m++$ // number of samples
    end for
    emit(Key Gene, Values ($sum$, $m$))
end map1
reduce1(Key Gene, IterableValues ($sum$, $m$))
    for each $sum\_val$ in $sum$ do
        $cum\_sum = cum\_sum + sum\_val.get()$
        $cum\_m = cum\_m + m.get()$//count sample values
    end for
    Calculate $\mu(cum\_sum, cum\_m)$
end reduce1
map2(Key Gene, IterableValues (GeneValues, $\mu$))
    Calculate $G_H$, $G_M$, $G_L$ based on $\mu$
    for each $g$ do
        if $GeneValues \in G_H$ then
            $sum\_G_H|G_L|G_M = sum\_G_H|G_L|G_M + GeneValue$
            $sum^2\_G_H|G_L|G_M = sum^2\_G_H|G_L|G_M + GeneValue^2$
            $m\_G_H|G_L|G_M = m\_G_H|G_L|G_M + 1$ //increment
        end if
    end for
    emit(Key Gene, Values ($sum\_G_H|G_L|G_M$, $sum^2\_G_H|G_L|G_M$, $m\_G_H|G_L|G_M$))
end map2
reduce2((Key Gene, IterableValues ($sum\_G_H|G_L|G_M$, $sum^2\_G_H|G_L|G_M$, $m\_G_H|G_L|G_M$))
    for each $val\_sum\_G_H|G_L|G_M \in sum\_G_H|G_L|G_M$ do
        $cum\_sum\_G_H|G_L|G_M = cum\_sum\_G_H|G_L|G_M + val\_sum\_G_H|G_L|G_M.get()$
        $cum\_m\_G_H|G_L|G_M = cum\_m\_G_H|G_L|G_M + m\_G_H|G_L|G_M.get()$
    end for
    Calculate $\mu\_G_H|G_L|G_M$, $\sigma\_G_H|G_L|G_M$, $\pi_{H|L|M}$, $P_{H|L|M}$, $FM$, $G_{sig}$, $G_{sev}$, $G_{mng}$
```

The *map* function takes input as a key and value pair. In our case, the *map*1 function computes the basic parameters, which are the sum of genes and the number of samples, using gene-ID as a key and gene values as a value attribute. The *map* function generates the gene-ID as a key and the sum of values and the number of samples as value attributes for each line for each gene and transfers them to the *reduce*1 function via the *emit* function call. All of the gene-IDs are used as keys in the *reduce*1 function, and the corresponding values are used as value attributes. *reduce*1 aggregates all the sums and number of samples and then calculates μ for each gene-ID. At next level, the *map*2 function is called for each *reduce*1 output. *map*2 takes the gene-ID as a key and gene values and corresponding μ values as a value parameter.

The *map* function determines the values of G_H, G_L, G_M based on μ value for each of the genes. Later, it calculates the basic parameters for each G_H, G_L, G_M. It calculates *sum*, *sum*^{2}, number of samples for each of G_H, G_L, G_M. The *map* function is executed on each line of the source data and generates all of these outputs for each individual line. Finally, the *reduce*2 function aggregates all the *map*2 outputs and computes other parameters that are used in finding top-ranked genes, such as μ_G_H, μ_G_L, μ_G_M, σ_G_H, σ_G_L, σ_G_M, G_{sig}, G_{sev}, G_{mng}, etc.

V. SIMULATION EXPERIMENTS AND EVALUATION

This section presents the details of experiments performed to validate the performance of the proposed MFI-CGPSO-FES approach—while considering the accuracy-interpretability-speedy tradeoff—using eleven gene-expression datasets. As a result, the proposed modeling approach's performance is evaluated and compared to state-of-the-art methods in terms of relevant gene selection, precision and perfection of the FES modeling, and accuracy of the designed expert system. In addition, the statistical validation of the proposed method is performed using several parameters and compared with state-of-the-art research.

A. SIMULATION ENVIRONMENT

All the simulations were implemented using MATLAB 7.11 in Intel Core i5 processor with a speed of 2.80GHz and 4 GB of RAM. Hadoop ecosystem is used along with the *MapReduce* programming paradigm in MATLAB. In addition, to study the biological relevance, the GO Sim package of 'R' software [22] was used.

B. GENE-EXPRESSION DATASETS

Existing approaches [5], [6], [12] were tested only on six microarray datasets and found to be non-claimable for other medical data. To confirm the robustness and scalability of the proposed method, five additional datasets on prostate, ovarian, breast, pancreatic, and lung cancer were included in the study. Table 1 display information about all of the datasets used in the simulation, sorted by dataset size. The datasets are summarized by the total number of genes (n) a dataset contains, total samples (m), the size of the dataset in terms of number genes-values (i.e., $n \times m$), total class-wise samples (m_c), and the disease it has classified (category labels).

C. RELEVANT GENES SELECTION

Reminding Section III-B, we identified relevant genes using MFI. For demonstration, values of fuzzy equivalence class (FEC) and fuzzy equivalence partition matrix (FEPM) for the individual gene 'PDX1'—considering 52 samples or patients—in the pancreatic cancer dataset are presented in

TABLE 1. Details of microarray datasets.

Dataset	n	m	$size$	m_c	Category labels
Colon cancer (col) [23]	2000	62	124000	40 22	Tumor Normal
Breast cancer (Bre) [24]	5776	27	155952	13 14	Breast Tumor Human mammary epithelia
Lymphoma (Lym) [25]	4026	45	181170	23 22	Germinal centre B-like(GCL) Activated B-like(ACL)
Leukemia (Leu) [26]	7129	72	513288	47 25	Acute lymphoblast leukemia Acute myeloid leukemia
RAO [27]	18432	31	571392	22 9	Rheumatoid arthritis(RA) Osteoarthritis(O)
Type 2 diabetes (T2D) [28]	22283	34	757622	17 17	Diabetes mellitus2(DM2) Normal glucose tolerance
RAC [29]	26000	35	910000	32 3	Rheumatoid arthritis(RA) Control (C)
Prostate cancer (Pro) [30]	12600	136	1713600	77 59	Prostate tumor Normal
Lung cancer (Lun) [31]	12600	181	2280600	150 31	Aclenocarcinoma (ADCA) Malignant pleural esothelioma (MPM)
Pancreatic cancer(Pan) [32]	54614	52	2839928	36 16	Tumor Normal
Ovary cancer (Ova) [33]	15154	253	3833962	162 91	Ovarian cancer Normal

Table 2. The values are grouped as low, middle, and high, calculated by the MFI procedure in Section III-B. Similarly, FEC and FEPM matrices are formed and a genes-group significance value is computed for each of the other genes. In the pancreatic cancer dataset, the 'PNLIPRP2' showed the highest significance value '0.4157', and was nominated as the most significant gene. Next, the gene-gene severance between the most significant gene 'PNLIPRP2' and all the other genes is quantified. The outcomes of genes-group significance G_{sig} and gene-gene severance G_{sev} are presented in Table 3. At the end, genes are ranked based on G_{sig} and G_{sev} values, aiming at maximizing G_{sig} and minimizing G_{sev}. The MFI values for the first-thirty genes of the pancreatic cancer dataset are presented in Figure 9.

The gene's relevancy is determined based on their ranks. Selecting a higher number of relevant genes negatively impacts the tradeoff between significance and severance [5], [6], [7], [8], [9], [10], [11]. As a result, at the CGPSO stage, we hardly considered the top ten genes to be relevant for expert system design. Table 4 lists the relevant genes chosen for each of the eleven datasets. However, these are not the final selected genes, as the CGPSO conditional statement provides the final meaningful genes with linguistics for their expression values.

D. PRECISE, ACCURATE, AND FAST FUZZY MODELING

The proposed MFI-CGPSO-based fuzzy medical expert system modeling is very fast and has a rich convergence behavior. It yields a very precise expert system model that is both accurate and general to any related problem. We present the results of our experiments to demonstrate the compactness and convergence behaviors, generalization and correctness, interpretability, and efficiency of the proposed method.

TABLE 2. FEC and FEPM values for PDX1 gene.

Fuzzy equivalence class for PDX1 gene					
FEC	S1	S2		S51	S52
Low Mediu High	0.0482 0.4516 0.8201	0.0145 0.5914 0.9361		0.1472 0.6147 0.8190	0.3811 0.6162 0.9445
Fuzzy equivalence partition matrix for PDX1 gene					
FEPM	S1	S2		S51	S52
Low Medium High	0.0974 0.6271 0.9362	0.0125 0.2572 0.8251		0.0462 0.5289 0.9153	0.2458 0.6147 0.8563

TABLE 3. Significance and severance values for prostate cancer dataset.

Gene-number	Gene-ID	G_{sig}	G_{sev}
G_1	AMY2A	0.193452	0.234561
G_2	GAD2	0.152567	0.343587
G_{54613}	PBCA	0.156722	0.145623
G_{54614}	CPA1	0.112345	0.532419

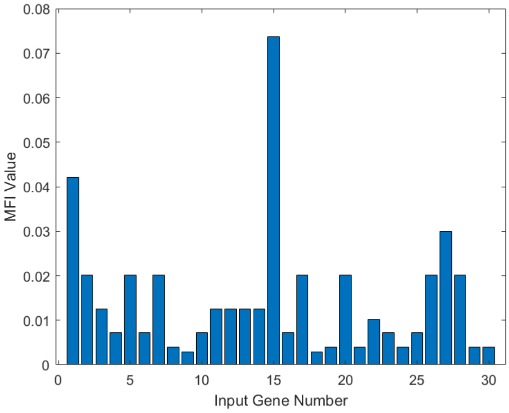


FIGURE 9. MFI values for pancreatic cancer dataset.

TABLE 4. Details of genes nominated by MFI.

Data	Gene-number
Col	1022, 870, 1158, 1077, 1358, 958,672,138,129,1238
Bre	8396,12580,559,4910,67, 4,15, 9639,181,289
Lym	1693, 2862, 1949, 2625, 1933, 1376, 222,189,964,131
Leu	2913,5198,98,37,176,3213, 4681, 103,567,4281
RAO	18117,8124,14774, 1293,9069, 16387, 5861, 9878, 636, 16
T2D	67,15,28,1216,1389,1761,138,69, 781,1667
RAC	268,1172,266,1275,981,123,1429,302,459,678
Pro	2047, 1829, 1089, 300,2168, 1713, 982,1035,1781,508
Lun	1250,6182,1615,136,611,2189, 6786,1346,1629,112
Pan	7864,8548,4456,4718, 8103,679, 7612,8321,4241,4678
Ova	5952,51071,120,461,1512, 6989, 613,781,137,817

1) CGPSO CONFIGURATION

To achieve an optimum fuzzy-based expert model to diagnose the disease, CGPSO is initially configured with a hybrid string (real and integer) to encode the MF and rules set. Each rule requires twelve integer numbers (one for R, ten for $I_1, I_2, \ldots, I_{10}$, one for O) in the solution pool. Seven membership-function points are used for each linguistic variable of the selected relevant genes. As a result, 70 floating-point numbers (i.e., 7 points × 10 relevant genes) are employed to come up with a compact rules set, starting from five to ten rules. In short, with ten rules, one hundred and

90 (10 rules × 12 integer numbers + 7 points × 10 relevant genes = 190) variables *var* are randomly initialized as a particle's position in the swarm. For each particle position, the velocity is randomly initialized between zero to V_{high}, where V_{high} is computed by (5).

$$V_{high} = \frac{maxLimit_i - minLimit_i}{2}, \quad i \in 1 \ldots var \qquad (5)$$

The size of the initial swarm space is kept within the range of 10-40. Each position in the swarm was evaluated by fitness function f_{min}, computed by (4), while changing the iterations from 10 to 100. The constant k varies from three to eight, depending on the selected number of rules SN_R. Around 50 independent experiments were conducted by varying the update-probability ɥ from 0.1 to 0.9, to examine the convergence performance of every particle inside the swarm, iteratively. Forty-three trials produced the best results for all the datasets with a swarm size of 25 and 70–110 iterations on 0.6 u.

2) COMPACTNESS AND CONVERGENCE

All of the designed fuzzy-based expert systems for disease diagnosis are very compact and accurate with the above-mentioned CGPSO settings. As an example, the designed expert system to confirm pancreatic cancer resulted in the selection of only seven genes and four fuzzy rules, as shown below.

1) If CELA2A is low and PPDPF is high, it is a tumor.
2) If FAM3B is high and C8orf22 is medium, it is a tumor.
3) If REG1A is low and is PBCA medium, it is normal.
4) If CELA2B is medium, it is normal.

Interestingly, the accuracy of the system is 98.07%. CGPSO found seven out of ten relevant genes to be highly responsible for cancer detection. Furthermore, the linguistic values for all seven genes are simple, allowing the physician to easily read the patient's genetic profile.

During the design process of the expert system for pancreatic cancer detection, the optimal membership-function points realized by the CGPSO for the CELA2A gene are shown in Figure 10. The values of the membership-function points are well adjusted and reasonable. This validates the CGPSO's faster-tuning capability, producing a MF with a partial mode connection and non-inclination toward the iterative dynamic ranges' boundary values within 100 generations.

The overall convergence behavior of the proposed CGPSO in the learning/training process of the pancreatic cancer diagnosis is compared with state-of-the-art FES optimization approaches, as shown in Figure 11. To justify the fair comparison, the relevant genes are selected with both MI and MFI procedures for all GA, PSO, GSA, ABA, and CGPSO optimization approaches. The results show that any optimization approach with MI-based genes selection performs worse than one with MFI-based genes selection. The genes nominated by MI may have been deprived due to discretization, and more generations are required to build a knowledge base that matches the input with the corresponding output class.

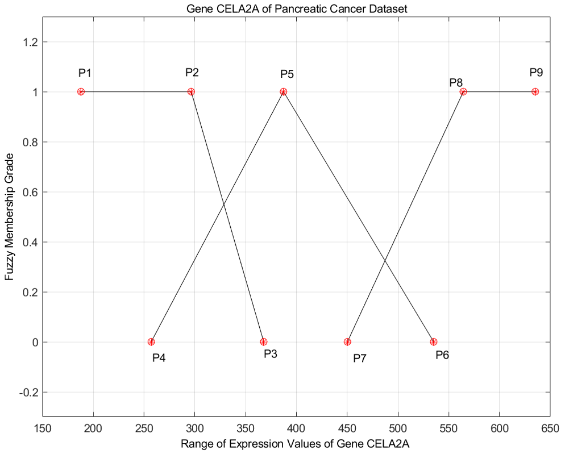


FIGURE 10. Optimal membership function formed by CGPSO for 'CELA2A' gene of pancreatic cancer dataset.

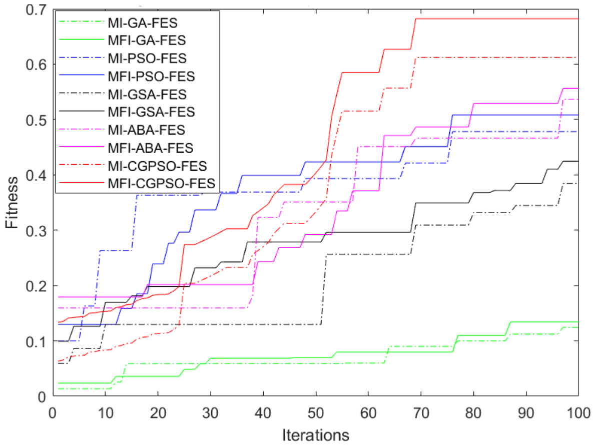


FIGURE 11. Convergence comparison of various approaches using pancreatic cancer dataset.

As a result, even if you use an optimization method other than CGPSO, we recommend using MFI for gene selection. With the offered MFI-based genes-selection and CGPSO-based optimization, we witnessed a faster convergence rate from the beginning and an optimal solution at 70^{th} iteration. We noticed PSO and CGPSO both showed an abrupt increase in the fitness value, whereas the GA, GSA, and ABA have a steady rise due to a large number of complex operations. In any case, regardless of whether MI or MFI is used to select relevant genes, CGPSO outperforms GA, PSO, GSA, and ABA approaches in terms of convergence behavior. The proposed MFI-based CGPSO demonstrated slightly better learning ability, with higher classification accuracy and fewer rules.

3) GENERALIZATION AND CORRECTNESS

The generalization ability of the MFI-CGPSO-FES was validated using the Monte Carlo cross validation (MCCV) [34]. For MCCV, S training sets $\{T_r\}_{(s=1,2,3,C,S)}^{s}$ are randomly formed by selecting records without replacement, from a dataset D containing m samples. For each training set $\{T_r\}$, samples for the testing set $\{T_e\}$ are randomly chosen from

TABLE 5. MCCV results using MFI-CGPSO-FES.

Datasets	$\mu\pm\sigma$
Col	0.28 ± 0.31
Bre	0.26 ± 0.12
Lym	0.22 ± 0.07
Leu	0.09 ± 0.11
RAO	0.24 ± 0.14
T2D	0.14 ±0.03
RAC	0.15 ± 0.03
Pro	0.24 ± 0.19
Lun	0.16 ± 0.11
Pan	0.12 ± 0.02
Ova	0.18 ± 0.06

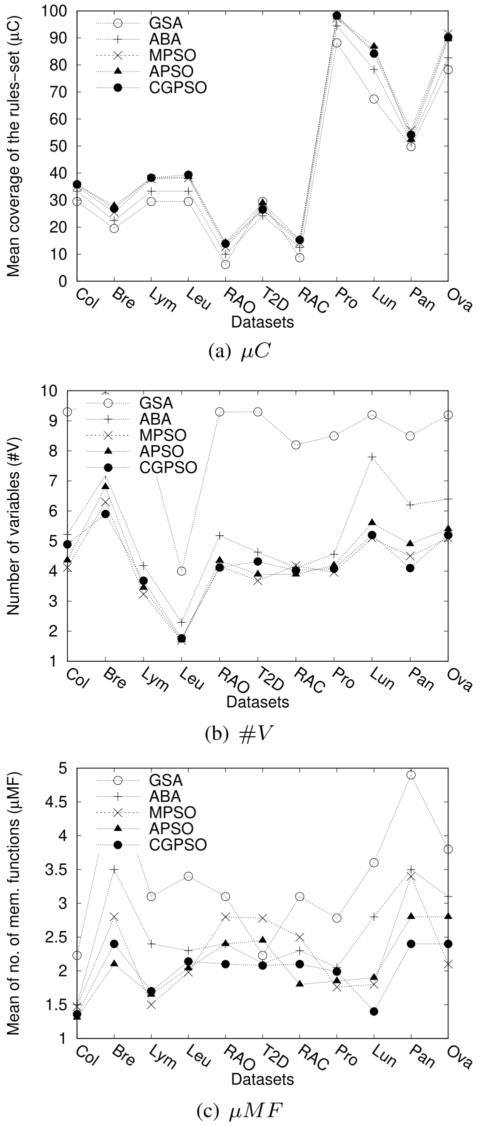
the remaining set $\{D - T_r\}$, meeting the ratio $mT_r^{(s)} : mT_e^{(s)}$. In this paper, we fix the ratio 4:1, which means the size of $\{T_r\}$ is 4 times the size of $\{T_e\}$. The error rate is computed using (6).

$$\hat{\epsilon}_{MCCV}(D(\{T_r\}^s)_{s=1,2,3,\ldots,m}) = \frac{1}{S}\sum_{s=1}^{S}\hat{\epsilon}_{TEST}(D,(\{T_r\}^s,\{T_e\}^s)) \qquad (6)$$

During the MCCV process, an individual FES is designed for each training dataset $\{T_r\}^s$ by employing the proposed MFI-CGPSO approach. Afterward, the FES is testing on the corresponding training dataset $\{T_e\}^s$. Inaccurate classified samples are counted at each of the S trials, and the average is taken as an error rate using (6). Table 5 presents the *MCCV* results for all the eleven datasets in terms of mean (μ) and standard deviation (σ) of error rates, considering $S = 50$ for each dataset. The error rate, expressed in terms of *mu* and *sigma*, is found to be the lowest in all iterations and across all test data sets. The minimum error rate demonstrates the suitability of MFI-based gene selection as part of our proposed CGPSO-based FES.

4) INTEPRETABILITY

The measure of interpretability is the ability to understand the operation of a fuzzy expert system. Many factors are used to calculate it, including the number of input variables, the number of fuzzy rules, the number of linguistic terms, and the shape of the fuzzy sets. In this paper, we computed interpretability in terms of the mean coverage of rules set (μC), the number of variables ($\#V$), and the average number of MFs (μMF). Figure 12 compares the interpretability of our system against state-of-the-art approaches, including GSA [5], ABA [6], adaptable velocity modified PSO (MPSO) [20], and Adaptable PSO (APSO) [19]. Overall, all the approaches produced competitive coverage values. Because MPSO and APSO are similar to the proposed CGPSO method in terms of updating the velocity adaptively, they produce nearly the same coverage value as CGPSO. The CGPSO, on the contrary, outperformed the GSA and ABA. The CGPSO produced a much smaller rules set with far fewer genes and MFs than the GSA and ABA. However, our approach has slightly inferior interpretability than MPSO and APSO in some of the cases. But, the computational overhead of MPSO and APSO is far larger than the CGPSO.



(a) μC

(b) $\#V$

(c) μMF

FIGURE 12. Interpretability comparison of CGPSO with other approaches.

5) EFFICIENCY

To analyze the efficiency aspect of the proposed FES modeling approach, we look at the proposed method to average processing time consumed on one gene value. We notice that the CPU time consumed to process a single gene value is less than 0.2 ms, except for col and bre datasets, as shown in Figure 13. Since the whole process is implemented on the big data processing platform, which is equipped with parallel processing mechanism of Hadoop and *MapReduce*, the average time decreases with the growing dataset size. The designed big data framework efficiently processes large datasets and makes algorithm implementation relatively faster with larger datasets. Furthermore, when working with a large dataset containing a large number of genes and samples, the CGPSO's chances of obtaining optimal value at an earlier

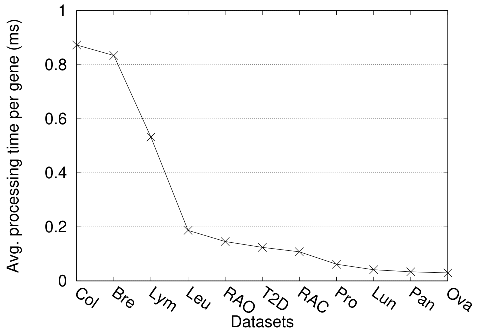
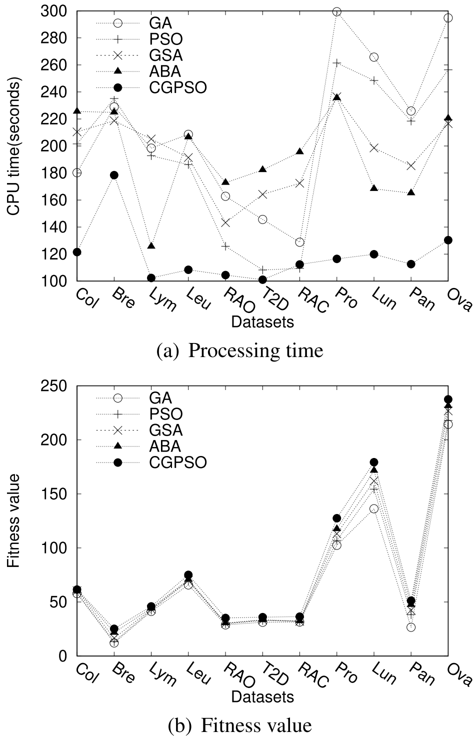


FIGURE 13. Processing time of genes in the proposed big data architecture.

stage are increased. In a nutshell, the results prove that the proposed MFI-CGPSO-based expert system modeling with a big data platform can handle massive datasets with a rate of less than .01ms per gene value.

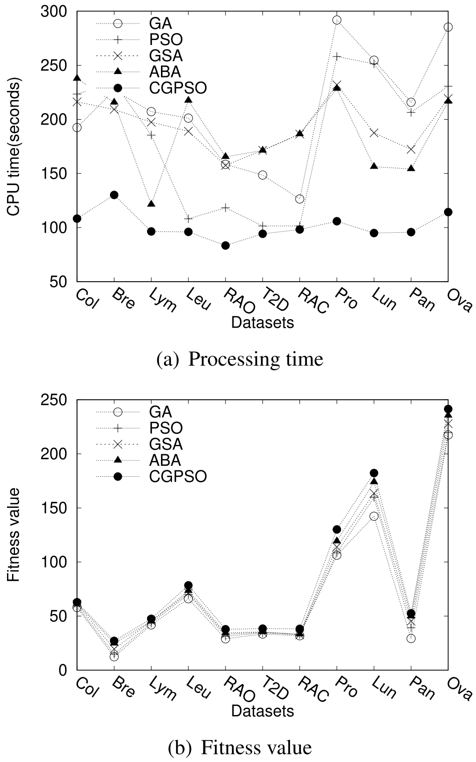
The proposed MFI-CGPSO-based FES modeling is very fast as compared to existing approaches. To fairly compare the efficiency of the proposed FES modeling with state-of-the-art methods, we tried both MI and MFI-based genes selection, individually, as a prerequisite of the FES design and optimization process. Figure 14 and Figure 15 present performance comparison between the proposed method and other state-of-the-art FES methods. The CPU time reflects the processing time consumed in the whole modeling process from a genes selection process to the final FES design. Figure 14 shows the results of the first trial, where the genes are selected using MI from all the datasets one by one, and the optimization approaches—such as GSA, ABA, MPSO, APSO, and our method CGPSO—are applied on the selected genes to form an individual FES design for each dataset. Whereas, Figure 15 presents the results of the 2nd trial, where the genes are selected by the proposed MFI-based algorithm. The corresponding fitness values selected by each method during the FES optimization are shown in part (b) of the figures for both trials. All of the methods performed well; the PSO was slightly faster, but the CGPSO is even faster due to its simplified operations, though it did not achieve the GSA and ABA's optimality. ABA produced interpretable rules competitively but consumed more CPU time due to the combinatorial operation in forming simple rules. Overall, the proposed CGPSO with a single control parameter achieved the desired fitness value quickly, using less CPU time, and with more accuracy for all the datasets. Furthermore, the novel idea of updating the velocity conditionally in CGPSO also simplified the standard PSO operations, allowing the rapid extraction of the rules set and MF from the data.



(a) Processing time

(b) Fitness value

FIGURE 14. Performance comparison with state-of-the-art methods with MI.



(a) Processing time

(b) Fitness value

FIGURE 15. Performance comparison with state-of-the-art methods with MFI.

E. STATISTICAL VALIDATION AND COMPARISON

To further evaluate and justify our approach, we identified differences in results produced by our method and state-of-the-art research. On this account, we employed Wilcoxon's signed-rank test [12], and analyzed the effects of the proposed MFI procedure against several information theory-based genes-selection approaches [7], [34], [35], [36], [37], [38], [39] and other popular state-of-the-art methods [16], [40], [41], [42], [43], [44], [45]. Using the same test, we compared outcomes of the proposed CGPSO with other competitive

TABLE 6. Results of Wilcoxon's test for genes selection.

Comparison	Genes				Accuracy				CPU time			
	R^+	R^-	$\mathcal{P}$	H	R^+	R^-	$\mathcal{P}$	H	R^+	R^-	$\mathcal{P}$	H
MI [35] Vs MFI	1	65	0.33	R^j	0	66	0.36	R^j	2	60	.28	R^j
CMI [36] Vs MFI	2	64	.48	R^j	4	62	.51	R^j	3	63	.49	R^j
mRMI [37] Vs MFI	2	64	.48	R^j	9	57	.92	R^j	5	61	.57	R^j
QMI [38] Vs MFI	5	61	.57	R^j	13	53	.99	R^j	9	57	.92	R^j
NMI [39] Vs MFI	6	60	.89	R^j	18	48	.13	R^j	12	54	.97	R^j
MSMI [34] Vs MFI	9	57	.92	R^j	21	45	.19	R^j	0	66	.36	R^j
FI [7] Vs MFI	8	58	.96	R^j	23	43	.11	R^j	6	60	.69	R^j
t-T [16] Vs MFI	12	53	.76	R^j	15	50	.58	R^j	22	43	.75	R^j
RF [40] Vs MFI	35	30	.63	R^a	22	40	.69	R^j	18	47	.67	R^j
MBF [41] Vs MFI	20	45	.83	R^j	40	25	.19	R^a	24	41	.88	R^j
SW [42] Vs MFI	15	52	.92	R^j	20	48	.74	R^j	15	50	.72	R^j
mAHP [43] Vs MFI	13	53	.96	R^j	15	47	.82	R^j	17	48	.69	R^j
GTA [44] Vs MFI	38	27	.54	R^a	44	21	.67	R^a	27	38	.94	R^j
mvlPSO [45] Vs MFI	22	40	.85	R^j	27	42	.95	R^j	30	35	.81	R^j

t-T:t-test, RF:random forest, MBF:Markov blanket filtering, SW:sample weighting, mAHP:modified analytic hierarchy process, GTA:graph theory approach, mvlPSO:multi-objective variable length PSO.

TABLE 7. Results of Wilcoxon's test for knowledge acquisition.

Comparison	Rules				Interpretability				CPU time			
	R^+	R^-	$\mathcal{P}$	H	R^+	R^-	$\mathcal{P}$	H	R^+	R^-	$\mathcal{P}$	H
GA [46] Vs CGPSO	2	51	.23	R^j	2	48	.27	R^j	1	54	.48	R^j
PSO [12] Vs CGPSO	4	49	.38	R^j	5	66	.49	R^j	4	51	.52	R^j
GSA [5] Vs CGPSO	3	46	.17	R^j	6	43	.24	R^j	5	53	.42	R^j
ABA [6] Vs CGPSO	10	32	.07	R^j	15	32	.09	R^j	2	57	.96	R^j
APSO [19] Vs CGPSO	12	34	.45	R^j	35	24	.41	R	11	46	.86	R^j
MPSO [20] Vs CGPSO	15	38	.59	R^j	38	18	.23	R	15	42	.94	R^j
bPSO [44] Vs CGPSO	37	12	.97	R	12	41	.48	R^j	18	51	.84	R^j
kFIS [16] Vs CGPSO	41	10	.87	R	10	38	.59	R^j	12	54	.75	R^j

bPSO:binary PSO, kFIS:Kernel-fuzzy inference system

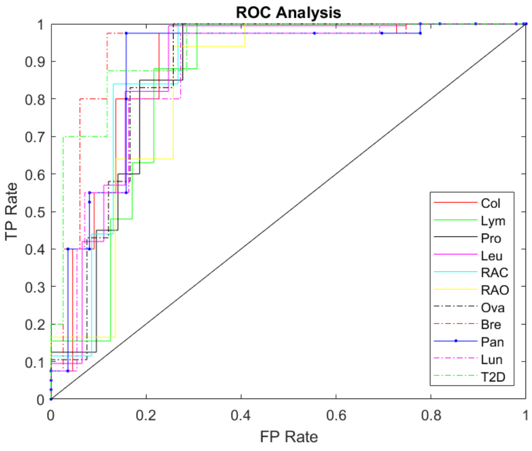


FIGURE 16. ROC curve of CGPSO using all datasets.

TABLE 8. Comparison of AUC for CGPSO and other approaches.

Data	GA	PSO	GSA	ABA	APSO	MPSO	bPSO	kFIS	CGPSO
Col	.748(5)	.753(4)	.763(3)	.776(2)	.741(6)	.735(8)	.738(7)	.73(9)	.785(1)
Bre	.512(5)	.618(1)	.595(3)	.586(4)	.412(8)	.399(9)	.475(6)	.435(7)	.599(2)
Lym	.615(5)	.675(4)	.747(3)	.768(2)	.572(7)	.596(6)	.489(9)	.570(8)	.782(1)
Leu	.584(5)	.647(3)	.695(1)	.632(4)	.549(9)	.573(7)	.561(8)	.581(6)	.661(2)
RAO	.426(5)	.462(4)	.483(3)	.525(2)	.423(6)	.374(7)	.371(8)	.368(9)	.561(1)
T2D	.552(5)	.612(2)	.562(4)	.583(2)	.476(8)	.498(7)	.513(6)	.46(9)	.642(1)
RAC	.741(4)	.618(5)	.748(2)	.759(3)	.728(6)	.623(8)	.656(7)	.621(9)	.793(1)
Pro	.381(5)	.458(4)	.463(3)	.490(2)	.377(6)	.365(7)	.362(8)	.360(9)	.507(1)
Lun	.614(4)	.587(5)	.629(3)	.656(2)	.564(6)	.524(7)	.487(8)	.463(9)	.678(1)
Pan	.432(4)	.412(5)	.442(3)	.519(2)	.375(6)	.343(9)	.351(8)	.366(7)	.526(1)
Ova	.589(5)	.613(4)	.642(2)	.636(3)	.512(9)	.574(6)	.553(7)	.532(8)	.669(1)
Avg. Rank	4.73	3.73	2.73	2.55	6.9	7.36	7.45	8.18	1.18

FES optimization algorithms [5], [6], [12], [19], [20], [44], [46] in the literature. Furthermore, to validation, the diagnostic capability of the designed expert systems, the true positive rate (TPR) is analyzed against the false positive rate (FPR) using the receiver operating characteristics (ROC) curve. Also, the area under the ROC curve (AUC) is compared for all the approaches.

Table 6 presents the result-summary of Wilcoxon's signed-rank test for all the genes-selection approaches. R^+ indicates ranks in the datasets where the first method is superior to the second one. Whereas, R^- indicates ranks with a conflicting outcome. The null hypothesis (H) associated with Wilcoxon's test was rejected (notified as R^j), since $\mathcal{P} < \alpha = 0.01$ in all cases is in the favor of MFI due to the difference between R^+ and R^-. Consequently, the newly formed expressions in the fuzzy approximation space—that are used to calculate the relevance and redundancy values—seemingly improved all the metrics, importantly the CPU time, when compared against the information theory-based genes-selection approaches. However, when compared to other methods, the proposed MFI performs slightly worse, particularly when compared to RF and GTA in terms of number of genes. Furthermore, the proposed MFI process has the same accuracy as MBF and SW. As a result, the resulting accepted hypothesis (represented as R^a). Although all genes-selection methods strive for the same goal of effective dimension reduction, the proposed MFI algorithm consumes less CPU time than any other method due to less overhead and compounding operations.

Table 7 presents the comparison of Wilcoxon's signed-rank test results for all related FES optimization algorithms and the proposed CGPSO. Wilcoxon's signed-rank test demonstrated that using a conditional statement with the PSO velocity updating process improves interpretability significantly more than GA, PSO, GSA, and ABA. Furthermore, random velocity updating quickly adjusts the MF and arranges the rules set, resulting in a simple and effective classifier. The CGPSO, on the contrary, generates more rules than bPSO and kFIS while having less interpretability than APSO and MPSO. Nevertheless, our proposed CGPSO is competitively faster than bPSO, kFIS, APSO, and MPSO.

Finally, the diagnostic test is validated using the ROC curve of the TPR against the FPR at diverse cut points. Figure 16 shows the plotted ROC curve for all the datasets used in this experiment. The ROC curve for the proposed method was closer to the upper left corner for all datasets, indicating that the proposed method has a higher sensitivity/specificity rate that is useful for a diagnostic-based decision support system. Furthermore, in Table 8, the AUC produced by our system for each dataset is compared to state-of-the-art systems. Despite the good sensitivity and specificity potentials of the proposed method, it produced slightly lower AUC values—although the difference is negligible—for the Leu and Bre datasets when compared with the GSA and PSO. Still, our approach demonstrated an ability to cover a wide area of nine datasets without varying significantly.

Overall, the refinement strength of the nominated genes and their linguistics provided by the proposed method is very effective for detecting disease. Furthermore, all statistical tests confirmed the proposed method's ability to generate an accurate, faster, and more interpretable FES for analyzing microarray data to diagnose diseases.

VI. CONCLUSION

Understanding big microarray data and designing an accurate expert system for disease diagnostic in a reasonable time, while meeting the accuracy-interpretability-speedy tradeoff, is one of the major challenges in bioinformatics. To meet this challenge, this paper proposed af-information modification by combining a fuzzy and rough set to identify relevant genes in large amounts of microarray data. Furthermore, to design the best fuzzy expert system, we propose a CGPSO for faster knowledge acquisition, in which the velocity is adjusted based on a predefined update probability, resulting in a faster search. To accelerate the implementation of the proposed method, a high-performance computing architecture based on the Hadoop ecosystem that efficiently handles microarray data is proposed. In extensive experiments with eleven datasets, the proposed MFI-CGPSO-FES approach with very few tunable parameters and complicated operations achieved a reasonably speedy tradeoff, and successfully handle the FES accuracy-interpretability conflict.

REFERENCES

[1] S. Jauhari and S. Rizvi, "Mining gene expression data focusing cancer therapeutics: A digest," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 3, pp. 533–547, May 2014.

[2] A. Kent, "We can change the future: Is genetic testing a powerful tool for determining the health prospects of our children?" *EMBO Rep.*, vol. 6, no. 9, pp. 801–804, Sep. 2005.

[3] O. Trifonova, V. I. In, E. Kolker, and A. Lisitsa, "Big data in biology and medicine," *Acta Naturae*, vol. 5, no. 3, p. 18, 2013.

[4] C. H. Lee and H.-J. Yoon, "Medical big data: Promise and challenges," *Kidney Res. Clin. Pract.*, vol. 36, no. 1, pp. 3–11, Mar. 2017.

[5] P. G. Kumar, T. A. A. Victoire, P. Renukadevi, and D. Devaraj, "Design of fuzzy expert system for microarray data classification using a novel genetic swarm algorithm," *Expert Syst. Appl.*, vol. 39, no. 2, pp. 1811–1821, Feb. 2012.

[6] P. GaneshKumar, C. Rani, D. Devaraj, and T. A. A. Victoire, "Hybrid ant bee algorithm for fuzzy expert system based sample classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 2, pp. 347–360, Mar. 2014.

[7] P. Maji, "f-information measures for efficient selection of discriminative genes from microarray data," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1063–1069, Apr. 2008.

[8] S. Foithong, O. Pinngern, and B. Attachoo, "Feature subset selection wrapper based on mutual information and rough sets," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 574–584, Jan. 2012.

[9] N. Kwak and C.-H. Choi, "Input feature selection by mutual information based on Parzen window," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1667–1671, Dec. 2002.

[10] P. G. Kumar, C. Rani, D. Mahibha, and T. A. A. Victoire, "Fuzzy–rough–neural–based f–information for gene selection and sample classification," *Int. J. Data Mining Bioinf.*, vol. 11, no. 1, pp. 31–52, 2015.

[11] P. Maji and S. K. Pal, "Fuzzy–rough sets for information measures and selection of relevant genes from microarray data," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 40, no. 3, pp. 741–752, Jun. 2010.

[12] S. A. A. Vijay and P. GaneshKumar, "Fuzzy system for classification of microarray data using a hybrid ant stem optimisation algorithm," *Int. J. Adv. Intell. Paradigms*, vol. 25, no. 1, pp. 154–170, 2021.

[13] Y. Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.

[14] Y.-C. Chang, Y.-L. Chen, Y. Xu, C.-H. Hsieh, C.-C. Chueh, Y.-T. Huang, and C.-T. Hsieh, "Particle swarm optimization with considering more locally best particles and Gaussian jumps," in *Proc. 10th Int. Conf. Natural Comput. (ICNC)*, Aug. 2014, pp. 285–290.

[15] X. Chen and Y. Li, "A modified PSO structure resulting in high exploration ability with convergence guaranteed," *IEEE Trans. Syst., Man, B, Cybern.*, vol. 37, no. 5, pp. 1271–1289, Oct. 2007.

[16] M. Kumar and S. K. Rath, "Classification of microarray data using kernel fuzzy inference system," *Int. Scholarly Res. Notices*, vol. 2014, pp. 1–18, Aug. 2014.

[17] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*, vol. 2, pp. 652–687, 2014.

[18] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[19] S. Helwig, F. Neumann, and R. Wanka, "Particle swarm optimization with velocity adaptation," in *Proc. Int. Conf. Adapt. Intell. Syst.*, Sep. 2009, pp. 146–151.

[20] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," *Appl. Math. Comput.*, vol. 189, no. 2, pp. 1205–1213, 2007.

[21] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets Syst.*, vol. 28, no. 1, pp. 15–33, Oct. 1988.

[22] H. Wickham and G. Grolemund, *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. Sebastopol, CA, USA: O'Reilly Media, 2016.

[23] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levin, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Nat. Acad. Sci. USA*, vol. 96, no. 12, pp. 6745–6750, 1999.

[24] C. M. Perou, S. S. Jeffrey, M. van de Rijn, C. A. Rees, M. B. Eisen, D. T. Ross, A. Pergamenschikov, C. F. Williams, S. X. Zhu, J. C. F. Lee, D. Lashkari, D. Shalon, P. O. Brown, and D. Botstein, "Distinctive gene expression patterns in human mammary epithelial cells and breast cancers," *Proc. Nat. Acad. Sci. USA*, vol. 96, no. 16, pp. 9212–9217, Aug. 1999.

[25] A. A. Alizadeh et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, Feb. 2000.

[26] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, and M. A. Caligiuri, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[27] V. H. Teixeira, R. Olaso, M.-L. Martin-Magniette, S. Lasbleiz, L. Jacq, C. R. Oliveira, P. Hilliquin, I. Gut, F. Cornelis, and E. Petit-Teixeira, "Transcriptome analysis describing new immunity and defense genes in peripheral blood mononuclear cells of rheumatoid arthritis patients," *PLoS ONE*, vol. 4, no. 8, p. e6803, Aug. 2009.

[28] V. K. Mootha et al., "PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes," *Nature Genet.*, vol. 34, no. 3, pp. 267–273, 2003.

[29] T. C. T. M. van der Pouw Kraan, F. A. van Gaalen, P. V. Kasperkovitz, N. L. Verbeet, T. J. M. Smeets, M. C. Kraan, M. Fero, P.-P. Tak, T. W. J. Huizinga, E. Pieterman, F. C. Breedveld, A. A. Alizadeh, and C. L. Verweij, "Rheumatoid arthritis is a heterogeneous disease: Evidence for differences in the activation of the STAT-1 pathway between rheumatoid tissues," *Arthritis Rheumatism*, vol. 48, no. 8, pp. 2132–2145, Aug. 2003.

[30] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, and J. P. Richie, "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.

[31] G. J. Gordon, R. V. Jensen, L.-L. Hsiao, S. R. Gullans, J. E. Blumenstock, S. Ramaswamy, W. G. Richards, D. J. Sugarbaker, and R. Bueno, "Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma," *Cancer Res.*, vol. 62, pp. 4963–4967, Sep. 2002.

[32] National Center for Biotechnology Information. (2015). *Pancreatic Cancer Dataset*. [Online]. Available: http://www.ncbi.nlm.nih.gov/

[33] E. F. Petricoin, A. M. Ardekani, B. A. Hitt, P. J. Levine, V. A. Fusaro, S. M. Steinberg, G. B. Mills, C. Simone, D. A. Fishman, and E. C. Kohn, "Use of proteomic patterns in serum to identify ovarian cancer," *Lancet*, vol. 359, no. 9306, pp. 572–577, Feb. 2002.

[34] P. G. Kumar and T. A. A. Victoire, "Multistage mutual information for informative gene selection," *J. Biol. Syst.*, vol. 19, no. 4, pp. 725–746, 2011.

[35] N. Hoque, D. Bhattacharyya, and J. K. Kalita, "MIFS-ND: A mutual information-based feature selection method," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6371–6385, 2014.

[36] A. Tsimpiris, I. Vlachos, and D. Kugiumtzis, "Nearest neighbor estimate of conditional mutual information in feature selection," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12697–12708, Nov. 2012.

[37] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *J. Bio. Comput. Biol.*, vol. 3, no. 2, pp. 185–205, 2005.

[38] K. E. Hild, D. Erdogmus, K. Torkkola, and J. C. Principe, "Feature extraction using information-theoretic learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1385–1392, Sep. 2006.

[39] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[40] R. Díaz-Uriarte and S. A. De Andrés, "Gene selection and classification of microarray data using random forest," *BMC Bioinf.*, vol. 7, no. 1, p. 3, Dec. 2006.

[41] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," in *Proc. ICML*, vol. 1, 2001, pp. 601–608.

[42] L. Yu, Y. Han, and M. E. Berens, "Stable gene selection from microarray data via sample weighting," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 9, no. 1, pp. 262–272, Jan. 2012.

[43] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "Hierarchical gene selection and genetic fuzzy system for cancer microarray data classification," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0120364.

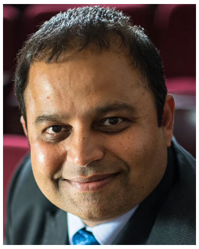
[44] M. Mandal and A. Mukhopadhyay, "A graph-theoretic approach for identifying non-redundant and relevant gene markers from microarray data using multiobjective binary PSO," *PLoS ONE*, vol. 9, no. 3, Mar. 2014, Art. no. e90949.

[45] A. Mukhopadhyay and M. Mandal, "Identifying non-redundant gene markers from microarray data: A multiobjective variable length PSO-based approach," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 6, pp. 1170–1183, Nov. 2014.

[46] P. G. Kumar and D. Devaraj, "Fuzzy classifier design using modified genetic algorithm," *Int. J. Comput. Intell. Syst.*, vol. 3, no. 3, pp. 334–342, Sep. 2010.

GANESHKUMAR PUGALENDHI (Senior Member, IEEE) received the B.Tech. degree in information technology from the University of Madras, in 2003, the M.S. (by research) degree in information technology from Anna University, Chennai, in 2008, and the Ph.D. degree in information technology from Anna University, Coimbatore, in 2012. From April 2015 to March 2016, he was a Postdoctoral Researcher with the School of Computer Science and Engineering, Kyungpook National University, South Korea. He is currently an Assistant Professor with the Department of Information Technology, Anna University Regional Campus, Coimbatore. His research interests include the application of soft computing techniques for data mining-based problems in bioinformatics, smart grid, and big data analytics. He was a recipient of the Student Scientist Award from TNSCST, in 2003, the IEEE Best Paper Award, in 2007, the IET Best Paper Award, in 2011, and the KIISE Best Paper Award, in 2015. He was the Track Chair of smart human–computer interaction at ACM SAC 2018.

M. MAZHAR RATHORE (Member, IEEE) received the master's degree in computer science and communication security from the National University of Sciences and Technology, Pakistan, in 2012, and the Ph.D. degree in computer science and engineering from Kyungpook National University, South Korea, in 2018. He is currently a Postdoctoral Researcher with the University of New Brunswick, Canada. His research interests include big data analytics, the Internet of Things, smart systems, network traffic analysis and monitoring, remote sensing, smart cities, urban planning, intrusion detection, and information security and privacy. He is a Professional Member of the IEEE and the ACM. For his paper "IoT-Based Smart City Development Using Big Data Analytical Approach," he received the Best Project/Paper Award at the Qualcomm Innovation Award 2016 at Kyungpook National University, South Korea. He was also the 2015 IEEE Communications Society Student Competition Best Project Award Nominee for his project "IoT-Based Smart City." He is serving as a reviewer for various reputed IEEE, ACM, Springer, and Elsevier journals.



DHIRENDRA SHUKLA is currently a Professor and the Dr. J. Herbert Smith ACOA Chair of technology management and entrepreneurship with the University of New Brunswick (UNB), Canada. He uses his telecom industry expertise and extensive academic background in entrepreneurial finance, business administration, and engineering to promote a bright future for New Brunswick. UNB's 2014 Award from Startup Canada as the "Most Entrepreneurial Post-Secondary Institution of the Year," his nomination as a Finalist for the Industry Champion by KIRA, and his nomination as a Finalist for the Progress Media's Innovation in Practice Award are all evidence of his tireless efforts and vision. He was nominated for the RBC Top 25 Canadian Immigrant Award and selected by the panel of judges as a top 75 finalist. Most recently, he received the Entrepreneur Promotion Award by Startup Canada, in 2017, as well as the Outstanding Educator Award by the Association of Professional Engineers and Geoscientists of New Brunswick, in 2018.

ANAND PAUL received the Ph.D. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2010. He is currently an Associate Professor with the School of Computer Science and Engineering, Kyungpook National University, South Korea. He is a delegate representing South Korea for the M2M focus group and MPEG. His research interests include algorithm and architecture reconfigurable embedded computing. He was the Track Chair of smart human–computer interaction at ACM SAC 2014 and 2015. He has guest-edited various international journals and he is also part of the Editorial Team of *Journal of Platform Technology*, *ACM Applied Computing Review*, and *Cyber-Physical Systems*.