Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1698323, 13 pages
https://doi.org/10.1155/2020/1698323

Research Article

Reinforcement Learning-Based Genetic Algorithm in Optimizing Multidimensional Data Discretization Scheme

Qiong Chen,[1,2] Mengxing Huang ,[1,2] Qiannan Xu,[3] Hao Wang,[4] and Jinghui Wang[1,2]

[1]*State Key Laboratory Marine Resource Utilization in South China Sea, Haikou 570228, China*
[2]*College of Information Science and Technology, Hainan University, Haikou 570228, China*
[3]*Dept. of Obstetrics and Gynecology, The First Affiliated Hospital of Hainan Medical University, Haikou 570102, China*
[4]*Big Data Lab, Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjøvik, Norway*

Correspondence should be addressed to Mengxing Huang; huangmx09@163.com

Received 22 November 2019; Revised 19 February 2020; Accepted 28 February 2020; Published 30 March 2020

Academic Editor: Bogdan Dumitrescu

Feature discretization can reduce the complexity of data and improve the efficiency of data mining and machine learning. However, in the process of multidimensional data discretization, limited by the complex correlation among features and the performance bottleneck of traditional discretization criteria, the schemes obtained by most algorithms are not optimal in specific application scenarios and can even fail to meet the accuracy requirements of the system. Although some swarm intelligence algorithms can achieve better results, it is difficult to formulate appropriate strategies without prior knowledge, which will make the search in multidimensional space inefficient, consume many computing resources, and easily fall into local optima. To solve these problems, this paper proposes a genetic algorithm based on reinforcement learning to optimize the discretization scheme of multidimensional data. We use rough sets to construct the individual fitness function, and we design the control function to dynamically adjust population diversity. In addition, we introduce a reinforcement learning mechanism to crossover and mutation to determine the crossover fragments and mutation points of the discretization scheme to be optimized. We conduct simulation experiments on Landsat 8 and Gaofen-2 images, and we compare our method to the traditional genetic algorithm and state-of-the-art discretization methods. Experimental results show that the proposed optimization method can further reduce the number of intervals and simplify the multidimensional dataset without decreasing the data consistency and classification accuracy of discretization.

1. Introduction

The rapid development of the internet of things has produced massive amounts of large-scale data [1–5]. These are mainly from various types of sensors, with high-dimensional, incomplete, random, fuzzy, and strong interference and other characteristics [6]. Despite the growing body of artificial intelligence research, how to extract and analyze valuable information from these massive amounts of complex sensor data is still a huge challenge in the field of artificial intelligence [7–9]. As one of the most influential data preprocessing technologies, feature discretization can reduce the complexity of data by transforming the continuous features in massive data to discrete features and obtain shorter, more accurate, and more comprehensible rules, so as to improve the efficiency of data mining and machine learning [10–16]. Therefore, feature discretization plays a key role in the application of artificial intelligence technology to these data.

With the development of artificial intelligence, more and more scholars are studying feature discretization [17–19]. Obtaining the optimal discretization scheme has been proved to be an NP complete problem [20]. The choice of a discretization method for a dataset will restrict the performance and accuracy of a posterior learning task. Discretization technology can be classified as either supervised or unsupervised, according to whether the data contain category information [21].

EqualWidth and EqualFrequency are two commonly used unsupervised discretization methods [22]. They divide the entire attribute according to a given interval length and frequency. Although they are simple and convenient, they lead to uneven data distribution and loss of some important information. Supervised discretization has the advantage of making full use of the class label and target attribute information, and it facilitates finding the appropriate breakpoint position compared to unsupervised discretization.

1R is the simplest supervised discretization algorithm [23]. It uses a greedy strategy to divide the attribute value range into intervals, each corresponding only to a decision class. However, because the partition standard of the interval is too simple, lacks flexibility, and does not consider the correlation between features, it cannot guarantee that the compatibility of an information system after discretization will not be destroyed.

The information entropy-based method is based on the minimum length description principle (MDLP), and it uses the measure of information gain to determine the breakpoint [24]. Although it can largely ensure the consistency of samples in the interval, it is difficult to filter the noise by setting the threshold value of the partition.

ChiMerge [25], Chi2 [26], and Extended Chi2 [27] discriminate and merge adjacent intervals by measuring the degree of association between condition attributes and target attributes. They have the advantage that the structures of adjacent intervals are distinct, but they are sensitive to parameters.

CADD [28], CAIM [29], and CACC [30] use statistics to quantify class attribute correlation, which they can maximize after discretization, and they can obtain the minimum number of discrete intervals as much as possible. However, they only achieve the best discretization for a single attribute interval, lack the description of all of the data, do not consider the consistency of the data either before or after discretization, and will inevitably lose important information from the original data.

These mainstream methods are obviously based on specific division criteria to achieve the discretization of continuous features. However, in the process of multidimensional data discretization, the distribution of target attribute values is usually difficult to know, and the features have complex correlations. In addition, the relatively fixed division criteria cannot provide a comprehensive measure of discrete intervals, and there will be some defects. Therefore, the discretization schemes obtained by most of the algorithms are not optimal in specific application scenarios, or they may even fail to meet the accuracy requirements of the system. A PSO algorithm was applied to feature selection based on discretization, which can generate more powerful and compact representations in high-dimensional datasets, and thus achieve better classification performance [31]. ACO [32] is used to solve the problem of discretization of continuous features to obtain more concise decision rules and higher prediction accuracy. RS-GA is a mature discretization method, which uses the individual fitness function based on rough sets to evaluate the uncertainty of an information system in a genetic algorithm and searches for the optimal discretization scheme through individual evolution [33]. Although these swarm intelligence algorithms can achieve better results, it is difficult to formulate appropriate strategies without prior knowledge, which will make the search in multidimensional space inefficient, consume computing resources, and easily fall into local optima.

Aiming at these problems, this paper proposes a reinforcement learning-based genetic algorithm (RLGA) to optimize the discretization scheme of multidimensional data. First, we binary code the attribute values of the multidimensional data and initialize the population. The binary code method can build an efficient mathematical model suitable for the problem of feature discretization. Second, it is difficult to form a proper strategy without prior knowledge as guidance, which causes the search space to easily fall into local optima. We use rough sets [34] to construct individual fitness functions and a design control function to dynamically adjust the diversity of the population. Then, we introduce a reinforcement learning mechanism [35] to crossover and mutation to determine the crossover fragments and mutation points of the discretization scheme to be optimized. We compare our method to state-of-the-art discretization methods on GF-2 and Landsat 8 images. Experimental results show that the proposed method can reduce the number of intervals and simplify the multidimensional dataset without decreasing the data consistency and classification accuracy of a discretization scheme.

The remainder of this paper is arranged as follows. Section 2 describes basic concepts and reviews some related work. Section 3 explains the algorithm flow of the proposed work. Section 4 introduces the experimental environment and datasets. We analyze and discuss the experimental results in Section 5. Section 6 summarizes this paper and discusses future research.

2. Background and Related Work

We introduce the concept of feature discretization and present simple descriptions of rough sets, genetic algorithms, and reinforcement learning mechanisms. We also analyze problems that occur in the discretization of multidimensional data in the optimization process.

2.1. Definitions of Feature Discretization. Feature discretization is the process of dividing a continuous attribute value (also called a continuous feature value) into a finite number of intervals according to a rule, and associating these intervals with a set of discrete values [10]. Considering the problem of m-dimensional data classification, the discretization algorithm divides the attribute values on the i-th dimension into n_i discrete, nonintersecting intervals:

$$D_i = \left\{ [d_0, d_1], (d_1, d_2], \ldots, \left(d_{n_i - 1}, d_{n_i}\right] \right\}, \tag{1}$$

where d_0 and d_{n_i} are the minimum and maximum attribute values. All of the values are arranged in ascending order in D_i, which is called a discretization scheme on the ith dimension. $D = \{D_1, D_2, \ldots, D_i, \ldots, D_m\}$ represents the whole discretization scheme of m-dimensional data. Obviously, the search space of m-dimensional data feature discretization is formed by all of the candidate breakpoints of each dimension, which are different attribute values of each dimension in the training set.

2.2. Genetic Algorithm Using Binary Coding. As a global optimization probability evolution algorithm, a genetic algorithm [36] has inherent implicit parallelism and a strong global search ability. It achieves good performance on many optimization problems [37]. We use binary encoding to encode the candidate breakpoint; the values 1 and 0 represent that the breakpoint is selected or discarded, respectively. Assuming that $BP_i = \{bp_i^1, bp_i^2, \ldots, bp_i^{n_i}\}(i = 1, 2, \ldots, m)$ is a candidate breakpoint set of m-dimensional data in the ith dimension, the chromosome structure in the genetic algorithm is shown in Figure 1, where colors represent different features in the m-dimensional data. The length of each chromosome is $\sum_{i=1}^{m} n_i$. The set of selected candidate breakpoints is a discretization scheme, according to which attribute values are classified into discrete intervals formed by the candidate breakpoints in the set.

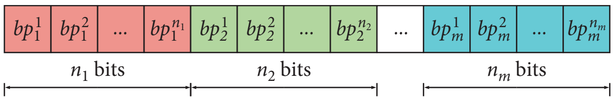


Figure 1: Chromosome structure.

2.3. Fitness Function Based on Rough Set. A rough set [38] is based on the classification mechanism. It interprets the classification as indiscernible relations in the space of features, and these relations form the division of the space. Given decision table $S = (U, R, V, f)$, where U is a finite set of objects, i.e., a domain, R is an attribute set including condition attribute set C and decision attribute set D. For each attribute subset $A \subseteq R$, the indiscernible binary relation $\mathrm{IND}(A)$ and the equivalent classes of attribute subset A in domain U are defined as

$$\begin{aligned} \mathrm{IND}(A) &= \left\{(x, y) \mid (x, y) \in U^2, \forall a \in A(a(x) = a(y))\right\}, \\ U \mid IND(A) &= \{X \mid X \subseteq U \wedge \forall x \in X \forall y \in X \forall a \in A(a(x) = a(y))\}. \end{aligned} \tag{2}$$

According to the abovementioned decision table S, for each subset A in U and equivalent class of attribute subset X in U, the lower and upper approximation sets of X are defined as

$$A_{-}(X) = \cup\{Y \mid Y \in U \mid \mathrm{IND}(A) \wedge Y \subseteq X\}, \tag{3}$$

$$A^{-}(X) = \cup\{Y \mid Y \in U \mid \mathrm{IND}(A) \wedge Y \cap X \neq \varnothing\}. \tag{4}$$

Since the principle of selecting the optimal breakpoint set is to minimize the number of breakpoints without changing the indiscernible relations of the decision table, the fitness function should be determined by the number of breakpoints and the indiscernible relations:

$$\mathrm{Fitness}(D) = \frac{(N_I - N_D)}{N_I} \times R_D, \tag{5}$$

where N_I is the number of breakpoints in the initial breakpoint set, N_D is the number of breakpoints obtained after chromosome decoding, and $\Delta N = N_I - N_D$ is the change of the number of breakpoints. R_D takes the value 0 or 1, which, respectively, mean that the indiscernibility of the decision table changes or does not change after discretization. The judgment process is shown in Algorithm 1.

2.4. Reinforcement Learning. Reinforcement learning is a goal-driven, highly adaptive machine learning technique in the field of artificial intelligence [39], in which there are two basic elements: state and action. Performing an action in a certain state is a strategy. The learner must constantly explore to generate an optimal strategy. Different from supervised and unsupervised learning, it regards learning as a process of interaction between agents and the environment through exploration and evaluation. The operation mechanism is shown in Figure 2. The agent selects an action to be applied to the environment by sensing the current state of the environment. After the environment accepts the action, the state changes, and a reward is given to the agent. According to the new state of the environment, the agent continues to select the next action, and this is repeated until it reaches the terminated state. The goal of reinforcement learning is to maximize the accumulated rewards by adjusting strategies.

Q-learning [40] is one of the most representative model-free reinforcement learning techniques. In the current state, the agent selects the next action according to the corresponding Q value of each action using the $\varepsilon -$ greedy strategy and updates Q at each step in the learning process as

$$Q(s, a) = R(s, a) + \gamma \cdot \max_{a' \in A}\{Q(s', a')\}, \tag{6}$$

where s and a are, respectively, the current state and behavior, s', a', and A are the next state and behavior, A is the action set, and $0 \leq \gamma \leq 1$. Q-learning has broad application prospects in solving complex control and decision-making problems [41, 42]. We use Q-learning to determine the cross fragments and mutation points of the discretization scheme to be optimized.

2.5. Main Challenges. The complexity of multidimensional data feature discretization increases sharply with the length of the attribute value interval and the association between attributes [36]. When using the genetic algorithm [43] to optimize the discretization scheme of multidimensional data, the main challenges are as follows.

(1) Improper control of population diversity causes premature convergence.

(2) Because multidimensional data contain many features, cross fragments and mutation points tend to focus on features with larger value intervals, which

```
Input: Discretization scheme, original decision table
Output: Indiscernible relationship change
initialize: $R_D = 1$;
begin
  for each category $i$ do
    Compute the lower approximation set $C_-(d_i)$ of the original decision table before discretization using equation (3);
    Compute the upper approximation set $C^-(d_i)$ of the original decision table before discretization using equation (4);
  end
  Discrete the original decision table by the discretization scheme;
  for each category $i$ do
    Compute the lower approximation set $C_-(d_i)'$ of the original decision table after discretization using equation (3);
    Compute the upper approximation set $C^-(d_i)'$ of the original decision table after discretization using equation (4);
  end
  for each category $i$ do
    if $C_-(d_i)' \neq C_-(d_i)$ or $C^-(d_i)' \neq C^-(d_i)$ do
      $R_D = 0$;
      break;
    end
  end
  Return $R_D$;
end
```

Algorithm 1: Indiscernible relationship change judgement.

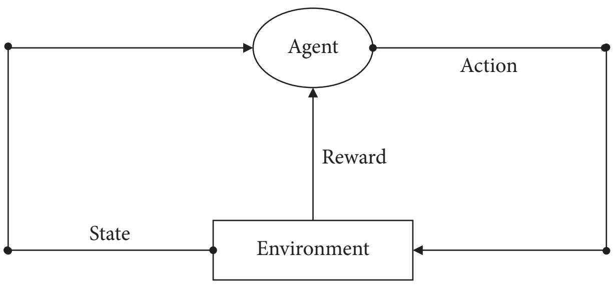


Figure 2: Operation mechanism of reinforcement learning.

decrease the opportunity for breakpoints on other features to evolve.

(3) Due to the complex correlations between features, the crossover and mutation operations are relatively blind without prior knowledge as guidance, making it highly likely that some high-quality fragments on features are destroyed in the next generation of evolutionary operations.

3. Reinforcement Learning-Based Genetic Algorithm

A genetic algorithm is a general method to search for the optimal solution in the field of artificial intelligence. We add a control function to the fitness function based on rough sets to dynamically adjust the diversity of the population. In addition, according to the characteristics of multidimensional data, we introduce a reinforcement learning mechanism to the crossover and mutation operation to determine the cross fragments and mutation points of the discretization scheme to be optimized, which greatly improves the accuracy and speed of convergence. Below, we discuss the flow of the algorithm.

3.1. Evolution of Discretization Scheme to Be Optimized. We perform evolutionary operations on the optimized discretization scheme and initial population, as shown in Figure 3. The global variable preserves the best individual of the population, while the local variable preserves the historical best individual of the discretization scheme to be optimized. In each iteration, the fitness of the current population is calculated, the optimal individual is obtained, and the global variable is updated. Then, the discretization scheme to be optimized and the optimal individual of the population carry out the cross and mutation operations based on reinforcement learning, and the local variable is updated. If the termination condition is not reached, then the population will continue to carry out ordinary evolutionary operations. Otherwise, by comparing the global variable and local variable, the individual with the largest fitness will be output.

3.2. Selection Operator Based on the Control Function. The selection operation is based on the evaluation of individual fitness in the population. Individuals with higher fitness are generally more likely to be selected. Roulette [44] is a simple, efficient, probability-based method that is often used to select individuals. If the population size is n and the fitness value of individual i is f_i, then the probability that individual i is selected is

$$P_i = \frac{f_i}{\sum_{j=1}^{n} f_j}. \quad (7)$$

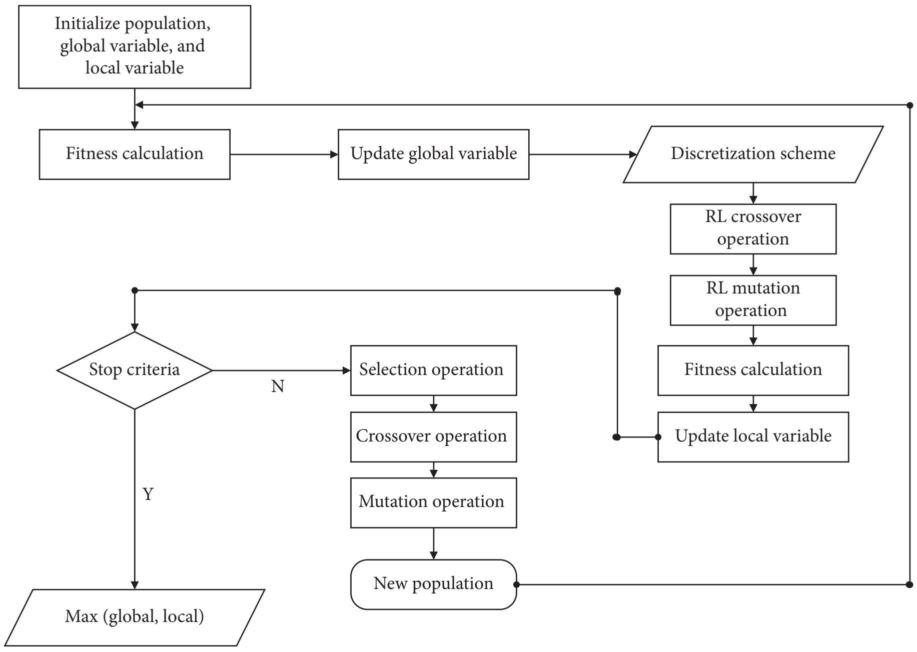


FIGURE 3: Evolution of the discretization scheme to be optimized.

However, individuals with large fitness are more likely to be selected, which may result in the destruction of population diversity. Since R_D takes the value 0 or 1, individuals with fitness value 0 cannot be selected in the evolution of the population. Although these individuals are not feasible solutions at the initial stage, they may be close to the optimal feasible solution, and may eventually evolve into it. However, considering the randomness and evenness of the initial population, feasible solutions with $R_D = 1$ and potential feasible solutions with $R_D = 0$ tend to be quantitatively equivalent. Even with a small population, because of the complex correlation among the features of multidimensional data, there are far more potential feasible solutions with $R_D = 0$ than with $R_D = 1$. Therefore, when roulette is used for individual selection, the potential feasible solutions that are the majority of the population will be eliminated, thus destroying the diversity of the population. According to the abovementioned analysis, to ensure the diversity of the population in the early stage and accelerate convergence of individuals to the optimal solution in the late stage of evolution, we expand the fitness function by adding the control function on the original basis, and we determine the control factors according to the proportion of feasible solutions in the population in the current stage of evolution. The fitness function of chromosome x is thus expanded to

$$\text{Fitness}'(x) = \frac{(N_I - N_D^x)}{N_I} \times R_D^x + \sigma(p) \times \varphi(x). \tag{8}$$

The control function consists of control item $\varphi(x)$ and control factor $\sigma(p)$, where $\sigma(p)$ is the proportion of the population consisting of potential feasible solutions with $R_D = 0$, and $\sigma(p)$ is a function with p as an independent variable. The expressions for $\varphi(x)$ and $\sigma(p)$ are

$$\begin{gathered}
\varphi(x) = \left[\frac{(N_I - N_D^x)}{N_I}\right]^z, \\
\sigma(p) = \begin{cases} 0, & \mu = 1, \\ \mu^{1-p}, & 0 < \mu < 1, \\ \left(\dfrac{1}{2k \times \log_2 N_I}\right)^{1-p}, & \mu = 0, \end{cases} \\
\mu = \frac{\sum_{i=1}^{k} \left(\text{setcmp}\left(C_-(d_i)', C_-(d_i)\right) + \text{setcmp}\left(C^-(d_i)', C^-(d_i)\right)\right)}{2k},
\end{gathered} \tag{9}$$

where N_I is the number of breakpoints in the initial breakpoint set, N_D^x is the number of breakpoints obtained after chromosome x decoding, R_D^x is the change of the indiscernible relationship of the decision table after discretization by chromosome x, k is the number of classes, d_i is the ith class, $C_-(d_i)$ and $C^-(d_i)$ are, respectively, the lower and upper approximation sets of d_i before discretization, $C_-(d_i)'$ and $C^-(d_i)$ are, respectively, the lower and upper approximation sets of d_i after discretization, and $\text{setcmp}\,(\text{set}A, \text{set}B)$ is a user-defined set comparison function. When set$A$ and set$B$ are equal, the function value is 1, and otherwise it is 0. Since each class corresponds to a lower and upper approximation set, the value range of μ is $0 \le \mu \le 1$. The following points are made regarding the extended fitness function.

(1) We want to prevent the unreasonable situation that when the number of breakpoints of the feasible solution with $R_D = 1$ is equal to that of the potential feasible solution with $R_D = 0$, the fitness value of the former is less than or equal to that of the latter. Hence we require that index z of the control item satisfies $z > 1$. In this paper, $z = 2$.

(2) It can be seen that the value range of μ is actually $2K + 1$ discrete points from 0 to 1, with the interval $1/2k$. When $\mu = 0$, the indiscernibility of the decision table is destroyed after discretization. If $\mu = 0$, then $\mu = 0$, and the significance of adding a control function for potential feasible solutions with $R_D = 0$ is lost, since the control function has a smaller value than when μ takes the minimum nonnegative value $\mu = 1/2K$. Consequently, we change the expression μ^{1-p} to $(1/(2k \times \log_2 N_1))^{1-p}$. So, when $\mu = 0$, the value of the control function is not 0, and it is smaller than its value with $\mu = 1/2k$. However, because the chromosome is binary coded, taking "2" as the base number and N_1 as the true number can ensure that the difference between the two control function values corresponding to $\mu = 0$ and $\mu = 1/2k$ is not too large, and this controls the selection of individuals relatively reasonably.

(3) When $\mu = 1$, $R_D = 1$ and $\sigma(p) = 0$, indicating that this individual is a feasible solution, and the value of the control item is 0. When $0 \le \mu < 1$, $0 \le \mu < 1$ and $\sigma(p)$ is an increasing function with p as the independent variable. Consider that in the early stage of evolution, there are few or no feasible solutions in the population, i.e., p is large. At this time, $\sigma(p)$ should be relatively large, so the potential feasible solution can be selected with a large probability in roulette, and the search is gradually guided to the area of feasible solutions. As evolution progresses, more and more feasible solutions appear in the population, i.e., p becomes smaller and smaller. Accordingly, $\sigma(p)$ should become smaller and smaller, accelerating the movement from feasible to optimal solutions.

3.3. Crossover Operator Based on Q-Learning. The crossover operation in a genetic algorithm is the exchange of some genes between two matched chromosomes in a certain way, so as to form two new individuals. However, multidimensional data contain many features, and cross fragments tend to focus on features with large value intervals; hence, the breakpoints on other features lose the chance to cross. However, there are complex correlations among features. Without prior knowledge as a guide, the cross operation of the discretization scheme to be optimized becomes blind, which causes some high-quality fragments to have a high probability of being destroyed. To this end, we use the decision-making ability of Q-learning to select some features of the discretization scheme to be optimized in each iteration for the cross operation.

(1) State: according to the abovementioned analysis, the crossover operation will give each feature a certain probability of change. We define the set of changing features in the crossover operation as a state. Assuming that the multidimensional data have N features, the search space is divided into $2N - 1$ states, each a combination of several features. For example, when $N = 3$, there are seven states: $\{f_1\}$, $\{f_2\}$, $\{f_3\}$, $\{f_1, f_2\}$, $\{f_1, f_3\}$, $\{f_2, f_3\}$, and $\{f_1, f_2, f_3\}$, where f_i is the ith feature of multidimensional data, $1 \le i \le N$, and the elements in $\{*\}$ represent the features corresponding to the loci of the most recent crossover or mutation operation.

(2) Action: since multidimensional data generally contain a large number of features, according to the abovementioned definition of states, the number of states will increase exponentially. The transition between two states corresponds to one action; so, many actions must be defined, which increases the computational complexity. According to the previous analysis, we mainly want to avoid the situation that in each crossover operation, cross segments focus on features with a larger value range, which makes some breakpoints lose the chance of crossing. In addition, some high-quality segments will be destroyed without prior knowledge as guidance. Therefore, for the current state, the next state to jump to after performing an action should be mainly considered from three aspects. First, the feature set of the next state is a subset of that of the previous state. Second, the feature set of the next state is a complement of that the previous state. Third, the intersection of the feature set of the next state and that of the previous state is not empty. Accordingly, there are three kinds of actions, represented by G, H, and I. The algorithm jumps to a new state by performing actions on the current state, and executes cross operations on all of the features contained in the new state. Suppose S_t is the current state and S_{t+1} is the next state. $G(S_t)$ represents a random jump to one of all of the subsets of the current state, $G(S_t)$ is a random jump

to one of all of the subsets of the complementary set of the current state, and $H(S_t)$ is a random jump to a set whose intersection with the current state is not empty and is not a subset of the current state, as shown below.

$$\begin{aligned} S_{t+1} &= G(S_t), \quad S_{t+1} \in \{S \mid S \subseteq S_t \wedge S \neq \varnothing\}, \\ S_{t+1} &= H(S_t), \quad S_{t+1} \in \{S \mid S \cap S_t = \varnothing \wedge S \neq \varnothing\}, \\ S_{t+1} &= I(S_t), \quad S_{t+1} \in \{S \mid S \not\subset S_t \wedge S \neq S_t \wedge S \cap S_t \neq \varnothing\}. \end{aligned} \tag{10}$$

It is easy to see that the range of values for $G(S_t)$, $H(S_t)$, and $I(S_t)$ covers all of the states.

(3) Reward: to make a correct decision when selecting a feature set in each crossover operation so as to more quickly approach the optimal solution, we set a reward value for each state-action combination. The reward value is based on the change of individual fitness, which is mainly used to evaluate the search for the optimal solution of the algorithm. In the formula mentioned below, $P(S_t \longrightarrow S_{t+1} \mid A_t)$ is the probability of jumping to the next state S_{t+1} after performing action A_t in the current state S_t. This is related to the number N_{state} of all of the possible states to jump to, which is $1/N_{\text{state}}$. $\text{fit}(S_t)$ and $\text{fit}(S_{t+1})$ are, respectively, the individual fitness of the current and next state, and lbest is the historical best fitness of the discretization scheme to be optimized.

$$\text{Reward} = \begin{cases} 5 \times P\left(S_t \longrightarrow S_{t+1} \mid A_t\right), & \text{fit}(S_{t+1}) > \text{lbest}, \\ 1 \times P\left(S_t \longrightarrow S_{t+1} \mid A_t\right), & \text{fit}(S_t) < \text{fit}(S_{t+1}) \leq \text{lbest}, \\ 0, & \text{fit}(S_{t+1}) = \text{fit}(S_t), \\ -1 \times P\left(S_t \longrightarrow S_{t+1} \mid A_t\right), & \text{fit}(S_{t+1}) < \text{fit}(S_t). \end{cases} \tag{11}$$

According to the designed reward value, we can update Q when the discretization scheme to be optimized has a cross operation.

3.4. Mutation Operator Based on Q-Learning. Like the crossover operator, we use the decision-making ability of Q-learning to select some features of the discretization scheme to be optimized in each iteration for the mutation operation. The state, action, and reward of the mutation and cross operation are consistent, the only difference being to change the cross operation on the feature to a mutation operation. They each maintain a Q-table. Figure 4 shows the update process of two Q-tables of a discretization scheme with three features in one iteration.

Both Q-tables are initialized to 0. After N iterations, the Q-tables corresponding to the cross and mutation operations are shown in Figures 4(a) and 4(b), respectively. Assuming that the current state is $\{f_1, f_3\}$, action H is selected for the cross operation. Accordingly, the state jumps to $\{f_2\}$, and Q's value of 1 is updated to 6 in cross operation. Then, state $\{f_2\}$ selects action $\{f_2, f_3\}$ for the mutation operation. Accordingly, the state randomly jumps to $\{f_2, f_3\}$, and Q' s value of 2 is updated to 3 in the mutation operation. In this way, the two Q-tables are updated in one iteration.

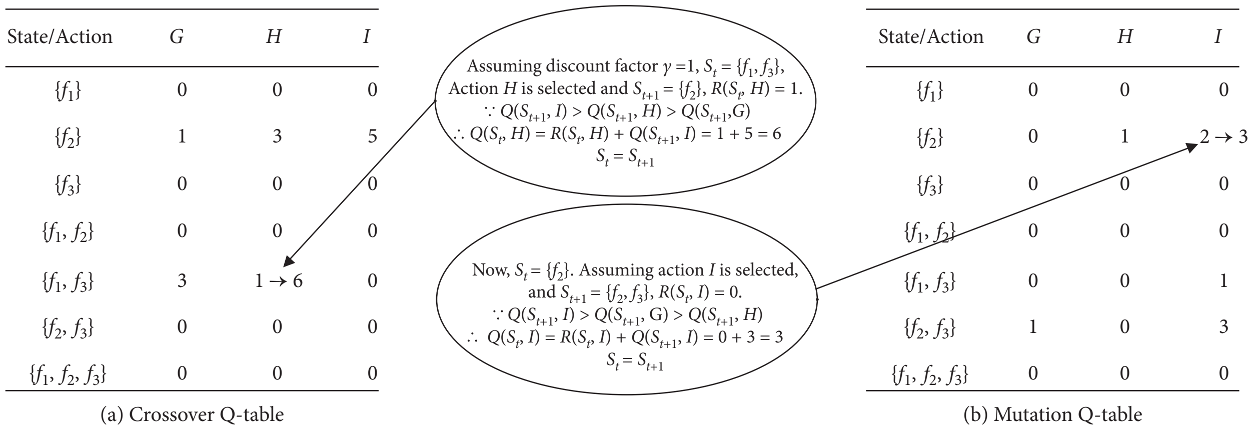
3.5. Flow of RLGA Algorithm. Algorithm 2 shows the flow of the proposed method. First, binary genetic coding is applied to the attribute values of the multidimensional data, and the state is generated according to the number of features. Then, in the current state of the discretization scheme to be optimized, the greedy algorithm is used to select the action, jump to the next state, and cross operate with the global optimal individual on the corresponding features. At the same time, the reward value is evaluated according to the fitness value after the cross operation, whose Q-table is then updated. Similarly, in the current state, an action is selected to continue the mutation operation, whose Q-table is updated. The population performs the conventional genetic operation and saves the global optimal individual in each iteration. Finally, the program outputs the maximum value of both the global and local variable. While the algorithm optimizes the given discretization scheme, other individuals of the population evolve to enlarge the search scope and improve the probability of obtaining the optimal solution.

4. Experimental Design

We introduce the experimental data source, experimental environment configuration, and dataset used in the experiment.

4.1. Data Source. The experimental data are from a Landsat 8 satellite image in the coastal area of the South China Sea on February 22, 2018, as shown in Figure 5. The image consists of seven bands. In the experiment, the objects on the image are divided into five categories: impervious surface, construction, bare land, water, and vegetation.

4.2. Configuration of Experimental Environment. To verify the effectiveness of the algorithm in this paper, comparative experiments are carried out using an Intel Core i5-5200U CPU@2.20GHZ, 12 GB memory, and 512 GB hard disk. The visualization, programming, simulation, testing, and calculation are realized in MATLAB R2016a. The radiometric



State/Action	G	H	I
$\{f_1\}$	0	0	0
$\{f_2\}$	1	3	5
$\{f_3\}$	0	0	0
$\{f_1, f_2\}$	0	0	0
$\{f_1, f_3\}$	3	$1 \rightarrow 6$	0
$\{f_2, f_3\}$	0	0	0
$\{f_1, f_2, f_3\}$	0	0	0

(a) Crossover Q-table

Assuming discount factor $\gamma = 1$, $S_t = \{f_1, f_3\}$, Action H is selected and $S_{t+1} = \{f_2\}$, $R(S_t, H) = 1$.
$\because Q(S_{t+1}, I) > Q(S_{t+1}, H) > Q(S_{t+1}, G)$
$\therefore Q(S_t, H) = R(S_t, H) + Q(S_{t+1}, I) = 1 + 5 = 6$
$S_t = S_{t+1}$

Now, $S_t = \{f_2\}$. Assuming action I is selected, and $S_{t+1} = \{f_2, f_3\}$, $R(S_t, I) = 0$.
$\because Q(S_{t+1}, I) > Q(S_{t+1}, G) > Q(S_{t+1}, H)$
$\therefore Q(S_t, I) = R(S_t, I) + Q(S_{t+1}, I) = 0 + 3 = 3$
$S_t = S_{t+1}$

State/Action	G	H	I
$\{f_1\}$	0	0	0
$\{f_2\}$	0	1	$2 \rightarrow 3$
$\{f_3\}$	0	0	0
$\{f_1, f_2\}$	0	0	0
$\{f_1, f_3\}$	0	0	1
$\{f_2, f_3\}$	1	0	3
$\{f_1, f_2, f_3\}$	0	0	0

(b) Mutation Q-table

FIGURE 4: Updating process of two Q-tables of a discretization scheme with three features in one iteration.

```
Input: Multidimensional data discretization scheme
Output: Optimal discretization scheme
Initialize: global variable = 0, local variable = 0, crossover Q-Table = null, mutation Q-Table = null, $t = 0$;
begin
  Get the initial breakpoints of the multidimensional data by sorting the values of each feature and removing duplicate values;
  Binary encode the initial breakpoints of multidimensional data according to the method in Part B of Section 2;
  Randomly generate initial population $P(t)$;
  Calculate the fitness of each individual in $P(t)$ using equation (8);
  Update global variable with the optimal individual fitness value in $P(t)$;
  Generate state set based on the number of features of multidimensional data according to the definition of state in Part C of
  Section 3;
  Choose a state from the state set as the initial state $S(t)$;
  while $t$ is less than the user's termination iterations do
    Choose an action from the action set $\{G, H, I\}$ by e-greedy strategy according to the definition of action in Part C of Section 3;
    Execute the selected action on the current state $S(t)$ to jump to the next state $S(t+1)$;
    Perform crossover operation with global variable on the features contained in state $S(t+1)$;
    Calculate the fitness of the multidimensional data discretization scheme after crossover operation using equation (5);
    Measure the corresponding reward using equation (11) according to the definition of reward in Part C of Section 3;
    Update crossover Q-Table using equation (6);
    if the fitness of the multidimensional data discretization scheme > local variable do
      Update local variable with the fitness of the multidimensional data discretization scheme;
    end
    Perform crossover operation in $P(t)$;
    Calculate the fitness of each individual in $P(t)$ using equation (8);
    Update global variable with the optimal individual fitness value in $P(t)$;
    $S(t) = S(t+1)$;
    Choose an action from the action set $\{G, H, I\}$ by e-greedy strategy according to the definition of action in Part C of Section 3;
    Execute the selected action on the current state $S(t)$ to jump to the next state $S(t+1)$;
    Perform mutation operation on the features contained in state $S(t+1)$
    Calculate the fitness of the multidimensional data discretization scheme after mutation operation using equation (5);
    Measure the corresponding reward using equation (11) according to the definition of reward in Part C of Section 3;
    Update mutation Q-Table using equation (6);
    if the fitness of the multidimensional data discretization scheme > local variable do
      Update local variable with the fitness of the multidimensional data discretization scheme;
    end
    Perform mutation operation in $P(t)$;
    Calculate the fitness of each individual in $P(t)$ using equation (8);
    Update global variable with the optimal individual fitness value in $P(t)$;
    $t = t + 1$;
  end
  Return Max(global variable, local variable);
end
```

ALGORITHM 2: RLGA algorithm process.

Figure 5: Area used for study.

calibration and atmospheric correction of images, training of classifiers for discrete results, and comparison of classification prediction accuracy are completed in an ENVI5.3 environment.

4.3. Preparation of Experimental Datasets. We randomly select several areas from the image, covering the five categories specified above and integrate them into a set of experimental samples containing training and test sets. The training set includes 6331 samples, consisting of 935 impervious surface samples, 936 construction samples, 958 bare land samples, 2324 water samples, and 1178 vegetation samples. We sort the pixel values of the training set and delete duplicate values in each band to obtain the initial breakpoints of the seven bands, which are 1403, 1429, 1680, 1869, 2402, 2530, and 2240, for a total of 13553 breakpoints. We discretize this initial set and carry out the comparative experiments.

We compare the proposed method to the classical genetic algorithm (GA) [33] based on the consistency principle of the decision system. Then, we compare the optimal set of breakpoints obtained by our method with those of current mainstream supervised discretization methods, such as EDiRa [45], ChiMerge [46], 1R [23], NCAIC [47], FUDC [48], Cramer's V-Test [49], and Chi2 [50], mainly on the evaluation of the number of intervals and data consistency.

In addition, we use the proposed method to optimize the discretization results of the MFD-mvtR algorithm [51]. Finally, we train the neural network classifiers with the discretized samples of all of the methods, and verify the effectiveness of our method by comparing the classification accuracy of each method.

5. Results and Discussion

We compare the performance of our method with that of the classical genetic algorithm based on the consistency principle of the decision system, and we evaluate the results of seven state-of-the-art discretization methods on the Landsat 8 image. We also use our method to optimize the discretization results of MFD-mvtR [51] on the Gaofen-2 image. Finally, the effectiveness of our method is verified by comparing the classification accuracy of each method.

5.1. RLGA versus GA. We set the population size of the two algorithms at 30 and the number of iterations at 500, and we run them 10 times independently. Figure 6 shows the number of iterations of the two algorithms to reach the theoretical optimal solution in 10 independent experiments. Table 1 compares the convergence rates of the two algorithms. The search efficiency is expressed as

$$E = 1 - \frac{A}{T}, \tag{12}$$

where A is the average number of iterations to obtain the optimal solution, T is the total number of iterations, and E is the search efficiency, i.e., the convergence speed. The larger the value, the faster the convergence.

It can be seen from Table 1 that the average number of iterations for GA to obtain the optimal solution is 425.3, and the search efficiency is 0.149. These numbers correspond, respectively, to 338.6 and 0.323 for our method, an obvious improvement in performance.

5.2. RLGA versus Mainstream Discretization Algorithms. We compare the optimal set of breakpoints obtained by our method to mainstream supervised discretization methods, including EDiRa [45], ChiMerge [46], 1R [23], NCAIC [47], FUDC [48], Cramer's V-Test [49], and Chi2 [50], on the number of intervals and data consistency.

Figure 7 shows the number of discrete intervals obtained by the eight algorithms in each band. RLGA obtains the minimum number of breakpoints in each band. Table 2 compares the overall number of intervals and data consistency of the eight algorithms. We can see that the number of discrete intervals obtained by RLGA is 2247, which is the least among all of the algorithms, and there is no data error. EDiRa obtains 3909 discrete intervals with 4 data errors. The number of discrete intervals obtained by ChiMerge is 4947, and the number of data errors is also 4. The number of discrete intervals obtained by 1R is 3053, which is the least excepted for RLGA, but it has the most data errors among all of the algorithms, at 31. The number of discrete intervals obtained by NCAIC is the largest among all of the algorithms, at 5041, with 4 data errors. FUDC obtains 4072 discrete intervals, with 4 data errors. Cramer's V-Test and Chi2 both obtain relatively small numbers of discrete intervals, 3858 and 3538, respectively, but there are also more data errors, 7 and 19, respectively. Considering the number of discrete intervals and the number of data errors, RLGA has the best discretization quality.

5.3. Optimization of Breakpoints on Gaofen-2 Image. We use RLGA to optimize the discretization results obtained by the MFD-mvtR algorithm on the Gaofen-2 image [51]. Considering that this image contains only four bands, the search space is divided into 15 states, which is less than the 127 states obtained from 7 bands of the Landsat 8 image. Therefore, we can set the number of iterations of RLGA to only 100 to allow the algorithm to fully learn. The number of optimized breakpoints is shown in Tables 3 and 4.

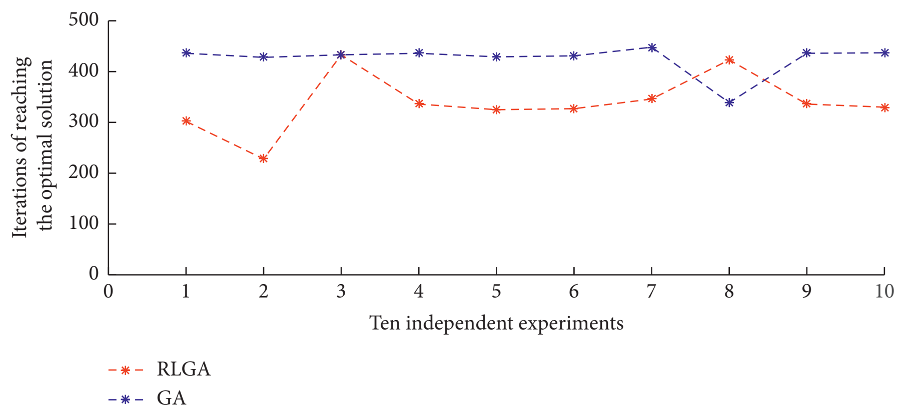
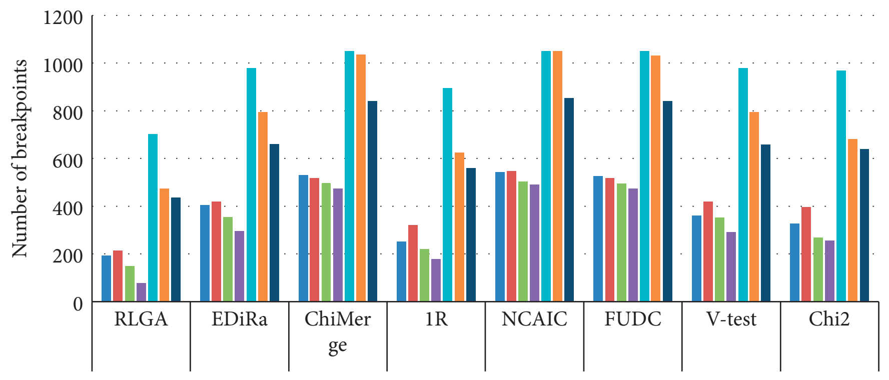


Figure 6: Iterations of the two algorithms to reach theoretical optimal solution.

Table 1: Comparison of running time in one iteration.

Method	Iterations	Average	Search efficiency
RLGA	500	338.6	0.323
GA	500	425.3	0.149



	RLGA	EDiRa	ChiMerge	1R	NCAIC	FUDC	V-test	Chi2
🟦 Band 1	193	406	531	252	542	526	361	328
🟥 Band 2	214	419	518	321	548	518	419	395
🟩 Band 3	148	354	497	220	505	496	352	268
🟪 Band 4	77	296	473	178	491	473	293	255
🟦 Band 5	703	979	1051	896	1051	1051	979	970
🟧 Band 6	475	795	1036	626	1051	1031	795	682
🟦 Band 7	437	660	841	560	853	841	659	640

Figure 7: Number of breakpoints of eight algorithms in different bands.

The MFD-mvtR algorithm [51] obtains 1345 breakpoints with 5 data errors on the Gaofen-2 image. Compared to other mainstream algorithms, the results of discretization are satisfactory. We take the results of MFD-mvtR as the discretization scheme to be optimized, use RLGA to perform the cross operation based on reinforcement learning with the optimal individuals in the population, and carry out its own mutation operation based on reinforcement learning. By further optimizing the results of MFD-mvtR, we can see that the numbers of breakpoints in the four bands are reduced to 310, 301, 222, and 198, respectively. The total number of breakpoints is 314 less than before optimization, and the number of data errors is reduced to 0.

Table 5 shows the classification accuracy of the neural network after training the discrete feature sets obtained by the abovementioned algorithms. We can see that after optimizing the discretization scheme of MFD-mvtR, the classification accuracy obtained by confusion matrix is about 6 percentage points higher than the original, and the kappa coefficient is 0.8925. Our method reduces the number of

Table 2: Comparison of the eight algorithms on intervals and inconsistencies.

Method	Intervals	Inconsistencies
RLGA	2247	0
EDiRa	3909	4
ChiMerge	4947	4
1R	3053	31
NCAIC	5041	4
FUDC	4936	4
V-Test	3858	7
Chi^2	3538	19

Table 3: Number of breakpoints optimized by RLGA.

Method	Band 1	Band 2	Band 3	Band 4
RLGA	310	301	222	198
MFD-mvtR	350	350	346	299
EDiRa	349	349	346	299
ChiMerge	442	433	298	299
1R	64	70	42	40
NCAIC	447	468	346	299
FUDC	397	439	312	258
V-Test	432	423	288	289
Chi^2	410	438	325	272

Table 4: Quality of the discretization scheme optimized by RLGA.

Method	Number of intervals	Inconsistencies
RLGA	1031	0
MFD-mvtR	1345	5
EDiRa	1343	16
ChiMerge	1472	7
1R	216	38
NCAIC	1560	7
FUDC	1406	7
V-Test	1432	7
Chi^2	1445	7

Table 5: Classification accuracy of discretization scheme optimized by RLGA.

Method	Overall accuracy (%)	Kappa coefficient
RLGA	91.0417	0.8925
MFD-mvtR	85.2083	0.8225
EDiRa	74.1667	0.6900
ChiMerge	80.2083	0.7625
1R	56.4583	0.4775
NCAIC	80.6250	0.7675
FUDC	76.2500	0.7150
V-Test	79.7917	0.7575
Chi^2	79.3750	0.7525

breakpoints in the discretization scheme of MFD-mvtR to a certain extent, while ensuring that there is no data error. It can simplify the dataset and effectively identify the five types of areas of impervious surface, construction, bare land, water, and vegetation on the image after training the classifier by the optimized discrete feature set, which improves the performance of the classifier.

6. Conclusion and Future Work

In the process of multidimensional data discretization, due to the complex correlation among features and the performance bottleneck of the traditional discretization criteria, most discretization schemes obtained by the algorithms are not optimal in specific application scenarios or they may even fail to meet the accuracy requirements of the system. Some swarm intelligence algorithms can achieve better results, but without prior knowledge as a guide, it is difficult to formulate appropriate strategies, which will cause an inefficient search in multidimensional space, consume many computing resources, and easily fall into local optimization. To solve these problems, this paper proposes a reinforcement learning-based genetic algorithm to optimize the discretization of multidimensional data. First, we binary code the attribute values of the multidimensional data and initialize the population. Second, we use rough sets to construct individual fitness functions and design control functions to dynamically adjust the diversity of the population. Then, we introduce the Q-learning reinforcement learning mechanism to the crossover and mutation operations to determine the crossover fragments and mutation points of the discretization to be optimized. We conduct simulation experiments on Landsat 8 and Gaofen-2 images to compare RLGA to the traditional genetic algorithm and state-of-the-art discretization methods. The experimental results show that our method can reduce the number of breakpoints and simplify the multidimensional dataset without decreasing the data consistency and classification accuracy of a discretization scheme.

Future research work includes the following: (1) test and improve the proposed method on different multidimensional datasets, expand its application scope, and make it more practical; (2) to ease the problems of high-dimensional datasets, improve the algorithm using deep Q-learning technology; and (3) improve the performance of the deep neural network by using RLGA to optimize its parameters.

Data Availability

The Landsat 8 and Gaofen-2 processing data used to support the findings of this study are included within the article.

Disclosure

The sponsors had no role in the design, execution, interpretation, or writing of the study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors would like to thank the support of the laboratory, university, and government. This work was supported by Hainan Provincial Natural Science Foundation of China (Grant no: 2019CXTD400) and the National Key Research and Development Program of China (Grant no: 2018YFB140400).

References

[1] X. Zeng, G. Xu, X. Zheng, Y. Xiang, and W. Zhou, "E-AUA: an efficient anonymous user authentication protocol for mobile IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1506–1519, 2019.

[2] L. Li, G. Xu, L. Jiao et al., "A secure random key distribution scheme against node replication attacks in industrial wireless sensor systems," *IEEE Transactions on Industrial Informatics (Early Access)*, vol. 16, no. 3, pp. 2091–2101, 2020.

[3] Y. Yin, L. Chen, Y. Xu, and J. Wan, "Location-aware service recommendation with enhanced probabilistic matrix factorization," *IEEE Access*, vol. 6, pp. 62815–62825, 2018.

[4] H. Kang and C. Chen, "Fruit detection and segmentation for apple harvesting using visual sensor in orchards," *Sensors*, vol. 19, no. 20, p. 4599, 2019.

[5] X. Yan, L. Wen, and L. Gao, "A fast and effective image preprocessing method for hot round steel surface," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–14, 2019.

[6] N. Kafli and K. Isa, "Internet of Things (IoT) for measuring and monitoring sensors data of water surface platform," in *Proceedings of the 2017 IEEE 7th International Conference on Underwater System Technology: Theory and Applications (USYS)*, Kuala Lumpur, Malaysia, December 2017.

[7] G. Xu, W. Wang, L. Jiao et al., "SoProtector: safeguard privacy for native SO files in evolving mobile IoT applications," *IEEE Internet of Things Journal*, p. 1, 2019.

[8] Y. Zhong, S. Fong, R. Hu, R. Wong, and W. Lin, "A novel sensor data pre-processing methodology for the internet of things using anomaly detection and transfer-by-subspace-similarity transformation," *Sensors*, vol. 19, no. 20, p. 4536, 2019.

[9] H. S. Hassanein and M. A. O. Sharief, "big sensed data challenges in the Internet of Things," in *Proceedings of the 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Ottawa, Canada, 2017.

[10] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization: an enabling technique," *Data Mining and Knowledge Discovery*, vol. 6, no. 4, pp. 393–423, 2002.

[11] S. Ramírez-Gallego, S. García, H. Mouriño-Talín et al., "Data discretization: taxonomy and big data challenge," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 5–21, 2016.

[12] M. J. Abadi, L. Luceri, M. Hassan, C. T. Chou, and M. Nicoli, "A cooperative machine learning approach for pedestrian navigation in indoor IoT," *Sensors*, vol. 19, no. 21, p. 4609, 2019.

[13] Y.-C. Lin, C.-L. Lin, M.-D. Tsai, and L.-S. Chou, "Discretization of object-based lidar features for land cover classification," in *Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Fort Worth, TX, USA, July 2017.

[14] A. Gomaa, M. M. Abdelwahab, M. Abo-Zahhad, T. Minematsu, and R. I. Taniguchi, "Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow," *Sensors*, vol. 19, no. 20, p. 4588, 2019.

[15] S. C. Tan, "Improving association rule mining using clustering-based discretization of numerical data," in *Proceedings of the 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, Plaine Magnien, Plaine Magnien, Mauritius, December 2018.

[16] D. S. D. Cunha, R. S. Xavier, D. G. Ferrari, F. G. Vilasbôas, and L. N. de Castro, "Bacterial colony algorithms for association rule mining in static and stream data," *Mathematical Problems in Engineering*, vol. 2018, Article ID 4676258, 14 pages, 2018.

[17] R. Zamudio-Reyes, N. Cruz-Ramírez, and E. Mezura-Montes, "A multivariate discretization algorithm based on multiobjective optimization," in *Proceedings of the 2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, December 2017.

[18] D. Maryono, P. Hatta, and R. Ariyuana, "Implementation of numerical attribute discretization for outlier detection on mixed attribute dataset," in *Proceedings of the 2018 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia, March 2018.

[19] A. E. Rad, H. Nezamabadi-pour, and M. Eftekhari, "Improving LAIM discretization method for multi-label data using evolution strategy," in *Proceedings of the 2018 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, Bam, Iran, March 2018.

[20] B. S. Chlebus and S. H. Nguyen, "On finding optimal discretizations for two attributes," *Rough Sets and Current Trends in Computing*, Springer, Berlin, Germany, pp. 537–544, 1998.

[21] S. Garcia, J. A. Luengo, J. A. Sáez, V. López, and F. Herrera, "A survey of discretization techniques: taxonomy and empirical analysis in supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 734–750, 2013.

[22] A. K. C. Wong and D. K. Y. Chiu, "Synthesizing statistical knowledge from incomplete mixed-mode data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 6, pp. 796–805, 1987.

[23] C. H. Robert, "Very simple classification rules perform well on most commonly used datasets," *Machine Learning*, vol. 11, no. 1, pp. 63–90, 1993.

[24] K. Shehzad, "EDISC: a class-tailored discretization technique for rule-based classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1435–1447, 2012.

[25] R. Kerber, "ChiMerge: discretization of numeric attributes," in *Proceedings of the AAAI'92 Proceedings of the tenth national conference on Artificial intelligence*, San Jose, CA, USA, July 1992.

[26] H. Liu and R. Setiono, "Feature selection via discretization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 9, no. 4, pp. 642–645, 1997.

[27] C.-T. Su and J.-H. Hsu, "An extended Chi2 algorithm for discretization of real value attributes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 437–441, 2005.

[28] J. Y. Ching, A. K. C. Wong, and K. C. C. Chan, "Class-dependent discretization for inductive learning from continuous and mixed-mode data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 641–651, 1995.

[29] L. A. Kurgan and K. J. Cios, "CAIM discretization algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 2, pp. 145–153, 2004.

[30] C.-J. Tsai, C.-I. Lee, and W.-P. Yang, "A discretization algorithm based on class-attribute contingency coefficient," *Information Sciences*, vol. 178, no. 3, pp. 714–731, 2008.

[31] B. Tran, B. Xue, and M. Zhang, "A new representation in PSO for discretization-based feature selection," *IEEE Transactions on Cybernetics*, vol. 48, no. 6, pp. 1733–1746, 2018.

[32] W. Zhu, J. Wang, Y. Zhang, and L. Jia, "A discretization algorithm based on information distance criterion and ant colony optimization algorithm for knowledge extracting on

industrial database," in *Proceedings of the 2010 IEEE International Conference on Mechatronics and Automation*, Xi'an, China, August 2010.

[33] C.-Y. Chen, Z.-G. Li, S.-Y. Qiao, and S.-P. Wen, "Study on discretization in rough set based on genetic algorithm," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693)*, Xi'an, China, November 2003.

[34] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[35] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[36] S. Ramirez-Gallego, S. Garcia, J. M. Benítez, and F. Herrera, "Multivariate discretization based on evolutionary cut points selection for classification," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 595–608, 2016.

[37] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, Boston, MA, USA, 1989.

[38] Z. Pawlak, "Rough set theory and its applications to data analysis," *Cybernetics and Systems*, vol. 29, no. 7, pp. 661–688, 1998.

[39] M. Kusy and R. Zajdel, "Application of reinforcement learning algorithms for the adaptive computation of the smoothing parameter for probabilistic neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2163–2175, 2015.

[40] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[41] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing atari with deep reinforcement learning," 2013, https://arxiv.org/abs/1312.5602.

[42] L. Lv, S. Zhang, D. Ding, and Y. Wang, "Path planning via an improved DQN-based learning policy," *IEEE Access*, vol. 7, pp. 67319–67330, 2019.

[43] K. Sriwanna, T. Boongoen, and N. Iam-On, "An evolutionary cut points search for graph clustering-based discretization," in *Proceedings of the 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Khon Kaen, Thailand, July 2016.

[44] L. Zhang, H. Chang, and R. Xu, "Equal-width partitioning roulette wheel selection in genetic algorithm," in *Proceedings of the 2012 Conference on Technologies and Applications of Artificial Intelligence*, Tainan, Taiwan, November 2012.

[45] C. R. de Sá, C. Soares, and A. Knobbe, "Entropy-based discretization methods for ranking data," *Information Sciences*, vol. 329, no. C, pp. 921–936, 2016.

[46] Z. Ali and W. Shahzad, "Comparative study of discretization methods on the performance of associative classifiers," in *Procedings of the 2016 International Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, December 2016.

[47] D. Yan, D. Liu, and Y. Sang, "A new approach for discretizing continuous attributes in learning systems," *Neurocomputing*, vol. 133, no. 10, pp. 507–511, 2014.

[48] G. Zhang, Z. Wu, and L. Yi, "A remote sensing feature discretization method accommodating uncertainty in classification systems," in *Proceedings of the 8th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, Shanghai, China, June 2008.

[49] B. Wu, L. Zhang, and Y. Zhao, "Feature selection via cramer's V-test discretization for remote-sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2593–2606, 2014.

[50] W. Qu, D. Yan, S. Yu, H. Liang, M. Kitsuregawa, and K. Li, "A novel Chi2 algorithm for discretization of continuous attributes," in *Proceedings of the APWeb'08 Proceedings of the 10th Asia-Pacific web conference on Progress in WWW research and development*, Shenyang, China, April 2008.

[51] M. Huang, Q. Chen, and H. Wang, "A multivariable optical remote sensing image feature discretization method applied to marine vessel targets recognition," *Multimedia Tools and Applications*, vol. 79, no. 7-8, pp. 1–22, 2019.