electronics

MDPI

*Review*

# A Survey on Particle Swarm Optimization for Association Rule Mining

**Guangquan Li [1], Ting Wang [1], Qi Chen [1,*], Peng Shao [1,*], Naixue Xiong [2] and Athanasios Vasilakos [3,4]**

[1] School of Computer and Information Engineering, Jiangxi Agriculture University, Nanchang 330045, China
[2] Department of Computer, Mathematical and Physical Sciences, Sul Ross State University, Alpine, TX 79830, USA
[3] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China
[4] Center for AI Research (CAIR), University of Agder (UiA), 4064 Grimstad, Norway
* Correspondence: chenqihao@jxau.edu.cn (Q.C.); pshao@whu.edu.cn (P.S.)

**Abstract:** Association rule mining (ARM) is one of the core techniques of data mining to discover potentially valuable association relationships from mixed datasets. In the current research, various heuristic algorithms have been introduced into ARM to address the high computation time of traditional ARM. Although a more detailed review of the heuristic algorithms based on ARM is available, this paper differs from the existing reviews in that we expected it to provide a more comprehensive and multi-faceted survey of emerging research, which could provide a reference for researchers in the field to help them understand the state-of-the-art PSO-based ARM algorithms. In this paper, we review the existing research results. Heuristic algorithms for ARM were divided into three main groups, including biologically inspired, physically inspired, and other algorithms. Additionally, different types of ARM and their evaluation metrics are described in this paper, and the current status of the improvement in PSO algorithms is discussed in stages, including swarm initialization, algorithm parameter optimization, optimal particle update, and velocity and position updates. Furthermore, we discuss the applications of PSO-based ARM algorithms and propose further research directions by exploring the existing problems.

**Keywords:** association rule mining; particle swarm optimization algorithm; algorithm optimization; heuristic algorithm

check for updates

**Citation:** Li, G.; Wang, T.; Chen, Q.; Shao, P.; Xiong, N.; Vasilakos, A. A Survey on Particle Swarm Optimization for Association Rule Mining. *Electronics* **2022**, *11*, 3044. https://doi.org/10.3390/electronics11193044

Academic Editor: José L. Abellán

Received: 23 August 2022
Accepted: 21 September 2022
Published: 24 September 2022

## 1. Introduction

Association rule mining (ARM), as one of the core techniques of data mining (DM), can discover potentially valuable association relationships between itemsets from complex and fuzzy mixed data. The Apriori algorithm is one of the classical algorithms for ARM, which was first proposed by Agrawal et al. [1] and has gained the attention and research of many scholars in the past two decades. Akbar et al. [2] reviewed the research on ARM algorithms based on evolutionary computation published before 2019 and classified the evolutionary ARM algorithms into four major categories: evolutionary-based, swarm intelligence-based, physics-inspired, and hybrid approaches. Minakshi et al. [3] systematically assessed numerical association rule mining (NARM) algorithms to help scholars select a suitable algorithm when performing NARM mining. Djenouri et al. [4] studied the application of various metaheuristics algorithms in frequent itemset mining (FIM) and high-utility itemset mining (HUIM), including genetic algorithms (GA), ant colony optimization (ACO) algorithms, artificial bee colony (ABC), and particle swarm optimization (PSO). Arani et al. [5] also used a metaheuristic-based clustering mechanism when analyzing cloud workloads. In addition, Gan et al. [6] combined the design of convolutional neural networks (CNN) to address the shortcomings of traditional mining models that cannot extract temporal data effectively by constructing a new temporal association rule mining model. Similar papers include [7,8]. Badhon et al. [9] developed a systematic and structured analysis of ARM algorithms in terms of different aspects, including chromosome representation, genetic operators, and initial swarms. In addition, Kheirollahi et al. [10] combined the clustering algorithm with the PSO algorithm, which can be applied to recognize effective, good clusters and extract strong rules among reservoir data. Datta et al. [11] introduced the concept of flexible dissociation to propose a significant ARM framework with high associativity. In response to the rapid growth of data, Sanjay et al. [12] proposed a distributed ARM algorithm called adaptive-miner, which can dynamically change the mining algorithm according to the characteristics of the dataset. Among them, the PSO algorithm has already received the attention of many scholars at home and abroad since it was proposed because of its few adjustable parameters, high efficiency, easy implementation, and fast convergence speed. The current research shows that ARM using the PSO algorithms has higher operational efficiency compared with other data mining algorithms and can generate more accurate and valuable rules [13–21].

However, recent papers reviewed only a small fraction of the PSO-based ARM algorithms; however, there have been many papers published in recent years. Therefore, we aimed to provide a more comprehensive and multi-faceted survey of emerging research. We expect this survey can provide a reference for researchers in the field to help them understand the state-of-the-art PSO-based ARM algorithms. We downloaded all the articles containing PSO or ARM keywords in recent years, removed the less cited and less relevant ones, and selected the remaining 117 papers for use in our survey.

To analyze these papers more systematically and comprehensively, we divided the ARM algorithms according to the different pattern types. They were fuzzy association rule mining, rare association rule mining, numerical association rule mining, quantitative association rule mining, binary association rule mining, high-utility association rule mining, and classification association rule mining, and they are shown in blue in the inner circle of Figure 1. In addition, we compared them from three perspectives: biologically inspired, physically inspired, and other algorithms, and listed the key information of each algorithm under the different perspectives, as shown in the outer circle of Figure 1. The specific description and summary of the characteristics of each algorithm can be found in Section 3 of the article.

The remaining paper is organized as follows: In Section 2, the concept of ARM is described in detail, and it is divided into three categories: classification association rule mining (CARM) [22], fuzzy association rule mining (FARM) [23,24] and numerical association rule mining (NARM) [15,25–27]; their algorithm evaluation metrics are described in three sections. In Section 3, different algorithms in ARM are compared and categorized into three groups: bio-inspired, physics-inspired, and other algorithms, and in each group, a paragraph describing the algorithms and a summary of their features are provided. The process of the PSO-based ARM algorithm is described in detail in Section 4, including the binary conversion of data and the encoding process. Section 5 discusses the current status of research on the ARM algorithms based on PSO algorithms from four perspectives: swarm initialization, algorithm parameter optimization, optimal particle update, and velocity and position update. The applications of the PSO algorithm in ARM are summarized in Section 6. Finally, the paper concludes with concluding statements and future aspects in Section 7.

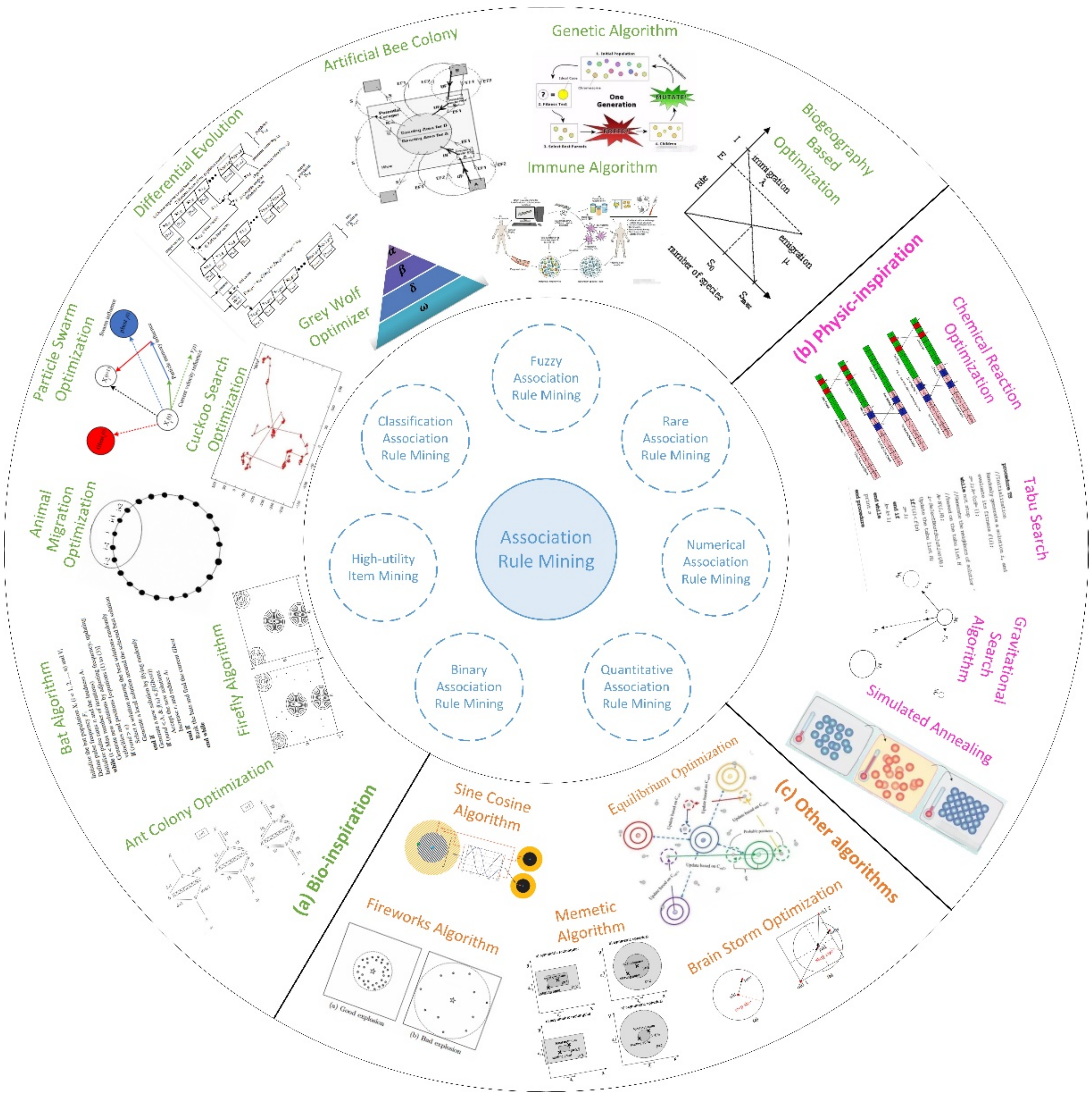


**Figure 1.** The research process of ARM.

## 2. Association Rule Mining

### 2.1. Definition of ARM

The definition of association rules was first introduced by Agrawal in 1993: $I = \{i_1, i_2, \cdots, i_n\}$ is called the itemsets, where $i_j \in \{0,1\}$ is called an item, $D = \{d_1, d_2, \cdots, d_m\}$ is called a database, and $d_k$ is called a transaction. A transaction is an itemset, i.e., a transaction is a subset $d_k \subseteq I$ of a transaction, and each transaction has a unique ID. Each rule consists of two different itemsets, $X$ and $Y$, where $X$ is called the premise, and Y is called the conclusion. Rules are represented as $X \rightarrow Y$, and $X, Y \subseteq I$.

### 2.2. Mode Types for ARM

#### 2.2.1. CARM

CARM is a widely used method in the real world. Compared with traditional ARM, CARM has the characteristics of high classification accuracy and adaptability, etc. Its mining is mainly divided into the following processes: in the preprocessing stage, if the data are continuous values, they need to be discretized; then, class association rules that satisfy the minimum support and confidence are mined in the database, and the generated association rules are pruned. Finally, the classifier is used to predict the data objects of unknown classes. Currently, this model is widely used in several fields, such as medicine and biology.

### 2.2.2. FARM

For traditional ARM, its itemsets are often discretized by dividing the intervals into continuous values [28], but the method suffers from the drawback of dividing the intervals too hard. Therefore, Chan et al. [23] proposed the concept of fuzzy association rules to divide the interval into multiple fuzzy sets by softening the division boundary with fuzzy sets and obtaining the probability of an element on a fuzzy set based on the fuzzy affiliation function to provide a smooth transition between the members and nonmembers in the itemset.

Let the itemset set $I = \{i_1, i_2, \cdots, i_n\}$, database $D = \{t_1, t_2, \cdots, t_m\}$, and each element $i_j$ in the itemset correspond to a fuzzy set $F_{ik} = \left\{f_{ik}^1, f_{ik}^2, \cdots, f_{ik}^l\right\}$. The fuzzy association rule for mining is that if $T$ is $A$, then $F$ is $B$. Where, $T = \{t_1, t_2, \cdots, t_p\}$, $F = \{f_1, f_2, \cdots, f_p\}$ are the itemsets, $A = \{a_1, a_2, \cdots, a_p\}$, $B = \{b_1, b_2, \cdots, b_p\}$ are the fuzzy sets corresponding to T and F, respectively. The rule premise $X = T$ is $A$, and the conclusion $Y = F$ is $B$. This is expressed as: when $X$ is satisfied, it can be concluded that $Y$ is also satisfied. For this purpose, we use the significance factor and the certainty factor to determine whether the rule is interesting or not.

Significance Factor: To generate a fuzzy association rule, it is first necessary to obtain the set of $k$ items with a significance factor above a user-specified threshold, denoted as $S_{<T,A>}$, which is calculated as follows:

$$S_{<T,A>} = \frac{\sum_{d_k \in D} \prod_{t_i \in T} \left\{\alpha_{a_j}(d_k[t_i])\right\}}{Total(D)} \tag{1}$$

$$\alpha_{a_j}(d_k[t_j]) = \begin{cases} m_{a_j} \in A(d_k[t_j]) & , if\ m_{a_j} > w \\ 0 & , otherwise \end{cases} \tag{2}$$

where $Total(D)$ indicates the total number of transactions in database $D$, $\alpha_{aj}(d_k[t_j])$ indicates the membership grade of $t_j$ in the $k$th records.

Certainty Factor: After the first step of calculation to obtain a large set of $K$ terms with significance factors above a given value $< Z, C >$, a deterministic factor is introduced to measure whether the rule is interesting or not, which is calculated as follows:

$$C_{<<T,A>,<F,B>>} = \frac{\sum_{d_k \in D} \prod_{z_i \in Z} \{\alpha_{c_i}(d_k[z_i])\}}{\sum_{d_k \in D} \prod_{t_j \in T} \left\{\alpha_{a_j}(d_k[t_j])\right\}} \tag{3}$$

$$\alpha_{c_i}(d_k[z_i]) = \begin{cases} m_{c_i} \in c(d_k[z_i]) & , if\ m_{c_i} > w \\ 0 & , otherwise \end{cases} \tag{4}$$

$$Z = T \cup F, C = A \cup B \tag{5}$$

### 2.2.3. NARM

In the real world, data are often represented in the form of numerical values, and the process of finding the association rules in numerical data is called NARM. In solving the NARM problem, most of the existing studies are divided into three methods: optimization, discretization, and distribution. Among them, the optimization methods can be divided into several sub-methods, such as swarm intelligence-based and evolution-based; the discretization methods are for discrete-continuous values, which can be further divided into supervised and unsupervised, static and dynamic, and other sub-methods; and the distribution methods include the mean, median, and variance. The division methods are shown in Figure 2.

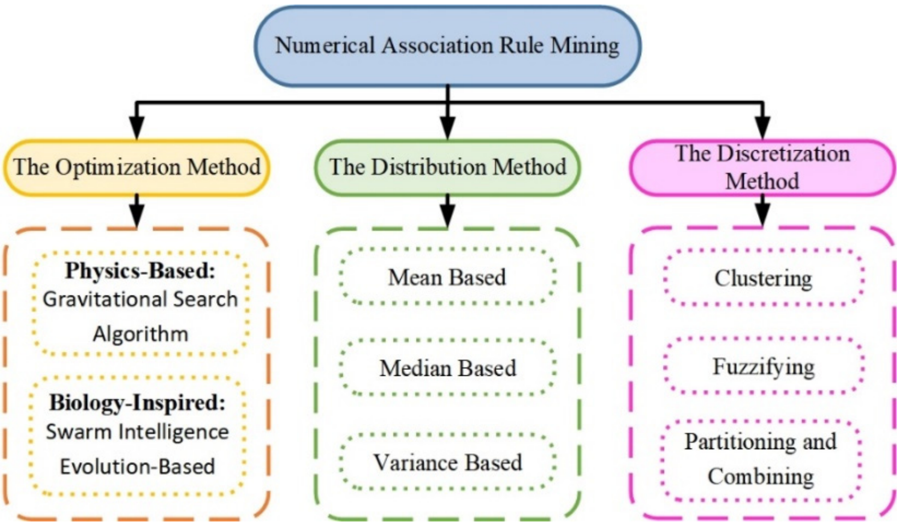

Numerical Association Rule Mining
The Optimization Method
The Distribution Method
The Discretization Method
Physics-Based: Gravitational Search Algorithm
Biology-Inspired: Swarm Intelligence Evolution-Based
Mean Based
Median Based
Variance Based
Clustering
Fuzzifying
Partitioning and Combining

**Figure 2.** Solution of NARM.

## 2.3. *Evaluation Metrics for ARM*

### 2.3.1. Basic Evaluation Indicators

To select the valuable rules from the possible frequent itemsets, some metrics are needed to filter and sift them. In most studies, the strength of a rule is described by support and confidence.

Support: Support indicates the frequency of the itemsets in the database. The support of the association rule $X \to Y$ is the ratio of the number of transactions in the database that also contain the itemsets $X$ and $Y$ to the total number of transactions.

$$Support(X \to Y) = P(X \cap Y) = \frac{X \cap Y}{all} \tag{6}$$

Confidence: The confidence level is used to measure the confidence level of a rule. The confidence level of the association rule $X \to Y$ is the ratio of the number of transactions in the database that contain both $X$ and $Y$ and the total number of transactions that contain $X$.

$$Confidence(X \to Y) = P(X|Y) = \frac{P(X \cap Y)}{P(X)} \tag{7}$$

When $Support(X \to Y)$ and $Confidence(X \to Y)$ are both greater than the minimum support threshold and the minimum confidence threshold, if $Lift(X \to Y) > 1$, then $X \to Y$ is a strong association rule, and vice versa.

### 2.3.2. Quantitative Evaluation Indicators

Most researchers use support and confidence to evaluate rules. The inherent drawback of this is that rare data items with less than the minimum support cannot be studied, and the support–confidence-based evaluation system generates a huge number of invalid rules; thus, other evaluation metrics need to be introduced to further mine the rules of interest to users [29].

Correlativity: Correlation reflects the degree of interdependence between $X$ and $Y$. If $Correlativity(X \to Y) < 1$, then it means that $X \to Y$ is an invalid strong correlation rule, and if $Correlativity(X \to Y) = 1$, then $X$ and $Y$ are independent of each other.

$$Correlativity(X \to Y) = \frac{P(X|Y)}{P(Y)} = \frac{Confidence(X \to Y)}{Support(Y)} \tag{8}$$

Effect: The degree of influence reflects the degree of correlation between $X$ and $Y$. It can filter invalid association rules and generate meaningful positive and negative association rules. $|Effect(X \to Y)|$ is larger, indicating a high degree of $X \to Y$ correlation.

$$Effect(X \to Y) = \frac{Confidence(X \to Y) - Support(Y)}{Max\{Confidence(X \to Y), Support(Y)\}} \tag{9}$$

Time: The timeliness reflects the value of time between $X$ and $Y$ and can help decision makers remove redundant or already outdated information. The temporal validity of the rules decreases exponentially with increasing time intervals. The temporal validity of support and confidence is calculated as shown in Equations (10) and (11), respectively.

$$Time\_Support(X \to Y) = \sum_{i=1}^{n} Support(X_i) \cdot e^{-\alpha \cdot |t_i - t_0|} \tag{10}$$

$$Time\_Confidence(X \to Y) = \sum_{i=1}^{n} Confidence(X_i) \cdot e^{-\alpha \cdot |t_i - t_0|} \tag{11}$$

where $X_i$ denotes the $i$th transaction in the $X$ events and $t_i$ denotes the time when the $i$th transaction in the $X$ events occurred.

#### 2.3.3. Qualitative Evaluation Indicators

Utility: Utility reflects the potential usefulness of some attributes in the rule and their importance, and $Utility \in [0,1]$. When $Utility = 1$, it means the attribute is the most important, and vice versa.

Concision: Simplicity is used to measure the comprehensibility of the association rule and is related to the arithmetic average of the number of items and the abstraction level of the antecedent items of the rule, and the weighted arithmetic average of the precedent items, whose weights depend on the importance of the antecedent and precedent items. The higher the number of items in a rule, the lower its comprehensibility, and the higher the level of abstraction contained in a rule, the lower its comprehensibility. The greater the simplicity of a rule, the more interesting it is to the user.

## 3. Comparison of Different ARM Algorithms

Due to the features of association rules in high-dimensional spaces, many scholars obtain better solutions through iterative heuristic processes. In this paper, we divided the ARM methods based on the heuristic algorithms into three main groups: bio-inspired, physics-inspired, and other algorithms, and provided a paragraph describing the algorithms and a summary of their features in each group.

### *3.1. Bio-Inspiration*

#### 3.1.1. Evolution-Based Algorithms

Some evolution-based algorithms are shown in Table 1.

**Table 1.** ARM, based on evolutionary algorithm.

| Algorithms | Authors | Technology | Advantages | Disadvantages |
|---|---|---|---|---|
| DE | Wang et al. [30] | Adaptive adjustment F and Cr | Enhanced algorithmic global search capability | High memory overhead |
| | Altay et al. [31] | DE-SCA, Multi-target, | Performs well on datasets with few attributes and many instances | High algorithmic complexity |
| | Guan et al. [32] | HGDE | Increased swarm diversity and high stability | Unable to handle high-dimensional data items |
| GA | Menaga et al. [33] | GAPPARM | High practicality | Easy to cause data loss |
| | Lin et al. [34] | Clustering the data | High accuracy and performance | Single objective |
| | Neysiani et al. [35] | Identify similarity between itemsets by association rules | High-quality rules | Long execution time |
| BBO | Giri et al. [36] | LGBBO-RuleMiner | High accuracy and a simple algorithm | Only for single target problems |
| | Arani et al. [37] | K-means, Bayesian learning | Reduce the delay, SLA violation ratio, cost, and energy consumption | High CPU usage |
| CSA | Mo et al. [38] | DMARICA | Short execution time, Highly scalable | Less integrity of generation rules |
| AIS | Husain et al. [39] | CLONALG | High accuracy | Accuracy is positively correlated with the number of iterations |
| | Danilo et al. [40] | CLONALG-GA | Performs better in sparse datasets | Not suitable for dense datasets |

Evolutionary algorithms are based on biological evolutionary mechanisms and are mostly inspired by the biological evolution of nature, including natural selection and genetics, which have higher robustness and wide adaptability, and efficiently solve complex problems that are difficult to solve in traditional optimization algorithms.

### 3.1.2. Swarm Intelligence-Based Algorithm

Some algorithms based on swarm intelligence are shown in Table 2.

**Table 2.** ARM, based on swarm intelligence algorithm.

| Algorithms | Authors | Technology | Advantages | Disadvantages |
|---|---|---|---|---|
| PSO | Tyagi et al. [41] | Multi-target, MOPSO-ARM | Performs well on sparse data | Calculated overload, low efficiency |
| | Kuo et al. [42] | Adaptive Archive Grid, MOPSO | Automatically finds the best interval between datasets; no data preprocessing is required. | Fewer targets to consider |
| | Baro et al. [14] | Guided search strategy | High-quality rules and short calculation times | Lower average fitness values |
| | Agarwal et al. [43] | Multi-objective | Balancing global and local search capabilities | Not applicable to quantitative association rules |
| | Moslehi et al. [44] | GA-PSO | No predefined minimum support and confidence levels are required | Generate redundant rules |

**Table 2.** *Cont.*

| Algorithms | Authors | Technology | Advantages | Disadvantages |
|---|---|---|---|---|
| CSO | Devi et al. [45] | HGWCSO-ETSVM, Min-Max | High accuracy and precision | Reduced overall system performance |
| AMO | Son et al. [46] | ARM-AMO | Reduces the time and memory required for frequent itemset generation | Lower quality rules |
| BAT | Kamel et al. [27] | MSBARM, loop strategy | Rule quality is higher than other ways algorithm | Unable to handle large databases |
| FA | Pradeep et al. [47] | BFFO-TA, Feature Selection | No redundant rules are created | Not applicable to multi-objective problems |
| GWO | Pazhaniraa et al. [48] | BGWO-HUI | Low time complexity | High memory overhead |
| | Yildirim et al. [49] | Adaptive multi-target intelligent search, MOGWO | For discrete, quantitative, and mixed datasets with high comprehensibility | Not suitable for distributed datasets |
| | Chantar et al. [50] | SVM, Elite Cross, BGWO | High accuracy | High time complexity |
| ABC | Akbar et al. [51] | ABC4ARH | Highly efficient and stable | Less scalability and higher complexity |
| HHO | Turabieh et al. [52] | HHO-KNN | Enhanced ability to think outside the local optimum | Less stability |
| ALO | Dong et al. [53] | ALO-ARM | The optimized search process has high efficiency | Not suitable for large datasets |

Inspired by the behavior of insect clusters, herds of animals, flocks of birds, etc., the swarm intelligence algorithm can change the search direction through collaboration and information sharing among individuals of the swarm; it has a faster convergence rate and can obtain a better solution than evolutionary algorithms; the poor performance of individuals does not affect the solution of the problem by the whole swarm.

### 3.2. Physics-Inspired

A physics-inspired algorithm is a swarm optimization algorithm that models certain physical objective laws. Individuals of the swarm collaborate and exchange information with each other in the search space through rules inspired by some physical principles. Some physically inspired algorithms are shown in Table 3.

**Table 3.** ARM, based on the physics-inspired algorithm.

| Algorithms | Authors | Technology | Advantages | Disadvantages |
|---|---|---|---|---|
| SA | Nawaz et al. [54] | HUIM-SA | HUIM-SA varies linearly with the number of iterations | Lack of mutation mechanisms |
| | Holman et al. [55] | GA-SA, Confusion Matrix | High accuracy | Accuracy depends on dataset size |
| TS | Chou et al. [56] | QTS | Obtain more rules | Algorithm performance degrades when the dataset is too small |
| CRO | Abir et al. [57] | CRO | Low algorithm complexity; no need to specify minimum support | Less stability, calculated overload |
| GSA | Taradeh et al. [58] | HGSA, Crossover, variation | Fast convergence and high-quality rules | Not applicable to mixed datasets |

### *3.3. Other Algorithms*

Some of the algorithms are shown in Table 4.

**Table 4.** ARM, based on other heuristic algorithms.

| Algorithms | Authors | Technology | Advantages | Disadvantages |
|---|---|---|---|---|
| FA | Eva et al. [59] | Combining k-means for web data clustering | Faster convergence; better performance at the minimum distance; shorter computation time | The placement strategy of the initial center of mass needs to be further optimized |
| MA | Ting et al. [24] | Optimizing the affiliation function in FARM | Improve the search capability of the algorithm and obtain high fuzzy support | Less scalability |
| SCA | Laith et al. [60] | SCAGA | Maximum classification accuracy with minimal attributes | The imbalance between global search capability and local search capability |
| BSO | Djenouris et al. [61] | HBSO-TS | Short execution time | Poor results in large datasets |
|  | Ma et al. [62] | PLBSO | Parallelized processing with low computational costs | Less performance when dealing with quantitative rules |
|  | Youcef et al. [63] | GBSO-Miner | Adaptable to large text and graphical databases | Thread divergence exists |
| EO | Malik et al. [64] | DEOA-CRM | High-quality rules and highly interpretable algorithms | Poor performance in some datasets |

Other heuristic algorithms are inspired by natural phenomena, mathematical principles, man-made activities, etc. They include the fireworks algorithm, brainstorming algorithm, backtracking search algorithm, and sine and cosine algorithms, etc.

## 4. Particle Swarm Optimization Algorithms

### *4.1. Standard PSO Algorithms*

The particle swarm optimization (PSO) algorithm [65] is an intelligent optimization algorithm studied by Kenney and Eberhart in 1995 for the swarm motion behavior of bird and fish flocks, and the basic idea is to seek the global optimal solution through mutual collaboration and information sharing among individuals. In this model, an individual is regarded as a particle and a flock of birds as a particle swarm. Suppose there are m particles in a D-dimensional target search space, where the position of the $i$th particle $(i = 1, 2, 3, \ldots, m)$ is denoted as $X_i = \left(x_i^1, x_i^2, \cdots, x_i^D\right)$, i.e., the position of the $i$th particle in the D-dimensional target search space is $X_i^D$. Substituting $X_i$ into the objective function calculates its fitness value and measures its merit. The best position experienced by each particle is denoted as $P_i = \left(p_i^1, p_i^2, \cdots, p_i^D\right)$, and the best position experienced by the whole particle swarm is denoted as $P_g = \left(p_g^1, p_g^2, \cdots, p_g^D\right)$. The velocity of particle $i$ is denoted as $V_i = \left(v_i^1, v_i^2, \cdots, v_i^D\right)$. Particle $i$ updates itself in each iteration by Equations (12) and (13) for $P_i$,$P_g$.

$$v_i^{d+1} = wv_i^d + c_1 r_1 \left(p_i - x_i^d\right) + c_2 r_2 \left(p_g - x_i^d\right) \tag{12}$$

$$x_i^{d+1} = x_i^d + \alpha v_i^d \tag{13}$$

where $i = 1, 2, 3, \ldots, m$, $d = 1, 2, 3, \ldots D$, $w$ is the inertia weight, $v_i^d$ is the velocity of the $i$th particle at the $d$th iteration, $c_1, c_2$ is a nonnegative constant, $r_1, r_2$ is an internally uniformly distributed random number, and $\alpha$ is a controlling factor that controls the velocity weight. The pseudocode is shown in Algorithm 1.

**Algorithm 1** Procedure of standard PSO algorithm

```
for each particle $i = 1, 2, \ldots, N$ do
    (a) Initialize the particle's position and velocity.
    (b) Evaluate the particle's fitness value;
    (c) Update the particle's Pbest;
    (d) Update the swarm's Gbest;
while termination criteria is not met do
    for each particle $i = 1, 2, \ldots, N$ do
        (a) Update particle's velocity using Equation (12);
        (b) Update particle's position using Equation (13);
        (c) Evaluate particle's fitness value;
        if (x[i]) < f(Pbest[i]) then
         Update the best known position of particle i: Pbest[i]=x[i];
         if (Pbest[i]) < f(Gbest) then
         Update the swarm's best known position: Gbest=Pbest[i];
        (d) t = t + 1;
return Gbest
```

Among them, we illustrate some of the more important parameters.

Maximum velocity $V_{\max}$: In each dimension, the velocity of the particle $v_i \in (-V_{\max}, V_{\max})$ reduces the possibility of the particle leaving the search space. Where $V_{\max} = k \cdot x_{\max}(0.1 < k < 1)$, the purpose is to limit the maximum distance the particle can move.

Inertia weight w: The larger the w, the greater the global searchability and the easier it is for the particle to escape the local extremes; however, it also reduces search efficiency and makes converging more difficult, and vice versa. $w$ usually varies linearly ($0 < w < 1$), and the variation equation is shown in Equation (14).

$$w_i = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \cdot iter \tag{14}$$

where $iter_{\max}$ is the maximum number of iterations and $iter$ is the number of current iterations.

### *4.2. Improved PSO Based on Algorithm Design*

#### 4.2.1. Swarm Initialization

The main function of swarm initialization is to generate several particles randomly, which has a significant impact on the diversity and convergence of the algorithm. Wang et al. [66] used the roulette wheel selection method and non-expectation coding vector construction strategy to initialize the swarm and select the initial optimized value of the next-generation swarm by a certain probability. Equation (15) represents the probability of itemset I being selected, Equation (16) represents the hetero set of the coding vector and Equations (17)–(19) represent the velocity update in the generated coding vector:

$$S_i = \frac{fitness_i}{\sum_{k=1}^{|SHUI|} fitness_k} \tag{15}$$

$$BitDiff(Vec_i, Vec_j) = \{num | 1 \le num \le len, b_{num}(Vec_i) \oplus b_{num}(Vec_j) = 1\} \tag{16}$$

$$v_i = v_{i1} + v_{i2} + v_{i3} \tag{17}$$

$$v_{i2} = [|BitDiff(p_i, pbest_i)| \cdot r_1] \tag{18}$$

$$v_{i3} = [|BitDiff(p_i, pbest_i)| \cdot r_2] \tag{19}$$

where $\mid SHUI \mid$ denotes the number of elements in the set $SHUI$, $Fitness_i$ denotes the adaptation value of the $i$th itemset, $b_{num}(Vec_i)$ denotes the value of $Vec_i$ in position num, and $\oplus$ denotes the iso-or operation.$v_{i1}$ defaults to 1.

Kalka et al. [67] introduced the chemical reaction optimization (CRO) algorithm to quickly generate the optimal set of solutions as the initial solution of the PSO algorithm and use the selection operator to improve the quality of the optimal solution and enhance the overall performance of the algorithm. Equation (20) is used to determine whether the generated latest solution is accepted after colliding through two chemical reactions:

$$f(R_j) = \varphi_1 \cdot MS_R + \varphi_2 \cdot EC_{Rj} + \varphi_3 \cdot \varepsilon \tag{20}$$

where $\varphi_1 + \varphi_2 + \varphi_3 = 1$.

Hematpour et al. [68] introduced a nonlinear dynamical chaotic system to initialize the swarm positions. Equation (21) represents the first class of Chebyshev polynomials, $\alpha \in (0, 2)$, and Equation (22) represents the second class of Chebyshev polynomials, which can be simplified to a logistic mapping, $\alpha \in (\frac{1}{2}, \infty)$, and $\alpha \in (\frac{1}{3}, 3)$ in Equation (23):

$$\Phi_2^{(1)} = \frac{\alpha^2 \cdot (2x-1)^2}{4x(1-x) + \alpha^2 \cdot (2x-1)^2} \tag{21}$$

$$\Phi_2^{(2)} = \frac{4 \cdot \alpha^2 \cdot x \cdot (1-x)}{1 + 4 \cdot (\alpha^2 - 1) \cdot x \cdot (1-x)} \tag{22}$$

$$\Phi_2^{(1,2)} = \frac{\alpha^2 \cdot x \cdot (4x-3)^2}{\alpha^2 \cdot x \cdot (4x-3)^2 + (1-x) \cdot (4x-1)^2} \tag{23}$$

Li et al. [69] used an S-type transfer function to initialize the swarm position by converting the inertia weights based on feature weighting into probability values. Let $z$ be the variable, Equation (24) represents the inertia weight normalization process, Equation (25) represents the conversion of $z$ to $z\prime$ by a standard logistic function, Equation (26) represents the conversion of $z\prime$ to reduce the upper limit of $z\prime$, and p represents the initial swarm position:

$$z = \frac{2 \cdot \sigma \cdot (w - w_{\min})}{w_{\max} - w_{\min}} - \sigma \tag{24}$$

$$z' = \frac{1}{1 + e^{-z}} \tag{25}$$

$$p = \lambda \cdot z' \tag{26}$$

where $p \in (0, \lambda)$, $\lambda \in (0.5, 1)$, $z \in [-\sigma, \sigma]$, $\sigma \in R^+$.

Wang et al. [70] used the principal component analysis (PCA) technique to map high-dimensional data to low-dimensional data, retaining variance information and ranking it by priority, and initializing the mixture by selecting significant features. Equation (27) represents the normalization process of the PCA mean and variance, Equation (28) represents the PCA mapping the original data into new coordinates to maximize the variance, and Equation (29) represents the swarm position obtained after eigendecomposition of the covariance matrix:

$$\begin{cases} u = \frac{1}{m} \cdot \sum_{i=1}^{m} x^{(i)} \\ \sigma_j^2 = \frac{1}{m} \cdot \sum_{i=1}^{m} x_j^{(i)} \end{cases} \tag{27}$$

$$\frac{1}{m} \cdot \sum_{i=1}^{m} \left(x^{(i)^T} \cdot u\right)^2 = \frac{1}{m} \cdot \sum_{i=1}^{m} \left(u^T \cdot x^{(i)} \cdot x^{(i)^T}\right)^2 = u^T \cdot \left(\frac{1}{m} \cdot \sum_{i=1}^{m} x^{(i)} \cdot x^{(i)^T}\right) \cdot u \tag{28}$$

$$y^{(i)} = \begin{bmatrix} u_1^T \cdot x^{(i)} \\ u_2^T \cdot x^{(i)} \\ \vdots \\ u_k^T \cdot x^{(i)} \end{bmatrix} \tag{29}$$

where $x^{(i)}$ denotes the position of the $i$th particle in the swarm, $j$ denotes the $j$th dimension of $x^{(i)}$, u denotes the mean of the $i$th particle, $\sigma^j$ denotes the standard deviation of the $j$th dimension, and $y^{(i)}$ denotes the new position generated by the $i$th particle.

Bangyal et al. [71] introduced the Well technique with the help of low deviation sequences to make it applicable to high-dimensional optimization problems. Sobia et al. [72] reviewed a variety of swarm initialization strategies for the PSO algorithms and analyzed and compared each strategy in detail by the QRS method, including the performance of SO-PSO, H-PSO, WE-PSO, etc. in 15 benchmark tests. It was found that KE-PSO performs better in high-dimensional problems.

### 4.2.2. Algorithm Parameter Optimization

In the PSO algorithms, algorithm parameters play a great role in balancing the global search ability and local searchability, which can effectively improve swarm diversity and speed up algorithm convergence. Common parameter optimization includes inertia weights w, learning factors c1 and c2, the introduction of other factors, etc. Tian et al. [73] adaptively controlled the inertia weights through the fitness function, and the interaction of the two helps to control the swarm diversity. Equation (30) is used to assess swarm diversity, and Equation (31) is used to calculate inertia weights:

$$f_{Div}(t) = \log_{10}(\frac{1}{m} \cdot \sum_{j=t-m+1}^{t} (f_{avg}(j) - f_{Asvg}(t))^2) \tag{30}$$

$$w(t) = \begin{cases} w_{\min} + (w_{\max} - w_{\min}) \cdot f_{Aavg}(t) & , f_{Div}(t) < \alpha \\ w_{\min} + (w_{\max} - w_{\min}) \cdot (\frac{T-t}{T})^2 & , f_{Div}(t) \geq \alpha \end{cases} \tag{31}$$

where $f_{avg}(t)$ denotes the mean value of adaptation in iteration $t$.

Li et al. [74] divided the swarm into three subsets by the fitness value and assigned different inertia weights, and Equation (32) represents the process of dynamically updating the inertia weights by introducing the cloud variation method when the particle fitness value is lower than the average fitness value of the swarm:

$$w = 0.9 - 0.7 \cdot e^{-\frac{(f_i - Ex)^2}{2(En')^2}} \tag{32}$$

Zhang et al. [75] introduced a multi-stage strategy to improve the inertia weights through the cosine function. Equations (33)–(35) show the improvement process:

$$w_{\cos}(t) = \frac{(w_{ini} + w_{fin})}{2} + \frac{(w_{ini} - w_{fin})}{2} \cdot \cos(\frac{I(t) \cdot \pi}{t_{\max}}) \tag{33}$$

$$I(t+1) = I(t) + a, I(1) = 0 \tag{34}$$

$$a = \begin{cases} a_1 & , \frac{I(t)}{6} \leq t_{\max} \\ a_2 & , \frac{t_{\max}}{6} < I(t) \leq \frac{5 \cdot t_{\max}}{6} \\ a_3 & , \frac{5 \cdot t_{\max}}{6} < I(t) \leq t_{\max} \end{cases} , \textit{where } \frac{1}{6 \cdot a_1} + \frac{1}{3 \cdot a_2} + \frac{1}{6 \cdot a_3} = 1 \tag{35}$$

Ankit et al. [76] introduced a mapping function based on binomial probability and adaptively updated the inertia weights. Equations (36) and (37) represent the updating process:

$$w = w_{\min} + (w_{\max} - w_{\min}) \cdot \sum_{k=0}^{X} \begin{pmatrix} N \\ k \end{pmatrix} \cdot p^k \cdot q^{N-k} \tag{36}$$

$$X = X(i,t) = \sum_{i=1}^{N} I(i,t) \tag{37}$$

where $I(i,t)$ denotes the $i$th particle at the time $t$ position function, $X$ denotes the total number of particles passing through the binomial probability improvement position, p denotes the improvement success probability, q denotes the failure probability, the particle swarm is independent of each other, and $p = q = 0.5$.

Komarudin et al. [77] combined the multi-dimensional direct fuzzy signature, parametric feedback success counter, and CBPE methods to transform the fuzzy process into aggregation equations for the adaptive updating of inertia weights. Equation (38) represents the structural hierarchy of the inertia weights, Equation (39) represents the best particle to reach the minimization goal, Equation (40) represents the best particle percentage, Equation (41) represents the particle distribution in the search space, and Equation (42) represents the *NCBPE* normalization process for measuring the best fitness value of the solution:

$$w = \begin{bmatrix} w_{\max} \\ \begin{bmatrix} w_{\min} \\ \begin{bmatrix} PS \\ \begin{bmatrix} diversity \\ div_{\max} \end{bmatrix} \\ NCBPE \end{bmatrix} \end{bmatrix} \end{bmatrix} \tag{38}$$

$$SC_i(t) = \begin{cases} 1 & , f(x_{id}(t)) < f(P_{id}(t)) \\ 0 & , else \end{cases} \tag{39}$$

$$PS(t) = \frac{1}{N} \cdot \sum_{i=1}^{N} SC_i(t) \tag{40}$$

$$diversity = \frac{1}{N} \cdot \sum_{i=1}^{N} \sqrt{\sum_{d=1}^{D} (x_{id} - g_d)^2} \tag{41}$$

$$NCBPE = \frac{CBPE - CBPE_{\min}}{CBPE_{\max} - CBPE_{\min}} \tag{42}$$

Fan et al. [78] analyzed a variety of improved inertia weights and proposed a time-varying adaptive inertia weighting strategy, and Equation (43) represents the update strategy:

$$w(t) = (w_{\max} - w_{\min}) \cdot \frac{f(P_{nbest})_{avg}}{f(P_{tbest})_{avg}} + \frac{w_{\max} \cdot t}{MaxIter} \tag{43}$$

where $f(P_{nbest})_{avg}$ and $f(P_{tbest})_{avg}$ denote the average fitness values of the global best position and the local best position in iteration $t$.

Chrouta et al. [79] proposed a factor selection strategy algorithm, FMSPSO, based on particle swarm number selection to adaptively adjust different parameter selection strategies during a swarm search to improve the algorithm's ability to find global search and avoid premature convergence, which is based on the problem that MSPSO fails to obtain the optimal solution due to premature convergence during the search process. Equation (44) represents the adaptive inertia weight updating strategy, and Equation (45) represents the learning factor updating strategy:

$$w_s = (w_{ini} - w_{end}) \cdot \operatorname{artanh}(\delta \cdot (1 - \frac{S}{S_{\max}})^{m}) + w_{ini} \tag{44}$$

$$c = (c_{ini} - c_{end}) \cdot \left[ 1 - \left( \frac{k}{k_{\max}} \right)^{\frac{s^m}{e^{s^m}}} \right] + c_{ini} \tag{45}$$

where $w_{ini} = 0.4$, $w_{end} = 0.9$, $\delta$ and $m$ are control factors, and $s$ is the number of current iterations.

Jeremiah et al. [80] introduced a shrinkage factor in the search process to ensure a smooth convergence of the algorithm to the global optimal solution to resolve the problem of premature convergence of the algorithm. They introduced a gravitational search algorithm to improve the global search ability of the algorithm and enhanced the stagnant particle activity through an adaptive response strategy. Equation (46) represents the shrinkage factor update strategy based on the cosine function, and Equation (47) represents the inertia weight update strategy. Equation (48) represents the relationship between the parameters:

$$k(t) = \frac{\cos(\frac{\pi}{G_{\max}} \cdot T + 2.5)}{4} \tag{46}$$

$$\begin{cases} w = w_{end} + (W_{start} - w_{end}) \cdot (1 - \frac{T}{G_{\max}}) & , if\ P_{gd} \neq x_{id} \\ w = w_{end} & , if\ P_{gd} = x_{id} \end{cases} \tag{47}$$

$$w \geq \frac{c_1 + c_2}{2} - \frac{1}{k(t)} \tag{48}$$

4.2.3. Optimal Particle Update

The self-renewal of particles in the PSO algorithm varies with the local optimal position $P_i$ and the global optimal position $P_g$, which has a greater ability to guide the swarm evolution. Li et al. [81] combined the gray wolf optimization (GWO) algorithm with the worst particle for the evolution strategy and the optimal particle perturbation strategy to improve the swarm diversity, and the elite swarm guidance strategy to guide the particle swarm for enveloping the search and enhance the global searchability. The perturbation strategy is shown in Equations (49) and (50):

$$N_{best}(t) = sign(rand - 0.5) \cdot e^{(-avg\_r) \cdot G_{best}(t) + G_{best}(t)} \tag{49}$$

$$G_{best}(t) = \begin{cases} N_{best}(t) & , if(fit(N_{best}(t)) < fit(G_{best}(t))) \\ G_{best}(t) & , otherwise \end{cases} \tag{50}$$

where $avg\_r$ denotes the average distance of all particles to the center position. $N_{best}(t)$ denotes the position after perturbation.

Keshavamurthy et al. [82] used a stochastic inertia weight strategy to balance the global and local search and adjusted the inertia weights by simulating the annealing probability, and the algorithm gradually approached the optimal solution as the temperature decreased. Equations (51)–(53) represent the inertia weight update strategy:

$$Tt = \frac{f_{avg}^t}{f_{best}^t} - 1 = \frac{\frac{1}{m}\sum_{i=1}^m f_i^t}{f_{best}^t} - 1 \tag{51}$$

$$p = \begin{cases} 1 & , if \min\limits_{1 \leq i \leq m} f_i^{t-k} \leq \min\limits_{1 \leq i \leq m} f_i^t \\ e^{-(\frac{\min\limits_{1 \leq i \leq m} f_i^{t-k} - \min\limits_{1 \leq i \leq m} f_i^t}{Tt})} & , if \min\limits_{1 \leq i \leq m} f_i^{t-k} > \min\limits_{1 \leq i \leq m} f_i^t \end{cases} \tag{52}$$

$$w = \begin{cases} \alpha_1 + \frac{r}{2} & , p \geq r \\ \alpha_2 + \frac{r}{2} & , p < r \end{cases} \tag{53}$$

where $Tt$ denotes the cooling temperature, $p$ denotes the simulated annealing probability, $k$ is an integer multiple of $t$, $r$ is a random number within [0,1], $\alpha_1$, $\alpha_2$ are constants within [0,1], and $\alpha_1 > \alpha_2$.

Han et al. [83] used the Karush–Kuhn–Tucker proximity measure to determine the contribution of optimally ranked particles to swarm convergence and introduced a gravitational search algorithm and an adaptive reset velocity strategy to enhance swarm diversity and avoid falling into local extremes. Equation (54) calculates the mass of the particle, and Equation (55) calculates the force of the $j$th particle on the $i$th particle to calculate the value of the particle's contribution to the swarm convergence in Equation (56) to decide whether to guide other particles as the optimal particle. Equations (57) and (58) represent the stalled particle generation process:

$$m_i^t = 1 - \frac{0.9 \cdot fit_i^t}{worst^t} \tag{54}$$

$$f_{ij}^d(t) = G(t) \cdot \frac{m_i^t + m_j^t}{R_{ij} + \varepsilon} \cdot (x_j^d(t) - x_i^d(t)) \tag{55}$$

$$a_i^d(t) = \frac{\sum_{j \in kbest, j \neq 1} rand_j \cdot f_{ij}^d(t)}{m_i^t} \tag{56}$$

$$pbest_{i,d} = \Phi + sdamp \cdot v_{mean} \tag{57}$$

$$v_{mean} = \frac{1}{N} \cdot \sum_{i=1}^N \sqrt{\frac{1}{D} \cdot \sum_{j=1}^D (v_{i,d})^2} \tag{58}$$

where $fit_i^t$ denotes the fitness value of the $i$th particle in the $t$th iteration, $worst^t$ denotes the fitness worst value, $R_{ij}$ denotes the Euclidean distance between particles $i$ and $j$, $G(t)$ denotes the gravitational coefficient, and $rand_j$ is a random number uniformly distributed between [0,1].

Miao et al. [84] sorted each particle in the current swarm by the nonDS function to obtain the kth iteration of the non-dominated particle set and optimally combined it with the kth iteration of the particle set to obtain the current non-dominated particle set $Arc_k$, where the particles choose their personal best position, $Pbest$, in $Arc_k$ according to the Euclidean distance. This effectively improves the quality of $P_{best}$ and prevents the algorithm from falling into local extremes. Equation (59) calculates the degree of influence of the optimal particles, and Equation (60) calculates Att, the size of which is proportional to the degree of influence:

$$dist(x,y) = \begin{cases} Att & , if\ x = y \\ -Att & , otherwise \end{cases} \tag{59}$$

$$Att = Att_{\max} - (Att_{\max} - Att_{\min}) \cdot (1 - \frac{gen}{G_{\max}})^5 \tag{60}$$

where $gen$ denotes the current iteration number and $G_{\max}$ denotes the maximum iteration number.

Zhu et al. [85] show that, based on the elastic collision principle and the law of momentum conservation, the worst particle of the swarm fitness value collides with the best particle of the fitness, and the worst particle obtains the best particle velocity. Equations (61) and (62) represent the calculation of the worst particle velocity, Equation (63) represents the particle velocity update strategy, and Equation (64) represents the particle position update strategy:

$$v_1 = Z \cdot e^{-(x_{worst} - x_{rand})^2} \tag{61}$$

$$Z = \frac{1}{2} \cdot ((1 - 2 \cdot \frac{t}{T})^3 + 1) \tag{62}$$

$$\begin{cases} m_1 v_1 = m_1 v_1' + m_2 v_2' \\ \frac{1}{2} \cdot m_1 v_1^2 = \frac{1}{2} \cdot m_1 v_1'^2 + \frac{1}{2} \cdot m_2 v_2'^2 \end{cases} \Rightarrow \begin{cases} v_1' = \frac{m_1 - m_2}{m_1 + m_2} \cdot v_1 \\ v_2' = \frac{2 \cdot m_1}{m_1 + m_2} \cdot v_2 \end{cases} \tag{63}$$

$$\begin{cases} X_{worst}' = X_{worst} + v_1' \\ X_{best}' = X_{best} + v_2' \end{cases} \tag{64}$$

where $X_{worst}$ denotes the particle with the worst adaptation value, $X_{rand}$ denotes the random particle position, $T$ denotes the nonlinear factor, $m_1$ denotes the adaptation value of the worst particle, and $m_2$ denotes the adaptation value of the best particle.

Fu et al. [86] introduced the RBF model to generate the global optimum and individual optimum through MATLAB's fmincon toolbox. While the existence of a large number of uncertain particles in the search space easily leads to local convergence of the algorithm, the authors also introduced the MaxEI method for pre-screening to measure the particle potential and uncertainty; thus, the global optimum has both maximum potential and maximum uncertainty.

### 4.2.4. Speed and Position Update

In the particle swarm optimization algorithms, velocity and position are properties specific to particles. Song et al. [87] combined successive higher Bessel curves and adaptive fractional-order velocities with particle evolution states to apply perturbations to the swarm to enhance its ability to jump out of local constraints. Equation (65) denotes the average distance of the $i$th particle from $g_{best}$ in the $k$ iterations, Equation (66) denotes the kth iteration and the swarm evolution state, Equation (67) denotes the swarm evolution adaptive adjustment factor, and Equations (68) and (69) denote the particle velocity and position update formulas:

$$d_i^k = \frac{1}{S-1}\sum\nolimits_{j=1,j\neq i}^{S}\sqrt{\sum\nolimits_{k=1}^{D}\left(x_i^k - x_j^k\right)^2} \tag{65}$$

$$E_f^k = \frac{d_{gb}^k - d_{\min}^k}{d_{\max}^k - d_{\min}^k} \tag{66}$$

$$\alpha = 0.9 - \frac{1}{1+e^{-E_f^k}} \cdot \frac{k}{k_{\max}} \tag{67}$$

$$\begin{aligned} v_i^{k+1} &= \alpha \cdot v_i^k + \tfrac{1}{2} \cdot \alpha \cdot (1-\alpha) \cdot v_i^{k-1} + \tfrac{1}{6} \cdot \alpha \cdot (1-\alpha) \cdot (2-\alpha) \cdot v_i^{k-2} \\ &+ \tfrac{1}{24} \cdot \alpha \cdot (1-\alpha) \cdot (2-\alpha) \cdot (3-\alpha) \cdot v_i^{k-3} \\ &+ c_1 \cdot r_1 \cdot (x_{ib}^k - x_i^k) + c_2 \cdot r_2 \cdot (x_{gb}^k - x_i^k) \end{aligned} \tag{68}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{69}$$

where $d_{\max}^k, d_{\min}^k$ denotes the maximum and minimum values of the distance from a particle to other particles in the swarm, and $d_{gb}^k$ denotes the average distance of the $g_{best}$.

Tian et al. [73] introduced the particle with the best comprehensive evaluation of the swarm fitness value and diversity in the current iteration as the latest guiding particle to guide the swarm evolution, together with the local and global optimum. Equation (70) represents the comprehensive evaluation of the $i$th particle, and Equation (71) represents the velocity update under the latest guiding particle:

$$f_{Com}(i) = d_1 \cdot f_{Fit}(i) + d_2 \cdot f_{Dis}(i) \tag{70}$$

$$v_i^d(t+1) = w \cdot v_i^d(t+1) + c_1 \cdot r_1 \cdot (p_i^d - x_i^d(t)) + c_2 \cdot r_2 \cdot (g^d - x_i^d(t)) + c_3 \cdot r_3 \cdot (f_{Com}^d(i) - x_i^d(t)) \tag{71}$$

where $d_1 = d_2 = 0.5$ and $d_1 + d_2 = 1$, $f_{Fit}(i)$ denotes the $i$th particle fitness value, and $f_{Dis}(i)$ denotes the $i$th particle diversity evaluation, i.e., the distance to the central particle $c_1 = c_2 = c_3 = 1.49445$.

Mohamed et al. [88] and Suman et al. [89] introduced the GWO algorithm to improve the global search ability of the algorithm. Equation (72) represents $\alpha, \beta, \delta$ the distance between the three wolves and other individuals, Equation (73) represents the current position of the three wolves, and Equation (74) represents the current gray wolf position:

$$\left\{ \begin{array}{l} \vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right| \\ \vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}\right| \\ \vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}\right| \end{array} \right\} \tag{72}$$

$$\left\{ \begin{array}{l} \vec{X}_1 = \left|\vec{X}_\alpha - \vec{A}_1 \cdot \left(\vec{D}_\alpha\right)\right| \\ \vec{X}_2 = \left|\vec{X}_\beta - \vec{A}_2 \cdot \left(\vec{D}_\beta\right)\right| \\ \vec{X}_3 = \left|\vec{X}_\delta - \vec{A}_3 \cdot \left(\vec{D}_\delta\right)\right| \end{array} \right\} \tag{73}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{74}$$

where $\vec{A} = 2 \cdot a \cdot r_1 - a$, $\vec{C} = 2 \cdot r_2$, and a is a convergence factor that decreases from 2 to 0 with iteration and $r_1, r_2$ are random numbers within [0,1].

Zhao et al. [84] used Euclidean distance to quantify the leading effect of the current optimal particle on evolution and to update the particle velocity and position. Equations (75) and (76) represent the velocity update process, and Equation (77) represents the particle position update probability:

$$V_{i(p,q)}^{k+1} = W \cdot V_{i(p,q)}^{k} + C_1 \cdot r_1 \cdot dist(Pbest_i^k(p), q) + C_2 \cdot r_2 \cdot dist(Gbest^k(p), q) \tag{75}$$

$$V_{i(p,q)}^{k+1} = \begin{cases} V_{\max} \cdot \frac{V_{i(p,q)}^{k+1}}{\left|V_{i(p,q)}^{k+1}\right|} & , if \left|V_{i(p,q)}^{k+1}\right| \geq V_{\max} \\ V_{i(p,q)}^{k+1} & , otherwise \end{cases} \tag{76}$$

$$P^{k+1}(x_{i,p}^{k+1} = q) = \frac{sigmoid(V_{i(p,q)}^{k+1})}{\sum_{l=1}^{M} sigmoid(V_{i(p,l)}^{k+1})} \tag{77}$$

where dist() denotes the distance between the global optimum and the individual optimum for evaluating the particle's importance.

Xu et al. [90] introduced variation, crossover, and selection operations of differential evolution to enhance the information exchange among swarms: Equation (78) for the variation factor, Equation (79) for the crossover operation, Equation (80) for the variation operation, and Equation (81) for the particle velocity update:

$$x_i^{son} = x_{ra} + 0.5 \cdot 2^{e^{1-\frac{it\max}{it\max+1-n}}} \cdot (x_{rb} - x_{rc})(i = 1, 2, \cdots, N) \tag{78}$$

$$u_{i,d}^{son} = \begin{cases} x_{i,d}^{son} & , if(rand < CR) \; or \; (d = d_{rand}) \\ x_{i,d} & , otherwise \end{cases} \tag{79}$$

$$x_i^{son} = \begin{cases} u_i^{son} & , if(f(u_i^{son}) < f(x_i)) \\ x_i & , otherwise \end{cases} \tag{80}$$

$$v_{id}^{n+1} = w \cdot v_{id}^{n} + c_1 \cdot r_1 \cdot (P_{id}^n - x_{id}^n) + c_2 \cdot r_2 \cdot (P_{gd}^n - x_{id}^n) + c_3 \cdot r_3 \cdot (P_{cd}^n - x_{id}^n) \tag{81}$$

where $ra$, $rb$, $rc$, and $d_{rand}$ denote the random numbers within $[0, N]$, CR denotes the crossover rate and $P_{cd}^n$ denotes the best communication location obtained after the variation, crossover, and selection.

Su et al. [91] introduced a time factor to overcome the problem that causes the traditional FOPSO algorithm to easily fall into local extremes, which reduces the overdependence of the algorithm on the fractional-order $\alpha$. The time factor, $T$, can change adaptively with the swarm parameters, and when $T$ is large, the algorithm has a strong global search capability and helps to obtain the global optimum, and vice versa. Equation (82) represents the time factor change process, and Equation (83) represents the position update process of the improved algorithm:

$$T = \frac{1}{2} \cdot 2^{e^{-\frac{a_t}{1-a_t}}}, \ where\ a_t = \frac{\min(f_{gd}, f_t)}{\max(f_{gd}, f_t)} \tag{82}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) \cdot T \tag{83}$$

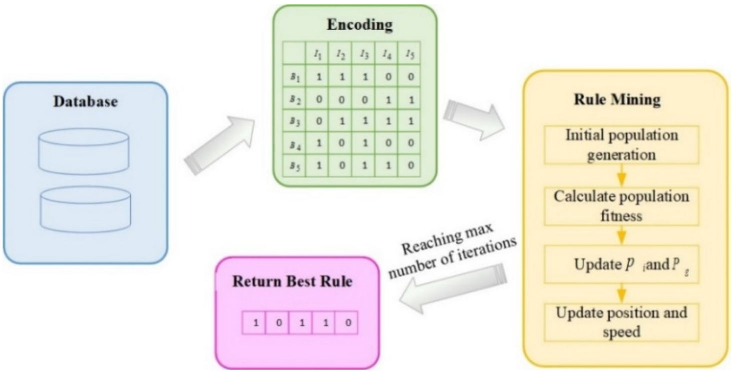
where $a_t$ denotes the swarm diversity, $f_{gd}$ denotes the global optimum, and $f_t$ denotes the average fitness value in iteration $t$.

## 5. Association Rule Mining Based on PSO

### *5.1. Algorithm Principle*

#### 5.1.1. ARM Process

The particle swarm optimization (PSO) algorithm for ARM mainly consists of two stages: preprocessing and rule mining. Its basic flow is shown in Figure 3.


Database
Encoding
Rule Mining
Initial population generation
Calculate population fitness
Update $p_i$ and $P_g$
Update position and speed
Reaching max number of iterations
Return Best Rule
1 0 1 1 0

| | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ |
|---|---|---|---|---|---|
| $B_1$ | 1 | 1 | 1 | 0 | 0 |
| $B_2$ | 0 | 0 | 0 | 1 | 1 |
| $B_3$ | 0 | 1 | 1 | 1 | 1 |
| $B_4$ | 1 | 0 | 1 | 0 | 0 |
| $B_5$ | 1 | 0 | 1 | 1 | 0 |

**Figure 3.** Association rule mining process.

The preprocessing stage mainly completes the binary data conversion. The rule mining stage is the process of an iterative search and the continuous updating of particle positions and velocities in the search space by the PSO algorithm after data encoding and initialization of the swarm to finally obtain the best rules.

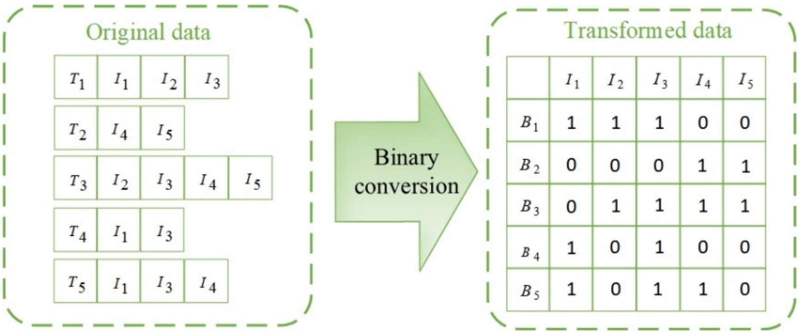
#### 5.1.2. Binary Conversion

The standard PSO algorithm is mainly used for continuous real-valued optimization problems, while in real life, it is mostly used for discrete or combinatorial optimization problems. To address this issue, Kennedy and Eberhart proposed the discrete binary particle swarm optimization algorithms (BPSO) in 1997, which use the sigmoid() function to limit the position and velocity to the {0,1} interval, and where the particle position is calculated iteratively, as shown in Equation (84).

$$x_i^{d+1} = \begin{cases} 1 & ,\ r < sigmoid(v_i^{d+1}); \\ 0 & ,\ r \geq sigmoid(v_i^{d+1}); \end{cases} \tag{84}$$

where $r$ is a random number between (0,1) and $sigmoid\left(v_i^{d+1}\right) = \frac{1}{1+e^{-v_i^{d+1}}}$.

The process of binary conversion of transaction data by the above means is shown in Figure 4.



| | | | | |
|---|---|---|---|---|
| $T_1$ | $I_1$ | $I_2$ | $I_3$ | |
| $T_2$ | $I_4$ | $I_5$ | | |
| $T_3$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ |
| $T_4$ | $I_1$ | $I_3$ | | |
| $T_5$ | $I_1$ | $I_3$ | $I_4$ | |

| | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ |
|---|---|---|---|---|---|
| $B_1$ | 1 | 1 | 1 | 0 | 0 |
| $B_2$ | 0 | 0 | 0 | 1 | 1 |
| $B_3$ | 0 | 1 | 1 | 1 | 1 |
| $B_4$ | 1 | 0 | 1 | 0 | 0 |
| $B_5$ | 1 | 0 | 1 | 1 | 0 |

**Figure 4.** Binary conversion of data.

Among them, the original database has five transaction records, which are $T_1, T_2, \cdots, T_5$, and these five transaction records are converted into binary form and stored. Take $T_5$ as an example, where the transaction record contains $I_1, I_3, I_4$, so its corresponding units are all 1, and the other units are 0. The converted transaction record is 10110.

### 5.1.3. Encoding

There are two main ways to encode data: the Michigan approach and the Pittsburgh approach. Among them, the Pittsburgh approach corresponds to a set of rules per chromosome and is suitable for solving classification problems. The Michigan approach corresponds to a single rule per chromosome and is more straightforward than the former, with a shorter rule syntax. Let us take the Michigan method as an example: Let there be N data items in the dataset, and each data item has two parts, and each part has two possible values that take the value of 1 or 0, as shown in Figure 5.

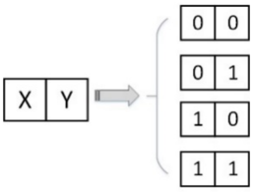


**Figure 5.** Data item representation of the Michigan method.

The dimension of each particle is 2N, and the data item X refers to whether the item is included in the rule. Y refers to whether the item is included in the result of the rule. For example, if the value $I_i$ is 11, then the $i$th item appears in the rule and is included in the result of the rule. The position representation of the rule is shown in Figure 6.

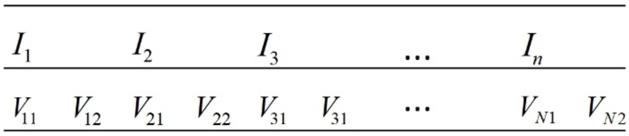


**Figure 6.** Location representation of the Michigan method rule.

### 5.1.4. Search for the Best Particle

First, we need to define a fitness function for evaluating the importance of each particle. By using Equation (84) as the objective function:

$$Fitness(i) = Confidence(i) + Support(i) \tag{85}$$

where $Confidence(i)$ and $Support(i)$ denote the confidence and support of the $i$th particle, respectively. The objective of this function is to maximize the confidence and support of the particle; the stronger this support, the stronger the strength of the association, implying that this is an important association rule.

Second, we also need to generate an initial population, and the particles in the population are called initial particles. Finally, we need to set boundary conditions to constrain the range of motion of the particles and obtain the global best particle when *Gbest* and *Pbest* are iterated until the termination condition is reached. The boundary conditions include calculating the Euclidean distance of the particles, etc. The termination conditions can be set by setting a fixed number of iterations.

### *5.2. Application*

In dealing with complex, realistic optimization problems, heuristic algorithms are inspired by nature and modeled by simulating the behavior of individuals or swarms, which have the advantage of not relying on specific problems, while such algorithms are not constrained by the form of optimization functions in the search process and are highly adaptable, efficient, and easy to implement. Compared with other algorithms, heuristic algorithms show obvious advantages in solving ARM problems. Among the heuristic algorithms, the genetic algorithm (GA) has a long convergence time due to the lack of individual memory and blind search without direction, and it shows inefficiency and difficulty in convergence in high-dimensional data. The ant colony optimization (ACO) algorithm uses pheromone to decide the search direction, which consumes a lot of time for a solution construction and also has the problem of a long convergence time and is prone to stagnation in high-dimensional data. The particle swarm optimization (PSO) algorithm is efficient and simple to implement, converges quickly, and can adjust the search direction by iteration, which is significantly better than other heuristics in ARM problems [92]. PSO algorithms are widely used in ARM, mainly in the fields of privacy protection, recommender systems, risk assessment, and medicine. The correlation is shown in Figure 7.

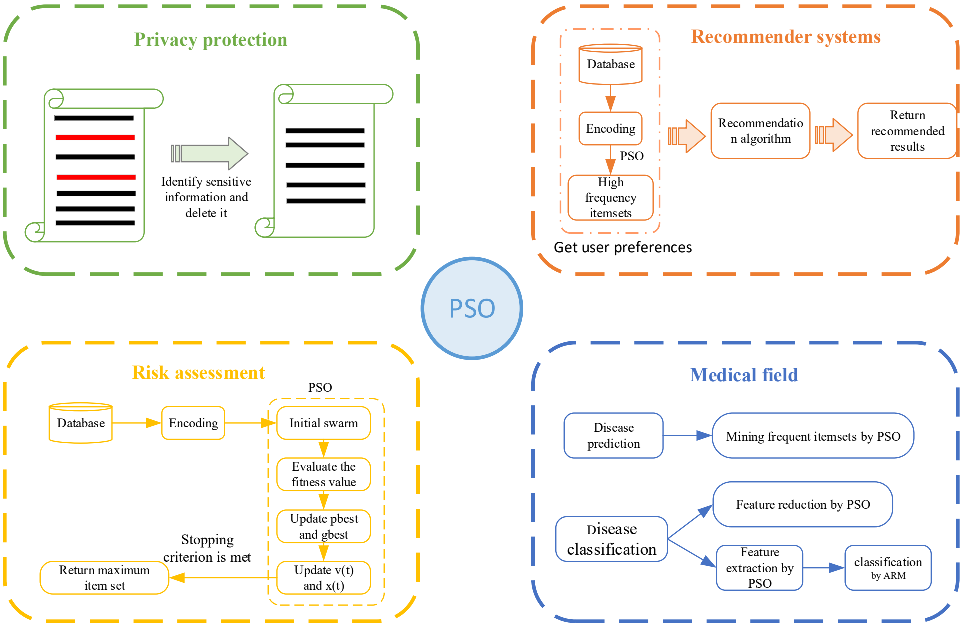


**Figure 7.** Application of particle swarm optimization algorithms.

#### 5.2.1. Privacy Protection

The rapid growth of the Web is accompanied by a dramatic increase in data, and more and more companies or organizations are sharing information by sharing data to gain more profits. To protect the sensitive information implicit in the data from leakage, users need to modify the individual numerical information of the database while sharing the data without compromising the validity of the data. Kalyani et al. [93] proposed an IFHARM algorithm based on PPARM, which includes identifying SAR, clustering

SAR by similarity metric, identifying sensitive transactions, removing and rescanning the converted database, and performing error rule verification. Among them, the authors use the result of each item deletion on non-sensitive data as its impact factor to identify SAR to minimize the loss of non-sensitive association rules and ensure the high utility of the data. Jangra et al. [94] proposed the VIDPSO algorithm to solve the rapid performance degradation of evolutionary algorithms in dense datasets, which prioritizes the presence of more sensitive patterns in data and identifies SAR based on its sensitive patterns to avoid the failure of sensitive information hiding due to the transformation of sensitive frequent itemsets into infrequent itemsets by removing too much data. Suma et al. [95] developed a new method by integrating fractional calculus (FC) and the salp swarm algorithm (SSA) for data sanitization using privacy-preserved data. Krisgnamoorthy et al. [96] used different methods to perturb the dataset to generate a large number of solutions without affecting the validity of the data, which were then sequentially optimized by the PSO algorithm until the support was lower than the minimum support, at which point, the loop was closed, effectively reducing the number of modifications and avoiding the generation of spurious rules.

#### 5.2.2. Recommender Systems

Analyzing consumers' previous behaviors can provide users with personalized products or valuable information, and recommendation systems attempt to recommend goods to potential customers by applying data mining techniques. Among them, the shopping basket problem is a typical example of ARM, and Guo et al. [97] proposed an improved PSO algorithm based on the gray model for dynamic ARM of supermarket shopping data. Furthermore, since the introduction of the background values of the gray model tends to lead to a decrease in prediction accuracy, the authors introduced a secondary search mechanism based on the PSO algorithm to optimally solve the background values of the gray model at different moments, which significantly improves the local search ability of the algorithm and avoids premature particle convergence. To obtain rich product manufacturing knowledge from enterprise historical data to reflect the mapping relationship between a product's design and manufacturing, Kou [98] proposed an ARM method based on a binary PSO algorithm, which encodes association rules and generates a certain number of particle swarms, obtains N rules through multiple iterations, integrates multiple indicators, such as reliability, relevance, and comprehension, and adds a similarity function to compare rules two by two, while eliminating the poor rules to achieve efficient rule mining.

Web data often implies a large amount of web search information. Kaur et al. [99] proposed a hybrid algorithm, HPSO-TS-ARM, to obtain the potential user information among them, which obtains web data through the PSO algorithm, arranges data information by the TAO algorithm, and then performs ARM. Its computation time and efficiency are better than other algorithms, and it has been validated in several datasets. Gangurde et al. [100] used the Weblog features related to Web mining to construct a Web recommendation model, introduced a PSO algorithm, and used a feedforward counter model to obtain association rules using a single data iteration technique, which was used as an adaptation value to execute the recommendation model, and the execution time was significantly reduced. Tyagi et al. [41] solved the low accuracy and data sparsity problems, considering support and confidence as the objectives for proposing the MOPSO algorithm. The algorithm can effectively filter out rules with significant association relationships, and its accuracy and precision are also better than other algorithms when the data is sparse. Jiang et al. [101] proposed an association rule method for generating sentiment designs based on online customer reviews for a large number of online user reviews on major websites by mining the sentiment dimension opinion of user reviews. This method can mine more reliable and interesting rules than the traditional PSO algorithm by performing sentiment analysis on customer reviews to achieve accurate user sentiment recommendations.

### 5.2.3. Risk Assessment

Credit risk management is one of the most important concerns of financial companies, including the collection of relevant information about borrowers, which is used to decide whether to approve their credit applications. Lanzarini et al. [102] proposed the LVQ-PSO algorithm by combining competitive neural networks with the PSO algorithm, which mines association rules by preprocessing the original data with the LVQ method and binary PSO algorithm. The simplified rules can assist the decision-maker in deciding whether to grant a loan or not. Priya et al. [103] proposed a novel PSO-based fuzzy associative classifier (PaSOFuAC) to detect phishing websites and avoid financial losses caused by users not recognizing phishing websites.

As the most convenient means of transportation, the flight process of aircraft is highly susceptible to environmental, equipment, and other complex factors, and the number of hazards increases. Li et al. [104] designed a weighted multiple swarm PSO algorithm for hazard source analyses based on the background of civil aviation safety risk assessments. The preprocessing stage assigns weights to items in the hazard source database and defines the candidate association rules for hazard sources; the analysis stage introduces the swarm interaction mechanism and uses a parallel search mode to generate the association rules for hazard sources and their generation causes, which provides important theoretical support for hazard source identification and prediction. The multiple swarms' parallel search mode provides an interaction mechanism between the swarms to avoid swarms falling into local extremes due to the lack of information exchanged. Inspired by an online sequence learning machine, She et al. [105] used a single hidden layer feedforward neural network to train hazard source data and update the output weights of the learning machine in real-time to realize the online identification of hazard sources. The paper discards the traditional ARM algorithm and uses the ACO-ARM algorithm to mine the most frequent term and the PSO algorithm to initialize the pheromone concentration, which effectively avoids the algorithm's initialization error and improves the system's response speed.

### 5.2.4. Medical Field

In the medical field, a pathological diagnosis is mostly derived from physicians' treatment experience due to the lack of quantitative criteria. ARM can extract features from multiple factors, classify and identify them, and extract the hidden information behind them from medical records to provide physicians with accurate scientific opinions. To improve the applicability and classification accuracy of the algorithm in processing medical data, Ripon et al. [106] combined the PSO algorithm with an artificial neural network to automatically analyze the dataset to achieve an accurate classification of tuberous sclerosis by ARM. To overcome the problem of excessive data dimensionality, Choubey et al. [107] introduced the PCA-PSO algorithm for a feature reduction of raw data to significantly reduce the computation time while ensuring prediction classification accuracy.

In terms of disease prediction, Karsidani et al. [108] proposed ANFIS-PSO to predict whether a patient has coronary artery disease effectively. To deal with the continuous and rapid growth of data volume problems, Mangat et al. [109] proposed a dynamic particle swarm-based rule mining classifier to achieve adaptive prediction of the expected life expectancy of patients with thoracic diseases by dynamically creating swarms and updating the algorithm parameters. Raja et al. [110] proposed a data mining strategy-based prediction model for type II diabetes mellitus by combining the PSO algorithm and fuzzy clustering mean with medical metrics, such as sensitivity, specificity, and accuracy, which are widely used in medicine to assess the performance of the system, and found that the prototype was 8.26% more accurate than traditional methods.

To address the existing problems of inefficient and time-consuming diagnoses of heart diseases, Alkeshuosh et al. [111] used a standard PSO algorithm to obtain heart disease rules from the original dataset and optimized them to generate the best rules. Mao Jie et al. [112] proposed the PSO-SVM algorithm based on support vector machines (SVM) for heart disease predisposing factor detection, which uses ARM to extract the disease features and train the feature dataset. Each PSO-SVM classifier performs the corresponding fitness function, formulated by a classified/total, to iteratively update the velocity and position of each particle until it obtains the global best value. The algorithm is tested on the UCI Cleveland dataset using confidence levels as an indicator and has better classification performance compared to traditional PSO algorithms.

## 6. Conclusions and Challenges

As an emerging field in ARM, PSO algorithms are becoming more mature through continuous exploration and research by many scholars and are widely used in privacy protection, recommender systems, risk assessment, and medicine. This section shows a comparative analysis of the different PSO algorithms, and the objective analysis of existing methods is shown in Table 5.

**Table 5.** An objective analysis of the PSO algorithm.

| Objective | Thesis |
|---|---|
| Convergence | [66,67,73,75,79,80,82,83,86,87,91,113] |
| Versatility | [66,68,73,81,83,84,87,90,113–117] |
| High-dimensional data | [70–72,115–117] |
| Result accuracy | [46,73,81,84,116,117] |
| Balance exploration and utilization | [67,69,74,77,78,80,82,85,86,114] |

The advantages of the particle swarm optimization (PSO) algorithm over other algorithms are: (1) It has uncertainty, which is significantly better than the deterministic algorithms in solving certain aspects of a problem and is more advantageous in finding the global optimal solution; (2) it has better adaptability—the PSO algorithm is a bionic optimization algorithm based on multiple intelligences and is highly robust; (3) it is evolutionary and can preserve the optimal solution information in each iteration round.

The above research shows that the PSO algorithm still has great development space and application value in ARM, but with the expansion of data scale, the difficulty of mining effective association rules from large-scale data using the PSO algorithm gradually increases, and this field will still face great challenges in future research.

## 7. PSO Future Prospects

The future research directions of particle swarm optimization algorithms may be as follows:

(1) PSO algorithms converge prematurely and are prone to falling into local extremes, which mainly lie in the imbalance between global search and local search capabilities. Nevertheless, many improvement algorithms have been proposed, such as the QPSO algorithm based on the cloud model, the adaptive multi-objective PSO algorithm based on Gaussian mixed variance and elite learning, etc. However, the two are relatively contradictory, and how to dynamically maintain the balance between them in the search process to obtain the optimal solution based on the actual search results or how to measure that the two have reached the best balance needs further study.

(2) Due to the exponential growth of high-dimensional data search space and low data relevance, search efficiency decreases rapidly. In addition to the existing "divide and conquer" method and the introduction of other algorithms, the use of feature extraction methods to remove redundant data and natural computational methods, such as nonlinear dimensionality reduction, can also be introduced to reduce redundancy using the maximum linear irrelevance group. Although significant progress has been made in handling large amounts of data, there is still much room for exploration in the ARM models for handling high-speed data.

(3) Inertia weights and learning factors are important parameters in PSO algorithms. Improvements to inertia weights include linearly decreasing inertia weights, fuzzy inertia weights, random inertia weights, etc. Improvements in learning factors include

shrinkage factors, synchronous learning factors, asynchronous learning factors, etc. In the future, we can continue to optimize these two parameters and consider whether the two affect each other and their mutual influence weights.

(4) Many scholars focus more on the swarm initialization problem, including the M-class random method, fixed-value setting method, two-step method, hybrid method, etc. However, the method is too simple in how to divide the swarm size and lacks a scientific and reasonable effective division, which will limit the speed of particle movement when the swarm is too small and, thus, lead to early convergence. When the swarm is too large, it leads to a too large search space and reduces the performance and efficiency of the algorithm.

(5) When there are missing or wrong datasets, there is difficulty in guaranteeing the PSO algorithm's accuracy and performance without artificial preprocessing, i.e., having certain fault tolerance.

**Author Contributions:** Conceptualization, G.L. and T.W.; methodology, G.L., T.W. and P.S.; validation, G.L., Q.C. and N.X.; writing—original draft preparation, T.W. and G.L.; writing—review and editing, G.L., T.W., P.S., Q.C. and A.V.; supervision, N.X. and A.V.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (#62041702); the Ministry of Education, Humanities and Social Sciences Project (#20YJA870010); the Jiangxi Provincial Social Science Planning Project (#19TQ05); the Key project of Education Science planning in Jiangxi Province (#19ZD024); the Jiangxi University Humanities and Social Science Planning Project (#TQ20105); the Basic Education Research Project of Jiangxi Province (#SZUNDZH2021-1143); and the Jiangxi Province Degree and Postgraduate Education and Teaching Reform Research Project (#JXYJG-2020-075).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Agrawal, R.; Tomasz, I.; Arun, S. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, 25–28 May 1993; pp. 207–216.
2. Telikani, A.; Amir, H.G.; Asadollah, S. A survey of evolutionary computation for association rule mining. *Inf. Sci.* **2020**, *524*, 318–352. [CrossRef]
3. Kaushik, M.; Sharma, R.; Peious, S.A.; Shahin, M.; Yahia, S.B.; Draheim, D. A systematic assessment of numerical association rule mining methods. *SN Comput. Sci.* **2021**, *2*, 1–13. [CrossRef]
4. Djenouri, Y.; Fournier, V.P.; Belhadi, A.; Chun, W.L. Metaheuristics for frequent and high-utility itemset mining. In *High-Utility Pattern Mining*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 261–278.
5. Ghobaei, A.M.; Shahidinejad, A. An efficient resource provisioning approach for analyzing cloud workloads: A metaheuristic-based clustering approach. *J. Supercomput.* **2021**, *77*, 711–750. [CrossRef]
6. Gan, X.; Gong, D.; Gao, X.; Zhang, Y.; Lin, L.; Lan, T. Research on Construction of Mining Model of Association Rules for Temporal Data based on CNN. In Proceedings of the 2021 IEEE International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB), Yilan County, Taiwan, 10–12 December 2021; pp. 233–235.
7. Chen, S.; Xi, J.; Chen, Y.; Zhao, J. Association mining of near misses in hydropower engineering construction based on convolutional neural network text classification. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–6. [CrossRef]
8. He, B.; Zhang, J. An Association Rule Mining Method Based on Named Entity Recognition and Text Classification. *Arab. J. Sci. Eng.* **2022**, *6*, 1–9. [CrossRef]
9. Badhon, B.; Kabir, M.M.; Xu, S.; Kabir, M. A survey on association rule mining based on evolutionary algorithms. *Int. J. Comput. Appl.* **2021**, *43*, 775–785. [CrossRef]
10. Kheirollahi, H.; Chahardowli, M.; Simjoo, M. A new method of well clustering and association rule mining. *J. Pet. Sci. Eng.* **2022**, *214*, 110479. [CrossRef]
11. Datta, S.; Mali, K. Significant association rule mining with high associability. In Proceedings of the IEEE 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 6–8 May 2021; pp. 1159–1164.
12. Rathee, S.; Kashyap, A. Adaptive-Miner: An efficient distributed association rule mining algorithm on Spark. *J. Big Data* **2018**, *5*, 1–17. [CrossRef]
13. Zhong, Q.Y.; Qian, Q.; Feng, Y.; Fu, Y. BPSO Algorithm with Opposition-Based Learning Method for Association Rule Mining. In *Advancements in Mechatronics and Intelligent Robotics*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 1220, pp. 351–358.

14. Baró, G.B.; Martínez, T.J.F.; Rosas, R.M.V.; Ochoa, J.A.C.; González, A.Y.R.; Cortés, M.S.L. A PSO-based algorithm for mining association rules using a guided exploration strategy. *Pattern Recognit. Lett.* **2020**, *138*, 8–15. [CrossRef]
15. Fister, J.I.; Fister, I. A Brief Overview of Swarm Intelligence-Based Algorithms for Numerical Association Rule Mining. *Appl. Optim. Swarm Intell.* **2021**, *8*, 47–59.
16. Wang, C.; Bian, W.; Wang, R.; Chen, H.; Ye, Z.; Yan, L. Association rules mining in parallel conditional tree based on grid computing inspired partition algorithm. *Int. J. Web Grid Serv.* **2020**, *16*, 321–339. [CrossRef]
17. Yuan, J. An anomaly data mining method for mass sensor networks using improved PSO algorithm based on spark parallel framework. *J. Grid Comput.* **2020**, *18*, 251–261. [CrossRef]
18. Sukanya, N.S.; Thangaiah, P.R.J. Customized Particle Swarm Optimization Algorithm for Frequent Itemset Mining. In Proceedings of the IEEE 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 22–24 January 2020; pp. 1–4.
19. Dubey, A.K.; Kumar, A.; Agrawal, R. An efficient ACO-PSO-based framework for data classification and preprocessing in big data. *Evol. Intell.* **2021**, *14*, 909–922. [CrossRef]
20. Tofighy, S.; Rahmanian, A.A.; Ghobaei, A.M. An ensemble CPU load prediction algorithm using a Bayesian information criterion and smooth filters in a cloud computing environment. *Softw. Pract. Exp.* **2018**, *48*, 2257–2277. [CrossRef]
21. Gad, A.G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Arch. Comput. Methods Eng.* **2022**, 1–31.
22. Nasr, M.; Hamdy, M.; Hegazy, D.; Bahnasy, K. An efficient algorithm for unique class association rule mining. *Expert Syst. Appl.* **2021**, *164*, 113978. [CrossRef]
23. Kuok, C.M.; Fu, A.; Wong, M.H. Mining fuzzy association rules in databases. *ACM SIGMOD Rec.* **1998**, *27*, 41–46. [CrossRef]
24. Ting, C.K.; Liaw, R.T.; Wang, T.C.; Hong, T.P. Mining fuzzy association rules using a memetic algorithm based on structure representation. *Memetic Comput.* **2018**, *10*, 15–28. [CrossRef]
25. Varol, A.E.; Alatas, B. Computing, Performance analysis of multi-objective artificial intelligence optimization algorithms in numerical association rule mining. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 3449–3469. [CrossRef]
26. Altay, E.V.; Alatas, B. Intelligent optimization algorithms for the problem of mining numerical association rules. *Phys. A Stat. Mech. Appl.* **2020**, *540*, 123142. [CrossRef]
27. Heraguemi, K.E.; Kamel, N.; Drias, H. Multi-objective bat algorithm for mining numerical association rules. *Int. J. Bio-Inspired Comput.* **2018**, *11*, 239–248. [CrossRef]
28. Srikant, R.; Agrawal, R. Mining quantitative association rules in large relational tables. In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, QC, Canada, 4–6 June 1996; pp. 1–12.
29. Song, X.D.; Zhai, K.; Gao, W. The Research of Evaulation index in Association Rules. *Microcomput. Inf.* **2007**, *23*, 174–176.
30. Wang, C.; Liu, Y.; Zhang, Q.; Guo, H.; Liang, X.; Chen, Y.; Wei, Y. Association rule mining based parameter adaptive strategy for differential evolution algorithms. *Expert Syst. Appl.* **2019**, *123*, 54–69. [CrossRef]
31. Altay, E.V.; Alatas, B. Differential evolution and sine cosine algorithm based novel hybrid multi-objective approaches for numerical association rule mining. *Inf. Sci.* **2021**, *554*, 198–221. [CrossRef]
32. Guan, B.; Zhao, Y.; Yin, Y.; Li, Y. A differential evolution based feature combination selection algorithm for high-dimensional data. *Inf. Sci.* **2021**, *547*, 870–886. [CrossRef]
33. Menaga, D.; Saravanan, S. GA-PPARM: Constraint-based objective function and genetic algorithm for privacy preserved association rule mining. *Evol. Intell.* **2021**, *15*, 1487–1498. [CrossRef]
34. Lin, J.C.W.; Djenouri, Y.; Srivastava, G.; Yun, U.; Fournier-Viger, P. A predictive GA-based model for closed high-utility itemset mining. *Appl. Soft Comput.* **2021**, *108*, 107422. [CrossRef]
35. Neysiani, B.S.; Soltani, N.; Mofidi, R.; NadimiShahraki, M.H. Improve performance of association rule-based collaborative filtering recommendation systems using genetic algorithm. *Int. J. Inf. Technol. Comput. Sci.* **2019**, *11*, 48–55. [CrossRef]
36. Giri, P.K.; De, S.S.; Dehuri, S.; Cho, S.B. Biogeography based optimization for mining rules to assess credit risk. *Intell. Syst. Account. Financ. Manag.* **2021**, *28*, 35–51. [CrossRef]
37. Ghobaei, A.M. A workload clustering based resource provisioning mechanism using Biogeography based optimization technique in the cloud based systems. *Soft Comput.* **2021**, *25*, 3813–3830. [CrossRef]
38. Mo, H.; Xu, L. Immune clone algorithm for mining association rules on dynamic databases. In Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), Hong Kong, China, 14–16 November 2005; Volume 5, pp. 202–206.
39. Husain, M.S. Exploiting Artificial Immune System to Optimize Association Rules for Word Sense Disambiguation. *Int. J. Intell. Syst. Appl. Eng.* **2021**, *9*, 184–190. [CrossRef]
40. Cunha, D.S.; Castro, L.N. Evolutionary and immune algorithms applied to association rule mining in static and stream data. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
41. Tyagi, S.; Bharadwaj, K.K. Enhancing collaborative filtering recommendations by utilizing multi-objective particle swarm optimization embedded association rule mining. *Swarm Evol. Comput.* **2013**, *13*, 1–12. [CrossRef]
42. Kuo, R.J.; Gosumolo, M.; Zulvia, F.E. Multi-objective particle swarm optimization algorithm using adaptive archive grid for numerical association rule mining. *Neural Comput. Appl.* **2019**, *31*, 3559–3572. [CrossRef]

43. Agarwal, A.; Nanavati, N. Association rule mining using hybrid GA-PSO for multi-objective optimisation. In Proceedings of the 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, India, 15–17 December 2016; pp. 1–7.
44. Moslehi, F.; Haeri, A.; Martínez-Álvarez, F. A novel hybrid GA–PSO framework for mining quantitative association rules. *Soft Comput.* **2020**, *24*, 4645–4666. [CrossRef]
45. Roopa Devi, E.M.; Suganthe, R.C. Enhanced transductive support vector machine classification with grey wolf optimizer cuckoo search optimization for intrusion detection system. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e4999. [CrossRef]
46. Chiclana, F.; Kumar, R.; Mittal, M.; Khari, M.; Chatterjee, J.M.; Baik, S.W. ARM–AMO: An efficient association rule mining algorithm based on animal migration optimization. *Knowl. Based Syst.* **2018**, *154*, 68–80.
47. Pradeep, G.; Ravi, V.; Krishna, G.J. Hybrid Evolutionary Computing-based Association Rule Mining. In *Soft Computing in Interdisciplinary Sciences*; Springer: Singapore, 2022; pp. 223–243.
48. Pazhaniraja, N.; Sountharrajan, S.; Sathis, K.B. High utility itemset mining: A Boolean operators-based modified grey wolf optimization algorithm. *Soft Comput.* **2020**, *24*, 16691–16704. [CrossRef]
49. Yildirim, G.; Alatas, B. New adaptive intelligent grey wolf optimizer based multi-objective quantitative classification rules mining approaches. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 9611–9635. [CrossRef]
50. Chantar, H.; Mafarja, M.; Alsawalqah, H.; Heidari, A.A.; Aljarah, I.; Faris, H. Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Comput. Appl.* **2020**, *32*, 12201–12220. [CrossRef]
51. Telikani, A.; Gandomi, A.H.; Shahbahrami, A.; Dehkordi, M.N. Privacy-preserving in association rule mining using an improved discrete binary artificial bee colony. *Expert Syst. Appl.* **2020**, *144*, 113097. [CrossRef]
52. Turabieh, H.; Azwari, S.A.; Rokaya, M.; Alosaimi, W.; Alharbi, A.; Alhakami, W.; Alnfiai, M. Enhanced Harris Hawks optimization as a feature selection for the prediction of student performance. *Computing* **2021**, *103*, 1417–1438. [CrossRef]
53. Dong, D.; Ye, Z.; Cao, Y.; Xie, S.; Wang, F.; Ming, W. An improved association rule mining algorithm based on ant lion optimizer algorithm and FP-growth. In Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, 18–21 September 2019; Volume 1, pp. 458–463.
54. Nawaz, M.S.; Fournier, V.P.; Yun, U.; Wu, Y.; Song, W. Mining high utility itemsets with hill climbing and simulated annealing. *ACM Trans. Manag. Inf. Syst. (TMIS)* **2021**, *13*, 1–22. [CrossRef]
55. Ospina, M.H.; Quintana, J.L.A.; Lopez, V.F.J.; Berrio, G.S.; Barrero, L.H.; Sana, S.S. Extraction of decision rules using genetic algorithms and simulated annealing for prediction of severity of traffic accidents by motorcyclists. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 10051–10072. [CrossRef]
56. Chou, Y.H.; Kuo, S.Y.; Chen, C.Y.; Chao, H.C. A rule-based dynamic decision-making stock trading system based on quantum-inspired tabu search algorithm. *IEEE Access* **2014**, *2*, 883–896. [CrossRef]
57. Derouiche, A.; Layeb, A.; Habbas, Z. Frequent Itemsets Mining with Chemical Reaction Optimization Metaheuristic. In Proceedings of the IEEE 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), Tebessa, Algeria, 24–25 October 2018; pp. 1–6.
58. Taradeh, M.; Mafarja, M.; Heidari, A.A.; Faris, H.; Aljarah, I.; Mirjalili, S.; Fujita, H. An evolutionary gravitational search-based feature selection. *Inf. Sci.* **2019**, *497*, 219–239. [CrossRef]
59. Tuba, E.; Jovanovic, R.; Hrosik, R.C.; Alihodzic, A.; Tuba, M. Web intelligence data clustering by bare bone fireworks algorithm combined with k-means. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, Novi Sad, Serbia, 25–27 June 2018; pp. 1–8.
60. Abualigah, L.; Dulaimi, A.J. A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm. *Clust. Comput.* **2021**, *24*, 2161–2176. [CrossRef]
61. Djenouri, Y.; Drias, H.; Chemchem, A. A hybrid bees swarm optimization and tabu search algorithm for association rule mining. In Proceedings of the IEEE 2013 World Congress on Nature and Biologically Inspired Computing, Fargo, ND, USA, 12–14 August 2013; pp. 120–125.
62. Ma, L.; Zhang, T.; Wang, R.; Yang, G.; Zhang, Y. Pbar: Parallelized brain storm optimization for association rule mining. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 1148–1156.
63. Djenouri, Y.; Djenouri, D.; Belhadi, A.; Fournier, V.P.; Lin, J.C.W.; Bendjoudi, A. Exploiting GPU parallelism in improving bees swarm optimization for mining big transactional databases. *Inf. Sci.* **2019**, *496*, 326–342. [CrossRef]
64. Malik, M.M.; Haouassi, H. Efficient sequential covering strategy for classification rules mining using a discrete equilibrium optimization algorithm. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *in press*. [CrossRef]
65. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the IEEE Sixth International Symposium on Micro Machine and Human Science, MHS'95, Nagoya, Japan, 4–6 October 1995; Volume 2, pp. 39–43.
66. Wang, C.W.; Yin, S.L.; Liu, W.Y.; Wei, X.M.; Zheng, H.J.; Yang, J.P. High utility itemset mining algorithm based on improved particle swarm optimization. *J. Chin. Comput. Syst.* **2020**, *41*, 1084–1090.
67. Dubey, K.; Sharma, S.C. A novel multi-objective CR-PSO task scheduling algorithm with deadline constraint in cloud computing. *Sustain. Comput. Inform. Syst.* **2021**, *32*, 100605. [CrossRef]

68. Hematpour, N.; Ahadpour, S. Execution examination of chaotic S-box dependent on improved PSO algorithm. *Neural Comput. Appl.* **2021**, *33*, 5111–5133. [CrossRef]
69. Li, A.D.; Xue, B.; Zhang, M. Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. *Appl. Soft Comput.* **2021**, *106*, 107302. [CrossRef]
70. Wang, S.; Yu, X.; Jia, W. A New Population Initialization of Particle Swarm Optimization Method Based on PCA for Feature Selection. *J. Big Data* **2021**, *3*, 1–9. [CrossRef]
71. Bangyal, W.H.; Hameed, A.; Alosaimi, W.; Alyami, H. A new initialization approach in particle swarm optimization for global optimization problems. *Comput. Intell. Neurosci.* **2021**, *2021*, 1–17. [CrossRef]
72. Pervaiz, S.; Haider, B.W.; Ashraf, A.; Nisar, K.; Haque, M.R.; Ibrahim, A.; Rawat, D.B. Comparative Research Directions of Population Initialization Techniques using PSO Algorithm. *Intell. Autom. Soft Comput.* **2022**, *32*, 1427–1444. [CrossRef]
73. Tian, S.; Li, Y.; Kang, Y.; Xia, J. Multi-robot path planning in wireless sensor networks based on jump mechanism PSO and safety gap obstacle avoidance. *Future Gener. Comput. Syst.* **2021**, *118*, 37–47. [CrossRef]
74. Xiang, L.I.; Chen, J. A modified PSO Algorithm based on Cloud Theory for optimizing the Fuzzy PID controller. *J. Phys. Conf. Ser.* **2022**, *2183*, 012014.
75. Zhang, J.; Sheng, J.; Lu, J.; Shen, L. UCPSO: A uniform initialized particle swarm optimization algorithm with cosine inertia weight. *Comput. Intell. Neurosci.* **2021**, *2021*, 41–59. [CrossRef]
76. Agrawal, A.; Tripathi, S. Particle swarm optimization with adaptive inertia weight based on cumulative binomial probability. *Evol. Intell.* **2021**, *14*, 305–313. [CrossRef]
77. Komarudin, A.; Setyawan, N.; Kamajaya, L.; Achmadiah, M.N.; Zulfatman, Z. Signature PSO: A novel inertia weight adjustment using fuzzy signature for LQR tuning. *Bull. Electr. Eng. Inform.* **2021**, *10*, 308–318. [CrossRef]
78. Fan, M.; Akhter, Y. A time-varying adaptive inertia weight based modified PSO algorithm for UAV path planning. In Proceedings of the IEEE 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 5–7 January 2021; pp. 573–576.
79. Chrouta, J.; Farhani, F.; Zaafouri, A. A modified multi swarm particle swarm optimization algorithm using an adaptive factor selection strategy. *Trans. Inst. Meas. Control.* **2021**, *3*, 01423312211029509. [CrossRef]
80. Osei, K.J.; Han, F.; Amponsah, A.A.; Ling, Q.; Abeo, T.A. A hybrid optimization method by incorporating adaptive response strategy for Feedforward neural network. *Connect. Sci.* **2022**, *34*, 578–607. [CrossRef]
81. Li, Z.; Wang, F.; Wang, R.J. An Improved Particle Swarm Optimization Algorithm. Available online: http://kns.cnki.net/kcms/detail/11.4762.TP.20210831.0841.008.html (accessed on 31 August 2021).
82. Keshavamurthy, B.N. Improved PSO for task scheduling in cloud computing. In *Evolution in Computational Intelligence*; Springer: Singapore, 2021; pp. 467–474.
83. Amponsah, A.A.; Han, F.; Osei, K.J.; Bonah, E.; Ling, Q.H. An improved multi-leader comprehensive learning particle swarm optimisation based on gravitational search algorithm. *Connect. Sci.* **2021**, *33*, 803–834. [CrossRef]
84. Miao, Z.; Yong, P.; Mei, Y.; Quanjun, Y.; Xu, X. A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment. *Future Gener. Comput. Syst.* **2021**, *115*, 497–516. [CrossRef]
85. Zhu, D.; Huang, Z.; Xie, L.; Zhou, C. Improved Particle Swarm Based on Elastic Collision for DNA Coding Optimization Design. *IEEE Access* **2022**, *10*, 63592–63605. [CrossRef]
86. Fu, K.; Cai, X.; Yuan, B.; Yang, Y.; Yao, X. An Efficient Surrogate Assisted Particle Swarm Optimization for Antenna Synthesis. *IEEE Trans. Antennas Propag.* **2022**, *70*, 4977–4984. [CrossRef]
87. Song, B.; Wang, Z.; Zou, L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. *Appl. Soft Comput.* **2021**, *100*, 106960. [CrossRef]
88. Shaheen, M.A.M.; Hasanien, H.M.; Alkuhayli, A. A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution. *Ain Shams Eng. J.* **2021**, *12*, 621–630. [CrossRef]
89. Suman, G.K.; Guerrero, J.M.; Roy, O.P. Optimisation of solar/wind/bio-generator/diesel/battery based microgrids for rural areas: A PSO-GWO approach. *Sustain. Cities Soc.* **2021**, *67*, 102723. [CrossRef]
90. Xu, L.; Huang, C.; Li, C.; Wang, J.; Liu, H.; Wang, X. Estimation of tool wear and optimization of cutting parameters based on novel ANFIS-PSO method toward intelligent machining. *J. Intell. Manuf.* **2021**, *32*, 77–90. [CrossRef]
91. Su, S.; Zhai, Z.; Wang, C.; Ding, K. Improved fractional-order PSO for PID tuning. *Int. J. Pattern Recognit. Artif. Intell.* **2021**, *35*, 2159016. [CrossRef]
92. Zhong, Q.Y.; Qian, Q.; Fu, Y.F.; Yong, F.E.N.G. Survey of particle swarm optimization algorithm for association rule mining. *J. Front. Comput. Sci. Technol.* **2021**, *15*, 777–793.
93. Kalyani, G.; Chandra, S.R.M.V.P.; Janakiramaiah, B. Particle swarm intelligence and impact factor-based privacy preserving association rule mining for balancing data utility and knowledge privacy. *Arab. J. Sci. Eng.* **2018**, *43*, 4161–4178. [CrossRef]
94. Jangra, S.; Toshniwal, D. VIDPSO: Victim item deletion based PSO inspired sensitive pattern hiding algorithm for dense datasets. *Inf. Process. Manag.* **2020**, *57*, 102255. [CrossRef]
95. Suma, B.; Shobha, G. Fractional salp swarm algorithm: An association rule based privacy-preserving strategy for data sanitization. *J. Inf. Secur. Appl.* **2022**, *68*, 103224.
96. Krishnamoorthy, S.; Sadasivam, G.S.; Rajalakshmi, M.; Kowsalyaa, K.; Dhivya, M. Privacy preserving fuzzy association rule mining in data clusters using particle swarm optimization. *Int. J. Intell. Inf. Technol. (IJIIT)* **2017**, *13*, 1–20. [CrossRef]

97. Guo, S.W.; Meng, Y.Y.; Chen, S.L. Application of improved PSOGM algorithm in dynamic association rule mining. *Comput. Eng. Appl.* **2018**, *54*, 160–165.
98. Kou, Z.C. Binary particle swarm optimization-based association rule mining for machine capabilities and part features. *J. Univ. Jinan Sci. Technol.* **2019**, *33*, 381–388.
99. Kaur, S.; Goyal, M. Fast and robust hybrid particle swarm optimization tabu search association rule mining (HPSO-ARM) algorithm for web data association rule mining (WDARM). *Int. J. Adv. Res. Comput. Sci. Manag. Stud.* **2014**, *2*, 448–451.
100. Gangurde, R.A.; Kumar, B. Next Web Page Prediction using Genetic Algorithm and Feed Forward Association Rule based on Web-Log Features. *Int. J. Perform. Eng.* **2020**, *16*, 10–18. [CrossRef]
101. Jiang, H.; Kwong, C.K.; Park, W.Y.; Yu, K.M. A multi-objective PSO approach of mining association rules for affective design based on online customer reviews. *J. Eng. Des.* **2018**, *29*, 381–403. [CrossRef]
102. Lanzarini, L.; Villa, M.A.; Aquino, G.; De, G.A. Obtaining classification rules using lvqPSO. In Proceedings of the ICSI 2015: Advances in Swarm and Computational Intelligence, Beijing, China, 25–28 June 2015; Springer: Cham, Switzerland, 2015; pp. 183–193.
103. Priya, S.; Selvakumar, S. PaSOFuAC: Particle Swarm Optimization Based Fuzzy Associative Classifier for Detecting Phishing Websites. *Wirel. Pers. Commun.* **2022**, *125*, 755–784. [CrossRef]
104. Li, S.Y.; Zhou, L.; Liu, H. Hazard ldentification Algorithm Based on Deep Extreme Learning Machine. *Comput. Sci.* **2017**, *44*, 89–94.
105. She, Y.L.; Zhou, L. Hazard ldentification Algorithm Based on Improved Online Sequential Extreme Learning Machine. *Comput. Technol. Dev.* **2018**, *28*, 72–77.
106. Ripon, S.; Sarowar, G.; Qasim, F.; Cynthia, S.T. An Efficient Classification of Tuberous Sclerosis Disease Using Nature Inspired PSO and ACO Based Optimized Neural Network. In *Nature Inspired Computing for Data Science*; Springer: Cham, Switzerland, 2020; pp. 1–28.
107. Choubey, D.K.; Kumar, P.; Tripathi, S.; Kumar, S. Performance evaluation of classification methods with PCA and PSO for diabetes. *Netw. Model. Anal. Health Inform. Bioinform.* **2020**, *9*, 1–30. [CrossRef]
108. Karsidani, S.D.; Farhadian, M.; Mahjub, H.; Mozayanimonfared, A. Prediction of Major Adverse Cardiovascular Events (MACCE) Following Percutaneous Coronary Intervention Using ANFIS-PSO Model. *BMC Cardiovasc. Disord.* **2022**, *22*, 389.
109. Mangat, V.; Vig, R. Novel associative classifier based on dynamic adaptive PSO: Application to determining candidates for thoracic surgery. *Expert Syst. Appl.* **2014**, *41*, 8234–8244. [CrossRef]
110. Raja, J.B.; Pandian, S.C. PSO-FCM based data mining model to predict diabetic disease. *Comput. Methods Programs Biomed.* **2020**, *196*, 105659. [CrossRef]
111. Alkeshuosh, A.H.; Moghadam, M.Z.; Al, M.I.; Abdar, M. Using PSO algorithm for producing best rules in diagnosis of heart disease. In Proceedings of the IEEE 2017 International Conference on Computer and Applications (ICCA), Doha, United Arab Emirates, 6–7 September 2017; pp. 306–311.
112. Mao, J.; Wang, H.Y. Association rule optimization for detection algorithm of factors inducing heart disease. *Control. Eng. China* **2017**, *24*, 1286–1290.
113. Shao, P.; Wu, Z.; Peng, H.; Wang, Y.; Li, G. An Adaptive Particle Swarm Optimization Using Hybrid Strategy. In Proceedings of the International Symposium on Intelligence Computation and Applications, Guangzhou, China, 20–21 November 2021; Springer: Singapore, 2017; pp. 26–39.
114. Liang, B.; Zhao, Y.; Li, Y. A hybrid particle swarm optimization with crisscross learning strategy. *Eng. Appl. Artif. Intell.* **2021**, *105*, 104418. [CrossRef]
115. Rostami, M.; Forouzandeh, S.; Berahmand, K.; Soltani, M. Integration of multi-objective PSO based feature selection and node centrality for medical datasets. *Genomics* **2020**, *112*, 4370–4384. [CrossRef] [PubMed]
116. Zhong, Q.Y.; Qian, Q.; Fu, Y.F. Association rule mining based on multi-strategy BPSO algorithm. *Bull. Sci. Technol.* **2021**, *37*, 40–46.
117. Dai, Y.B. Multi-Objective Particle Swarm Algorithm Based on Quasi-Circular Mapping. *Appl. Res. Comput.* **2021**, *38*, 3673–3677.