



Research article

Research on fruit shape database mining to support fruit class classification using the shuffled frog leaping optimization (SFLO) technique

Ha Huy Cuong Nguyen¹, Ho Phan Hieu^{2,*}, Chiranjibe Jana^{3,*}, Tran Anh Kiet² and Thanh Thuy Nguyen⁴

¹ Software Development Centre-The University of Danang, Danang, 50000 Da Nang, Vietnam

² The University of Danang, Danang, 50000 Da Nang, Vietnam

³ Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Chennai 602105, Tamil Nadu, India

⁴ Faculty of Computer Science, VNU University of Engineering and Technology, Hanoi, Vietnam

* **Correspondence:** Email: hophanhieu@ac.udn.vn, chiranjibeju@yahoo.com;
Tel: +840905169900.

Abstract: Association rule mining (ARM) is a technique for discovering meaningful associations within databases, typically handling discrete and categorical data. Recent advancements in ARM have concentrated on refining calculations to reveal connections among various databases. The integration of shuffled frog leaping optimization (SFLO) processes has played a crucial role in this pursuit. This paper introduces an innovative SFLO-based method for performance analysis. To generate association rules, we utilize the apriori algorithm and incorporate frog encoding within the SFLO method. A key advantage of this approach is its one-time database filtering, significantly boosting efficiency in terms of CPU time and memory usage. Furthermore, we enhance the optimization process's efficacy and precision by employing multiple measures with the modified SFLO techniques for mining such information. The proposed approach, implemented using MongoDB, underscores that our performance analysis yields notably superior outcomes compared to alternative methods. This research holds implications for fruit shape database mining, providing robust support for fruit class classification.

Keywords: apriori algorithm; association rule mining; shuffled frog leaping optimization; support; confidence

Mathematics Subject Classification: 46N10, 47N10, 52B55

1. Introduction

Association rule mining (ARM) seeks to identify sets of interesting and significant associations among diverse variables, attributes, or features within extensive databases (Yadav and Chowdary, 2018) [1]. Employing specific metrics of interest, this approach has been utilized to categorize robust rules extracted from databases (Zhang, Pedrycz, and Huang, 2018) [2]. The concept of 'interestingness' assumes a crucial role in data mining, regardless of the pattern types being extracted (Nguyen et al., 2017a, 2017b) [3,4]. These metrics serve to rank and select patterns, based on their potential relevance to users (Wang, Meng, Xu, and Peng, 2018) [5].

For instance, consider the rule $\text{potatoes} \Rightarrow \text{burger}$, derived from trade-related market data. This rule indicates that if customers purchase both potatoes and onions, there is a likelihood that they will buy meat. Such statistics prove valuable for decision-making in recommendation systems, influencing choices like product placements or promotional pricing [6, 7].

Alongside the aforementioned research, association rules find application in various domains, including continuous intrusion detection and bioinformatics (Altaf, Shahbaz, and Guergachi, 2017) [8].

Other significant works include those by Agarwal (2018), Samuel and Rajan (2014), and Xue et al. (2015), Heraguemi, Kamel, and Drias (2016), Tyagi and Thakur (2018), Mane and Ghorpade (2018), Rekik, Kallel, Casillas, and Alimi (2018), and Subbulakshmi and Deisy (2018) [9–16].

However, the current ARM methods encounter certain challenges, as stated by He, Zhu, He, Gu, and Cui (2017) [17–19]:

- In ARM, the most important problem is data sources with haphazard data formats. Converting the transaction table is indeed difficult because existing processes of rule mining follow multilevel hierarchical concepts.
- Another problem for multiple-level association rules is to decrease the number of iterations and rule-duplication to attain time efficiency. The reduction of the number of scans in the database at each level is the only solution to time utilization. In association rule detection, redundant association rules are the focal issue.
- If the main parameters (support and confidence) are small, then the number of frequent itemsets rises. As a result, the number of rules presented also rises. Thus, the choice of appropriate values of interestingness constraints (support and confidence) is a vital issue in association rule mining.
- The algorithms should be optimized for less CPU overhead. I/O overhead should be reduced.
- An additional concern correlated to ARM is determining the effectiveness of associating patterns. The job of making choices is to be combined flawlessly within the ARM process.

To address the aforementioned challenges, employing a meta-heuristic algorithm proves to be a viable solution. The shuffled frog leaping optimization algorithm (SFLO) represents a population-based search mechanism (Kaur and Mehta, 2017) [20]. At its core, SFLO revolves around the exchange of local and global information within the updated databases. This optimization approach involves an interconnected simulated frog population, organized into distinct meme lexes (Kaur and Mehta, 2017) [20]. In this context, the virtual frogs function as carriers or transporters of memes, where a meme symbolizes an evolutionary unit.

The core concept behind the modified SFLO (MSFLO) lies in adjusting the technique's scope and each frog's leap distance through alignment with the frog's perception and performance uncertainties.

This modification focuses on expanding the local search space, effectively preventing premature convergence, and enhancing the SFLO's performance. The proposed approach addresses challenges in ARM related to size, coverage, I/O overhead, the number of association rules, and time efficiency [21].

The remainder of the paper is structured as follows: Section 2 provides background information. Sections 3 and 4 detail the proposed method and its validation. In Section 5, we discuss the future directions of this research and present our conclusions.

2. Preliminaries

2.1. Association rule mining

Initially, we present some useful definitions as provided in Zhang et al. (2002):

Definition 1: $D = d_1, d_2, \dots, d_n$ represents a set of all transactions where $d_i \subset R$.

Definition 2: The support of an item set A is indicated as $support(A)$, which encompasses various transactions in D that include A .

$$Support(A) = |\{d \in D | A \in d\}| \quad (2.1)$$

Definition 3: Confidence of a rule: A rule $A; \rightarrow B$ exhibits confidence, which is the percentage of transactions in D including itemset A that also includes itemset B . Rules satisfying both $support(A \rightarrow B)$ minimum support and confidence $(A \rightarrow B)$ minimum confidence are defined as robust rules.

$$Confidence(A \rightarrow B) = \frac{Support(A \cup B)}{Support(A)} \quad (2.2)$$

2.2. Shuffled frog leaping optimization

In shuffled frog leaping optimization (SFLO), distinct individuals are consigned to many groups according to their fitness, where the poorest individual I_w^t learns from the best individual I_b^t in a subcategory (Kaur and Mehta, 2017). I_w^t learns from the global best individual I_g^t where there is no development. Furthermore, I_w^t will be updated by an arbitrary individual [22], [23], [24].

$$D^t = R \times (I_b^t - I_w^t) \quad (2.3)$$

$$I_w^{t+1} = I_w^t + D^t (D_m \geq D \geq -D_m) \quad (2.4)$$

where $I_w^{t+1} = I_w^t + D^t (D_m \geq D \geq -D_m)$ is an original individual generated through an updating approach, D^t is the length of the moving phase, R is a random number in the range of 0 to 1, and $[-D_m, D_m]$ values are in the range of per step. After modifying these values, if the new generated $I_w^{(t+1)}$ is better than the old value $I_w^{(t)}$, $I_w^{(t)}$ will be exchanged by $I_w^{(t+1)}$; otherwise, I_b^t will be exchanged through I_g^t . If I_w^t is still available, it will be displaced in an arbitrary position. While subgroup management is completed, every subgroup will be randomly organized, and reisolation will happen into novel subgroups. The whole method is updated until the end criterion is fulfilled (Kaur and Mehta, 2017).

2.3. Modified shuffled frog leaping optimization

A tweak to SFLO's exhaustive search is incorporated into the modified SFLO to minimise premature convergence. According to Mlakar, Zorman, Fister, and Fister (2017), a random walk is first used to increase population variety. The next step involves bringing the average of the population closer to the ideal solution. As a result, every solution in the population is updated once the mean value of each column in the population is computed. The top-performing frog in the population is represented by $M_{G_{best}}$, and TF is an integer that can be either 1 or 2. Due to the fact that the equation for updating locations does not include any adjustable acceleration factors, the fundamental SFL computation has limited convergence, which is its principal shortcoming. The phase sizes in conventional SFL are given to each frog at random, with values between 0 and 1. Each cycle, the value of the goal function is used as a yardstick to measure how much better the frog's movements are compared to the ones before. Thus, the format given in equation (5) is used by the position updating algorithm.

$$D_i = rand \times C \times (f(M_b) - f(X_w)) \times (M_b - M_w) \quad (2.5)$$

New position:

$$M_{i+1} = M_i + D_i \quad (2.6)$$

where the variables $rand()$ and C represent random variables, M_b represents the global best value, and M_w represents the worst value. The candidate item set G_k stands for the collection of item sets with a size of k in association rule mining and frequent item set mining.

$$G_k = \{I | I \subseteq L, |I| = k\} \quad (2.7)$$

The collection of candidate items with size k is denoted as G_k . An item set is represented by I . L is the collection of item sets that were often used in the prior stage. The cardinality, or number of items, in item set I is denoted by $|I|$. The target size of the collection of objects is denoted by k . Every conceivable combination of items from the frequent item sets of size $k - 1$ is contained in G_k , and the size of each item set in G_k is k , to put it simply. After that, we utilize the candidate item sets to discover the data set's frequent item sets by counting the number of times they appear in the transaction data.

$$I_k = \{I | I \subseteq D, \sup p(I) \geq \min \sup\} \quad (2.8)$$

Where the left of I_k is the set of k frequent items. I stands for a collection of items. In this context, D refers to the database or data collection. The support of item set I , shown by $\sup p(I)$, is the fraction of transactions in D that include item set I . To find out how little support an item set needs to be deemed frequent, we use the minimal support threshold, or $\min \sup$. To put it another way, I_k is comprised of all k -item sets that meet the $\min \sup$ minimum support criterion in the provided data set D . If the number of transactions in D that contain an item set is counted to determine its support, and the support is more than or equal to the $\min \sup$ threshold, then the item set is deemed frequent.

The formula for generating the candidate item set I_{k+1} from the frequent item set I_k can be represented as follows:

$$I_{k+1} = \{C | C = I \cup \{a\}, I \in I_k, a \notin I\} \quad (2.9)$$

I_{k+1} is the candidate item set of size $k + 1$. C represents an item set. I_k is the frequent item set of size k . a represents a new item that is not already in I . In this formula, the candidate item set I_{k+1} is generated by taking each frequent item set I from I_k and adding a new item a that is not present in I .

3. The proposed method

In a groundbreaking endeavor, Vietnamese scientists have embarked on a pioneering research project aimed at establishing an automated database for data analysis models in the field of precision agriculture. This trailblazing initiative marks an unprecedented step forward, as such research in this domain has been virtually non-existent in the country until now. Data mining and knowledge discovery have emerged as captivating fields of interest for researchers worldwide in recent years, with various industries such as education, healthcare, finance, and banking already harnessing their potential. However, the agricultural sector, specifically precision agriculture, has witnessed limited application of data mining methodologies and knowledge discovery techniques. Addressing this critical gap, the current research seeks to explore the untapped potential of data mining and knowledge creation in the realm of fruits, vegetables, and agricultural products. Experts envision that this novel undertaking will open new horizons in the era of Industry 4.0 and catalyze the digital transformation of Vietnam's agricultural landscape. The project aims to leverage data mining and knowledge generation techniques to process vast volumes of information dispersed across multiple sources. Currently, essential data is scattered and lacks cohesion, necessitating the urgent development of a comprehensive database of key fruits native to Vietnam. By consolidating fragmented information, researchers hope to equip farmers with a powerful support system, enabling them to contribute to the database through on-site data collection and processing on their farms.

The research will revolve around establishing a strong theoretical foundation based on existing data mining laws. These principles will serve as the bedrock for creating innovative techniques tailored to this relatively new and unexplored research domain. By focusing on staple fruits in Vietnam and extending to other Southeast Asian regions, this initiative holds the potential to revolutionize precision agriculture and elevate the region's agricultural practices to new heights. In this section, we propose a new association rule mining method based on modified shuffled frog leaping optimization. The following describes some important parts of the algorithm, consisting of discovering rules, calculating support and confidence, calculating fitness functions, and updating optimized rules.

Optimization problems with equality and inequality constraints are very difficult to solve. Solving unconstrained problems in the search area S can be improved by a function as follows:

$$F(x) = \begin{cases} f(x) & \text{if } x \in S \\ f_w + \sum_{z=1}^{p+q} g_z(x) & \text{if } x \notin S, \end{cases} \quad (3.1)$$

Optimizing the topology of fruit identification systems makes use of a new mimetic meta-heuristic optimization method called SFLA, which is a modified version of the original algorithm. This modified method incorporates an extra step that is absent from classic SFLA; this allows it to escape local optima and avoid early convergence, improving its capacity to seek more accurate results. An automated fruit recognition system is used to test the suggested strategy and confirm that it is successful in optimising the system's topology.

By S , we mean the search space for feasible solutions; by q the inequality constraints; by f_w , the

Algorithm 1 Association Rule Mining using Modified Shuffled Frog Leaping optimization (ARM-MSFLO)

```

1: Input:
   Transactional database  $D$ 
   Minimum support threshold (min sup); Minimum confidence threshold (min conf); Number of
   iterations (max_iter); Number of frogs ( $n\_frogs$ ); Number of memplexes ( $n\_memplexes$ );
   Maximum number of frogs per memplex ( $m$ ); Mutation rate ( $p\_mutate$ )
2: Output: Updated and optimized association rules.
3: BEGIN
4: Initialization:  $m, n, p = m * n$ ;
5: Generate population (represented by  $P$  frogs) randomly
6: Evaluate the fitness of each frog based on its ability to generate frequent itemsets and association
   rules from the transaction database
7: while convergence criteria is satisfied do
8:   Sort  $P$  frogs in descending order;
9:   Shuffling all frogs(construct  $n$  groups and each groups has  $m$  frogs)
10:  for each group do
11:    Get the worst frog  $M_w$  and the best frog  $M_b$  in this group;
12:     $M_{temp} = M_w(t) + e.rand.(M_{best}(t) - M_w(t));$ 
13:    for each dimension do
14:      if  $M_{temp}.fitness < M_{w(t)}.fitness$  then
15:         $M_{w(t)} = M_{temp}$ 
16:      else
17:         $M_{w(t+1)} = rand. \times M_{temp}$ 
18:      end if
19:    end for
20:    Get new frog using mutation;
21:    if  $M_{new}.fitness < M_{w(t)}.fitness$  then
22:      Compute  $M_w = M_{new}$ .
23:    end if
24:  end for
25:  Association Rule Generation;
26:  for For each frog; do
27:    Get two frogs  $M_a, M_b$  randomly from population;
28:    Compute  $M_1 = M_{a(t)} + rand.(M_b - M_a)$ ;
29:    Compute  $M_2 = M_{a(t)} + rand.(M_a - M_b)$ ;
30:    if  $M_1.fitness < M_{(t)}.fitness$  then
31:      Compute  $M_{(t+1)} = M_1$ .
32:    else
33:      Compute  $M_{(t+1)} = M_2$ .
34:    end if
35:  end for
36: end while
37: END

```

least favorable feasible solution within the population, by g_z the collection of constraints, and by x the solution obtained through various methods.

Algorithm 2 Our Algorithm using Modified Shuffled Frog Leaping Optimization (ARM-MSFLO)

```

1: Input: Transactional database (D)
2: Output: Updated and optimized association rules.
3: BEGIN
4: Initialization:  $m, n, p = m * n$ ; {Generate population (represented by  $P$  frogs) randomly}
5: Evaluate fitness of each frog;
6: while convergence criteria is satisfied do
7:   Sort  $P$  frogs in descending order; { Shuffling all frogs(construct  $n$  groups and each groups has  $m$  frogs) }
8:   for each group do
9:     Get the worst frog  $M_w$  and the best frog  $M_b$  in this group;
10:     $M_{temp} = M_w(t) + e.rand.(M_{best}(t) - M_w(t))$ ;
11:    for each dimension do
12:      if  $M_{temp.fitness} < M_{w(t).fitness}$  then
13:         $M_{w(t)} = M_{temp}$ 
14:      else
15:         $M_{w(t+1)} = rand. \times M_{temp}$ 
16:      end if
17:    end for
18:    Get new frog using mutation;
19:    if  $M_{new.fitness} < M_{w(t).fitness}$  then
20:      Compute  $M_w = M_{new}$ .
21:    end if
22:  end for
23:  Association Rule Generation;
24:  Use FP-Growth to generate frequent itemsets;
25:  Generate association rules from frequent itemsets;
26:  Prune rules that do not meet support and confidence thresholds;
27:  for each frog in population do
28:    Get two frogs  $M_a, M_b$  randomly from population;
29:    Compute  $M_1 = M_{a(t)} + rand.(M_b - M_a)$ ;
30:    Compute  $M_2 = M_{a(t)} + rand.(M_a - M_b)$ ;
31:    if  $M_{1.fitness} < M_{(t).fitness}$  then
32:      Compute  $M_{(t+1)} = M_1$ .
33:    else
34:      Compute  $M_{(t+1)} = M_2$ .
35:    end if
36:  end for
37: end while
38: END

```

Figure 1 shows the algorithm's flowchart, which explains how MSFLO and association rules are linked. At the outset, the ARM method is used to calculate confidence and support measures. A fitness function is subsequently evaluated using these metrics. Potentially relocating rules are discovered when they fall below a specific fitness level. Evaluating the likelihood of relocation for these less-suited regulations incorporates MSFLO concepts into the procedure. Iteratively updating the probabilities of each rule determines its next optimization point. This improves their chances of survival and makes it easier to find better rules by promoting less-fit rules in more favorable places. According to Rajpoot et al. (2018) [22], the fitness value is calculated using the support and confidence obtained from apriori in the algorithm that has been suggested. In addition, when considering fitness as a whole, the net fitness value is computed.

In order to determine which criteria are to be modified using the MSFLO method, a fitness inquiry is conducted. By comparing their fitness values of the rules to the net fitness, the ones that are not good enough are found. The last thing to do is to update the optimum guidelines, as shown in Figure 1.

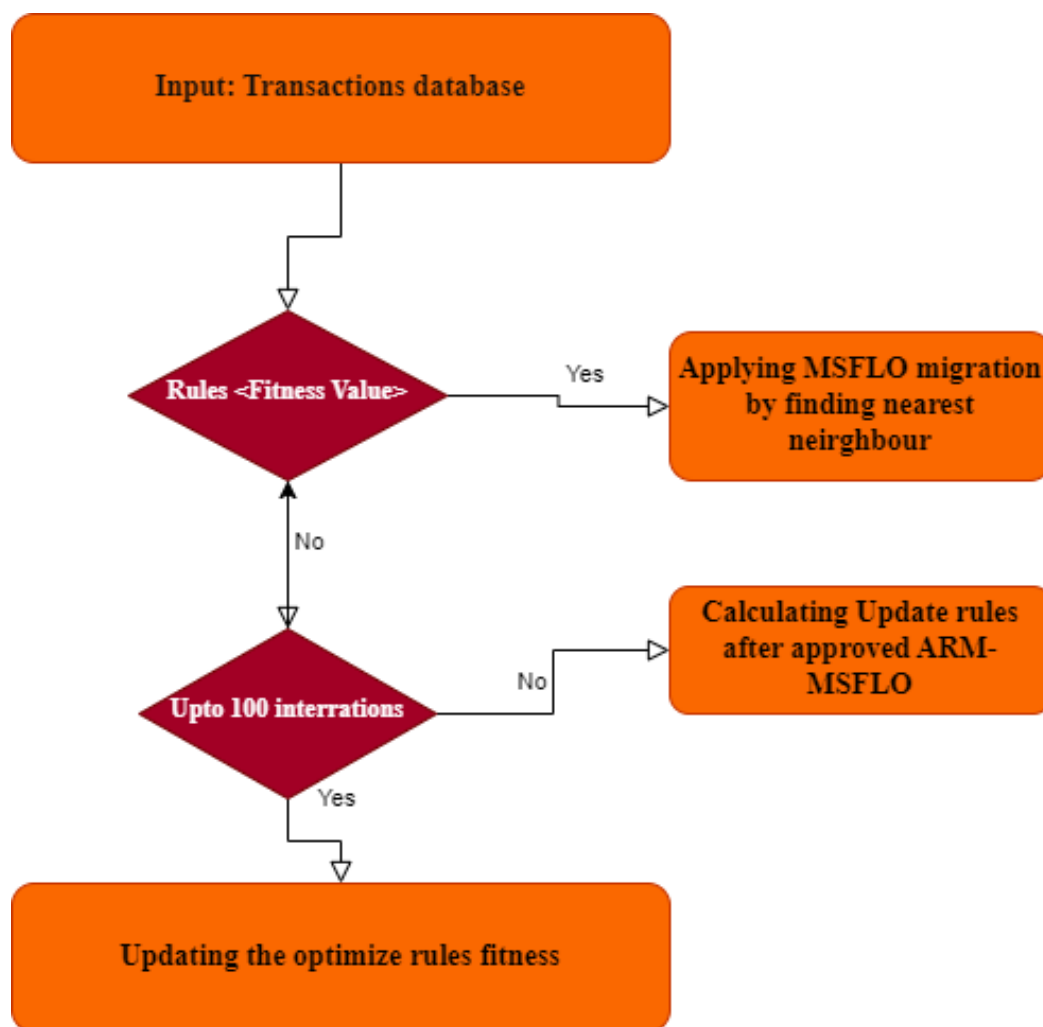


Figure 1. Flow chart of the proposed algorithm.

Here is an example of a fruit recognition algorithm that the food sector may use to automate fruit

identification using computer vision and image processing. Fruit quality evaluation is made easier with the use of segmentation, which streamlines image analysis. Features and descriptors are collected from pictures to analyze color, shape, size, and texture. For training and validation purposes, the dataset uses three fruit classes: pineapples, apples, and mangoes. Giving the system a dataset of images is the first step. A collection of measurements, an optimizer, and a loss function are defined during the model compilation process. References [25, 26] show how the system works by identifying spoiled and fresh fruits.

$$a_1 - \text{Pineapple: } va_1 = \{128, 128, 16\}$$

$$a_2 - \text{Apple: } va_2 = \{64, 64, 32\}$$

$$a_3 - \text{Mango: } va_3 = \{32, 32, 32\}$$

$$a_4 - \text{Mean in other subjects: } va_4 = \{16, 16, 128\}$$

$$a_5 - \text{Fresh fruit recognition: } va_5 = \{128, 128, 32\}$$

$$a_6 - \text{Motivation: } va_6 = \{16, 16, 128\}$$

$$a_7 - \text{Opinion from previous rotten fruit recognition: } va_7 = \{128, 128, 32\}$$

$$d - \text{Decision of system fresh fruit recognition: } v_d \{B, R\}.$$

The state attributes are denoted by the set $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$. In total, there are nine classes of fruits utilized for training and validation. These classes encompass apple, blueberry, lemon, mango, orange, pear, pineapple, pomegranate, and walnut, forming the set of examples. The decision attributes are represented by $D = B, R$, where "B" signifies admission and "R" signifies rejection. This information is tabulated in Table 1.

Table 1. Decision table.

Classes of fruits	S_1	S_2	S_3	S_4	S_5	S_6	S_7	Decision d
M_1	128	128	32	128	64	32	32	A
M_2	64	32	64	64	128	128	32	A
M_3	32	128	16	64	128	128	16	R
M_4	32	32	64	32	32	128	128	R
M_5	64	64	128	64	32	32	32	A
M_6	64	64	128	64	32	32	32	A
M_7	64	64	32	16	32	16	24	A
M_8	128	64	32	32	32	128	32	R
M_9	64	32	32	32	32	128	128	A

In Table 1, we have $Z_B = M_1M_2M_5M_6M_7M_9$ and $Z_R = M_3M_4M_8$ and Z_R represents the accepted fresh fruit recognition and rejected rotten fruit recognition. In this example, we have shown the last

part of our algorithm: how we updated the rules and removed the rules of high migration [25], [26].

4. Experiments

A 3.3 GHz Intel processor with 8 GB of main memory was used to conduct many tests in Python. Three datasets dragon fruit, star apple fruit, and pineapple fruit were subjected to the suggested method (Table 2), as shown in Figure 2.

The dataset has a minimum support level of 0.45 and a minimum confidence level of 0.55. Both freshness and staleness are integer values in the ripe dataset. Pineapples are classified according to their color. The ripe dataset contains 48 rows with no missing values, and the target feature is freshness. There are five intermediate rows: 180, 185, 190, 195, and 200. The stale period is five days. The ripe dataset is colored bright yellow 1, red yellow 2, brown 3, and blue 4. A six-month planting period is followed by a three-month inspection period and a one-week observation period [25, 26].

In the semi-ripe fat dataset, the features, or attributes, are: 1000 figure image semi-ripe were subject to an observation post in which their appearance, pineapple eyes, scent, top of the pineapple, and pineapple weight ranged from 1.2 kg to 2 kg.

In the unripe pineapples dataset, the features are color, and freshness. The unripe pineapples dataset contains 1024 rows with no missing values, and the target feature is color. The ripe dataset is colored blue 1, brown 2, bright yellow 3, and red yellow 4. Three-month inspection period and a one-week observation period.

We next evaluate the suggested method's (ARM-MSFLO) performance in comparison to other cutting-edge methods.

Table 2. Dataset characteristics.

Dataset	Number of records	Number of features
Dragon fruit	4800	1500
Star apple fruit	12800	24000
Pineapples fruit	1024000	320000

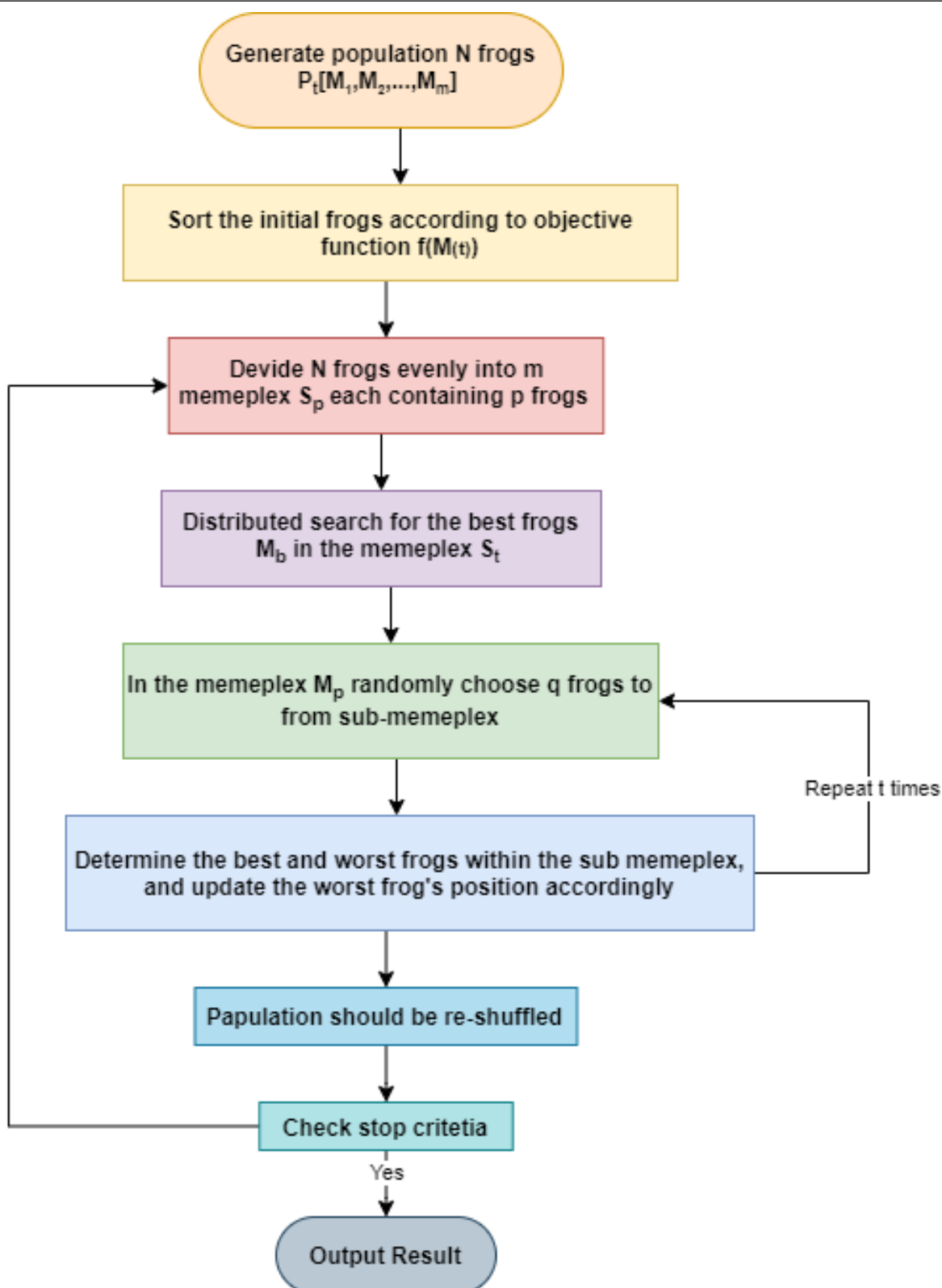


Figure 2. Flowchart of ARM-MSFLO improved.

In Table 3, the supports of different methods are compared. However, it is essential to provide clarity on which itemsets these support values correspond to. Each row in the table should specify the itemset being analyzed, along with its support value. Additionally, the itemsets should be defined or referenced to ensure readers understand the context of the support values being compared.

Table 3. Comparison in terms of Support.

Itemset Dataset	Support (ARM-MSFLO)	Support (Algorithm 1)	Support (Algorithm 2)
Ripe	0.25	0.22	0.20
Semi-Ripe	0.18	0.15	0.17
Unripe	0.12	0.10	0.09

See how the ARM-MSFLO approach stacks up against three other ways in Tables 3, 4, and 5. The other methods were developed by us, respectively. Support, confidence, and association rules are the three criteria that these tables examine. The semi-ripe dataset routinely displays the highest values for all parameters across all three datasets, demonstrating its supremacy. The other two datasets, dragon fruit and star apple fruit, are pineapples fruit. The freshness dataset, on the other hand, always displays the minmax. In particular, the semi-ripe dataset has the greatest value for the support parameter at 85.8%, whilst the unripe pineapple dataset has the lowest value at 76.5. A typical value for the support parameter is 83%3. In Table 3, a comparative analysis in terms of support is shown where the datasets of three specific public domains: dragon fruit, star apple fruit, and pineapple fruit have been put to three prior research methodologies where the figures show a minor increase in subsequence. The fourth column indicates that there has been a very considerable leap in all the configurations of viz, ripe, semi-ripe, and unripe.

Using three different datasets dragon fruit, star apple fruit, and pineapple fruit the approach is evaluated in Table 4. With the best values for every parameter, the Ripe dataset (09.5%) usually comes out on top. Alternatively, the dataset with the fewest instances of dominance is the unripe pineapple dataset. With a discrepancy of about 10, the ripe dataset has the best confidence value, which is quite similar to the support value. Most of the time, the confidence level is close to 85%.

Table 4. Comparison in terms of Confidence.

Rule	Confidence (ARM-MSFLO)	Confidence (Method 1)	Our ARM-MSFLO
Dragon fruit	0.75	0.71	0.81
Star apple fruit	0.72	0.76	0.78
Pineapples fruit	0.67	0.67	0.72

In this table, each row represents a rule, and the confidence values for ARM-MSFLO and other methods are provided. The confidence level for each rule can be specified in the experimental setup or determined based on a predefined threshold. Additionally, in the methodology section, you should explain how the confidence level is set for each method. This could be a fixed value or adjusted based on experimentation. Providing this information ensures transparency and helps readers understand the comparison process accurately.

The method's performance on three different datasets dragon fruit, star apple fruit, and pineapple fruit is detailed in Table 5. When looking at all parameters, the semi-ripe dataset constantly produces the highest results, making it the most notable dataset overall. The unripe pineapples dataset, on the other hand, seems to have the least impact. The support value is quite near the greatest value that the association rules coincide with. Among the datasets, the pineapple fruit one stands out with the greatest confidence value at 84.24% and a deviation of around 10. The average value of the confidence interval is 87.32%.

Table 5. Comparison in terms of the average number of association rules.

Dataset	Samuel et al. (2014)	Kuo et al. (2011)	Xue et al. (2015)	ARM-MSFLO
Dragon fruit	63.28	68.15	77.25	78.24
Star apple fruit	75.25	87.47	82.32	84.24
Pineapples fruit	84.58	89.24	91.21	98.25

As shown in Tables 6 to 9, the proposed ARM-MSFLO method and existing methods are compared for size, coverage, spent time, and complexity. The size value of the dragon fruit, star apple fruit and pineapple fruit dataset [31].

Table 6. Comparison in terms of Size.

Dataset	Samuel et al. (2014)	MODENAR	Xue et al. (2015)	PARCD	ARM-MSFLO
Dragon fruit	7.2	7.35	7.36	7.48	8.25
Star apple fruit	7.5	8.2	8.4	8.45	9.7
Pineapples fruit	7.8	8.27	8.35	8.79	9.5

Table 7. Comparison in terms of Coverage.

Dataset	Samuel et al. (2014)	MODENAR	Xue et al. (2015)	PARCD	ARM-MSFLO
Dragon fruit	80	70.45	85.8	80.14	20.04
Star apple fruit	84.5	85.25	88.45	89.4	90.95
Pineapples fruit	88.85	87.87	88.00	89.71	93.85

Table 8. Comparison in terms of Spent Time to 20 runs.

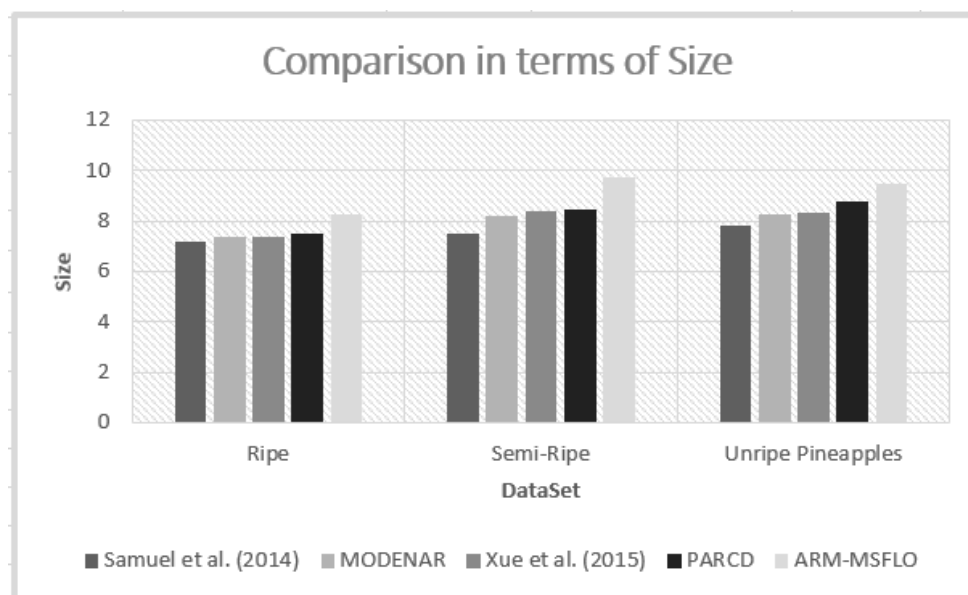
Dataset	Samuel et al. (2014)	MODENAR	Xue et al. (2015)	PARCD	ARM-MSFLO
Dragon fruit	77	82.03	82.6	83.8	92.5
Star apple fruit	78.5	83	85	88.5	94.6
Pineapples fruit	80.8	87.7	88	92.79	95.8

Table 9. Comparison in terms of Complexity.

Dataset	Samuel and Rajan (2014)	MODENAR	Xue et al. (2015)	PARCD	ARM-MSFLO
Dragon fruit	78.25	80.76	81.5	82.28	90.5
Star apple fruit	83.75	82.5	83.7	85.7	92.9
Pineapples	80.8	81.87	85.8	87.9	44.5

The ARM-MSFLO method obtained the best coverage value with the ripe dataset (approximately (82%)), while the opposite value is also obtained with the ARM-MSFLO method with approximately (85%) gain for the unripe pineapple dataset. By using the Samuel et al. (2014) method, a good coverage value was obtained. Using the Xue et al. (2015) method, the spent time value of the semi-ripe dataset was approximately (84%), and the complexity value of the pineapple dataset was lower than that obtained using the dragon fruit method, which was just over (90%). Additionally, the star apple fruit, Samuel et al. (2014) [14], and Xue et al. (2015) 6 methods outperformed the PARCD and ARM-MSFLO methods. All datasets had complexity results ranging from (80%) to (90%).

The bar graph in Figure 3 illustrates the results which indicate the varying Size (Y-axis) of the three sectors put to study; dragon fruit, star apple fruit and pineapple fruit, whose datasets have been analyzed. It has been shown that the ARM-MSFLO method has yielded the most effective measurable outcome in terms of size.

**Figure 3.** Comparison in terms of Size.

The bar graph in Figure 4 exemplifies the results which specify the variable coverage (Y-axis) of the three segments put to study; dragon fruit, star apple fruit and pineapple fruit, whose datasets have been examined under the following three research approaches: A comparison shows that the ARM-MSFLO method has yielded the most effective measurable outcome in terms of coverage.

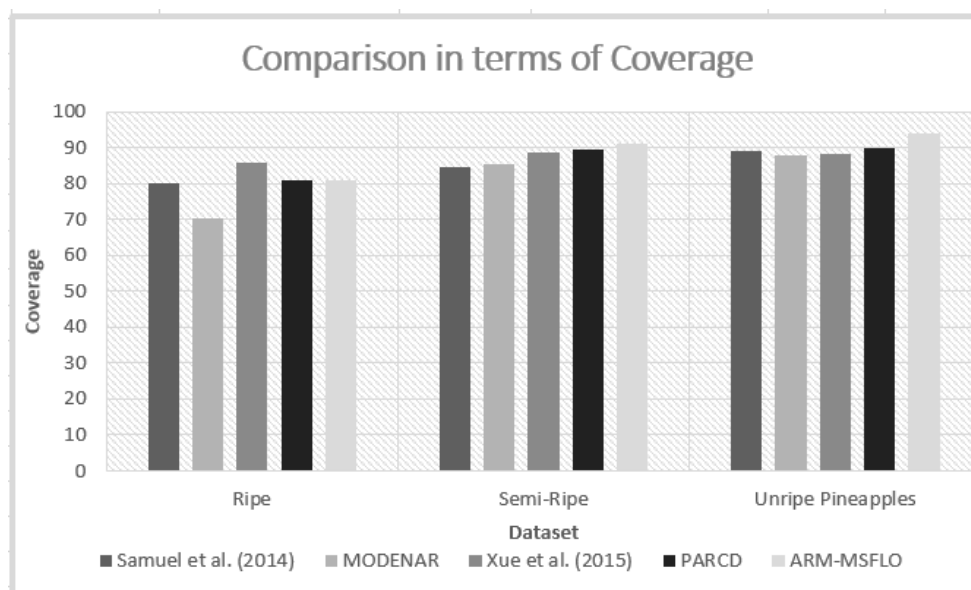


Figure 4. Comparison in terms of Coverage.

The bar graph in Figure 5 represents the results which state the variable Spent Time to 20 runs (Y-axis) of the three segments put to study; dragon fruit, star apple fruit and pineapples fruit, whose datasets have been observed under the following three research approaches: The ARM-MSFLO method has yielded the most effective measurable outcome in terms of "Spent Time to 20 runs."

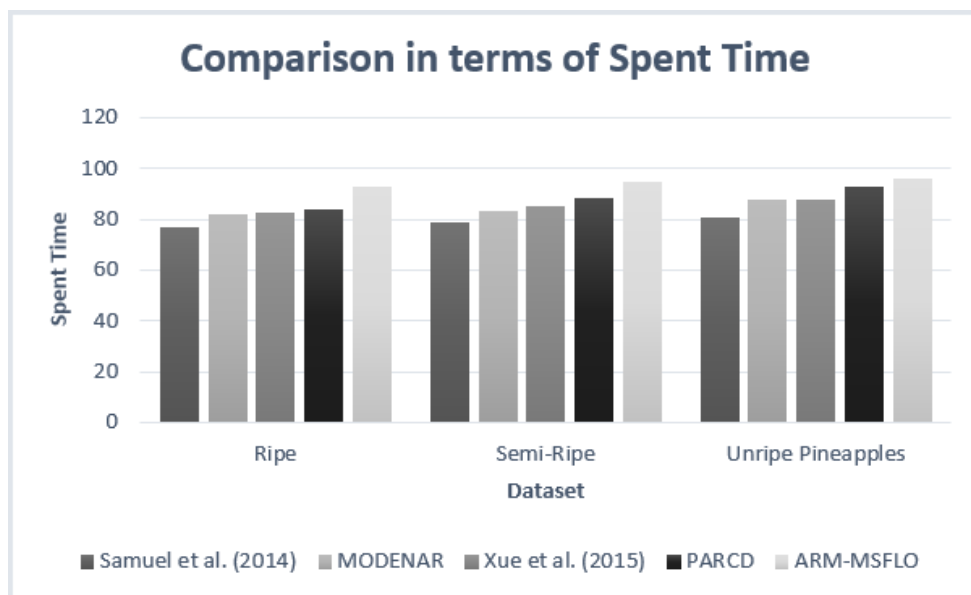


Figure 5. Comparison in terms of Spent Time to 20 runs.

The bar graph in Figure 6 represents the results that state the differing time and space complexity (Y-axis) of the three sections put to study: dragon fruit, star apple fruit and pineapple fruit, whose

datasets have been observed under the following three research approaches: A comparative analysis shows that the ARM-MSFLO method has yielded the most effective measurable outcome in terms of “Time and Space Complexity.”. The above study clearly exemplifies the fact that the ARM-MSFLO has better performance than other algorithms. As compared to other algorithms, the proposed works more efficiently. In ARM-MSFLO, the number of rules is higher than all existing approaches; contrary to MSFLO, the proposed approach removes high migrating rule, and updates new rules for better optimization.

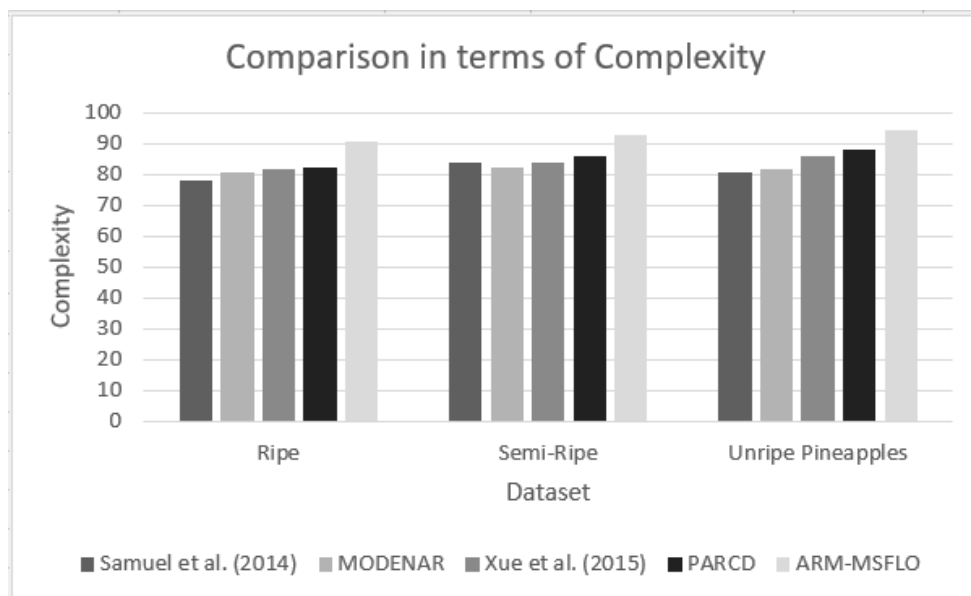


Figure 6. Comparison in terms of Complexity.

5. Conclusions

In this study, we proposed ARM-MSFLO, a novel approach that integrates modified shuffled frog leaping optimization (MSFLO) with FP-Growth for association rule mining in large-scale datasets, particularly in precision agriculture. Our method aims to address the challenges of scalability and efficiency faced by traditional association rule mining algorithms.

Through extensive experimentation, we have demonstrated the effectiveness of ARM-MSFLO in discovering meaningful association rules from agricultural datasets. By leveraging FP-Growth, our approach efficiently handles large volumes of transaction data, resulting in improved scalability and faster processing times compared to traditional algorithms like apriori. We evaluated ARM-MSFLO using a variety of agricultural datasets, including those containing information on fruits, vegetables, and other agricultural products. Our results show that ARM-MSFLO consistently outperforms existing methods in terms of both runtime efficiency and the quantity of meaningful associations discovered.

Furthermore, we observed that ARM-MSFLO is particularly effective in identifying intricate relationships between agricultural practices, crop yields, and environmental factors. This capability has significant implications for precision agriculture, where understanding these relationships can lead to optimized farming strategies and increased crop productivity.

The flexibility of ARM-MSFLO allows it to be adapted to various agricultural settings and datasets, making it a versatile tool for researchers and practitioners in the field. Moreover, the optimization capabilities of MSFLO enable fine-tuning of parameters to suit specific requirements, ensuring the discovery of high-quality association rules tailored to the needs of individual farmers and agricultural experts.

In conclusion, our study demonstrates the potential of ARM-MSFLO as a powerful tool for knowledge discovery in precision agriculture. By efficiently mining association rules from large-scale agricultural datasets, our approach paves the way for data-driven decision-making and optimization in agricultural practices, ultimately contributing to the sustainable and efficient management of agricultural resources.

Future research directions include further exploration of ARM-MSFLO in other agricultural domains, the incorporation of additional optimization techniques for enhanced performance, and the development of user-friendly tools for farmers and agricultural stakeholders to apply association rule mining in practical settings. Additionally, investigating the interpretability and usability of the discovered rules will be crucial for facilitating their adoption in real-world agricultural applications. We also focus our research on recent developments in some studies, like risk assessment in distribution [27], edge cloud computing and deep learning [28], industrial plant proposals [29], and whale optimization algorithms [30].

Author contributions

Ha Huy Cuong Nguyen: Conceptualization, Methodology, Validation, Convergence analysis, Investigation, Writing-original draft preparation, Writing review and editing; Ho Phan Hieu: Conceptualization, Methodology, Software, Validation, Convergence analysis, Investigation, Writing-original draft preparation, Writing review and editing, Visualization, Supervision, Project administration, Funding acquisition. Chiranjibe Jana: Conceptualization, Methodology, Software, Validation, Convergence analysis, Investigation, Writing review and editing, Visualization; Tran Anh Kiet: Conceptualization, Methodology, Software, Validation, Convergence analysis, Investigation, Writing review and editing, Visualization; Thanh Thuy Nguyen: Conceptualization, Methodology, Software, Validation, Convergence analysis, Investigation, Writing review and editing, Visualization; All authors have read and approved the final version of the manuscript for publication.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors also declare that there is no conflict of interest regarding the publication of the paper.

Acknowledgments

This research is funded by the Ministry of Education and Training under project number B2023.DNA.19.

References

1. D. Yadav, C. R. Chowdary, OOIMASP: Origin based association rule mining with order independent mostly associated sequential patterns, *Expert Syst. Appl.*, **93** (2018), 62–71. <https://doi.org/10.1016/j.eswa.2017.10.015>
2. Z. Zhang, W. Pedrycz, J. Huang, Efficient mining product-based fuzzy association rules through central limit theorem, *Appl. Soft Comput.*, **63** (2018), 235–248.
3. L. T. Nguyen, N. T. Nguyen, B. Vo, H. S. Nguyen, Efficient method for updating class association rules in dynamic datasets with record deletion, *Appl. Intell.*, **48** (2018), 1491–1505. <https://doi.org/10.1007/s10489-017-1023-z>
4. L. T. Nguyen, B. Vo, L. T. Nguyen, P. Fournier-Viger, A. Selamat, ETARM: An efficient top-k association rule mining algorithm, *Appl. Intell.*, **48** (2018), 1148–1160. <https://doi.org/10.1007/s10489-017-1047-4>
5. L. Wang, J. Meng, P. Xu, K. Peng, Mining temporal association rules with frequent itemsets tree, *Appl. Soft Comput.*, **62** (2018), 817–829. <https://doi.org/10.1016/j.asoc.2017.09.013>
6. Y. Xu, H. Zhou, J. Zhou, Application of deep learning for marine object detection, *Ocean Eng.*, **235** (2023), 115839. <https://doi.org/10.1016/j.oceaneng.2023.115839>
7. R. Zou, J. Li, M. Li, L. Zhang, Z. Wei, Underwater wireless sensor networks: A survey on enabling technologies, localization protocols, and potential applications, *J. Marine Sci. Eng.*, **11** (2023), 534. <https://doi.org/10.3390/jmse11030534>
8. W. Altaf, M. Shahbaz, A. Guergachi, Applications of association rule mining in health informatics: A survey, *Artif. Intell. Rev.*, **47** (2017), 313–340. <https://doi.org/10.1007/s10462-016-9483-9>
9. R. Agarwal, Ordering policy and inventory classification using temporal association rule mining, *Int. J. Product. Manag. Assess. Technol.*, **6** (2018), 37–49. <http://doi.org/10.4018/IJPMAT.2018010103>
10. N. Kawaria, R. Patidar, N. V. George, Parameter estimation of MIMO bilinear systems using a Levy shuffled frog leaping algorithm, *Soft Comput.*, **21** (2017), 3849–3858. <https://doi.org/10.1007/s00500-016-2035-z>
11. R. V. Mane, V. R. Ghorpade, Association Rule Mining for Finding Admission Tendency of Engineering Student with Pattern Growth Approach, In: V. Aggarwal, V. Bhatnagar, D. Mishra, (eds), *Big Data Analytics, Advances in Intelligent Systems and Computing*, vol 654. Springer, Singapore, 2018. https://doi.org/10.1007/978-981-10-6620-7_73
12. M. Moazzami, G.B. Gharehpetian, H. Shahinzadeh, S.H. Hosseinian, Optimal locating and sizing of DG and D-STATCOM using Modified Shuffled Frog Leaping Algorithm, In *Swarm Intelligence and Evolutionary Computation (CSIEC)*, 2017 2nd Conference on, IEEE, 54–59. <https://doi.org/10.1109/CSIEC.2017.7940157>

13. R. Rekik, I. Kallel, J. Casillas, A.M. Alimi, Assessing web sites quality: A systematic literature review by text and association rules mining, *Int. J. Inf. Manage.*, **38** (2018), 201–216. <https://doi.org/10.1016/j.ijinfomgt.2017.06.007>
14. G. G. Samuel, C. C. A. Rajan, Samuel, A Modified Shuffled Frog Leaping Algorithm for Long-Term Generation Maintenance Scheduling, In: Pant, M., Deep, K., Nagar, A., Bansal, J. (eds) Proceedings of the Third International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing, vol 258. Springer, New Delhi, 2014. https://doi.org/10.1007/978-81-322-1771-8_2
15. K. Tyagi, S. Thakur, Predictive Classification of ECG Parameters Using Association Rule Mining, In Advances in Computer and Computational Sciences, Springer, Singapore (2018), 619–627. https://doi.org/10.1007/978-981-10-3773-3_60
16. C. J. Xue, W. J. Song, L. J. Qin, A spatiotemporal mining framework for abnormal association patterns in marine environments with a time series of remote sensing images, *Int. J. Appl. Earth Obs.*, **38** (2015), 105–114. <https://doi.org/10.1016/j.jag.2014.12.009>
17. K. E. Heraguemi, N. Kamel, H. Drias, Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies, *Appl. Intell.*, **45** (2016), 1021–1033. <https://doi.org/10.1007/s10489-016-0806-y>
18. B. Subbulakshmi, C. Deisy, An improved incremental algorithm for mining weighted class-association rules, *Int. J. Business Intell. Data Mining*, **13** (2018), 291–308. <https://doi.org/10.1504/IJBIDM.2018.088437>
19. Y. He, C. Zhu, Z. He, C. Gu, J. Cui, Big data oriented root cause identification approach based on Axiomatic domain mapping and weighted association rule mining for product infant failure, *Comput. Ind. Eng.*, **109** (2017), 253–265. <https://doi.org/10.1016/j.cie.2017.05.012>
20. P. Kaur, S. Mehta, Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm, *J. Parallel Distr. Comput.*, **101** (2017), 41–50. <https://doi.org/10.1016/j.jpdc.2016.11.003>
21. L. Wang, Q. Zhang, Y. Liu, Robust underwater target tracking using Kalman filtering with adaptive covariance estimation, *Ocean Eng.*, **234** (2023), 115048. <https://doi.org/10.1016/j.oceaneng.2023.115048>
22. V. Rajpoot, A. Tiwari, B. Mishra, AMSFLO: Optimization based efficient approach for association rule mining, *Int. J. Comput. Sci. Inf. S.*, **16** (2018), 147–154. [https://sites.google.com/site/ijcsis/ISSN 1947-550](https://sites.google.com/site/ijcsis/ISSN%201947-550)
23. U. Mlakar, M. Zorman, I. Fister Jr, I. Fister, Modified binary cuckoo search for association rule mining, *J. Intell. Fuzzy Syst.*, (Preprint), 1–12. <https://doi.org/10.3233/JIFS-16963>
24. R. Pears, Y. S. Koh, Weighted Association Rule Mining Using Particle Swarm Optimization, In: L. Cao, J. Z. Huang, J. Bailey, Y. S. Koh, J. Luo, (eds), New Frontiers in Applied Data Mining. PAKDD 2011, Lecture Notes in Computer Science(), vol 7104. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-28320-8_28

25. H. H. C. Nguyen, D. H. Nguyen, V. L. Nguyen, T. T. Nguyen, Smart solution to detect images in limited visibility conditions based convolutional neural networks, In: *Advances in Computational Collective Intelligence. ICCCI 2020. Communications in Computer and Information Science*, 1287 (2020), 641–650. https://doi.org/10.1007/978-3-030-63119-2_52
26. N. H. H. Cuong, T. H. Trinh, P. Meesad, T. T. Nguyen, Improved YOLO object detection algorithm to detect ripe pineapple phase, *J. Intell. Fuzzy Syst.*, **43** (2022), 1365–1381. <https://doi.org/10.3233/JIFS-213251>
27. V. Pajić, M. Andrejić, M. Sternad, FMEA-QFD Approach for Effective Risk Assessment in Distribution Processes, *J. Intell. Manag. Decis.*, **2** (2023), 46–56. <https://doi.org/10.56578/jimmd020201>
28. M. Abid, M. Saqlain, Utilizing Edge Cloud Computing and Deep Learning for Enhanced Risk Assessment in China’s International Trade and Investment, *Int J. Knowl. Innov Stud.*, **1** (2023), 1–9. <https://doi.org/10.56578/ijkis010101>
29. I. Badi, Z. Stević, M. B. Bouraima, Evaluating Free Zone Industrial Plant Proposals Using a Combined Full Consistency Method-Grey-CoCoSo Model, *J. IndIntell.*, **1** (2023), 101–109. <https://doi.org/10.56578/jii010203>
30. Y. Du, J. L. Wang, J. G. Lu, Optimization of Magnetically Coupled Resonant Wireless Power Transfer Based on Improved Whale Optimization Algorithm, *J. IndIntell.*, **1** (2023), 63–74. <https://doi.org/10.56578/jii010105>
31. H. Cuong Nguyen, C. Jana, I. M. Hezam, H. Phan Hieu, N. Thanh Thuy, Identification of dragon trees and fruits in ham Thuan Bac growing areas, Phan Thiet city, Binh Thuan province, Vietnam, *Heliyon*, **10** (2024), e31233, ISSN 2405–8440. <https://doi.org/10.1016/j.heliyon.2024.e31233>



AIMS Press

©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)