IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 May 2022, accepted 30 June 2022, date of publication 5 July 2022, date of current version 13 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3188689

RESEARCH ARTICLE

Hybrid Knowledge Extraction Framework Using Modified Adaptive Genetic Algorithm and BPNN

YUN OU[1], SHAO-QIANG YE[1], LEI DING[1], KAI-QING ZHOU[1], AND AZLAN MOHD ZAIN[2], (Member, IEEE)

[1]College of Information Science and Engineering, Jishou University, Jishou, Hunan 416000, China
[2]Big Data Centre, Universiti Teknologi Malaysia, Skudai, Johor 80310, Malaysia

Corresponding author: Yun Ou (yun_ou@jsu.edu.cn)

This work was supported in part by the Research Foundation of Education Bureau of Hunan Province, China, under Grant 21C0383; in part by the National Natural Science Foundation of China, under Grant 62066016 and Grant 61966014; and in part by the Natural Science Foundation of Hunan Province, China, under Grant 2020JJ5458.

ABSTRACT Fault diagnosis based on the expert system (ES) is still a research topic of manufacturing in Industry 4.0 because of the stronger interpretability. As the core component of the ES, fault diagnosis accuracy is positively correlated to the precision of the knowledge base. But it is difficult for users to understand the knowledge obtained from the original dataset utilizing the existing knowledge extraction method. Therefore, it is of great significance to extract easy-to-understand and exact rules from the NN framework. This paper proposes a hybrid extraction framework to perform the rule extraction for overcoming this drawback. First, an improved adaptive genetic algorithm (GA) using a logistic function, namely LAGA, is proposed to solve the traditional GA's insufficient prediction performance issue. Compared with the other three mainstream adaptive GAs, the experiment results of optimizing six selected test functions by these GA variants show that the LAGA algorithm's convergence accuracy and speed have been greatly improved, especially for high latitude functions. On this basis, a rule extraction method based on the symbol rule and NN, namely the LAGA-BP framework, is discussed in this manuscript to classify the real-valued attributes. This framework obtains hidden knowledge (knowledge refinement process) by NN and further transforms the acquired hidden knowledge into more easy-to-understand rule knowledge (rule extraction process). The execution of the LAGA-BP framework could be separated into two phases. The first phase is to optimize a back propagation NN (BPNN) using the LAGA and refine prediction classification knowledge over the optimized BPNN. In the second phase, an attribute reduction algorithm using multi-layered NN (SD algorithm) based on two different superposed networks is used in this framework to reduce data-set attributes and then uses the K-means clustering algorithm to extract the if-then rule from the simplified attributes. The Wisconsin breast cancer dataset is used as a case study to reveal the correctness and robustness of the proposed LAGA-BP method. Consulting relevant medical personnel and referencing relevant data shows that the rules extracted using this method help verify the diagnosis results, thus verifying the proposed framework's feasibility and practicality.

INDEX TERMS Adaptive genetic algorithm, back propagation neural network, attribute reduction, classification prediction, knowledge extraction.

I. INTRODUCTION

In the context of Industry 4.0, an essential topic in the manufacturing area is to detect the fault diagnosis and recover the current status from potential faults as soon as possible for reducing maintenance costs and preventing unscheduled downtime [1]. Although various fault diagnosis approaches are proposed in recent years from different viewpoints and achieved fruitful results, which are knowledge-driven, data-driven, and value-driven methods [2]–[4]. Among them, the

The associate editor coordinating the review of this manuscript and approving it for publication was Zhan-Li Sun.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

knowledge-driven approach, which depends on expert experience, is still widely applied in various areas because of the stronger interpretability [5]–[7].

The basis of the knowledge-driven diagnosis fault is the expert system (ES). ES could be considered as an intelligent system to analyze and solve a specific problem using expert-level knowledge and experience in past decades. Being the basis of the knowledge base of ES, fuzzy production rules (FPRs) have been widely manipulated to represent expert knowledge with vague factors in an ES utilizing an 'IF–THEN' configuration to implement the pointed reasoning tasks for capturing data and obtaining the possible diagnosis outcome [8]. Although the ES can help people reduce the burden of learning in some specific fields, a series of drawbacks such as knowledge source, knowledge representation, and knowledge extraction efficiency seriously hamper the development of the expert system. For example, the fault diagnosis accuracy is positively correlated to the precise of the extracted FPRs from the original dataset. Therefore, the knowledge extraction operation is the key to building an expert system and one of the ''bottleneck'' problems that the expert system needs to solve.

As the core process of the expert system, knowledge extraction is one of the most crucial parts of this module. The existing knowledge extraction methods could be classified into the following four aspects: knowledge extraction combining NN and symbol rule, knowledge extraction combining NN and soft computing, classifier-based knowledge extraction, and system-or-model-based knowledge extraction.

Knowledge extraction combining NN and symbol rule mainly uses a particular rule to represent the learning structure of NNs to enhance its understanding ability and realize the rules (knowledge) extraction of NNs. Gallant [9] constructed an if-then symbol rule by discussing the relationship between the connectionist model and the analysis paradigm, a prototype of the NN rule extraction method. Sestito and Dillon [10] proposed an attribute reduction algorithm using multi-layered NN (SD algorithm) to extract if-then rules. The SD algorithm is helpful for feed-forward NN, but it needs a high implementation cost. Setiono [11] proposed an If (m of a set of conditions) then conclusion' (M-of-N) algorithm for extracting M-of-N rules from feed-forward networks. The algorithm only needs to train and trim the network, and hidden layer neuron activation values are clustered, and the conditional part of obtained classification rules is replaced with M-of-N conditions to obtain M-of-N rules. Policicchio *et al.* [12] extended negation and disjunction functions of the M-of-N rule and proposed a task-specific GA based on the expanded M-of-N$\{\neg, \vee\}$ rule. The experimental results show that the proposed method can be better extended to realistic large domains than C4.5 and Ripper. Veronica and Christine [13] proposed and compared two versions of NN rule extraction models (Piece-Wise Linear Artificial Neural Network (PWL-ANN) and enhanced Piece-Wise Linear Artificial Neural Network (enhanced PWL-AN)). Among them, the enhanced PWL-ANN model can generate comprehensible if-then rules about the relative problem domain.

The knowledge extraction method combining NN and soft computing uses various soft computing techniques to optimize NN and extract better rules (knowledge) from NN. Hruschka and Ebecken [14] used a GA to cluster the neurons' activation values in NN's hidden layer and analyzed the relationship between input and output neurons to generate conjunction rules. Ding [15] successfully combined rough set theory and NN to simplify attributes of a dataset and then used the simplified attributes as the input of NN for rule extraction. Zhang *et al.* [16] used a clustering algorithm to improve NN for rule extraction. They were clustering the nodes' output values in the network's hidden layer, and all extracted symbol rules with high accuracy and good comprehensibility. Marghny [17] obtained a simple network structure of neither comprehensible form nor black box by the constructive method and then used GA to identify the input variable set to produce the required output values to extract rules. Compared with the standard classifier, this method has higher accuracy.

Classifier-based knowledge extraction makes the extraction of rules (knowledge) no longer limited to relying on NN. It also can use support vector machine (SVM) and other types of classifiers to replace the role of NN in the process of knowledge extraction. Singh *et al.* [18] proposed a hybrid method to extract rules from the support vector machine, introduced the feature selection mechanism, and transformed SVM black box model into an easy-to-understand model with the XGBoost tool, which solved a problem that the existing rule-based models could not generate coherent ruleset. Wang *et al.* [19] used the random forest to build many decision tree models to generate a large number of available decision rules, then designed an improved rule extraction method based on a random forest to separate decision rules from the trained tree, and finally used the improved multi-objective evolutionary algorithm to find optimal rules. This method can derive accurate and interpretable classification rules from the decision tree set.

Because of its good fault tolerance and scalability and robust functionality, system-or-model-based knowledge extraction is more comfortable to meet real-life knowledge acquisition needs. It makes the research object of rule (knowledge) extraction transform from a simple NN or classifier to a complex learning system or model. Benitez *et al.* [20] proved that multi-layer feed-forward NN is equivalent to a fuzzy system so that NN can be interpreted by its corresponding fuzzy rules of the fuzzy system. Singh and Biswas [21] proposed a fuzzy neural system for feature selection and classification. It can determine a large number of linguistic features for each input and use the importance of input features and the certainty of rules to extract understandable rules easily. For the classification of hydrology, Vidyarthi and Jain [22] use the input-output relationship of the trained NN model and introduce decision tree to extract classification rules from the

model. The knowledge of hydrology extracted by this method is practical and straightforward.

Since NNs have many advantages to meet the needs of rule extraction, hybrid knowledge extraction methods by combining NN and other algorithms are likely to be still the hot spots of the knowledge extraction domain. In the past decades, researchers have been trying to improve the intelligibility of trained NNs through knowledge extraction to improve the reliability and acceptability of the decision-makers for operational purposes. Inspired by similar thinking, this paper proposes a knowledge extraction framework based on if-then rules, NNs, and clustering. First, an improved adaptive genetic algorithm using a logistic function, namely LAGA, is used to optimize back propagation NN (BPNN) to achieve knowledge refinement of a dataset. The SD algorithm of double-layer NN is used to achieve attribute reduction, and finally, the K-means clustering algorithm is used to extract the rules of a dataset. It solves the ''black box'' problem of NNs and realizes the effective extraction of dataset knowledge.

The highlights of this manuscript could be classified into the following aspects.

Firstly, an improved adaptive GA based on the logistic function (LAGA) is proposed and compared with the other three improved adaptive genetic algorithms to demonstrate its preponderance.

Next, a framework of knowledge acquisition based on the if-then rule and LAGA-BPNN is proposed, and its design idea and implementation method are discussed.

Finally, the proposed framework's feasibility and effectiveness are verified with the Wisconsin Breast Cancer Data Set as an example.

This paper is divided into six sections, and its organizational structure is as follows. Section 2 recalls the concepts and related algorithms of knowledge extraction briefly. Section 3 introduces the proposed LAGA in detail and verifies the robustness of LAGA by testing some classical functions. Section 4 discusses the proposed knowledge extraction framework using LAGA and BPNN and explains each framework's stage. Section 5 simulates the knowledge extraction process using the proposed framework and verifies the proposed framework's effectiveness. Finally, section 6 summarizes the whole manuscript and reveals future research work.

II. RELATED NOTIONS AND ALGORITHMS

As a core part of the expert system design and realization, knowledge extraction is also called the ''bottleneck'' problem of the expert system because of its cumbersome process and low efficiency. There has not found a unified and effective method of knowledge acquisition yet. The early knowledge extract methods are that experts dictate their accumulated practical experience and let knowledge engineers enter it into the expert knowledge base. However, some experiences and strategies are difficult to express directly in the language, and expert knowledge is unconscious, so it is challenging to obtain expert knowledge. Therefore, the method of knowledge extraction cannot merely transform the knowledge representation form. It should be to establish a model for solving a specific professional problem. At present, the leading knowledge extraction methods are based on rules, NNs, classifiers, and others [23], [24]. The related notions in this manuscript are demonstrated briefly below.

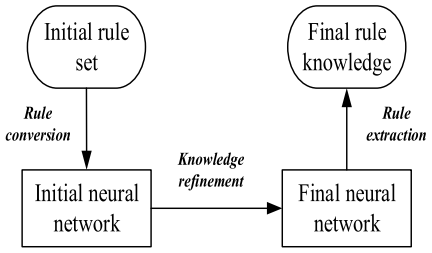


FIGURE 1. The general process of knowledge extraction using NN.

A. BPNN AND HEBB LEARNING RULES

Because of the vital multi-dimensional function mapping function, NN is used to solve many nonlinear related problems such as prediction classification, function fitting, and pattern recognition. The Hebb rule [25] realizes network learning by judging the activation signal between input and output neurons to change the connection weight.

BPNN uses an error back-propagation algorithm to realize network learning and training. By learning and training the network through the BP algorithm, a set of connection weights with an implicit knowledge and implicit production rules could be obtained to simulate and predict some unknown samples [26].

B. KNOWLEDGE EXTRACTIONS BY USING NN

A general knowledge extraction framework using NN can divided into three parts: knowledge transfer, sample training, and rule extraction. The knowledge extraction process is represented as shown in Figure 1.

According to Fig.1, it easily found that the first step of knowledge extraction using NN is to generate a NN structure from the corresponding knowledge base, then convert hidden knowledge in the neural network into explicit rules, and finally generate and acquire explicit rules (knowledge).

Compared with the traditional knowledge extraction method knowledge extraction by NN uses machine learning algorithms to accumulate knowledge and then operate the knowledge base, so that the time and efficiency of knowledge acquisition have been greatly improved [27].

C. GENETIC ALGORITHM

A genetic algorithm (GA) is a global random search and adaptive optimization method to explore the mechanism of biological heredity and evolution. A better approximate optimal solution could be gained in GA by employing the survival of the fittest principle among a set of potential solutions. GA is one of the general algorithms for solving complex system problems. Due to its unrestricted and robust universal

framework, coupled with its simple principle and easy-to-operate characteristics, GA has become a practical tool for solving problems such as function optimization and combinatorial optimization [28].

III. IMPROVED ADAPTIVE GA USING A LOGISTIC FUNCTION

In the traditional GA, the probability values of crossover and mutation are often constant. Thus, the direct genetic operation has seriously hindered the convergence speed of traditional GA on the population without distinguishing individual characteristics. To overcome this drawback, Srinivas [29] proposed a modified GA (AGA) that adapt changes the probability values of crossover and mutation based on the fitness values of the population. This algorithm can improve the convergence speed and accuracy of genetic evolution and effectively enhance its global search ability and avoid the problem of falling into local optimal solutions. To ensure the diversity of the population and keep the best individuals in the population as possible, the AGA uses $f_{\max} - \bar{f}$ as an index to measure the degree of population convergence, where $f_{\max}$ is the maximum fitness value of the population, and $\bar{f}$ is the average fitness value of the population. Equations (1) and (2) reveal the mathematical mechanism of the AGA operation, which are.

$$P_c = \begin{cases} k_1(f_{\max} - f')/(f_{\max} - \bar{f}), & f' \geq \bar{f} \\ k_3, & f' < \bar{f} \end{cases} \tag{1}$$

$$P_m = \begin{cases} k_2(f_{\max} - f)/(f_{\max} - \bar{f}), & f \geq \bar{f} \\ k_4, & f < \bar{f} \end{cases} \tag{2}$$

where, p_c and p_m must be a value in the range of 0 to 1; f' is the maximum fitness value of the two individuals to be cross-operated; f is the individual fitness value of the mutation to be performed; k_1, k_2, k_3 and k_4 are arbitrary constants less than 1.

It can be seen from Equation (1) and Equation (2) that when the values of f' and f are close to the values of $\bar{f}$, the values of p_c and p_m become very small. This processing method of the AGA will cause the excellent individuals of the population to lose the opportunities for evolution in the early stages of evolution. Therefore, Ren *et al.* [30] proposed an improved AGA (IAGA). The probabilities of crossover and mutation can be obtained by Equations (3) and (4)

$$P_c = \begin{cases} P_{c1} - \dfrac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{\max} - f_{avg}}, & f' \geq f_{avg} \\ P_{c1}, & f' < f_{avg} \end{cases} \tag{3}$$

$$P_m = \begin{cases} P_{m1} - \dfrac{(P_{m1} - P_{m2})(f - f_{avg})}{f_{\max} - f_{avg}}, & f \geq f_{avg} \\ P_{m1}, & f < f_{avg} \end{cases} \tag{4}$$

where $f_{\max}$ is the maximum fitness of the population, and f_{avg} is the average fitness of the population; f' is the maximum fitness value of the two individuals to be cross-operated; f is the individual fitness value of the mutation to be performed.

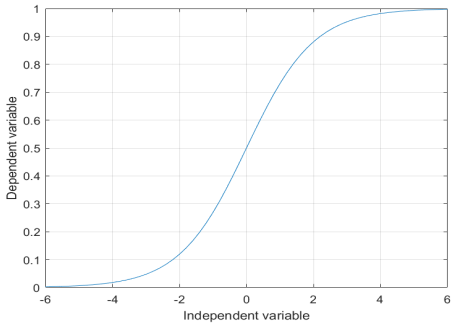


FIGURE 2. The graph of the sigmoid function.

In IAGA, the crossover and mutation probabilities of excellent individuals in the population are appointed as P_{c2} and P_{m2} to evolve all the time, but the convergence speed effect of the algorithm is not very good. To solve this issue, Chen *et al.* [31] adopted an adaptive genetic algorithm (SAGA) based on the sigmoid function (a special form of the logistic function) to solve the slow convergence issue of the IAGA. The graph of the sigmoid function is shown in Figure 2, and the process of approaching 0 and approaching 1 is smooth. As can be seen from Figure 2, the value of the sigmoid function can maintain a smooth change with the change of the independent variable, and the range of the function value is [0, 1], which is helpful for the adaptive construction of crossover and mutation probability, so this function is used to optimize the GA. The mathematical expression of the sigmoid function is shown in Equation (5).

$$y = 1/(1 + e^{-x}) \tag{5}$$

The mathematical description of the SAGA is shown in Equation (6) and Equation (7). In these two functions, $\Delta$is the ''precocity'', which is used to evaluate whether the population is precocious.

$$P_c = \frac{1}{1 + e^{-c_1\Delta}} \tag{6}$$

$$P_m = -\frac{1}{1 + e^{-c_2\Delta}} + 1.0 \tag{7}$$

In the two equations, c_1, c_2 are constants greater than 0; $\Delta = f_{\max} - f_{avg}$, where $f_{\max}$ represents the maximum fitness value of the population, and f_{avg} represents the average fitness value of all individuals in the population whose fitness value is greater than the average fitness value of the population.

A. AN IMPROVED IDEA OF ADAPTIVE GENETIC ALGORITHM

In the several adaptive GAs introduced in the previous section, AGA and IAGA have higher convergence accuracy than SAGA, but the convergence rate of SAGA is faster than AGA and IAGA. Based on this finding, an improved adaptive genetic algorithm based on the logistic function (LAGA) is proposed to weigh the convergence accuracy and speed.

The logistic function is an optimal mathematical model to describe population growth law under the condition of limited

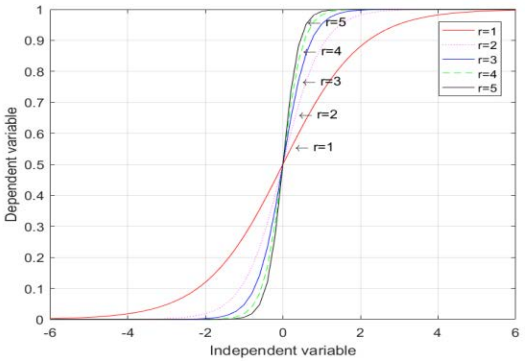


FIGURE 3. The graph of the logistic function with different r value.

resources. Because its graph is S-shaped, it is also called an ''S-shaped function.'' It grows exponentially in the initial stage, then gradually slows down, and finally until the growth stops. The differential form of the logistic function (i.e., the logistic Equation) is

$$\frac{dP}{dt} = rP(1 - \frac{p}{k}) \tag{8}$$

From the Equation (8), the logical function's mathematical expression can be derived as Equation (9).

$$P(t) = \frac{KP_0e^{rt}}{K + P_0(e^{rt} - 1)} \tag{9}$$

where P_0 is the initial value, K is the final value, and r measures how fast the curve changes.

The integral form of the logistic function is

$$P = \frac{K}{1 + e^{a-rt}} \tag{10}$$

If the parameter $K = 1$ and $\alpha = 0$, the graph of the logistic function with different r values is shown in Figure 3.

It can be seen from Figure 3 that the change of the logistic function value is consistent with the adaptive change law of the crossover and mutation probability of the genetic algorithm. Hence, the logistic function is selected to optimize the adaptive genetic algorithm. The LAGA requires that the crossover and mutation probabilities of genetic operations fluctuate within the range of [0.4, 0.9] and [0.01, 0.1] respectively, and the following conditions must be met as the evolutionary algebra increases, the maximum fitness value and average fitness value of population gradually tend to be the same, and the change law of the crossover and mutation probability decreases gradually [32]–[34].

Because of the logistic curve Equation, let $K = 1, \alpha = 0, r = 1, p = y, t = \pm x$, two simplified function equations could be obtained, what are $y = 1/(1+e^{-x})$, $y = 1/(1+e^{x})$, and $1/(1 + e^{-x}) + 1/(1 + e^{x}) = 1$. From this, the mathematical expressions of the crossover and mutation probability changes of the LAGA can be derived: Equation (11) and Equation (12).

$$P_c = \frac{1}{1 + e^{-k_1\Delta f}} - 0.1 \tag{11}$$

$$P_m = \frac{1}{10}(1 - \frac{k}{1 + e^{k_2\Delta f}}) \tag{12}$$

where $\Delta f = f_{\max} - \overline{f}$, $f_{\max}$ is the maximum fitness value of individuals and $\overline{f}$ is the average fitness value of individuals in the population. k is a constant parameter, and $k > 0$; $k_1 > 0$, $k_2 > 0$, are probability adjustment rates.

B. IMPLEMENTATION OF THE LAGA

According to the basic principles and improvement measures of genetic algorithm, this paper uses binary coding to create a population, and then initializes the binary-coded population to a real-valued population, and calculates the target function value of the population, and then converts the target function value of the population into the fitness value of the population using the linear sorting method. Finally, three genetic operators are used for genetic operations until a satisfactory approximate optimal solution is obtained. The selection operation uses a random traversal sampling method, the crossover operation uses a single-point crossover operator, and the mutation operation uses a discrete mutation operator. The implementation steps of the LAGA could be classified into the following steps.

Step 1: Determining the target value of the problem and the fitness value of the population;

Step 2: Creating an initial binary population;

Step 3: Converting the binary-coded population into the real-valued population;

Step 4: Evaluating the individuals according to the fitness value of the population;

Step 5: Selecting individuals as offspring population according to the fitness value of the population;

Step 6: Calculating the crossover probability according to Equation (11), and then performing crossover operations on random individuals in the offspring population to generate a new population according to the fitness value of the population;

Step 7: Calculating the mutation probability according to Equation (12), and then performing mutation the population on random individuals in the old population to generate a new population according to the fitness value of the population;

Step 8: Re-inserting the new population and re-evaluating the performance of chromosomes in the new population;

Step 9: Whether the iteration is reached or not? If yes, output the result. Else, repeat Steps 4 to 8.

C. PERFORMANCE TEST OF THE LAGA

Six standard test functions are selected to evaluate the validity of four adaptive genetic algorithms [35]. Table 1 records the details of the six selected functions. The experimental environment is a 1.90GHz processor and a 64-bit Windows 7 computer with 8GB of running memory. The four algorithms are implemented in the MATLAB R2016a compilation environment, and the parameter settings of each algorithm are shown in Table 2.

To reduce the random impact of the experiment, for each task in this work, the experiment was independently repeated

TABLE 1. Benchmark functions.

function	expression	dim	range	f_{min}
Schaffer（f1）	$f_1(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	2	[-100,100]	0
Shubert(f2)	$f_2(x) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$	2	[-10,10]	-186.7309
Dejong's(f3)	$f_3(x) = \sum_{i=1}^{d} x_i^2$	20	[-512,512]	0
Rosenbrock's(f4)	$f_4(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	2	[-10,10]	0
Rastrigins(f5)	$f_5(x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$	20	[-5.12,5.12]	0
Griewangk's(f6)	$f_6(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos(\frac{x_i}{\sqrt{i}}) + 1$	10	[-600,600]	0

TABLE 2. Parameter's design of four modified GAs.

Method	Parameter
LAGA	*NIND=40, MAXGEN=500, k=9/5, k_1=1, k_2=1*
AGA	*NIND=40, MAXGEN=500, k_1=0.6, k_2=0.01, k_3=0.9, k_4=0.1*
IAGA	*NIND=40, MAXGEN=500, P_{C1}=0.9, P_{C2}=0.6, P_{m1}=0.1, P_{m2}=0.01*
SAGA	*NIND=40, MAXGEN=500, k=9/5, c_1=1, c_2=1*

20 times for calculating the mean of results. Table 3 shows the comparative experimental results of four adaptive genetic algorithms. ''Best'', ''Worst'', ''Mean'' and ''Std'' represent the best, worst, mean, and overall variance of the optimization results of the four methods in the six test functions [36], [37].

The following are a few pictures of the optimal solution's evolution process in one of 20 repeated experiments, including four algorithms and six test functions, as shown in Figure 4.

From Table 3 and Figure 6, the performance of the presented LAGA is superior to the other three improved algorithms in optimizing the high-dimensional function extremums. However, in optimizing the low-level function extreme value, the LAGA and the other three improved algorithms have their strengths and weaknesses.

Relative to the other three improved algorithms (AGA, IAGA, and SAGA), LAGA improves the optimal solution's accuracy and dramatically improves the convergence speed of evolution. Experimental results show that the LAGA optimized the BPNN because LAGA improved the NN's weights and thresholds.

IV. KNOWLEDGE EXTRACTION FRAMEWORK USING LAGA-BPNN

A novel knowledge extraction by combining BPNN with LAGA, namely LAGA-BP framework, is proposed to obtain the fuzzy production rules from the dataset. The entire extraction process could be divided into two parts. One is to obtain hidden knowledge using NN; the second is to convert the acquired hidden knowledge into more understandable knowledge (rule). In the former module, the discussed LAGA in the previous section is utilized to optimize the weights and thresholds of BPNN for improving the accuracy of prediction classification of the known dataset and obtaining more precise implicit knowledge. In the later module, it is divided into two sub-processes. First, the SD algorithm extracts rules from dataset attributes (Attribute reduction) to obtain the causal relationship between input attributes and output attributes. It is a prototype of the IF-THEN rule and the prerequisite for successfully extracting attribute value rules in the following process. The second is to use the K-means clustering algorithm to achieve rules extraction of dataset attribute values (Rules extraction), and then easy-to-understand IF-THEN rules can be obtained.

A. STRUCTURE OF LAGA-BP FRAMEWORK

The LAGA-BP framework structure mainly includes acquiring and preprocessing the dataset, optimizing BPNN by LAGA, knowledge refinement of neural network (predictive classification), rules extraction of sample attribute relation (attributes reduction), and rules extraction of the sample attribute value, etc.

TABLE 3. Experimental results of six function optimization using four GA variants.

Function	*Method*	*Best*	*Worst*	*Mean*	*Std*	*Optimal*
f_1	LAGA (ours)	**1.819E-11**	1.723E-05	4.030E-06	6.649E-06	0
	AGA	**1.819E-11**	1.167E-05	3.152E-06	4.490E-06	
	IAGA	**1.819E-11**	7.996E-06	**2.085E-06**	**3.101E-06**	
	SAGA	3.965E-05	**1.987E-04**	1.028E-04	6.387E-05	
f_2	LAGA (ours)	**-1.86731E+02**	-1.867E+02	**-1.867E+02**	**0**	-1.867E+02
	AGA	**-1.86731E+02**	-1.867E+02	**-1.867E+02**	**0**	
	IAGA	**-1.86731E+02**	-1.867E+02	**-1.867E+02**	**0**	
	SAGA	-1.86728E+02	**-1.865E+02**	-1.867E+02	9.491E-02	
f_3	LAGA (ours)	**8.962E+03**	1.859E+04	**1.359E+04**	**3.176E+03**	0
	AGA	9.179E+03	2.172E+04	1.584E+04	4.944E+03	
	IAGA	3.627E+04	6.171E+04	4.708E+04	9.109E+03	
	SAGA	2.645E+05	**4.015E+05**	3.124E+05	4.939E+04	
f_4	LAGA (ours)	**4.114E-09**	7.814E-04	1.767E-04	3.032E-04	0
	AGA	4.114E-07	8.151E-04	2.031E-04	3.089E-04	
	IAGA	1.702E-05	6.720E-04	**1.593E-04**	**2.566E-04**	
	SAGA	1.348E-03	**3.635E-02**	9.177E-03	1.360E-02	
f_5	LAGA (ours)	**5.490E+01**	6.934E+01	5.963E+01	5.267E+00	0
	AGA	5.493E+01	6.808E+01	6.491E+01	5.029E+00	
	IAGA	7.440E+01	1.041E+02	**8.493E+01**	**1.082E+01**	
	SAGA	1.635E+02	**1.851E+02**	1.779E+02	7.785E+00	
f_6	LAGA (ours)	**6.467E-01**	8.872E-01	**8.146E-01**	**8.743E-02**	0
	AGA	1.025E+00	1.176E+00	1.064E+00	5.647E-02	
	IAGA	1.108E+00	1.614E+00	1.371E+00	1.706E-01	
	SAGA	1.543E+01	**2.230E+01**	1.887E+01	2.878E+00	

TABLE 4. Parameters of BPNN.

Input layer nodes	Hidden layer nodes	Output layer nodes	Iterations	Training accuracy	Learning rate
10	25	1	500	0.01	0.1

TABLE 5. Parameters of GA.

Individuals	Genetic algebra	Binary digits	Select generation gap	Cross probability	Mutation probability
40	20	20	0.95	0.7	0.01

TABLE 6. Classification accuracy of test-set samples.

Number / Result	1	2	3	4	5
BP	0.8029	0.9781	0.9635	0.8686	0.9416
GA+BP	0.9781	0.9781	0.9489	0.9781	0.9708
LAGA+BP	0.9781	0.9854	0.9781	0.9854	0.9781

The discussed structure is completely shown in Figure 5.

The LAGA is used to optimize the initial weights and thresholds of given BPNN to realize the dataset classification. The proposed method is mainly divided into three parts: the determination of network structure, the optimization of the network, and the network classification. The flow chart

TABLE 7. Parameters of single layer network.

Input layer nodes	Output layer nodes	Transfer function	Iterations	Training accuracy	Learning rate
11	1	logsig	1000	0.35	0.1

TABLE 8. Values of SSE'_{ab}.

SSE'_{ab}	1	2	3	4	5	6	7	8	9	10
1	1.74E+01	1.98E+01	2.30E+01	2.11E+01	2.40E+01	2.70E+01	2.53E+01	2.06E+01	2.06E+01	2.43E+01

TABLE 9. Values of $Weight_{ab}$.

$Weight_{ab}$	1	2	3	4	5	6	7	8	9	10	11
1	7.35E+10	3.03E+05	2.15E+05	2.20E+05	1.93E+05	2.21E+05	2.42E+05	2.35E+05	1.96E+05	1.10E+05	2.39E+04

TABLE 10. Values of $Product_{ab}$.

$Product_{ab}$	1	2	3	4	5	6	7	8	9	10	11
1	1.28E+12	5.27E+06	3.74E+06	3.82E+06	3.36E+06	3.84E+06	4.21E+06	4.09E+06	3.41E+06	1.90E+06	4.15E+05
2	1.46E+12	6.01E+06	4.26E+06	4.35E+06	3.83E+06	4.38E+06	4.80E+06	4.66E+06	3.88E+06	2.17E+06	4.74E+05
3	1.69E+12	6.97E+06	4.94E+06	5.04E+06	4.44E+06	5.07E+06	5.56E+06	5.41E+06	4.50E+06	2.52E+06	5.49E+05
4	1.55E+12	6.39E+06	4.53E+06	4.63E+06	4.07E+06	4.65E+06	5.10E+06	4.96E+06	4.13E+06	2.31E+06	5.04E+05
5	1.76E+12	7.27E+06	5.16E+06	5.26E+06	4.63E+06	5.30E+06	5.80E+06	5.64E+06	4.70E+06	2.62E+06	5.73E+05
6	1.99E+12	8.19E+06	5.81E+06	5.93E+06	5.22E+06	5.97E+06	6.54E+06	6.35E+06	5.29E+06	2.96E+06	6.45E+05
7	1.86E+12	7.66E+06	5.44E+06	5.55E+06	4.88E+06	5.58E+06	6.12E+06	5.94E+06	4.95E+06	2.77E+06	6.04E+05
8	1.51E+12	6.25E+06	4.43E+06	4.52E+06	3.98E+06	4.55E+06	4.99E+06	4.85E+06	4.04E+06	2.26E+06	4.92E+05
9	1.51E+12	6.24E+06	4.42E+06	4.51E+06	3.97E+06	4.54E+06	4.98E+06	4.84E+06	4.03E+06	2.25E+06	4.91E+05
10	1.78E+12	7.36E+06	5.22E+06	5.33E+06	4.69E+06	5.36E+06	5.87E+06	5.71E+06	4.75E+06	2.66E+06	5.80E+05

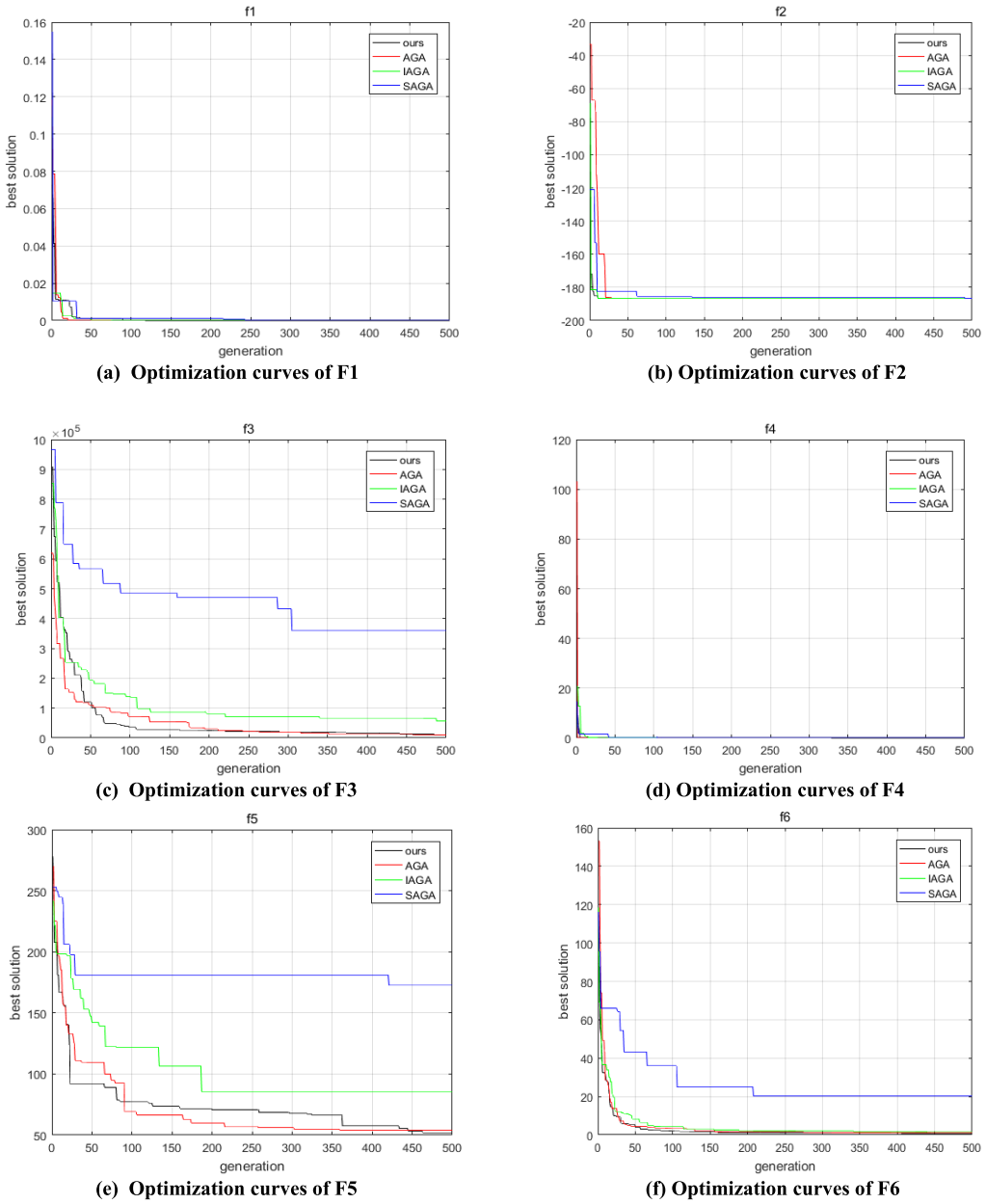
is shown in Figure 8. According to the number of sample attributes and class attributes of the dataset, the determination part is to determine the network's topology and the individual coding length of the LAGA population. In the optimization process, the norm of the test sample's prediction error trained by BPNN is used as the LAGA algorithm's target function value for further implementation. For gaining higher prediction accuracy, the LAGA is used to predict the theoretical optimal weights and thresholds of the network. On the other hand, the LAGA used in the classification part is to optimize the network structure, train it, and finally obtain the dataset's classification result. Figure 6 illustrates the whole BPNN training process using the LAGA.

B. THREE ALGORITHMS USED IN THE LAGA-BPNN FRAMEWORK

As mentioned above, the LAGA is used to solve the optimal weights and thresholds of the network, then use these optimal parameters' value to optimize the initial structure of a new network, and finally train and predict the new network to realize the classification of all samples. The detailed implementation process of the LAGA-BP framework is shown in algorithm 1.

Due to the "black box" of neural networks, the knowledge we acquire through NN learning and training is contained in the connection rights of the network. Therefore, it is necessary to express the implicit knowledge contained in the network explicitly. Thus, the second phase implements the SD algorithm to realize the dataset attributes simplicity (e.g., rule extraction from NN). The SD algorithm uses a suppressive single-layer network and a single hidden layer network to extract the hidden rule (knowledge) from the original neural network. Algorithm 2 introduces the implementation process of the SD algorithm.

Finally, K-means clustering algorithm implements the rule extraction of sample attribute values. According to the association attributes extracted by SD algorithm, K-means clustering algorithm is used to divide the samples containing only the association attributes into K categories. The divided K-type samples are compared with the actual class attribute samples to verify sample division accuracy. If the accuracy rate of this method is high, the final cluster center point can be approximated as a basis for classification, i.e. the IF-THEN rule of sample attribute values. The process of rule extraction using K-means clustering algorithm is given in Algorithm 3.



(a) Optimization curves of F1 (b) Optimization curves of F2

(c) Optimization curves of F3 (d) Optimization curves of F4

(e) Optimization curves of F5 (f) Optimization curves of F6

FIGURE 4. The evolution graph of the optimal solution of four algorithms for six test functions.

TABLE 11. Classification results of K-means clustering algorithm.

K_name	*K*	*Classification*	*Correct_Number*	*Class*	*Correct_rate*
Benign	1	453	435	444	97.97%
Malignant	2	230	221	239	92.47%
total		683	656	683	96.05%

V. KNOWLEDGE EXTRACTION CASE UTILIZING THE PROPOSED FRAMEWORK

A. DESCRIPTION OF DATASET

To verify the feasibility and effectiveness of LAGA-BP framework, this paper realizes an example of knowledge extraction based on the proposed framework for disease-assisted diagnosis. A data set of Wisconsin Breast Cancer [38] in the UCI machine learning library is used to reveal the feasibility of the proposed framework. The pre-processed data set contains a total of 683 complete samples. Each sample has ten

Algorithm 1 LAGA-BP Algorithm

Step 1: Structural construction of BPNN and determination of the number of each layer nodes in the network;
Step 2: LAGA algorithm optimizes the structure of BPNN and initializes the network;
Step 3: Dataset preprocessing and sample division;
Step 4: Normalization of data;
Step 5: Setting of BPNN training parameters;
Step 6: For each training set sample, the forward calculation of network ideal output and actual output error, and the reverse calculation and correction of network weights and thresholds;
Step 7: If the accuracy requirements or other end conditions are met, stop training; otherwise, go to Step 6 and continue training;
Step 8: Simulation of BPNN;
Step 9: Result analysis of BPNN.

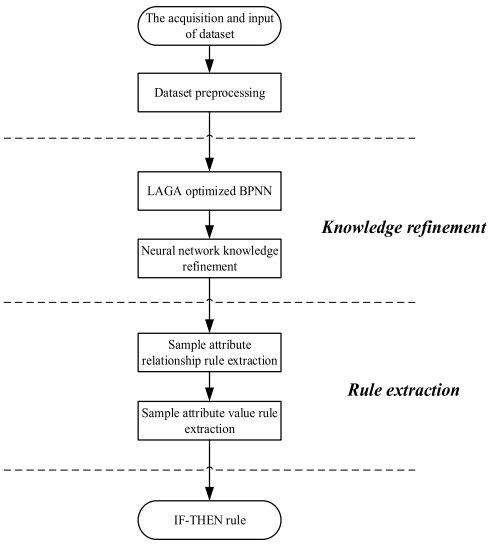


FIGURE 5. Structure diagram of knowledge acquisition framework based on LAGA-BPNN.

attributes and a class attribute (benign or malignant diagnosis result). The experiment allocates train and test sets of BPNN by the ratio of 8: 2.

B. COMPARISON AND ANALYSIS OF BPNN CLASSIFICATION RESULTS

In this experiment, BPNN is used to predict a data set of Wisconsin Breast Cancer samples classification results for refining the knowledge of the disease-assisted diagnosis. Firstly, the structure of BPNN is determined to be 10-25-1 according to the data set sample attributes. The number of hidden layer nodes is obtained by cut and trial. The specific training parameters of BPNN are shown in Table 4, and the other remaining parameters are default. Then, the norm of the NN's sample prediction error is used as the individual target value of the population in the GA, and the value range of the norm is $[-0.5, 0.5]$. The LAGA is used to search an approximate optimal solution of NN, including weights and thresholds. The specific parameters of GA are shown in Table 5. The crossover and mutation probability of the conventional GA are set as 0.7 and 0.01, respectively, and the crossover and mutation probability of the LAGA are dynamically obtained from individual fitness values of the population by Equation (11) and Equation (12). Finally, the LAGA's optimal weights and thresholds are given initial values of BPNN used for prediction and classification, and the network is trained and predicted. The structure and training parameters of this network are given in Table 4.

GA and LAGA algorithms are used to optimize the weight and threshold of BPNN, respectively. According to a contrast experiment, the minimum evolutionary error and implementation time of GA are 1.7032 and 185.223388 seconds, while that of the LAGA algorithm are 1.6337 and 118.907102 seconds. Two graphs of genetic evolution produced by the optimization of two algorithms are shown in Figure 7. According

Algorithm 2 SD Algorithm

Step 1: SSE_{ab} value calculation. First, it uses the original network's output neurons as the new network's additional input neurons, the expanded input neurons, and the original output neurons to build a new single hidden layer network and train it with the BP algorithm SSE_{ab} is calculated with the weights between input layer and hidden layer, and then using it to determine the dependence of output nodes on the input nodes.
Step 2: $Weight_{ab}$ value calculation. First, it uses the expanded input neurons and original output neurons to build a suppressive single-layer network and uses the Hebb rule to adjust the weights and determine the connection weight between neurons ($Weight_{ab}$), respectively.
Step 3: $Product_{ab}$ value calculation. First, it uses $SSE_{ab} \times Weight_{ab}$ to calculate the value of $Product_{ab}$, and sorting the value of $Product_{ab}$ in descending order. Then finding the cutoff point of $Product_{ab}$ table (The difference of $Product_{ab}$ value must be more than $2 \sim 3$ times as the cutoff point), statistics of the attribute distribution relationship of table two sides, and finally determining the IF-THEN rule between attributes and classes

Algorithm 3 K-Means Clustering Algorithm

Step 1: Choosing k samples from n data samples as the initial cluster center points;
Step 2: Calculating the distance between each sample and these center points according to the mean of each cluster (the center-points), and dividing the samples according to the minimum distance (the distance measure uses the square Euclidean distance);
Step 3: Recalculating the mean of each cluster that has changed;
Step 4: Output result if each cluster no longer changes; Else, move to step 2.

TABLE 12. Locations of the center of mass of K cluster.

b	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}
1	3.0552	1.2980	1.4283	1.3532	2.0949	1.3179	2.0927	1.2605	1.1126
2	7.1739	6.8000	6.7348	5.7391	5.4783	7.9304	6.1087	6.0391	2.5696

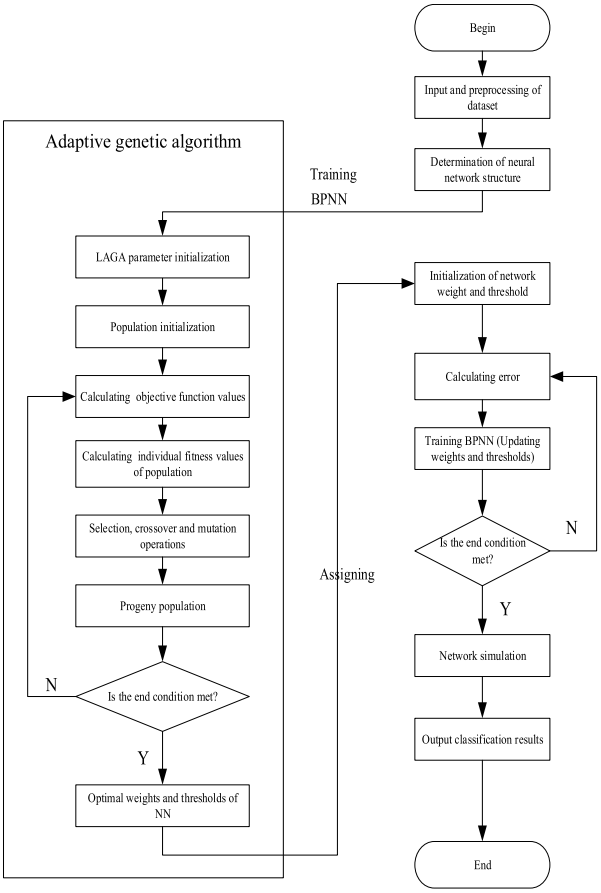


FIGURE 6. The flow chart of BPNN training probes by the LAGA.

to the experimental results, the LAGA is superior to the conventional GA in terms of weight and threshold optimization of NN.

To reduce the influence of randomness of NN, five repeated experiments are implemented to reveal the corresponding findings. Table 6 shows the classification accuracy rate of the test set samples of three BPNN variants.

Among them, initial weights and thresholds of NNs are optimized using random, GA optimized, and LAGA optimized, respectively.

Figure 8 includes three scatter diagrams, which is used to demonstrate the predicted and actual categories of test set samples before and after the optimization of weights and thresholds of BPNN in one of the five repeated experiments.

From the Fig. 8, compared with other two frameworks, it further indicated that predicted and actual categories is more accurate over the LAGA+BP framework. To sum up, the optimized BPNN is better than the non-optimized in predicting and classification from Table 6 and Fig. 8.

C. RESULTS OF RULE EXTRACTION AND ITS ANALYSIS

As mentioned before, the entire knowledge extraction process could be separated into two phases, which are "Attribute reduction" and "Rule extraction", respectively.

In the 'Attribute reduction' phase, the SD algorithm is used to analyze the correlation between sample attributes and sample classes. SD algorithm uses a single-layer inhibitory NN and a single-hidden layer NN to reduce Wisconsin Breast Cancer dataset samples' attributes. Parameters of single hidden layer feed-forward network are shown in Table 4, and parameters of the single-layer network are shown in Table 7.

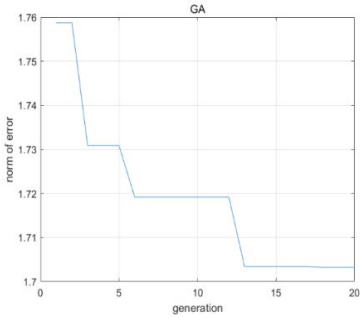
The following tables are some results obtained by the SD algorithm. Table 8 is a transposed matrix of the sum of squared weight errors between the input layer and hidden layer neurons in a single hidden layer network. Table 9 is a matrix of inhibitory connection weights between the input layer and output layer neurons in a single layer network, and Table 10 is a product matrix of the sum of squared weight errors and inhibitory connection weights.

By comparing Wisconsin Breast Cancer dataset (background knowledge base) and the similarity data of connection weight between input layer and hidden layer in extended NN (see Table 9), combined with the congruent relationship of "attribute-neuron" established during network initialization, we find that nodes corresponding to $Weight_{ab}$ happens to be attributes corresponding to rule. In fact, this "coincidence" is a correspondence between the input and output vectors, which is rule and knowledge designed by expert. It can be clearly seen that values of the first column of the $Product_{ab}$ are two to three times higher more than second to eleventh columns from Table 10, so here should be a cutoff point. Therefore, we can get an if-then rule of the attribute relationship of data set.

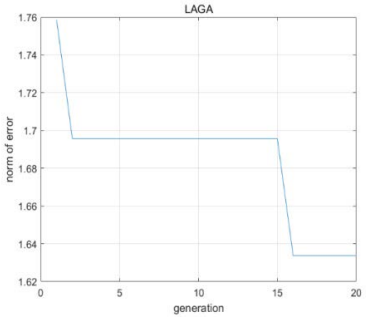
If Clump Thickness and Uniformity of Cell Size and Uniformity of Cell Shape and Marginal Adhesion and Single Epithelial Cell Size and Bare Nuclei and Bland Chromatin and Normal Nucleoli and Mitoses, then Class of Breast Cancer.

In the 'Rules extraction' phase, the K-means clustering algorithm is used to classify the dataset with reduced attributes, and its results are shown in Table 11. It further indicates that the K cluster's centroid is the needed rule (knowledge) by analyzing the dataset and verifying the clustering results. The classification results are given in Table 12.

In table 12, the Symbol "b" stands for Class (1 for benign, 2 for malignant). Symbols "a_2, a_3, a_4, a_5, a_6, a_7, a_2, a_8, a_9, a_{10}" represent attributes of Clump Thickness (1-10), Uniformity of Cell Size (1- 10), Uniformity of Cell Shape (1-10), Marginal Adhesion (1-10), Single Epithelial Cell Size

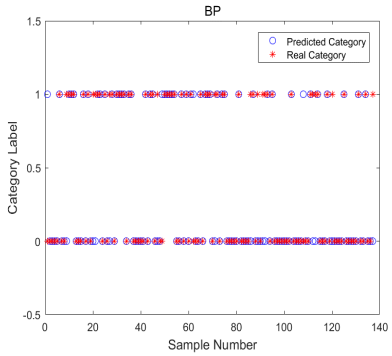


(a) Genetic evolution graphs of GA

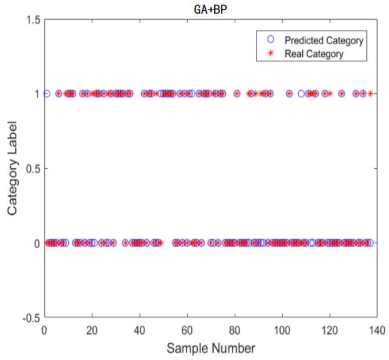


(b) Genetic evolution graphs of LAGA

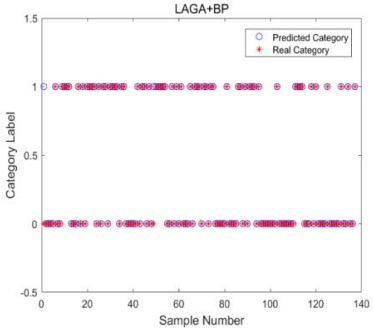
FIGURE 7. Genetic evolution graphs of GA and LAGA.



(a) Predicted and actual categories without/with BPNN



(b) Predicted and actual categories with/without GA-BPNN



(c) Predicted and actual categories with/without LAGA-BPNN

FIGURE 8. Scatter-plots of predicted and actual categories of test-set samples with/without BPNN variants.

(1-10), Bare Nuclei (1-10), Bland Chromatin (1-10), Normal Nucleoli (1-10), Mitoses (1-10), respectively.

Since the center of mass of K cluster is the required rule, we obtain two if-then rules based on the relationship between the sample attributes and classes of breast cancer dataset from Table 12. Two rules as following.

Rule 1:

If $a_2 \approx 3$ and $a_3 \approx 1$ and $a_4 \approx 1$ and $a_5 \approx 1$ and $a_6 \approx 2$ and $a_7 \approx 1$ and $a_8 \approx 2$ and $a_9 \approx 1$ and $a_{10} \approx 1$, then $b = 1$.

Rule 2:

If $a_2 \approx 7$ and $a_3 \approx 7$ and $a_4 \approx 7$ and $a_5 \approx 6$ and $a_6 \approx 5$ and $a_7 \approx 8$ and $a_8 \approx 6$ and $a_9 \approx 6$ and $a_{10} \approx 3$, then $b = 2$.

To verify the feasibility of the proposed framework of knowledge acquisition, we accurately matched rule 1 with the breast cancer dataset, matching a total of twenty samples that were the same as the class $b = 1$. Similarly, rule 2 was hazily matched with the dataset, a total of two samples matching the class $b = 2$, i.e., "a_2-a_{10}" is equal to (7, 8, 7, 6, 4, 3, 8, 8, 4) or (7, 8, 8, 7, 3, 10, 7, 2, 3). By analyzing two rules and referring to relevant materials, it is found that rule 1 and rule 2 have important reference values for the auxiliary diagnosis of breast cancer.

VI. CONCLUSION

Fault diagnosis is key operation to avoid economical losses and enhance machine availability in manufacturing domain. Due to the interpretability, the fault daignosis based on the knowledge-driven is still widely applied and discussed in various domains. As the basic component of the ES, there exists a positive correlation between the reasoning result's accuracy and the precise of the gained FPRs. Hence, how to obtain the FPRs accurately is a valuable topic in the fault diagnosis research.

In this manuscript, a novel knowledge extraction framework is proposed in this manuscript to extract FPRs from the original dataset effectively and accurately. Firstly, by combining the basic principle of GA and analyzing the deficiency of several adaptive GAs, an improved adaptive genetic algorithm, namely LAGA, is proposed in this manuscript to improve both convergence accuracy and speed, especially for high latitude functions. Then, a rule extraction method by combining BPNN and LAGA is discussed to refine knowledge of the data set, and an SD algorithm is used for attribute reduction. Finally, the K-means clustering algorithm is utilized to extract rules (knowledge) from the simplified attribute to extract the classification rules of the data set. A case study is used as an example for a knowledge extraction experiment, which verified the feasibility and effectiveness of the proposed knowledge extraction framework. Experimental results reveal that the proposed framework can extract the corresponding 'IF-THEN' knowledge more precisely.

REFERENCES

[1] Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, "Industrial big data for fault diagnosis: Taxonomy, review, and applications," *IEEE Access*, vol. 5, pp. 17368–17380, 2017.

[2] L. Kou, C. Liu, G.-W. Cai, J.-N. Zhou, Q.-D. Yuan, and S.-M. Pang, "Fault diagnosis for open-circuit faults in NPC inverter based on knowledge-driven and data-driven approaches," *IET Power Electron.*, vol. 13, no. 6, pp. 1236–1245, 2020.

[3] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2017.

[4] M. Sánchez, E. Exposito, and J. Aguilar, "Implementing self-autonomic properties in self-coordinated manufacturing processes for the industry 4.0 context," *Comput. Ind.*, vol. 121, Oct. 2020, Art. no. 103247.

[5] Q. Zhou, P. Yan, H. Liu, and Y. Xin, "A hybrid fault diagnosis method for mechanical components based on ontology and signal analysis," *J. Intell. Manuf.*, vol. 30, no. 4, pp. 1693–1715, Apr. 2019.

[6] A. R. Ramos, C. D. Acosta, P. J. R. Torres, E. I. S. Mercado, G. B. Baez, L. A. Rifón, and O. Llanes-Santiago, "An approach to multiple fault diagnosis using fuzzy logic," *J. Intell. Manuf.*, vol. 30, no. 1, pp. 429–439, Jan. 2019.

[7] G. Wang, F. Zhang, B. Cheng, and F. Fang, "DAMER: A novel diagnosis aggregation method with evidential reasoning rule for bearing fault diagnosis," *J. Intell. Manuf.*, vol. 32, no. 1, pp. 1–20, Jan. 2021.

[8] K.-Q. Zhou, L.-P. Mo, J. Jin, and A. M. Zain, "An equivalent generating algorithm to model fuzzy Petri net for knowledge-based system," *J. Intell. Manuf.*, vol. 30, no. 4, pp. 1831–1842, Apr. 2019.

[9] S. I. Gallant, "Connectionist expert systems," *Commun. ACM*, vol. 31, no. 2, pp. 152–169, Feb. 1988.

[10] S. Sestito and T. Dillon, "Knowledge acquisition of conjunctive rules using multilayered neural networks," *Int. J. Intell. Syst.*, vol. 8, no. 7, pp. 779–805, 1993.

[11] R. Setiono, "Extracting M-of-N rules from trained neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 512–519, Mar. 2000.

[12] V. L. Policicchio, A. Pietramala, and P. Rullo, "GAMoN: Discovering hypotheses for text classification by a lattice-based genetic algorithm," *Artif. Intell.*, vol. 191, pp. 61–95, Nov. 2012.

[13] V. K. H. Chan and C. W. Chan, "Towards explicit representation of an artificial neural network model: Comparison of two artificial neural network rule extraction approaches," *Petroleum*, vol. 6, no. 4, pp. 329–339, 2020.

[14] E. R. Hruschka and N. F. F. Ebecken, "Applying a clustering genetic algorithm for extracting rules from a supervised neural network," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. (IJCNN), Neural Comput., New Challenges Perspect. New Millennium*, vol. 3, Jul. 2000, pp. 407–412.

[15] X. L. Ding, "Research on knowledge acquisition of expert system based on rough set and ANN," College Inf. Sci. Eng., Lanzhou Univ., Lanzhou, China, 2008.

[16] Z. G. Zhang, M. G. Yu, and D. W. Guo, "Neural network rule extraction algorithm based on clustering," *J. Jilin Univ., Inf. Sci. Ed.*, vol. 28, no. 5, pp. 506–512, 2010.

[17] M. H. Mohamed, "Rules extraction from constructively trained neural networks based on genetic algorithms," *Neurocomputing*, vol. 74, no. 17, pp. 3180–3192, Oct. 2011.

[18] N. Singh, P. Singh, and D. Bhagat, "A rule extraction approach from support vector machines for diagnosing hypertension among diabetics," *Expert Syst. Appl.*, vol. 130, pp. 188–205, Sep. 2019.

[19] S. Wang, Y. Wang, D. Wang, Y. Yin, Y. Wang, and Y. Jin, "An improved random forest-based rule extraction method for breast cancer diagnosis," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105941.

[20] J. M. Benítez, J. L. Castro, and I. Requena, "Are artificial neural networks black boxes?" *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1156–1164, Sep. 1997.

[21] H. R. Singh and S. K. Biswas, "Rule extraction from neuro-fuzzy system for classification using feature weights: Neuro-fuzzy system for classification," *Int. J. Fuzzy Syst. Appl.*, vol. 9, no. 2, pp. 59–79, Apr. 2020.

[22] V. K. Vidyarthi and A. Jain, "Knowledge extraction from trained ANN drought classification model," *J. Hydrol.*, vol. 585, Jun. 2020, Art. no. 124804.

[23] Y. Q. Zhang, "Research on knowledge acquisition based on neural network algorithm," College Educ. Sci., Nanjing Normal Univ., Jiangsu, China, 2010.

[24] B. Wu and X. MaY, *Expert System*, 2nd ed. Beijing, China: Beijing Institute of Technology Press, 2000.

[25] T. H. Martin, *Neural Network Design*, 2nd ed. Beijing, China: China Machine Press, 2018.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[27] Z. K. Liu and Z. H. Gui, "Knowledge acquisition method based on artificial neural network," *Comput. Appl. Res.*, vol. 16, no. 5, pp. 7–9, 1999.

[28] B. Su, N. Xie, and Y. Yang, "Hybrid genetic algorithm based on bin packing strategy for the unrelated parallel workgroup scheduling problem," *J. Intell. Manuf.*, vol. 32, pp. 957–969, May 2021.

[29] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 4, pp. 656–667, Apr. 1994.

[30] Z. W. Ren and Y. Umbrella, "Improvement of adaptive genetic algorithm and its application in system identification," *J. Syst. Simul.*, vol. 18, no. 1, pp. 41–43 and 66, 2006.

[31] C. Chen, "Research and application of improved adaptive genetic algorithm," College Sci., South China Univ. Technol., Guangzhou, China, 2011.

[32] Z. Yuping, "Research and application of adaptive genetic algorithm," Dept. Math., Harbin Inst. Technol., Heilongjiang, China, 2009.

[33] Z. Yanzhao, "Improvement and application research of adaptive genetic algorithm," College Sci., Liaoning Tech. Univ., Liaoning, China, 2014.

[34] W. Siyan, "Research on adaptive genetic algorithm," College Math. Phys., North China Electr. Power Univ., Baoding, China, 2009.

[35] S. Surjanovic and D. Bingham, "Optimization test problems," Virtual Library Simul. Exp., Test Function Datasets. Accessed: Mar. 5, 2021. [Online]. Available: http://www.sfu.ca/~ssurjano/index.html

[36] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[37] Y. Zhu, W. Xu, G. Luo, H. Wang, J. Yang, and W. Lu, "Random forest enhancement using improved artificial fish swarm for the medial knee contact force prediction," *Artif. Intell. Med.*, vol. 103, Mar. 2020, Art. no. 101811.

[38] *Breast Cancer Wisconsin Data Set*. Accessed: Sep. 5, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/

YUN OU was born in Hengyang, China, in 1979. He is currently a Lecturer and a CAA Member. His research interests include soft computing technology and fuzzy Petri net and its applications.

SHAO-QIANG YE was born in Changsha, China, in 1996. He received the B.S. degree in network engineering from Jishou University, in 2019, where he is currently pursuing the master's degree in intelligent computing and its applications. His research interests include soft computing techniques and cloud computing.

KAI-QING ZHOU was born in Changsha, China, in 1984. He received the B.S. degree in computer science and technology from Jishou University, in 2006, the M.S. degree in computer applied techniques from the Changsha University of Science and Technology, in 2011, and the Ph.D. degree in computer science from Universiti Teknologi Malaysia, in 2016. He was a Postdoctoral Fellow with the College of Information and Engineering, Central South University, from 2016 to 2018. He is currently an Associate Professor with the Department of Data Science and Big Data Technology, College of Information and Engineering, Jishou University. His research interests include fuzzy Petri net and its applications, clinical decision support systems, and soft computing techniques.

LEI DING was born in Yueyang, China, in 1972. He received the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2009. He is currently a Professor with the College of Information Science and Engineering, Jishou University, Jishou, China. His research interests include neural networks, network security, and intelligent computing.

AZLAN MOHD ZAIN (Member, IEEE) received the Ph.D. degree in computer science from Universiti Teknologi Malaysia, in 2010. He is currently a Professor in computer science with the Big Data Center, Universiti Teknologi Malaysia. His research interests include artificial intelligence, modeling and optimization, machining, and statistical process control.