



ARTICLE

A Quarterly High RFM Mining Algorithm for Big Data Management

Cuiwei Peng¹, Jiahui Chen^{2,*}, Shicheng Wan³ and Guotao Xu⁴

¹School of International Education, Guangdong University of Technology, Guangzhou, 510006, China

²School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, 510006, China

³Business Administration, South China University of Technology, Guangzhou, 510641, China

⁴Guangzhou Institute of Technology, Xidian University, Guangzhou, 710071, China

*Corresponding Author: Jiahui Chen. Email: csjhchen@gmail.com

Received: 19 May 2024 Accepted: 09 August 2024 Published: 12 September 2024

ABSTRACT

In today's highly competitive retail industry, offline stores face increasing pressure on profitability. They hope to improve their ability in shelf management with the help of big data technology. For this, on-shelf availability is an essential indicator of shelf data management and closely relates to customer purchase behavior. RFM (recency, frequency, and monetary) pattern mining is a powerful tool to evaluate the value of customer behavior. However, the existing RFM pattern mining algorithms do not consider the quarterly nature of goods, resulting in unreasonable shelf availability and difficulty in profit-making. To solve this problem, we propose a quarterly RFM mining algorithm for On-shelf products named OS-RFM. Our algorithm mines the high recency, high frequency, and high monetary patterns and considers the period of the on-shelf goods in quarterly units. We conducted experiments using two real datasets for numerical and graphical analysis to prove the algorithm's effectiveness. Compared with the state-of-the-art RFM mining algorithm, our algorithm can identify more patterns and performs well in terms of precision, recall, and F1-score, with the recall rate nearing 100%. Also, the novel algorithm operates with significantly shorter running times and more stable memory usage than existing mining algorithms. Additionally, we analyze the sales trends of products in different quarters and seasonal variations. The analysis assists businesses in maintaining reasonable on-shelf availability and achieving greater profitability.

KEYWORDS

Data mining; recency pattern; high-utility itemset; RFM pattern mining; on-shelf management

1 Introduction

Retail has become increasingly competitive, prompting offline stores to confront mounting profit pressures. Effective shelf management is a critical determinant of store performance, as it directly influences sales and ultimately shapes profit potential. In response, many retailers have leveraged big data technologies to enhance their shelf management capabilities. For instance, applying frequent pattern mining techniques [1] enables retailers to identify frequently occurring itemsets in transaction databases, optimize shelf layout for customer convenience, and contribute to increased profitability.



However, frequent pattern mining techniques solely focus on item frequency, overlooking other crucial factors such as profit margins. Subsequently, utility pattern mining methods have been employed to identify high-profit itemsets from transaction databases [2]. Placing these high-value products in prominent shelf locations where customers are most likely to see them can optimize shelf layout, maximizing the benefits of on-shelf products [3]. Nevertheless, most utility pattern mining algorithms only focus on identifying the profitability of products without regard for shelf occupancy time. This oversight is impractical since shelf occupancy time varies: some products are seasonal or influenced by holidays, performing efficiently only during specific seasons [4]. We illustrate the inadequacy of utility pattern mining results by considering two scenarios. In the first scenario, low-priced items like ice cream cause high utility only during summer, not throughout the year. Consequently, the mining results might mislabel ice cream as low-utility items. In the second scenario, products with high prices, like down jackets, may show significant annual turnover in utility pattern mining results. However, these products are only bestsellers during specific seasons (e.g., summer and winter). Therefore, mining results overlook the temporal factor, fail to reveal the actual sales patterns of products comprehensively, and are unable to assist retailers in determining the optimal selling time for products. Consequently, traditional utility pattern mining algorithms lack applicability in this regard. The study [5] proposed on-shelf utility pattern mining to address the issue of uneven on-shelf time for products. Although they accounted for product on-shelf time, their method was insufficiently rational and practical, as it merely divided the total time into several time slots without considering the practical application in offline stores.

A key metric in shelf management is On-Shelf Availability (OSA), which refers to the ratio of actual stocked items on the shelf to the total capacity of the shelf, indicating the proportion of non-empty products on the shelf. A low OSA indicates stockouts, where consumers cannot find the products they plan to buy on the supermarket shelves. Stockouts inevitably lead to unpleasant customer shopping experiences, reducing customer satisfaction and loyalty while forfeiting valuable profit opportunities [6]. This outcome is undesirable for retailers. Hence, many retailers have realized the importance and necessity of improving OSA and have taken measures to enhance it. However, unquestioningly striving for high OSA on all shelves may not maximize profits. Maintaining full-shelf OSA requires constantly stocking sufficient inventory, incurring significant replenishment costs. Additionally, in smaller stores, excessive stocking can lead to product clutter and a crowded shopping environment, adversely affecting the customer experience. Moreover, excessively high OSA indicates slow-moving products, suggesting a failure to stock items aligned with customer preferences and inefficient use of limited shelf resources, resulting in wastage. Therefore, a suitable OSA is crucial and necessary. It not only enhances the shopping experience for customers but also increases the potential for business profitability.

In the retail industry, RFM pattern mining is a commonly used and effective method that segments and evaluates customers based on their behavior. According to RFM analysis, retailers can better understand various customer segments' purchasing habits, enabling them to formulate targeted marketing strategies to increase sales volume and profitability. However, RFM analysis assumes the availability of accurate customer identity information, which is often challenging for retailers to obtain. For instance, some customers do not use membership cards or share their cards with others, and thus, it is difficult for retailers to static precise customer identity information. Therefore, the RFM pattern was proposed to find interesting results without requiring customer identity information. RFM patterns refer to sets of patterns within recency (R), frequency (F), and monetary (M) values that are all higher than predefined minimum thresholds. RFM pattern mining holds practical and effective significance for the retail industry. In detail, recency reflects the recent activity level of

the product, with higher recency indicating recent popularity. Frequency denotes the number of times the product has been purchased, indicating its popularity. Monetary represents the sales amount of the product, reflecting its utility value. It aims to discover customer purchasing preferences and potential interesting patterns based on transaction records. This helps retailers manage products on shelves, maintain reasonable shelf availability, and ultimately, greater business profitability. However, most RFM pattern mining algorithms do not consider OSA application. The core contribution of these heuristic approaches is how to be more efficient in mining RFM patterns.

To address the above issues, we propose a novel OS-RFM pattern mining algorithm considering seasonality. Compared with existing RFM algorithms that do not consider time factors, the newly designed OS-RFM pattern considers the shelf occupancy time of products every quarter. This allows us to analyze the sales trends of products in different quarters and seasonal variations, thereby adjusting the products on shelves. Thus, the issue of shelf availability in the retail industry is tackled. In addition, compared to the existing on-shelf utility pattern mining, the superiority of the proposed method lies in its more precise and practical consideration of the shelf occupancy time of products, which is more directly applicable to offline stores. However, it is essential to note that since the on-shelf time is measured every quarter, the R and F values calculation is also quarterly-based. This study aims to utilize the mining results of the OS-RFM patterns to analyze the sales trends of products in different quarters and seasonal variations, thereby adjusting the products on shelves to maintain reasonable shelf availability, enhance the sales capability of retailers, and achieve greater profitability. The major contributions of our work are summarized as follows:

1. We introduce a quarterly RFM pattern mining algorithm for on-shelf datasets, termed OS-RFM, aimed at mining comprehensive OS-RFM patterns from transaction databases containing temporal records. Our algorithm can mine the patterns of high recency, high frequency, and high monetary utility and consider the period of the on-shelf goods in quarterly units. Thus, it can solve the problem of on-shelf availability in the retail industry.
2. We conduct experimental evaluations and comparisons on real datasets and synthetic datasets to assess the effectiveness and performance of our proposed algorithm. Compared with the state-of-the-art RFM mining algorithm, our algorithm can mine more patterns and is effective in Precision, Recall, and F1-score features. Also, our algorithm outperforms the compared algorithms regarding running time and memory usage.
3. We perform performance testing on a synthetic dataset, obtaining information on products with high recency, frequency, and monetary value across all quarters. Additionally, we analyze product sales trends in different quarters and seasonal variations. The analysis assists businesses in maintaining reasonable on-shelf availability and achieving greater profitability.

The rest of the content of this paper is structured as follows: [Section 2](#) discusses related work. Then, [Section 3](#) introduces basic definitions and problem statements of the OS-RFM pattern mining task, and [Section 4](#) elaborates on the new algorithm proposed in this study. Furthermore, [Section 5](#) shows the experimental results. Finally, [Section 6](#) provides conclusions and an outlook for future work.

2 Related Work

2.1 On-Shelf Utility Mining

In the face of increasing profit pressures in the retail industry, many researchers have begun to focus on solving the problem of maximizing shelf profitability. Yang et al. [7] proposed a comprehensive model to optimize shelf space allocation and management, which enhances the effectiveness of

allocation while reducing inventory levels and improving customer satisfaction. Satapathy et al. [8] introduced simple selfie cameras for real-time shelf checks, enabling restocking alerts and increasing shelf availability. Additionally, some researchers have integrated shelf management with big data technology applications. For instance, Chan et al. [2] utilized utility pattern mining to discover high-profit itemsets from transaction databases, optimizing shelf displays for maximizing on-shelf profitability. However, most utility mining algorithms assume uniform shelf occupancy time for all items, whereas in reality, shelf occupancy time varies, influenced by factors such as seasonality and holidays. For example, ice cream sells well in summer, while turtleneck sweaters are popular in winter. To address the issue of uneven shelf occupancy time, Lan et al. [5] introduced the concept of high-utility itemsets on shelves, considering both profit and shelf occupancy time, and proposed a two-stage algorithm for mining high-utility itemsets. Recently, Chen et al. [9] utilized a heuristic pattern mining algorithm to discover on-shelf high utility itemsets from transaction databases. This helps retailers optimize shelf layout by grouping itemsets to increase profitability.

All the above studies considered products' shelf occupancy time. However, compared with previous studies, the algorithm proposed in this paper can provide adequate quarterly sales data for retailers by considering the quarterly shelf occupancy time, which is conducive to direct application by enterprises and can quickly adjust shelf layout to meet the demand for goods in different seasons, ultimately increasing the profitability of enterprises.

2.2 RFM Pattern Mining

RFM analysis is a data-driven marketing technique based on customer purchasing behavior, which serves as a powerful and effective method for clustering or classifying customers. Tavakoli et al. [10] pointed out that RFM analysis can provide helpful information for retailers, such as predicting customers' behavior, including whether they will revisit the market soon, their shopping frequency, and expenditure amounts. Heldt et al. [11] proposed a new model called RFM/P to predict the customer value of each product. This model combines product and customer marketing perspectives, giving managers a more complete strategic perspective on the product portfolio and customers. Tasabat et al. [12] proposed the RFMS model and developed a more efficient classification method for managing customer relationships, which improved the efficiency process of the company. However, RFM analysis provides only a basic understanding of customer behavior and may not reveal underlying patterns or relationships. Data mining technology, on the other hand, can assist businesses in gaining a deeper insight into customer behavior by analyzing transaction data. This approach can uncover hidden patterns and relationships, categorize customers into different levels, and facilitate customer segmentation, ultimately enhancing customer engagement. Therefore, RFM analysis is often combined with data mining techniques [13] to discover customer behavior patterns, thereby assisting companies in making better business decisions. More specifically, RFM analysis calculates the corresponding scores for each customer based on their recency, frequency, and monetary values to derive each customer's RFM score. Subsequently, customers are segmented into different groups by taking collected scores as a basis. Companies can thus develop more targeted and personalized marketing strategies to meet the unique needs of different customer groups. Liao et al. [14] proposed a multi-behavior RFM based on the self-organizing map algorithm to analyze various customer behaviors. Chen et al. [15] developed a data-driven recommendation model based on customer purchase and product sales behavior using improved RFM analysis and association rule mining to enhance retail transactions.

However, the above studies consider customer identity to be crucial information. In reality, customer identity information is sensitive and often undisclosed. Moreover, retailers often struggle to

obtain accurate customer identity information because some customers do not possess membership cards or potentially lend them to others, thereby complicating the process of obtaining accurate customer identity information. Hu et al. [16] first applied RFM analysis to pattern mining without customer identity information. They proposed the RFM pattern concept and demonstrated that “RFM customers” are subsets of patterns in some cases. Subsequently, Wan et al. [17] proposed a more efficient and scalable RFM mining algorithm called RFMUL. Besides, Wan et al. [18] also proposed another efficient mining algorithm to address the RFM pattern mining task, which performs better than previous approaches.

For comprehensive and practical considerations, this paper also explores RFM patterns and discovers attractive customer purchasing preferences and sales patterns of products, solely based on customers’ product transaction records, without requiring customer identity information. Although RFM analysis can be conducted without customer identity information, the quality and integrity of data remain challenging. However, our algorithm can provide adequate quarterly sales data, which is beneficial for businesses to analyze product sales trends in different quarters and seasonal variations, thereby maintaining reasonable on-shelf availability and achieving greater profitability.

3 Preliminaries and Problem Definition

In this section, we introduce the preliminaries and definitions related to on-shelf RFM pattern mining task. Some basic definitions are as follows.

Definition 1. (*Transaction Database*) Let $I = \{x_1, x_2, \dots, x_m\}$ be a finite set of items consisting of m distinct items. Let D be a transaction database collecting a set of transactions, $D = \{T_1, T_2, \dots, T_n\}$ where n is the number of transactions in the database. Each transaction T_c in D has a unique identifier c called its transaction *tid*. For each item i within a transaction T_c , there exists a positive number $q(i, T_c)$ known as its internal utility and a positive number $p(i)$ denoting its external utility. Periods are represented as positive integers in a *PE set*. Every transaction T_c is linked to a period $P_i(T_c)$, indicating the period when the transaction takes place. Table 1 serves as the running example in this paper, and Table 2 presents the external utility of these items.

Table 1: An example transaction database with periods

<i>Tid</i>	Transaction	Period
T_1	$(A, 2)(C, 3)(D, 2)$	h_1
T_2	$(A, 3)(C, 5)(F, 6)(G, 4)$	h_1
T_3	$(A, 3)(B, 2)(C, 1)(D, 4)(E, 2)(F, 5)(G, 1)$	h_2
T_4	$(B, 1)(C, 3)(F, 4)(G, 2)$	h_2
T_5	$(B, 3)(D, 4)(E, 5)(G, 1)$	h_3

Table 2: External utility (unit profit) of each item

Item	A	B	C	D	E	F	G
Utility (\$)	2	1	3	4	1	2	2

Definition 2. (*Utility of an Itemset in a period*) Let p_i denote a set of positive integers representing periods.

$$p_i(X) = \{pt(T_c | T_c \in D \wedge X \subseteq T_c)\} \quad (1)$$

During a period $h \in (X)$, the utility of an itemset X is defined as

$$u(X, h) = \sum_{T_c \in D \wedge h = pt(T_c) \wedge X \subseteq T_c} \sum_{i \in X \wedge X \subseteq T_c} u(i, T_c) \quad (2)$$

The overall utility of an itemset X in a D is denoted as

$$u(X) = \sum_{h \in p_i(X)} u(X, h) \quad (3)$$

Definition 3. (*Relative Utility of an Itemset*) For a transaction T_c , its utility is calculated as the sum of individual item utilities in T_c , which is denoted as

$$TU(T_c) = \sum_{i \in T_c} u(i, T_c) \quad (4)$$

The total utility of the periods of X is computed as

$$to(X) = \sum_{h \in p_i(X) \wedge T_c \in D \wedge h = pt(T_c)} TU(T_c). \quad (5)$$

Besides, the utility of a period h is denoted as

$$pto(h) = \sum_{T_c \in D \wedge h = pt(T_c)} TU(T_c) \quad (6)$$

The relative utility of an itemset X for a period h is

$$ru(X, h) = u(X, h) / pto(h) \quad (7)$$

The above content is all about the definitions related to on-shelf pattern mining. Since this paper integrates the RFM pattern, the following will introduce the relevant definitions of RFM pattern.

Definition 4. (*Transaction-Weighted Utilization [19]*) For a specific period h , TWU of an itemset X is the sum of the transaction utilities of transactions from h that contain X , and denoted as

$$TWU(X, h) = \sum_{T_c \in D \wedge X \subseteq T_c \wedge pt(T_c) = h} TU(T_c) \quad (8)$$

Property 1. For a given period h , the TWU of an itemset X represents the upper-bound on the utility of X , $TWU(X, h) \geq u(X, h)$. In addition, TWU exhibits anti-monotonicity. When considering two itemsets X and Y , if $X \subset Y$, there is a relationship such that $TWU(X, h) \geq TWU(Y, h)$. Based on the definition of relative utility, it can find that $TWU(X, h) / pto(h) \geq u(X, h) / pto(h)$. Hence, $TWU(X, h) / pto(h)$ serves as an upper bound on $ru(X, h)$. Furthermore, the proof details were presented in [19].

Strategy 1. (*TWU Pruning Strategy*) For an itemset X , if there does not exist a period h for which $TWU(X, h) / pto(h) \geq \beta$, $TWU(Y, h) / pto(h)$ must also less than β according to Property 1, then the itemset X and all its supersets are not considered OS-RFM itemsets.

Definition 5. (*Recency pattern*) Consider an itemset X within a transaction T_c ,

$$R(X, T_c) = (1 - \delta)^{|D| - T_c} \quad (9)$$

where $|D|$ denotes the total number of transactions in D . Moreover, the sum of recency values across all transactions in D containing X is given by

$$R(X) = \sum_{T_c \subseteq D} R(X, T_c) \quad (10)$$

For a period h , the sum of recency values across all transactions containing X in h is denoted as

$$R(X, h) = \sum_{T_c \subseteq D \wedge h} R(X, T_c) \quad (11)$$

An itemset is classified as a recency pattern only if $R(X, h)$ is not less than the user-specified minimum recency threshold γ ($0 \leq \gamma$). It is important to note that OS-RFM-patterns is the subject of recency patterns.

Property 2. According to the Definition 5, the recency value is related to the number of the transactions containing the itemset. The inequality $R(X, h) \leq R(X)$ is always true because the number of the transactions containing this itemset in h is less than in D .

Proof. Assuming two itemsets X and Y ($X \subseteq Y$), the number of transactions containing X is more than that containing Y . Therefore, if $R(X)$ does not satisfy the condition, then $R(Y)$ must also be lower than γ . Similarly, for a period h , if $R(X, h) \geq \gamma$, then $R(Y, h) \geq \gamma$.

Strategy 2. (*Recency Pruning Strategy*). Based on Property 2, if $R(X)$ is less than the minimum recency threshold, it is unnecessary to explore itself and its supersets in the search space. It should be noted that this pruning condition is used to filter out the 1-itemset before mining n -itemset ($n \geq 2$). Additionally, if $R(X, h)$ less than the minimum recency threshold, it is unnecessary to explore X and its supersets in the search space too.

Definition 6. (*Frequency Pattern*) The frequency value reflects the count of occurrences of an itemset, and it is equal to the number of the transactions that contain X . In a database D , it is denoted as

$$F(X) = \sum_{X \subseteq T_c \wedge T_c \subseteq D} Count(T_c) \quad (12)$$

For a period h , it is denoted as

$$F(X, h) = \sum_{X \subseteq T_c \wedge h \in \text{pi}(T_c)} Count(T_c) \quad (13)$$

In particular, only if $F(X, h)$ not less than $|D| \times \alpha$ ($0 \leq \alpha \leq 1$), which is the minimum frequent threshold specified by the user and $0 \leq \alpha \leq 1$, then X is a frequency pattern. It should be noted that OS-RFM-patterns is also the frequency patterns.

Property 3. According to the Definition 6, the frequency value is related to the occurrence times of the itemset. The inequality $F(X, h) \leq F(X)$, because the occurrence of this item set in h is less than in D .

Proof. Assuming two itemsets X and Y ($X \subseteq Y$), the occurrence of X is more than Y , if $F(X)$ does not satisfy the condition $F(X) \geq |D| \times \alpha$, then $F(Y)$ must also be lower than $|D| \times \alpha$. Similarly, for a period h , if $F(X, h) \geq |D| \times \alpha$, then $F(Y, h) \geq |D| \times \alpha$.

Strategy 3. (Frequency Pruning Strategy) Based on Property 3, if $F(X)$ is less than $|D| \times \alpha$, then there is no need to explore X and all its supersets in the search space. It is important to note that this pruning strategy is employed to eliminate 1-itemsets before mining n -itemset ($n \geq 2$). Besides, if $F(X, h)$ is less than $|D| \times \alpha$, it is unnecessary to explore X and its supersets in the search space too.

In this study, the global order $<$ on set I (e.g., the lexicographical order) is defined as an ascending order based on the TWU value (we will introduce TWU value. In Table 3, the order “ $E < B < D < F < A < C < G$ ” indicates that items are arranged in increasing order of their TWU values. The proposed algorithm sequentially adds one item at a time to an itemset, following the global order $<$ to generate high-level itemsets.

Table 3: Transaction-weighted utilization of items

Item	A	B	C	D	E	F	G
TWU (\$)	98	85	118	86	65	97	122

Definition 7. (Remaining utility) For a given period h , the remaining utility of X in h is defined as

$$re(X, h) = \sum_{T_c \in D \wedge pt(T_c) = h \wedge X \subseteq T_c} u(X, T_c) \quad (14)$$

Definition 8. (Remaining Utility Upper-Bound [9]) Given an itemset X and a period h , the sum of $iutil$ and $rutil$ values in the h -based period from the OS-RFM-list of X can be defined as

$$reu(X, h) = u(X, h) + re(X, h) \quad (15)$$

Property 4. Let X be an itemset and h be a period. $TWU(X, h) \geq reu(X, h) \geq u(X, h)$, that means $reu(X, h)$ is a tighter upper bound than $TWU(X, h)$ and the upper bound on $u(X, h)$. In addition, Considering another itemsets Y and $X \subset Y$, then $reu(X, h) \geq reu(Y, h)$. According to the definition of relative utility, it can find that $reu(X, h)/pto(h) \geq u(X, h)/pto(h)$, then $reu(X, h)/pto(h)$ is an upper bound on $ru(X, h)$.

Strategy 4. (Remaining Utility Upper-Bound Pruning Strategy). For an itemset X , if there is no period h such that $reu(X, h)/pto(X, h) < \beta$, $reu(Y, h)/pto(h)$ must also less than β (Property 4), then X and all its extensions are unnecessary to be explored.

Definition 9. (pOS-RFM pattern) Given a transaction database D and three user-specified thresholds α , β and γ . For an itemset X in D , X is called an pOS -RFM pattern if it satisfies all the following conditions: (1) $R(X) \geq \gamma$; (2) $F(X) \geq |D| \times \alpha$; and (3) $\exists h \in PE \wedge TWU(\{i\}, h)/pto(h) \geq \beta$. This means that X is a potential pOS -RFM-pattern.

Definition 10. (OS-RFM pattern). If an itemset X satisfies the three following constraints: $ru(X) \geq \beta$; (2) $R(X, h) \geq \gamma$; and (3) $F(X, h) \geq |D| \times \alpha$. Then, X is an OS-RFM pattern.

Definition 11. (OS-RFM-List) The OS-RFM-list of an itemset X in D consists of four rows, and each row is a quadruple ($period, tid, iutil, rutil$), where $period$ represents the identification of a period h that contains X , tid represents the identification of a transaction that contains X , the $iutil$ element

corresponds to the utility of X in T_{tid} (i.e., (X, T_{tid})), and the *rutil* element represents the remaining utility of X in T_{tid} (i.e., $re(X, T_{tid})$).

To create the OS-RFM-List structure for an l -itemset ($l \geq 2$) without scanning the database multiple times, the lists of two different $(l-1)$ -itemsets that have some same *tid* in the same period h are directly joined. Subsequently, in relation to the utility of the l -itemset, the term *iutil* represents the sum of the respective utilities of the two component itemsets, where *rutil* is based on the smaller one. In the Fig. 1 the OS-RFM-lists $\{A\}$ -List and $\{B\}$ -List have the same *tids* (i.e., T_2 and T_3). The mathematical formula is $iutil(\{A, F\}, T_2) = iutil(\{A\}, T_2) + iutil(\{F\}, T_2) = \18 . In addition, the *rutil* of $\{A, F\}$ is $\$19$. Therefore, the $\{A, F\}$ -List have been constructed as shown in Fig. 2.

$\{A\}$				$\{F\}$			
<i>period</i>	<i>tid</i>	<i>iutil</i>	<i>rutil</i>	<i>period</i>	<i>tid</i>	<i>iutil</i>	<i>rutil</i>
h_1	T_1	4	9	h_1	T_2	12	25
h_1	T_2	6	19	h_2	T_3	10	10
h_2	T_3	6	4	h_3	T_4	8	11

Figure 1: The OS-RFM-List of 1-itemsets

$\{A, F\}$			
<i>period</i>	<i>tid</i>	<i>iutil</i>	<i>rutil</i>
h_1	T_2	18	19
h_2	T_3	16	4

Figure 2: The OS-RFM-List of a 2-itemset

Problem Statement. Based on the definitions, the problem of the OS-RFM pattern mining task involves identifying a comprehensive set of OS-RFM patterns from a transaction database D , considering three minimal thresholds (i.e., recency, frequency, and monetary).

4 The Proposal Algorithm

In this section, the detail about the proposal OS-RFM algorithm and time complexiyy analysis are presented below.

4.1 Mining Algorithm

The main procedure (Algorithm 1) begins by requiring several key input parameters, including a transaction database, a minimum recency threshold, a delayed recency rate, a minimum frequency threshold, and a minimum utility threshold. The algorithm initially iterates through the entire database to calculate the TWU of each item i in h , the frequency and recency of i and $pto(h)$ of each period $h \in PE$ (lines 1–3). Subsequently, the algorithm filters out 1-itemsets unlikely to be OS-RFM patterns in advance (line 4). This is achieved by judging whether they meet the following conditions: (1) $R(X) \geq \gamma$; (2) $F(X) \geq |D| \times \alpha$; and (3) $\exists h \in PE$ and $TWU(\{i\}, h)/pto(h) \geq \beta$ according to Denifition 9. If they satisfy the above conditions, they belong to $pOS - RFM^*$, the set of potential OS-RFM patterns, collected at line 4. Otherwise, they will be filtered out. This step significantly reduces the number of itemsets that the algorithm needs to examine and explore, thereby lowering the time complexity. Simultaneously, the number of itemsets that the algorithm needs to store in memory is also reduced, thus decreasing the algorithm's space complexity. To further enhance the algorithm's efficiency, it sorts

the items in $pOS - RFM^*$ based on the TWU values of the 1-itemsets in ascending order at line 5. This sorting method has been proven to reduce the search space when using depth-first search in HUI mining [16]. After that, the algorithm performs a second database scan (line 6). During this scan, it rearranges the transactions according to the ascending order of TWU and constructs the OS-RFM-List for each item $i \subseteq pOS - RFM^*$ at line 6. Finally, in line 7, the algorithm calls a recursive method to analyze further and identify real OS-RFM patterns.

Algorithm 1: The OS-RFM algorithm

Input: D : a transaction database; γ : the minimal recency threshold; δ : the delay recency rate; α : the minimal frequency rate threshold, β : a minimal utility threshold.

Output: the set of OS-RFM patterns/

1. **foreach** transaction in D **do**
 2. Compute $TWU(\{i\}, h)$, $pto(h)$, $F(i)$, $R(i)$;
 3. **end**
 4. Collect $pOS - RFM^*$ set; // $pOS - RFM^*$ donates the set of potential OS-RFM patterns.
 5. Let \prec be the global TWU ascending order on $pOS - RFM^*$;
 6. Scan D to construct the OS-RFM-list of each item $i \in pOS - RFM^*$;
 7. **Search** (ϕ , ϕ , P^*_{List} , β , γ , α);
-

The **Search** procedure (Algorithm 2) takes seven parameters as input: (1) a prefix itemset P ; (2) the OS-RFM-Lists of a prefix itemset P_{List} ; (3) the set of OS-RFM-List of P 's 1-extensions P^*_{List} ; (4) the delay recency rate; and three minimal thresholds. For each of X in P^*_{List} , the algorithm first scans the OS-RFM-List of X to calculate $R(X, h)$, $F(X, h)$, $reu(X, h)$ for each period h and $ru(X)$ (line 3). Then, for lines 4 to 6, it checks whether X is an OS-RFM pattern based on the following three conditions: (1) $R(X, h) \geq \gamma$; (2) $F(X, h) \geq |D| \times \alpha$; and (3) $ru(X) \geq \beta$. If the conditions are true, X will be output as an OS-RFM pattern (line 5). Otherwise, it should be checked whether $pOS-RFM$ patterns exist (lines 7–19). If X does not satisfy these three conditions: (1) $R(X, h) \geq \gamma$; (2) $F(X, h) \geq |D| \times \alpha$; and (3) $reu(X, h)/pto(h) \geq \beta$ (line 7), X and all its extensions do not need to be explored (by Strategies 2, 3, and 4). Then, the algorithm will select the next itemset after X and repeat the checking steps above. Otherwise, X and all its extensions need to be explored and initialize X^*_{List} , the set of OS-RFM-List of X 's 1-extensions, as an empty set (line 8). Then, the algorithm will explore the extension of X (lines 9–16). The OS-RFM-List of XY is constructed by calling the **Construct** procedure (Algorithm 3) to join the OS-RFM-Lists of X_{List} and Y_{List} in line 10. The latter procedure is like that in FHM [20] and is thus not described in more detail. After constructing XY_{List} , the procedure determines whether XY and its extensions may be OS-RFM patterns according to the following conditions: (1) $R(XY) \geq \gamma$; (2) $F(X, h) \geq |D| \times \alpha$; and (3) $\exists h \in p_i(XY)$ and $TWU(XY, h)/pto(h) \geq \beta$ (lines 11–15). Specifically, if the conditions are proper, the extensions of XY should be further explored, and then the OS-RFM-List of XY will be added to X^*_{List} (line 13). Finally, X will be added to the prefix itemset P (line 17), and then the algorithm call the **Search** procedure in line 18. In addition, although the condition in line 12 may appear repetitive, considering that these three conditions impose a less stringent upper limit than that in line 7, a justified rationale exists for its inclusion. The purpose of this validation in line 12 is to preemptively prune the itemset XY before initiating the recursive call to **Search**. This preemptive pruning circumvents the necessity of comparing XY with potentially numerous other extensions of X within the subsequent recursive call, thereby enhancing the algorithm's efficiency.

Algorithm 2: The search procedure

Input: P : the prefix itemset, P_{List} : the OS-RFM-List of P ; P^*_{List} : the set of OS-RFM-List of P 's 1-extensions; γ : the minimal recency threshold; δ : the delay recency rate; α : the minimal frequency threshold; β : a minimal utility threshold.

Output: the set of OS-RFM patterns.

```

1. foreach  $X_{List}$  of  $X$  in  $P^*_{List}$  do
2.   foreach period where  $X$  appears do
3.     Compute  $R(X, h), F(X, h), reu(X, h), ru(X)$ ;
4.     if  $R(X, h) \geq \gamma$  and  $F(X, h) \geq |D| \times \alpha$  and  $ru \geq \beta$  then
5.       OS-RFM pattern  $\leftarrow X$ ;
6.     end
7.     if  $R(X, h) \geq \gamma$  and  $F(X, h) \geq |D| \times \alpha$  and  $reu(X, h)/pto(h) \geq \beta$  then
8.        $X^*_{List} \leftarrow \phi$ ; //  $X^*_{List}$  donates the set of OS-RFM-List of  $X$ 's 1-extensions.
9.       for each  $Y_{List}$  of  $Y$  after  $X$  in  $P^*_{List}$  do
10.         $XY_{List} \leftarrow$  Construct ( $h, P_{List}, X_{List}, Y_{List}$ );
11.        for each period where  $XY$  appears do
12.          if  $R(XY) \geq \gamma$  and  $F(XY) \geq |D| \times \alpha$  and  $\exists h \in p_i(XY)$  and
             $TWU(XY, h)/pto(h) \geq \beta$  then
13.             $X^*_{List} \leftarrow X^*_{List} \cup XY_{List}$ ;
14.          end
15.        end
16.      end
17.       $P = P \cup X$ ;
18.      Search ( $P, X_{List}, X^*_{List}, \beta, \gamma, \alpha$ );
19.    end
20.  end
21. end

```

Algorithm 3: The construct procedure

Input: PE : the set of periods; P_{List} : the OS-RFM-List of a prefix itemset P ; X_{List} : the OS-RFM List of itemset X ; Y_{List} : the OS-RFM-List of itemset Y .

Output: a. OS-RFM-List of super-itemset XY .

```

1. for each  $h \in PE$  do
2.   foreach element  $X_e \in X_{List}$  and  $X_h == h$  do
3.     if  $\exists Y_e \in Y_{List}$  and  $Y_h == h$  and  $X_e.tid == Y_e.tid$  then
4.       if  $R(X, h) \geq \gamma$  and  $F(X, h) \geq |D| \times \alpha$  and  $reu(X, h)/pto(h) \geq \beta$  then
5.         if  $P_{List} \neq \phi$  then
6.           foreach  $P_e \in P_{List}$  do
7.             if  $P_{e,h} == h \wedge P_{e,tid} == X_e.tid$  then
8.                $XY = (X_e.tid, X_{e,iutil} + Y_{e,iutil} - P_{e,iutil}, Y_{e,rutil})$ ;
9.             end
10.          end
11.        end
12.      else

```

(Continued)

Algorithm 3 (continued)

```

13.          $XY = (X_{e.tid}, X_{e.iutil} + Y_{e.iutil}, Y_{e.rutil});$ 
14.     end
15.      $XY_{List} \leftarrow \{h, XY_e\};$ 
16. end
17. end
18. end
19. return  $XY_{List}$ 

```

4.2 Time Complexity Analysis

This section presents a series of experiments designed to assess the effectiveness of the proposed algorithm. The algorithms are implemented in Java and tested on Windows. In this section, we will analyze the time complexity of Algorithm 2. In a time period when X appears in the 1-extensions itemset, item X must traverse that period to find Y ($X < Y$) that meets the criteria. In the worst-case scenario, XY appears in all transactions, and each transaction belongs to a distinct period. Because the items in the 1-extensions itemset are arranged in order, the time complexity of constructing an XY_{List} for item X and any subsequent item Y at a specific period is $O(|I| - 1)$, where $|I|$ represents the number of items in the transaction database. After constructing the OS-RFM list for the 2-itemset, Algorithm 2 performs **Search** self-recursion to explore the extended itemset for XY . The number of recursive calls depends on the length of the OS-RFM pattern. In the worst-case scenario, the length of the OS-RFM pattern is $|I|$, then the number of recursive calls is $(|I| - 1)$. Therefore, during a specific period, the time complexity of exploring the complete OS-RFM patterns containing X is $O((|I| - 1)^2)$, that is, $O(|I|^2)$. It should be noted that not all OS-RFM patterns have the maximum length, so the actual time complexity will be lower than $O(|I|^2)$. In all periods, the time complexity of exploring OS-RFM patterns containing X is $O(|D| * |I|^2)$. In the worst-case scenario, all items in the dataset belong to the 1-extensions itemset. Therefore, the complexity of searching for all OS-RFM patterns for all 1-extensions items in all periods is $O(|D| * |I|^3)$. It should be noted that not all 1-extensions items have $(|I| - 1)$ items after them, so the actual time complexity $O(|D| * |I|^3)$ of Algorithm 2 will decrease exponentially.

5 Experiments Conduction

We present a series of experiments designed to assess the effectiveness of the proposed algorithm. The algorithms are implemented in Java and tested on Windows 10 with 16 GB memory and Intel(R) Core (TM) i7-8550U CPU@1.80 GHz.

Experiments were performed using a set of standard datasets, namely Foodmart, Online Retail II, and Retail. These datasets are chosen because they all belong to the retail field and each transaction records the transaction time, which aligns with the OS-RFM algorithm's application scenario. In addition, these datasets have different characteristics such as size, source, and type. Conducting experiments using these datasets can more effectively demonstrate the algorithm's effectiveness. Table 4 summarizes their characteristics. Among them, the two real datasets, Foodmart, and Online Retail II, were used in the empirical evaluation. The Foodmart transaction dataset, provided by Microsoft SQL Server 2000, contains sales records from 24 stores from 1997 to 1998. It is a small and sparse dataset with short transactions. In contrast, Online Retail II is a large and dense dataset with moderately long transactions, comprising all transaction records of a non-brick-and-mortar online retailer registered

in the UK between 1 December, 2009 and 9 December 2011. As this paper does not consider negative profits and does not require customer identity information, we preprocessed the datasets by removing irrelevant information such as product descriptions, transactions with negative profits, customer IDs, and transaction countries. Another synthetic dataset called “Retail” is used to perform an ablation study. It is a small and sparse dataset with moderately short transactions sourced from customer transactions at an anonymous retail store in Belgium.

Table 4: Statistics of the experimental datasets

Dataset	#Trans	#Items	#AvgLen	#Density	#Type	#Source
Foodmart	4141	1559	4.42	0.28%	Sparse	Real
Online ratalail II	39533	4916	25.88	5.26%	Dense	Real
Retail	8162	16470	10.30	0.06%	Sparse	Synthetic

5.1 Overall Performance Evaluation

Table 5 provides a comparative analysis of the OS-RFM and RFM algorithms [17] regarding numbers of generated patterns, Precision, Recall, and F1-score, using various threshold parameter settings on the Foodmart dataset. Specially, we assume the delay rate is fixed at 0.1. Notably, both algorithms demonstrate a decrease in the number of results as threshold parameters increase. As previously content introduced, the RFM algorithm considers patterns relative to the entire year, but the OS-RFM algorithm considers each period individually. Consequently, OS-RFM will output more interesting patterns than that of RFM algorithms. As shown in Table 5, the high recall rate of OS-RFM, which is close to 100%, indicates its capability to identify the most significant patterns recognized by RFM. Although OS-RFM has relatively low precision, with the highest accuracy below 40%, this does not imply inferior performance; instead, it highlights its effectiveness in uncovering additional patterns characterized by high recency, frequency, and utility within specific periods. This is a challenging task for RFM pattern mining algorithms due to their low sensitivity to periods. However, the OS-RFM algorithm evaluates patterns per period, while the RFM algorithm evaluates them across entire periods, resulting in different outcomes. The OS-RFM algorithm partially replicates the effects of RFM, thus leading to a lower F1-score, averaging 28%.

Table 5: Performance comparison between OS-RFM and RFM algorithms on foodmart dataset

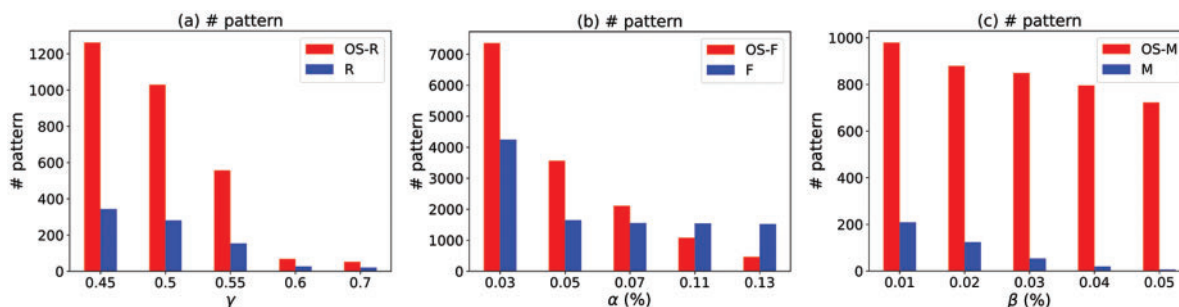
#Patterns of OS-RFM	#Patterns of RFM	α (%)	β (%)	γ (%)	Recall (%)	F1-score (%)	Precision (%)
2249	64	0.01	0.10	0.02	100.00	5.53	2.85
1018	32	0.01	0.12	0.02	100.00	6.10	3.14
908	29	0.01	0.12	0.03	100.00	6.19	3.19
221	64	0.03	0.10	0.02	100.00	45.00	29.00
173	32	0.03	0.12	0.02	100.00	31.22	18.50
162	29	0.03	0.12	0.03	100.00	30.37	17.90
156	64	0.05	0.10	0.02	93.75	54.55	38.46

(Continued)

Table 5 (continued)

#Patterns of OS-RFM	#Patterns of RFM	α (%)	β (%)	γ (%)	Recall (%)	F1-score (%)	Precision (%)
129	32	0.05	0.12	0.02	93.75	37.27	23.26
122	29	0.05	0.12	0.03	93.10	35.76	22.13

To better compare OS-RFM with RFM and highlight the effectiveness of the OS-RFM algorithm, we conducted a comparative experiment between the sub-algorithms of OS-RFM and RFM [17] regarding the number of mining results on the Foodmart dataset. OS-R, OS-F, and OS-M are the sub-algorithms of OS-RFM algorithm. Similarly, R, F, and M are the sub-algorithms of RFM algorithm. The OS-R algorithm only considers recency factors and aims to discover patterns that meet the minimum reception threshold within each period. Therefore, when implemented, the frequency and utility thresholds must be set to 0. OS-F, OS-M, R, F, and M work similarly. Additionally, OS-R compares with R, OS-F with F, and OS-M with M. As shown in Fig. 3, the number of mining results for all algorithms decreases as the threshold increases. In addition, although the OS-R algorithm has fewer mining results than the R algorithm in Fig. 3b when the threshold is set higher, it can be seen that it has a more significant number of mining results than the R algorithm in other cases. The number of results from OS-F and OS-M algorithms is much greater than from F and M algorithms, particularly in Fig. 3c.

**Figure 3:** Comparison of OS-RFM and RFM algorithms in mining result patterns

In summary, the proposed algorithm effectively explores patterns meeting recency, frequency, and utility thresholds, leveraging product on-shelf time considerations. Under reasonable parameters, the OS-RFM algorithm identifies more valuable patterns than RFM algorithm, such as seasonal items, providing actionable insights for businesses to optimize stocking strategies, ensure on-shelf availability, and enhance profitability while satisfying customer demand. We explore the performance of our algorithm and the compared algorithm to show the efficiency analysis. It should be noted that the accuracy, recall, and F-score results are all because the OS-RFM algorithm mines more patterns than the RFM algorithm.

Next, we explore the runtime and memory usage of our algorithm and the compared baseline algorithm [19] to show the efficiency analysis. The experiments are based on the Retail dataset, generating a dataset Retail_p4 for performance testing with four periods. As far as we know, current RFM mining algorithms cannot deal with on-shelf datasets, and we found OS-M [19] a suitable mining

algorithm for, so we use it as our benchmark algorithm and compared our algorithm with it. The OS-R and OS-F are all sub-algorithms of the OS-RFM algorithm. The OS-R algorithm only considers recency factors and aims to discover patterns that meet the minimum recency threshold within each period. Therefore, when implemented, the frequency and utility thresholds must be set to 0. OS-F and OS-M work similarly. The experimental parameter settings are shown in Table 6. Besides, “variable” denotes the threshold used for testing. In particular, the delay rate of tested algorithms is set as 0.1.

Table 6: The threshold setting details

Pattern	α (%)	β (%)	γ (%)
OS-RFM	0.1	0.045	Variable
OS-R	0	0	Variable
OS-RFM	Variable	0.045	0.1
OS-F	Variable	0	0
OS-RFM	0.1	Variable	0.1
OS-M	0	Variable	0

As shown in Figs. 4 and 5, the OS-RFM algorithm performs significantly more efficiently than the other algorithms in terms of runtime and memory consumption. In Fig. 4, as the threshold increases, the runtime of all algorithms decreases. Additionally, it can be observed that the OS-RFM algorithm has a notably shorter runtime, all of which are less than those of the OS-R, OS-F, and OS-M algorithms. Due to the OS-RFM list structure and some efficient pruning strategies we adopted, such as the strategies of recency, frequency, and TWU, the OS-RFM algorithm can strikingly reduce the runtime. In Fig. 5, although the OS-RFM algorithm consumes more memory than the OS-R algorithm under different thresholds, OS-RFM costs less than the other two algorithms. Furthermore, the memory consumption of the OS-RFM algorithm is relatively stable, while that of the OS-R, OS-F, and OS-M algorithms fluctuates significantly.

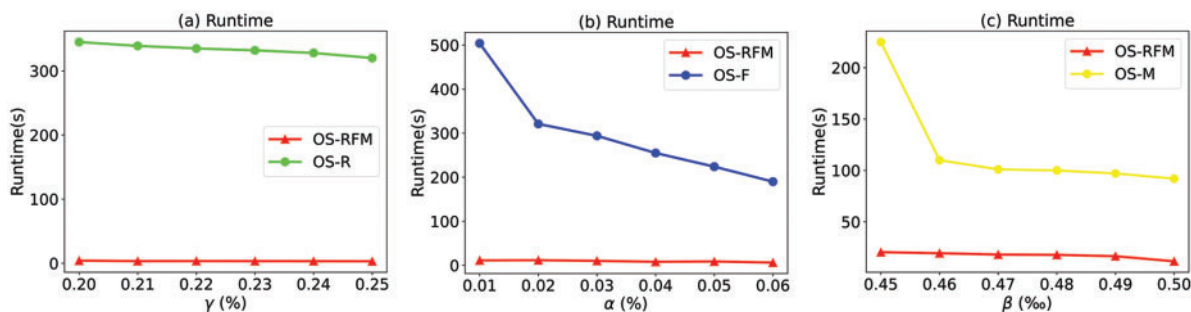


Figure 4: Runtime cost on Retail dataset (a) Runtime cost comparison between OS-RFM and OS-R algorithm; (b) Runtime cost comparison between OS-RFM and OS-F algorithm; (c) Runtime cost comparison between OS-RFM and OS-M algorithm

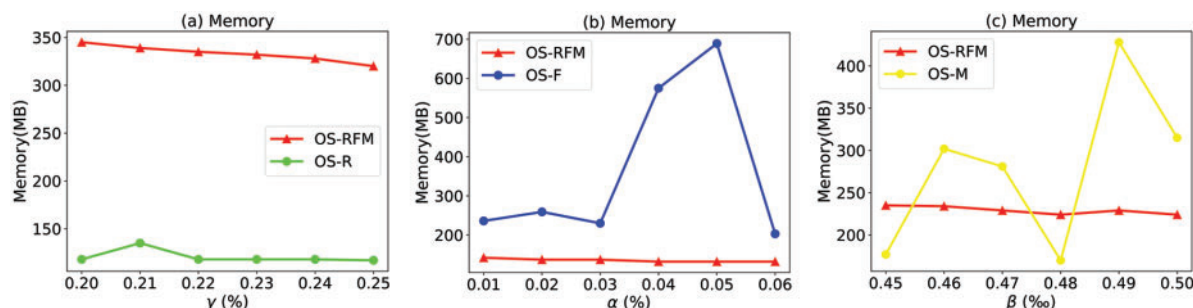


Figure 5: Memory usage on Retail dataset (a) Memory usage comparison between OS-RFM and OS-R algorithm; (b) Memory usage comparison between OS-RFM and OS-F algorithm; (c) Memory usage comparison between OS-RFM and OS-M algorithm

Thereafter, a summary table that compares OS-RFM with other algorithms in terms of the number of mining result patterns, runtime, and memory usage is given in Table 7. It can be seen that OS-RFM performs well in all comparisons.

Table 7: A summary comparison table between OS-RFM and other algorithms

	OS-RFM	RFM [17]	OS-R	OS-F	OS-M [19]
#Patterns	Large	Small	–	–	–
Runtime	Fast	–	Slow	Slow	Slow
Memory	Medium	–	Small	Large	Large

5.2 A Simple Case Study

In this section, we try to enable enterprises to gain further insights into the monthly changes in the recent activeness, popularity, and sales revenue of products throughout the year, allowing them to implement shelf management strategies flexibly. This experimental study employs the Online Retail II dataset, divided into 12 periods based on months. The recency, frequency, and utility values of products in each period (month) are calculated and plotted into graphs to analyze the sales situation of products over time. To further reveal the interesting and valuable insights about products reflected by the graphs using the OS-RFM algorithm, the beach bag and parasol products will be taken as an example to elaborate on the insights revealed by the OS-RFM algorithm.

Fig. 6a reveals the recency of beach bag and parasol products over a year, reflecting the product's recent activity level change throughout the year. Note that next we will give the study analysis of parasol products in the bracket. It can be seen from the figure that the recency value of the beach bag (parasol) is higher from June to August (in June and December), reaching its peak value of 1.85 (2.4) in June (December). However, the recency value of beach bag (parasol) is lower from September to November (January to February and also in November), with the lowest value being less than 0.02 (0.3). The difference between the maximum and minimum values is quite significant. This suggests that the product's sales fluctuate wildly, and the beach bag is most active and popular with customers from June to August in the summer. However, fewer people purchase it from September to November. Enterprises can utilize the recency information on the beach bag to restock in advance and boost the quantity of the beach bag on shelves in June, fulfilling customer demand and enhancing profitability.

By examining the trends in the recency value of items like beach bags and parasol over time, enterprises can identify the peak popularity periods and the less favorable ones. This enables them to restock beforehand, adjust the timing and quantity of products on shelves, ensure reasonable stock availability, prevent profit opportunities from being missed due to stock-outs during peak demand, and also avoid stocking up excessively during periods of low demand, which could lead to overcrowding and impact the sales of other more valuable items on the shelves.

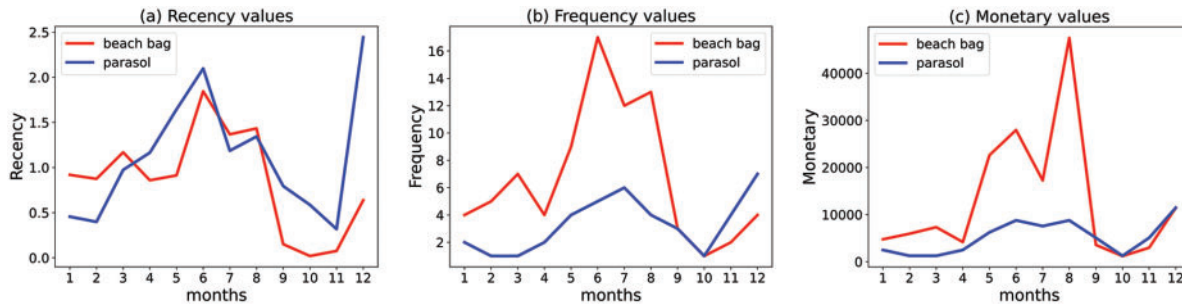


Figure 6: The variation in recency, frequency, and monetary values of beach bags and parasol throughout the year (a) The variation in recency values of beach bags and parasol throughout the year; (b) The variation in frequency values of beach bags and parasol throughout the year; (c) The variation in monetary values of beach bags and parasol throughout the year

Fig. 6b depicts the change in the frequency of beach bag and parasol products over a year, reflecting the changes in the product’s popularity over time. A higher frequency indicates that the product is more popular and has been purchased more frequently. As shown in the figure, the frequency value of the beach bag (parasol) has been increasing from April to June (March to July and October to December), with a sharp growth rate, reaching its maximum value in June (December). Besides, the frequency values of beach bag from September to December are relatively low. This indicates that the weather dramatically influences the beach bag. June to August is the peak sales period for the beach bag, as it is summer when the weather is scorching, which is suitable for people to purchase beach bags for the beach. However, from September to December, during the autumn and winter, when the weather is colder, fewer people purchase it, marking the off-season for the product. Retailers can make informed stocking decisions based on this information, increasing inventory in early May to add more beach bags to the shelves for customers to select. Additionally, they can appropriately reduce the number of items on shelves in September to prevent overcrowding and interfere with the sales of autumn and winter products. Retailers can maximize profits by accurately identifying the peak and off-peak products sales periods. Similarly, retailers can gain insight into their store’s monthly sales by analyzing the trends in similar products’ frequent value changes. This helps to pinpoint the peak sales periods and the off-peak periods. This information is crucial for retailers to scientifically and efficiently plan their inventory levels, ensuring adequate products are available on shelves for customers while preventing overstocking or understocking. It also enables retailers to use their limited funds effectively for procurement, avoiding the risks of overstocking and tying up funds, which can impact cash flow or lead to missed profit opportunities during the limited peak seasons.

Fig. 6c introduces the change in the utility of beach bag and parasol products throughout the year. It can be observed from the figure that the sales revenue of the product fluctuates greatly over time. The sales revenue of beach bag (parasol) shows an increasing trend from April to August (March to August and October to December), with a slight decrease in July (July) but still maintaining an

average level for the year, reaching its peak value of nearly \$48,000 (\$11,000) in August (December). Overall, the sales peak for the beach bag (parasol) is from May to August (June to August and in December), while the sales revenue in other months is relatively low, around \$4000 (\$3000). Enterprises can leverage the intricate sales patterns of the product to flexibly manage inventory and adjust the quantity of items displayed on shelves. For instance, from May onwards, gradually and appropriately boost both inventory and the number of items on shelves to satisfy customer demand, while making suitable reductions in September. This approach ensures the capture of valuable profit opportunities and maximizes the return on products displayed. Likewise, by analyzing trend graphs of comparable product sales over time, enterprises can scrutinize the actual profits generated by their products. If profits during a specific period are insufficient (making a minor contribution to the overall corporate profit). In this case, enterprises might consider removing the product or reducing its display quantity for that period to prevent the squandering of shelf space by low-value items. Concurrently, products with more significant profit potential can be introduced, or their display numbers on shelves can be increased, sustaining a reasonable stocking rate to maximize the return on displayed products.

As evident from the analysis of the beach bag and parasol product, considering the trend of the product's recency value, frequency value, and utility value over time is meaningful and effective. The new approach can provide enterprises with more valuable sales information, guiding them to adopt flexible and accurate strategies to adjust the shelving of products rather than temporarily stocking or shelving products due to unknown market fluctuations. This helps maintain a reasonable in-stock rate on shelves and brings more profits. Besides, it demonstrates that considering recency, frequency, or monetary loneliness will make it challenging to reflect the actual sales situation accurately. Moreover, the on-shelf pattern mining task only reflects the sale trend of products. This is not conducive to enterprises comprehensively grasping the profitability of a product. In conclusion, the novel algorithm in this paper can provide enterprises with more comprehensive and sufficient sales information, helping them maintain a reasonable in-stock rate on shelves and achieve greater profits.

6 Conclusion

In conclusion, the OS-RFM algorithm introduced in this paper considers quarterly patterns. It aims to discover OS-RFM patterns that exhibit high recency, frequency, and monetary value within specific periods. The effectiveness of this algorithm has been demonstrated through the utilization of real-life datasets and numerical and curve analyses. The numerical study comparing the accuracy, recall, and F1-score of OS-RFM and RFM using the real Foodmart dataset indicates that OS-RFM discovers most patterns identified by RFM and uncovers more meaningful patterns within specific periods. Additionally, experiments illustrate the variation of a product's recency, frequency, and monetary value over the year, providing valuable insights for businesses to maintain reasonable shelf availability. Performance comparison between OS-RFM and other algorithms based on the Retail dataset shows that OS-RFM exhibits higher efficiency. In conclusion, the proposed OS-RFM algorithm can assist businesses in maintaining reasonable shelf availability and achieving greater profitability. The algorithm in this paper is mainly based on historical data analysis and may not respond more quickly to the changing market environment and customer behavior in real time. Future research should explore how to make algorithms more dynamic and real-time to better adapt to rapidly changing market demands. In addition, multi-source data fusion, careful consideration of multiple factors, data privacy, and security are essential for further development in this field.

Acknowledgement: This research was partially supported by the Foundation of State Key Laboratory of Public Big Data.

Funding Statement: This research was partially supported by the Foundation of State Key Laboratory of Public Big Data (No. PBD2022-01).

Author Contributions: Cuiwei Peng: Software design, Writing—original draft. Jiahui Chen: Conceptualization, Supervision, Funding. Shicheng Wan: Methodology, Writing—reviewing. Guotao Xu: Data curation, Software Programming. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, Santiago de Chile, Chile, Sep. 1994, vol. 1215, pp. 487–499.
- [2] R. Chan, Q. Yang, and Y. D. Shen, “Mining high utility itemsets,” in *Proc. Third IEEE Int. Conf. Data Min.*, Melbourne, Florida, USA, 2003. doi: [10.1142/S0219622010004159](https://doi.org/10.1142/S0219622010004159).
- [3] T. Flamand, A. Ghoniem, and B. Maddah, “Store-wide shelf-space allocation with ripple effects driving traffic,” *Oper. Res.*, vol. 71, no. 4, pp. 1073–1092, 2023. doi: [10.1287/opre.2023.2437](https://doi.org/10.1287/opre.2023.2437).
- [4] G. P. Soysal and L. Krishnamurthi, “Demand dynamics in the seasonal goods industry: An empirical analysis,” *Mark. Sci.*, vol. 31, no. 2, pp. 293–316, 2012. doi: [10.1287/mksc.1110.0693](https://doi.org/10.1287/mksc.1110.0693).
- [5] G. C. Lan, T. P. Hong, and V. S. Tseng, “Discovery of high utility itemsets from on-shelf periods of products,” *Expert. Syst. Appl.*, vol. 38, no. 5, pp. 5851–5857, 2011. doi: [10.1016/j.eswa.2010.11.040](https://doi.org/10.1016/j.eswa.2010.11.040).
- [6] J. Fernie and D. B. Grant, “On-shelf availability: The case of a UK grocery retailer,” *Int. J. Logist. Manage.*, vol. 19, no. 3, pp. 293–308, 2008. doi: [10.1108/09574090810919170](https://doi.org/10.1108/09574090810919170).
- [7] M. H. Yang and W. C. Chen, “A study on shelf space allocation and management,” *Int. J. Prod. Econ.*, vol. 60, pp. 309–317, 1999. doi: [10.1016/S0925-5273\(98\)00134-0](https://doi.org/10.1016/S0925-5273(98)00134-0).
- [8] R. Satapathy, S. Prahlad, and V. Kaulgud, “Smart shelfie—Internet of shelves: For higher on-shelf availability,” in *2015 IEEE Region 10 Symp.*, Ahmedabad, India, May 2015, pp. 70–73.
- [9] J. Chen, X. Guo, W. Gan, C. M. Chen, W. Ding and G. Chen, “On-shelf utility mining from transaction database,” *Eng. Appl. Artif. Intell.*, vol. 107, 2022, Art. no. 104516. doi: [10.1016/j.engappai.2021.104516](https://doi.org/10.1016/j.engappai.2021.104516).
- [10] M. Tavakoli, M. Molavi, V. Masoumi, M. Mobini, S. Etemad and R. Rahmani, “Customer segmentation and strategy development based on user behavior analysis, RFM model and data mining techniques: A case study,” in *IEEE 15th Int. Conf. e-Bus. Eng.*, Xi’an, China, Oct. 2018, pp. 119–126.
- [11] R. Heldt, C. S. Silveira, and F. B. Luce, “Predicting customer value per product: From RFM to RFM/P,” *J. Bus. Res.*, vol. 127, pp. 444–453, 2021. doi: [10.1016/j.jbusres.2019.05.001](https://doi.org/10.1016/j.jbusres.2019.05.001).
- [12] S. E. Taşabat, T. Özçay, S. Sertbaş, and E. Akca, A new RFM model approach: RFMS. in *Industry 4.0 and the Digital Transformation of International Business*. Singapore: Springer, pp. 143–172, 2023.
- [13] D. R. Liu, C. H. Lai, and W. J. Lee, “A hybrid of sequential rules and collaborative filtering for product recommendation,” *Inf. Sci.*, vol. 179, no. 20, pp. 3505–3519, 2009. doi: [10.1016/j.ins.2009.06.004](https://doi.org/10.1016/j.ins.2009.06.004).
- [14] J. Liao, A. Jantan, Y. Ruan, and C. Zhou, “Multi-behavior RFM model based on improved SOM neural network algorithm for customer segmentation,” *IEEE Access*, vol. 10, pp. 122501–122512, 2022. doi: [10.1109/ACCESS.2022.3223361](https://doi.org/10.1109/ACCESS.2022.3223361).

- [15] A. H. L. Chen and S. Gunawan, “Enhancing retail transactions: A datadriven recommendation using modified RFM analysis and association rules mining,” *Appl. Sci.*, vol. 13, 2023, Art. no. 10057. doi: [10.3390/app131810057](https://doi.org/10.3390/app131810057).
- [16] Y. H. Hu and T. W. Yeh, “Discovering valuable frequent patterns based on RFM analysis without customer identification information,” *Knowl. Based Syst.*, vol. 61, pp. 76–88, 2014. doi: [10.1016/j.knosys.2014.02.009](https://doi.org/10.1016/j.knosys.2014.02.009).
- [17] S. Wan, J. Chen, Z. Qi, W. Gan, and L. Tang, “Fast RFM model for customer segmentation,” in *Companion Proc. Web Conf. 2022*, Lyon, France, 2022, pp. 965–972.
- [18] S. Wan, J. Deng, W. Gan, J. Chen, and S. Y. Philip, “Fast mining RFM patterns for behavioral analytics,” in *IEEE 9th Int. Conf. Data Sci. Adv. Anal.*, Shenzhen, China, 2022, pp. 1–10.
- [19] P. Fournier-Viger and S. Zida, “FOSHU: Faster on-shelf high utility itemset mining—with or without negative unit profit,” in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, Salamanca, Spain, 2015, pp. 857–864.
- [20] P. Fournier-Viger, C. W. Wu, S. Zida, and V. S. Tseng, “FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning,” in *Found. Intell. Syst.: 21st Int. Symp., ISMIS 2014*, Roskilde, Denmark, Jun. 25–27, 2014. pp. 83–92.