An Efficient Approach for Mining Reliable High Utility Patterns

MOHAMMED A. FOUAD, WEDAD HUSSEIN, SHERINE RADY, PHILIP S. YU, AND TAREK F. Gharib

Abstract: Utility mining is one of the most thriving research topics with a wide range of real-world applications. High utility pattern mining uses a utility function to extract all desired patterns that exceed a minimum utility threshold. However, a significant number of patterns will be generated if this threshold is set too low, which is an inherent limitation of these algorithms. This may cause the mining process to be inefficient as it would be difficult to analyze the patterns found. Furthermore, most of these patterns are unreliable and hard to be employed in making decisions. This paper proposed a novel problem of mining reliable high utility patterns by adapting the concept of reliability to mine a significant type of pattern called reliable high utility patterns. To address this issue, an efficient approach named RUPM (Reliable Utility-based Pattern Mining) is presented. RUPM introduces three novel measurements for estimating the reliability of utility-based patterns and proposes several strategies to efficiently handle reliable patterns with high utility values. Experimental results suggest that up to 99% of the patterns discovered by existing traditional high utility pattern mining algorithms were, in fact, unreliable. In contrast, the average reliability proportion in the resultant patterns obtained from the RUPM approach is at least 47.6% higher. Moreover, the proposed pruning strategies provide a reduction in both the runtime and memory usage.

Index Terms: Data mining, reliable high-utility itemset, utility mining, pruning strategy.

I. INTRODUCTION

Frequent Patterns Mining (FPM) is an analytical process that uses a co-occurrence measurement as a sole criterion to extract valuable patterns (i.e., itemset, sequence, rule, etc.) from a transaction database [1]. This process aims to discover valuable associations between items in transactions, which are employed in numerous real-world applications in different domains [2]. However, it disregards significant criteria required in many real-world applications such as profit, importance, risk, etc. These criteria can be called utility parameters, while the process of extracting the patterns that satisfy all or some utility measurement from transactional databases is named utility mining [3]. High Utility Pattern Mining (HUPM) is classified based on the desired pattern into high-utility itemsets, high-utility episodes, high-utility rules, and high-utility sequential patterns [2].

High Utility Itemset Mining (HUIM) refers to extracting all itemsets that exceed a predefined minimum utility threshold minUtil set by the user using a utility function [4], [5]. HUIM has been extensively used for process model extraction in different applications, such as recommendation systems, retail market analysis, and medical applications [2], [6]. Furthermore, the model’s performance has been improved by developing a set of pruning strategies and data structures to reduce the search space and significantly speed up the mining process [7], [8]. Recently, researchers have designed a set of multi-objective HUIM algorithms to retrieve the most relevant patterns, such as Frequent HUIM [9], Periodic HUIM [10], Correlated HUIM [11], Stable HUIM [12], and Closed HUIM [6]. While, the key objective of the proposed approach is finding itemsets that can repeat their significant high value produced from the utility function in unavailable data in the future, which is called Reliable HUIM.

The reliability expresses our expectation to get a similar value when repeating the utility function in unseen data. It is usually measured in terms of probability as a
function of time [13]. Reliability mining is essential for financial markets, predicting product demand, and the retail industry because it analyses consumption behaviors in buying the products and predicting interesting products in the future [14]. It is thus a critical research problem to design an algorithm for mining itemsets that have reliable behavior and generate a high profit. In light of this objective, current HUIM algorithms have three significant limitations. First, traditional HUIM algorithms use the utility measurement as the sole criterion to extract valuable patterns from the available data while disregarding interestingness objective measurements [9]. Second, in the event that the minUtil is set too low, it produces a tremendous number of patterns, and it can pose certain problems such as the difficulty of analysis as well as the time factor [3]. Third and most importantly, the majority of the patterns discovered by HUIM algorithms are often unreliable. Therefore, the user’s ability to depend on this result in making decisions is diminished. It is thus a challenge to produce an approach for mining those reliable utility-based patterns. As such, the major contributions of this paper can be outlined as follows:

- An effective approach called Reliable Utility-based Pattern Mining (RUPM) is introduced to mine itemsets with potentially high utility in unseen data.
- Design and development of an algorithm for mining reliable high-utility itemsets from transactional databases.
- Several pruning strategies are developed for searching reliable high utility items to reduce the search space and improve the performance.
- Three useful measurements for evaluating the reliability of the high utility itemsets named Trimmed Utility, Internal Consistency Coefficient, and Utility Consistency Coefficient are introduced.

The rest of this paper is organized as follows. In the next section, a brief overview of the related work of HUPM and the concept of reliability in pattern mining is discussed. The essential preliminaries and a problem statement are introduced in Section III. In Section IV, the proposed approach is explained. Section V presents experimental evaluations of the proposed approach. Finally, Section VI draws the conclusions.

II. RELATED WORK

This section briefs related works in HUPM algorithms and their subsequent development considering different forms of reliability, and, consequently, the broad concept of reliability in pattern mining is introduced.

Utility-based mining is proposed to determine all patterns that satisfy a minimum utility threshold. The classical HUPM uses utility measurement as the sole interestingness measure to evaluate the importance of itemsets, such as HUI-Miner [15], HUIM-IGA [4], EFIM [16], HIMU [17], and CBPM [18]. These algorithms are similar in input and output, with only differences in the data structures or strategies applied for reducing search space, memory consumption, and runtime. Overall, the efficiency and scalability of the HUPM algorithms are well studied. Nevertheless, the runtime and memory are not the sole factors that measure the quality of the HUPM approach. The effectiveness of these algorithms is also critical because it is related to the usefulness scale of discovered patterns. Reliability is one of the effectiveness measurements, which estimate the item’s importance over time.

The reliability in the pattern mining process has gained significant attention because it enhances the accuracy of prediction. Many researchers have proposed various algorithms and models to study the probabilities of an item’s importance happening by chance [19]. Moreover, several measurements from statistics probability and information retrieval have been suggested to evaluate reliability [20]. Overall several works have been presented and discussed to provide the reliability investigating periodicity, frequency, correlation, or stability models. The following paragraphs regard these interesting criteria to produce more reliable patterns.

Geng and Hamilton [20] described reliability as one of the main objective measurements of pattern interestingness, based on the probability of pattern’s occurrence. Also, a frequent pattern is deemed reliable if the pattern’s interestingness measurements occur in a high percentage or regularly appear in a sequence of events. Many measurements based on generality are introduced to measure the reliability of association rules such as IS measure, weighted relative accuracy, Recall, and Jaccard [20]. Prajapati et al. [21] identified association rules as consistent if they are as locally and globally frequent in large data. However, Shyur et al. [14] label a sequential pattern as reliable if the inter-arrival time of consecutive items probability satisfies the user-specified minimum threshold. Consequently, the SPP-Growth algorithm is proposed to find more predictable patterns by finding all stable periodic patterns in a transaction database with timestamps using a new periodicity measurement named Lability [22]. Additionally, the TSPIN algorithm introduces a concept of stability to find periodic patterns with stable periodic behavior [12]. Although periodic patterns carry interesting knowledge, considering the patterns’ utility values would reveal more practical information [23]. Fournier-Viger et al. [10] introduced the concept of periodic high-utility sequential pattern mining, where the utility measurement is integrated with periodicity measurements to avoid inconsistent itemsets having periods that vary widely in an algorithm called PHM.

A correlation factor such as the lift or added value is separately used in some studies to represent the reliability of the association rules [20]. Therefore, in order to extract more interesting patterns from classic HUPM algorithms while avoiding meaningless or non-discriminative patterns (which occur by chance), several works utilized both utility and correlation measures to present more reliable and profitable patterns, such as FCHM all-confidence [11], FCHM-bond [11], DHUP-Miner [24], and CoUPM [25], [26] based
on the following measurements: all confidence, bond, frequency affinity, and Kulc, respectively. At the same time, all-confidence is becoming a popular measure to find correlated patterns since it satisfies both the null-invariance and anti-monotonic properties [27]. Furthermore, correlation is consolidated with periodicity to make the correlated pattern mining more practicable in real-world applications [27], [28]. For example, the all-confidence measurement is combined with the periodicity measurement to sign the predictive behavior of the customers’ purchases using a new measurement called periodic-all-confidence [27].

On the other hand, frequency, correlation, stability, and periodicity deliver considerable benefits in different domains, but they do not cover the comprehensive notion of reliability. The domain of reliability consists of the degree of consistency and freedom from error [29], [30]. In other words, the reliable pattern should be: free from measurement error, including random error and systematic bias [30], internally consistent, which reflects the strong interrelatedness degree among the items, and reproducible, which demonstrates the limited total variance proportion in the measurements [29]. In summary, observing the related works, it can be concluded that most of the proposed pattern mining literature provided relevant measurements and employed them to extract more applicable patterns. The problem with these works is that they might be seen as incorrect or imprecise to define reliable patterns. It would be desirable to discover such reliable utility-based patterns that are essentially internally reliable, reliable across time, and free from error in practice.

III. PRELIMINARIES AND PROBLEM STATEMENT

This section provides some basic preliminaries of mining high utility itemsets from transaction databases, which are useful for describing our proposed algorithm and its analysis.

The term “transaction database” refers to a collection of transactions, \( D = \{T_1, T_2, \ldots, T_n\} \). Tid is a unique identifier associated with a single transaction. While each transaction is made up of a group of items I and each item \( i \in I \) has internal utility (e.g., quantity) is indicated as \( q(i, T_e) \) and external utility (e.g., profit) is indicated as \( p(i) \). Table 1 is used as a running example database containing ten transactions \( \{T_1, T_2 \ldots T_{10}\} \). The transaction \( T_2 \) consists of a group of items a, b, c, d, e, f and g with internal utility values 2, 2, 9, 2, 1, 3 and 5, respectively, while the external utility values of these items are 2, 3, 7, 5, 1, 1 and 6, respectively, are indicated in Table 2.

**Definition 1 (Utility of an Itemset):** The utility value of an itemset \( I \) in a transaction \( T_e \) is defined as \( u(I, T_e) = p(i) \times q(i, T_e) \). While \( u(X, T_e) \) indicates the utility value of an itemset \( X \) in a transaction \( T_e \), \( u(I, T_e) = \sum_{i \in I} u(i, T_e) \), and \( u(X) \) indicates the utility value of an itemset \( X \) in a database \( D \), \( u(X) = \sum_{T_e \in D} u(X, T_e) \), where \( q(X) \) is the collection of transactions including itemset \( X \). An itemset \( X \) is considered a high-utility itemset if minUtil less than \( u(X) \) [2]. For example, the utility value of the itemset \{a, b\} in a transaction \( T_7 \) is \( u(a, b) = u(a, T_7) + u(b, T_7) = 2 \times 4 + 3 \times 5 = 23 \). While, in the database, the utility value of the itemset \{d\} is \( u(d) = u(d, T_1) + \ldots + u(d, T_{10}) = 5 + 10 + 45 + 10 + 10 + 10 + 5 + 15 = 120 \).

**Definition 2 (Transaction Weighted Utilization):** The term “transaction utility (TU)” of a transaction refers to the summation of all utility values of the items involved in this transaction. While, the term “transaction weighted utilization (TWU)” of an itemset \( X \) refers to the summation of the transaction utility of all transactions comprising \( X \) [3], while relativeminUtil = minUtil/ \( \sum_{T_e \in D} TU(T_e) \).

**TABLE 1. A transaction database.**

| TID | Transaction |
|-----|-------------|
| \( T_1 \) | (a,3),(b,4),(c,8),(d,1) |
| \( T_2 \) | (a,2),(b,2),(c,9),(d,2),(e,1),(f,3),(g,5) |
| \( T_3 \) | (a,7),(b,6),(c,7),(d,1),(e,2) |
| \( T_4 \) | (a,4),(b,9),(c,2),(d,9) |
| \( T_5 \) | (a,5),(b,1),(c,8),(d,2),(f,4) |
| \( T_6 \) | (a,6),(b,7),(c,6),(d,1) |
| \( T_7 \) | (a,4),(b,5),(c,4),(d,2) |
| \( T_8 \) | (a,2),(b,4),(c,1),(d,2),(g,6),(f,1) |
| \( T_9 \) | (a,5),(b,6),(c,1),(d,1) |
| \( T_{10} \) | (a,3),(b,1),(c,2),(d,3) |

**TABLE 2. External utility table.**

| Item | a | b | c | d | e | f | g |
|------|---|---|---|---|---|---|---|
| Profit of Unit | 2 | 3 | 7 | 5 | 1 | 6 | 8 |

**Definition 3 (Window of Transactions):** Let there be two transactions having the following positions \( i \) and \( j \) such that \( i \leq j \) in a transactions database \( D \). The set of transactions from \( T_i \) to \( T_j \) is indicated as \( T_{i:j} = \{ T | i \leq p(T) \leq j \wedge T \in D \} \). A range of transactions \( T_{i:j} \) has a length (\( T_{i:j} \)) = \( j - i + 1 \). The size of window maxPer is defined by the user as a predefined threshold, while considering patterns that have not occurred at least once in consecutive transactions with range length maxPer is undesirable [10].

**Definition 4 (Reliable Utility-Based Pattern):** According to the definition of reliability discussed in section II, a Pattern X is called a reliable utility-based pattern if three conditions are met in X. First, \( u(X) \) is no less than minUtil, excluding high outlier values. Second, X is internally consistent or has a correlation degree no less than a predefined minimum positive correlation threshold given by the user. Third, \( u(X) \) is reproducible or has a stability degree no less than a predefined minimum stability threshold provided by the user.

**Definition 5 (Predictive Accuracy):** Predictive accuracy is a commonly used measurement for evaluating the quality of
the classification rule [20], which is defined as

\[
\text{Predictive Accuracy} = \frac{\text{True Positive}}{\text{True Positive} + \text{false Positive}}
\]

In this paper, three objective measurements are proposed to estimate the reliability, while the utility measurement is used as a classification rule to predict the desired pattern in unseen data. Consequently, True Positives refer to the model that produces the pattern with a high utility value from the training set, and the discovered pattern actually has a high utility in the test set. False Positives refer to the model that produces the pattern with a high utility value from the training set, but the discovered pattern does not have a high utility value in the test set.

Based on the reliability concept and utility theory, the importance of utility and reliability is taken into account to extract highly reliable and profitable patterns named Reliable Utility-based Patterns (RUPs).

IV. PROPOSED APPROACH FOR MINING RUPS

This section introduces the proposed approach for mining RUPs. The first subsection presents three effectiveness measurements to estimate the reliability of high utility itemsets. Secondly, efficient pruning strategies for searching reliable high utility items are presented, while the last subsection suggests an algorithm that employs the proposed measurements to find RUPs.

RUPM is a solution that many businesses can use to get more accurate results and forecast product situations in the future by applying multiple measurements on known past values. The novelty of this approach lies in proposing new pruning strategies for dramatically reducing the search space, then introduce some novel measurements for verifying the different aspects of reliability. Fig. 1 shows an architectural overview diagram containing the main modules of the RUPM approach. This diagram is divided into two parts: the first part includes the first six processes, which are responsible for generating potential reliable high utility patterns by implementing the proposed pruning strategies, whereas the second part includes the last three processes, which describe how to apply the proposed measurements to identify reliable utility-based itemsets.

A. MEASURING THE RELIABILITY OF HIGH-UTILITY ITEMSETS

The proposed RUPM approach merges a semantic measurement (utility) with an objective measurement (reliability) to discover a type of pattern called RUPs. A pattern is considered a RUP if it yields a high utility with an acceptable degree of reliability. However, there is a significant challenge that no single test is a perfect measurement of reliability [29]. Therefore, three objective measurements are introduced to produce RUPs:

- **Trimmed Utility Measurement** is proposed to avoid the outlier impact.
- **Internal Consistency Coefficient** aims at evaluating the internal consistency.
- **Utility Consistency Coefficient** is introduced to assess the consistency of utility over time.

1) HIGH UTILITY OUTLIER ADJUSTMENT

The first step of the proposed approach is called high utility outlier adjustment, and is built to detect and adjust the high outlier values from the time series of the pattern’s utility. The outlier values were considered unusual events or non-repetitive events such as natural disasters, special occasions, etc. Outliers may generate a random high utility value, which is difficult to replicate with a random sample of data. Interquartile Range Method IQR is a standard objective method in the statistical literature for identifying outliers by setting up a “fence” outside the first quartile and the third quartile [31]. Consequently, \( u(i, T_c) \) is considered an outlier if its value is higher than the third quartile or less than the first quartile. A trimmed utility of a pattern is determined by replacing high outlier values using linear interpolation.
of neighboring (non-outlier values) and then computing the Trimmed Utility.

**Definition 6 (Trimmed Utility of an Itemset):** The trimmed utility of an itemset \( X \) is indicated as \( tu_{\text{d}}(X) \), \( tu_{\text{d}}(X) = u(X) - \epsilon \), where \( \epsilon \) is the difference between the outlier value and linear interpolation of neighboring (non-outlier values). \( tu_{\text{d}} \) is a measurement to determine the less affected pattern by the presence of outliers compared to the expected utility value. For example, the trimmed utility of item \( \{d\} \) is \( tu_{\text{d}}(d) \). \( tu_{\text{d}}(\{d\}, D) = 120. \) Since \( u(d), T_d = 45 \) is considered as an outlier value, \( \epsilon = 45 - ((10 + 5)/2) = 37.5, tu_{\text{d}}(\{d\}) = 120 - 37.5 = 82.5 \), if \( \text{minUtil} = 100 \) and utility of \( \{d\} = 120 \), it cannot be considered a high utility pattern. Fig. 2 shows the utility distribution of item \( \{d\} \) and clarifies \( u(d), T_d = 45 \) as an outlier value according to interquartile range.

![FIGURE 2. Utility distribution and frequency of item d.](image)

2) **EVALUATE THE INTERNAL RELIABILITY**

The second step of our proposed approach to affirm reliability is evaluating internal consistency/reliability across items. Internal consistency is defined as the interrelatedness between the (sub)items [29]. Many correlation measurements have been mentioned in Section II to evaluate the correlation in HUPM algorithms. However, most of the current measurements fail to extract accurate patterns properly [32]. On the other hand, the Spearman-Brown prophecy formula (SB formula) has been documented in the statistical literature as more accurate in predicting reliability [13], [31]. As a result, the second step aims to use the average correlation coefficient of a correlation matrix to evaluate the internal reliability of the set of variables in the matrix based on the SB formula.

Using the same example introduced in section III, Fig. 3 shows a correlation matrix of itemset \( \{a, b, c\} \) and displays how each item correlates to all of the other items belonging to the itemset, and how each item correlates to itemset \( \{a, b, c\} \) using Pearson’s correlation coefficient. For example, if two variables of interest have \( n \) sample values, \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\), a Pearson’s product-moment correlation of the two variables, \( r \), is calculated according to Equation 1 [31].

\[
r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}
\]  

**FIGURE 3. Correlation matrix of itemset \{a, b, c\}.

The internal consistency coefficient of an itemset \( X \) is indicated as \( \text{icc}(X) \). To evaluate \( \text{icc}(X) \), assuming that \( \bar{r} \) is the mean of the correlation coefficient matrix of \( X \), and \( k \) is the number of items or itemsets in the correlation matrix of \( X \), \( \text{icc}(X) \) is calculated according to SB formula shown in Equation 2 [13].

\[
\text{icc}(X) = \frac{k \bar{r}}{1 + (k - 1) \bar{r}}
\]  

A pattern \( X \) is internally consistent if \( \text{minICC} \) is less than \( \text{icc}(X) \). Here, \( \text{minICC} \) is a minimum internal consistency threshold given by the user. In general, \( \text{icc}(X) \) can be interpreted as follows: less than 0.60 is unacceptable, a range of 0.60 to 0.65 is undesirable, a range of 0.65 to 0.70 is minimally acceptable, while greater than 0.70 is acceptable [13], [31]. For example, the internal consistency coefficient of itemset \{a, b, c\} is defined as \( \text{icc}\{a, b, c\} \) and can be calculated using equation (2). \( \bar{r} = (0.44 + 0.15 + 0.46 + 0.10 + 0.90 - 0.32)/6 = 0.29, k = 4 \) and \( \alpha = 4*0.29/(1 + (3*0.29)) = 0.62 \). If \( \text{minICC} = 0.7 \), itemset \{a, b, c\} cannot be considered as an internally consistent pattern.

Fig. 4 illustrates the difference between internally consistent and inconsistent patterns \{a, b\} and \{b, c\} by representing utility distribution of the two itemsets and their sub-items, \( \text{icc}\{a, b\} = 0.87 \) and \( \text{icc}\{b, c\} = 0.45 \).

3) **EVALUATING RELIABILITY OVER TIME**

Although the internal consistency coefficient efficiently discovers strongly correlated itemsets, it cannot be used as a direct measurement of reliability. The direct reliability measurements should include reliability over time measure, which addresses reproducibility or stability [29]. Therefore, the third step evaluates the reliability of utility over time as an indicator of reproducibility or stability on the utility distribution of patterns. The term “consistent pattern” refers to patterns that are locally frequent as well as globally frequent [21]. Similarly, the pattern is said to be a consistent utility-based pattern if the average utility value of a pattern in any period is similar to the global average utility.

**Definition 7 (Average Window Utility):** Let a window of transactions \( T_{i,j} \) includes itemset \( X \). The average window
utility of \( X \) in \( T_{ij} \) is computed from Equation (3).

\[
\text{awu}(X, T_{ij}) = \frac{\sum_{k=i}^{j} u(X, T_k)}{\maxPer}
\]

\( \text{Definition 8 (Global Average Utility):} \) Consider \( n \) number of transactions in a database \( D \). The global average utility of itemset \( X \) in \( D \) is computed from Equation (4).

\[
gau(X) = \frac{\sum_{i=1}^{n-\maxPer+1} \text{awu}(X, T_{i:s+i+\maxPer})}{n - \maxPer + 1}
\]

\( \text{Definition 9 (Utility Consistency Coefficient):} \) Consider a pattern \( X \) in a database \( D \). The utility consistency coefficient is denoted as \( ucc(X) \); the value of \( ucc(X) \) is calculated as the minimum average period utility of \( X \) divided by the global average utility of \( X \) in \( D \), \( ucc(X) \) can be calculated from Equation (5).

\[
ucc(X) = \frac{\min(\text{awu}(X, D))}{\text{gau}(X)}
\]

The \( ucc(X) \) value should be no less than the \( \minucc \) threshold to deem \( X \) as utility consistent, where \( \minucc \) is the minimum stability of utility threshold. For example, the utility consistency coefficient of itemset \{a, b\} in \( D \) is defined as \( ucc(\{a, b\}) \) and can be calculated using equation (5), when \( \maxlength = 3 \), minimum \( \text{awu}(\{a, b\}, D) \) = 17.67, \( \text{gau}(X) \) = 22.67 and \( ucc(X) \) = 0.78. If \( \minucc = 0.5 \), itemset \{a, b\} is regarded a consistent utility-based pattern. However, \( ucc(\{b, c\}) = 0.41 \) is considered an inconsistent utility-based pattern. As was shown in Fig. 4 which illustrates the utility distribution for two itemsets \{a, b\} as a reliable utility-based pattern and \{b, c\} as unreliable utility-based pattern. It is evident from the figure that itemset \{b, c\} has uncorrelated subsets and unstable behavior, and it is difficult to replicate the utility value with a new sample of data or consider it a reliable pattern.

\[\text{FIGURE 4. (a) Utility distribution of } \{a, b\} \text{ and its subsets and (b) utility distribution of } \{b, c\} \text{ and its subsets.}\]

**B. PRUNING STRATEGIES FOR RELIABLE HIGH UTILITY ITEMSETS MINING**

Exhaustive search can find all reliable itemsets with high utility values, but this method is excessively time-consuming because most databases have extensive items. For a database with \( n \) items, a complete search has to check \( 2^n - 1 \) itemsets. The following pruning strategies employ the reliability measurements for pruning the search space and return only those Reliable High Utility Itemsets RHUIs.

**Strategy 1 (Pruning Search Space Using the Trimmed TWU Property):**

According to TWU definition [33], if TWU(X) is less than \( \minUtil \), \( X \) and all its supersets are not high utility itemsets likewise are not RHUIs. However, the consideration of robustness is vital in reliability mining; thus, there is a need to exclude outlier values or unusual events. As a result, if Trimmed TWU(X) less than \( \minUtil \), the search space containing \( X \) and its supersets can be eliminated.

\[\text{Lemma 1: If Trimmed TWU(X) < } \minUtil, \text{ then } X' \text{ is not RHUIs forall } X' \supseteq X.\]

**Proof:** Based on the definition of trimmed utility, Trimmed TWU(X) = TWU(X) – \( \epsilon \). At the same time, the value of \( \epsilon \) is considered an uncertain value and should not be achieved again. This implies that in a new sample of data \( \epsilon \) equals zero and TWU(X) = Trimmed TWU(X). Consequently, if Trimmed TWU(X) < \( \minUtil \), then TWU(X) < \( \minUtil \) when reliability is considered. As a result of those concerns, in a new sample of data, \( X \) is not a high utility itemset, by extension, not RHUIs. Moreover, according to the reliability consideration and the transaction-weighted downward closure property [33], any superset of an item with a low trimmed TWU is an un-reliable utility-based itemset.

**Strategy 2 (Pruning Items With Significant Downtrend Behavior):**

Assume the general tendency of itemset \( X \) is a downtrend or has a negative slope, which means the utility value of itemset \( X \) in time \( t \) is less than the previous value in time \( t - 1 \). Thus, in some time, the utility value of itemset \( X \) will be zero,
and X and its supersets could never be reproducible and are not RHUIs. These unpromising patterns can be considered irrelevant patterns and pruned directly.

**Lemma 2:** If Tendency (X) is downtrend, then X’ is not RHUIs forall X’ $\geq X$.

**Proof:** Predicting the value of $u(X)$ in time $t$, using time-series forecasting methods, is based on the previous utility values of $X[u(X_{t-1}), u(X_{t-2}) \ldots u(X_1)]$. Building upon this concept if $u(X_t) < u(X_{t-1})$ then $u(X_{t+1}) < u(X_t)$. Consequently, if there is a negative slope between $u(X_{t-1})$ and $u(X_t)$, there is a utility value of $X$ equal to zero in time $n, u(X_n) = 0$. As a result, $awu(X, T_{n,n+maxPer}) = 0$ and $ucc(X) = 0$. According to the third condition of RUPs, X and its supersets are not RHUIs. Moreover, the failure of the utility function to repeat its significant value in any zone of the current dataset or in the next sample of data is conflicted with the concept of reliability.

**Strategy 3 (Pruning Items With Impermissible Maximum Periodicity):**

Making decisions based on a periodic pattern can be marked as less risky than based on unusual patterns [22]. The reason is that a periodic pattern will be more reproducible (reliable over time) as its periods will not vary significantly. Thus the non-periodic items and all their supersets can be discarded.

**Lemma 3:** If maximum periodicity (X) > maxPer, then X’ is not RHUIs forall X’ $\geq X$.

**Proof:** In the case of maximum periodicity (X) > maxPer, there is at minimum one window where itemset X has not occurred at least once in consecutive transactions with range length maxPer. Thus, in a window $T_{i,j}$, where X has not occurred, $\sum_{k=i}^{j} u(X, T_k) = 0$ and $awu(X, T_{i,j}) = 0$. Consequently, the utility consistency coefficient of X and its supersets is zero, so X and its supersets are not RHUIs.

**C. RUPM-BASED ALGORITHM**

In summary, observing the running example regarding reliability, it can be concluded that high $icc$ and high $ucc$ values can indicate reliable utility-based patterns. Therefore, the user can determine RUPs by assigning the minimum acceptable internal consistency coefficient and utility consistency coefficient. Consequently, an itemset X is considered RUP if $icc(X)$ no less than $minIcc$, $ucc(X)$ no less than $minUcc$, and $tu(X)$ no less than $minUtil$.

The following algorithms adopt the diagram in Fig. 1 using the utility-list structure [15] to store information about the utility of the items and adopt the RUPM approach to discover RUPs. Algorithm 1 describes the main procedures of finding reliable high utility itemsets, whereas applying the proposed reliability measurements to define RHUIs from the candidates is described in Algorithm 2. In order to produce the potential high utility itemsets without exploring the whole search space, the proposed algorithm adopts a set of pruning strategies. Through the first scan of the database, the frequency and TWU of items are accumulated. According to the pruning strategies, the unpromising items are eliminated, while the candidate items are arranged in TWU ascending order. However, initial utility-lists of 1-itemset are constructed by the second scan of the database, while k-itemset utility-lists are obtained by utility-lists intersection operation. The pseudo-code of these procedures is described in algorithm 1.

**Algorithm 1 RUP-Miner Algorithm**

**input:** D: a transaction database, $minIcc$, $minUcc$, $minUtil$, and $maxPer$: the user defined thresholds

**output:** reliable high-utility itemsets

1. Scan D to obtain TWU((i)), Support ((i)) for each item \( i \in I \);
2. For each item i in which Trimmed TWU(i) $\geq minUtil$ and maximum periodicity(i) $\leq maxPer$ and tendency(i) is not downtrend;
3. Order the items in I* according to the ascending of TWU;
4. Scan D to obtain the utility-list of item i;
5. For each itemset $P \in$ Supersets of item i do
   a. $P$ is a k-itemset constructed based on item i.
   b. If (TWU(P) $\geq minUtil$ and maximum periodicity(P) $< maxPer$) then
      i. Call Search-RUPs($P$, $minUtil$, $minIcc$, $minUcc$);
      ii. If endorsement $\geq minUtil$ and $minIcc$, then $P$ is RUP;
5. End for

To clarify our methodology, the pseudo-code of the main RUPM procedures is described in Algorithm 2, the input of which is the potential reliable high utility itemsets. The first step is to accumulate $\varepsilon$, which is the subtraction between the outlier value and linear interpolation of neighboring non-outlier values, to filter the itemsets with Trimmed Utility less than $minUtil$. The second step calculates correlation coefficients for all subsets of candidate itemsets to recognize the itemsets having an Internal Consistency Coefficient no less than $minIcc$. The itemsets with a Utility Consistency Coefficient no less than $minUcc$ are subsequently identified in the third set, using Average Window Utility and Global Average Utility. Finally, the algorithm returns the RUPs.

The complexity analysis of the proposed algorithms is as follows. Suppose the database has $m$ transactions, and $I$ is the number of different items. The existing HUPM algorithms first scan the database to find the candidate high utility items $n$, which takes a time of $O (m \times \log(I))$. Generating a set of k-itemsets results in $2^n − 1$ candidates. The worst-case for computing the utility value of each itemset needs $O (2^n)$, which is related to the number of candidate 1-itemsets. Hence, the worst-case scenario expects the time complexity as $O (m \times \log(I) + 2^n − 1)$. Therefore, the search space can be vast, and the runtime and memory used may increase exponentially. As the number of the candidate 1-itemsets $n$ decreases, the time to compute decreases. Consequently, the execution time will be further reduced due to the effect of pruning strategies reducing the value of $n$ by the number
Utility mining approaches. Table 3 contains information about these datasets, including the name of the dataset, the number of transactions, the total number of items, the average number of items in a transaction, the density of dataset, and the utility type. All datasets are obtained from the SPMF Repository [34]. All the algorithms are implemented in Java and conducted on a computer with 64 bit and Intel Core i7-6820HQ CPU @ 2.70GHz processor running Windows 10 and 16 GB of memory.

To estimate the reliability of HUIM algorithms, the dataset is split into 70% for training and 30% for testing. All algorithms are run using the training set to recognize high utility itemsets, and results are used to predict high utility itemsets in the testing set. The high utility itemsets obtained from the testing set were used to estimate the accuracy of the algorithms based on the relevant predictable itemsets using the predictive accuracy metric, which is defined in details in Section III.

A. EFFECTIVENESS EVALUATION

In this section, the predictive accuracy metrics are accumulated and compared to test the following algorithms: Periodic HUIM (PHM) [10], Correlated HUIM (FCHM all-confidence) [11], and the proposed Reliable HUIM(RUPM) in terms of prediction accuracy through a series of experiments.

Table 3. Characteristics of the experimental datasets.

| Dataset     | No. Transactions | No. Items | Avg. Len (A) | Density (A/I)% | Type    |
|-------------|------------------|-----------|--------------|----------------|---------|
| Retail      | 88162            | 16470     | 10.3         | 0.0625         | synthetic |
| Ecommerce   | 14975            | 3468      | 11.71        | 0.338          | Real    |
| Mushroom    | 8124             | 119       | 23           | 19.327         | synthetic |
| Fruithut    | 181970           | 1265      | 3.58         | 0.283          | Real    |

In these experiments, parameter values should be guaranteed to produce a wide range of candidate itemsets by setting a value of minUtil and a maxPer according to the particular situation of each dataset. For all experiments, the problem of mining correlated high utility itemsets CoHUIs can be solved using the FCHM all-confidence algorithm [11] by simply setting the minimum all-confidence threshold \( \beta = 0.5 \). At the same time, the problem of mining periodic high utility itemsets PHUIs can be solved using the PHM algorithm [10]. The minPer and minAvg thresholds are set to 0, While the maxPer and maxAvg thresholds are set to 1000, 3000, 500, and 5000 for dataset Retail, Ecommerce, Mushroom, and Fruithut, respectively. While relative minUtil ranging to present a wide range of possible scenarios. Fig. 5 shows the results of these experiments and displays the predictive accuracy values of an algorithm on a dataset across multiple values of minUtil, with \( minIcc = 0.5 \) and \( minUcc = 0.3 \).
Three observations emerge from the process of analyzing the results of experiments: First of all, the proposed RUPM approach outperforms the correlated and periodic HUIM approaches by incorporating all aspects of the reliability to discover such patterns that are essentially internally reliable, reliable across time, and free from error.

Second, datasets with a relatively small number of transactions, such as Ecommerce and Mushroom, suggest that the FCHM all-confidence algorithm outperforms the periodic-based algorithm in terms of reliability. The reason is that, in the case of a small number of transactions, the reliability challenge is the interrelatedness between the sub-items. Thus, the reliability should include correlation, internal consistency, or symmetric measure.

Third, datasets with a relatively large number of transactions, such as Retail and Fruithut, suggest that the PHM algorithm outperforms the correlation-based algorithm in terms of reliability. The reason is that in the case of a large number of transactions, the reliability should include stability, consistency over time, or symmetric measure to determine the ability of the pattern to be profitable for a long time.

Table 4 shows the number of reported desirable high utility itemsets HUIs, PHUIs, CoHUIs, and RHUIs in the four datasets described above. There are two observations: First, the number of RHUIs can be significantly lower than the number of HUIs, PHUIs, or CoHUIs. For example, on Fruithut, 10833 HUIs are found for minUtil = 0.03%. However, only 63 itemsets that have a high chance of reproducibility produced by the RUPM algorithm, including 62 itemsets, were, in fact, reproduced high utility value in the testing set. However, the resultant HUIs, including 3516 itemsets, were non-reproducible in the testing set, which may be misleading for dictions makers. A similar pattern reduction is observed on the PHM algorithm, which finds 901, including 66 itemsets, were non-reproducible in the testing set. In comparison, The FCHM all-confidence algorithm finds 286, including 55 itemsets, were non-reproducible in the testing set. Second, the experiment verifies the impact of change in the minUtil on reliability proportion in the resultant patterns obtained from a traditional HUIM algorithm. The results show that decreasing minUtil reduces reliability proportion by increasing the search space in this case. This demonstrates that the RUPM algorithm effectively filters unreliable patterns and that many unreliable patterns are found in real-life datasets.

The following experiment demonstrates the usefulness of the internal consistency coefficient measurement.
Applying the RUPM based algorithm using \textit{minUcc} is set to 0, which means that the stability factor has not been considered. Relative \textit{minUtil} is set to 0.001\%, 0.005\%, 0.05\%, and 0.0005\%, for dataset Retail, Ecommerce, Mushroom, and Fruithut, respectively. These figures guarantee to generate a wide range of candidate itemsets. Fig. 6 (a) shows a positive correlation between internal consistency coefficient value and predictive accuracy, while the \textit{minIcc} value varies from 0.00 to 0.80. The average correlation coefficient for the experimental evaluation is 86\%.

On the other hand, \textit{minIcc} is set to 0, disregarding the correlation factor, and the RUPM pruning strategies have been deactivated for verifying the usefulness of the utility consistency coefficient measurement. The predictive accuracy was measured using the RUPM based algorithm. Fig. 6 (b) shows a strong positive correlation between the utility consistency coefficient and predictive accuracy at the \textit{minUcc} value varies from 0.0 to 0.80. The average correlation coefficient for the experimental evaluation = 90\%.

**B. PERFORMANCE EVALUATION**

Other experiments have been carried out to investigate the efficiency of the proposed algorithm by comparing the running time and peak memory consumption of each algorithm with different utility threshold values. A clear trend was observed in the following experiment to compare the execution times while decreasing the \textit{minUtil} threshold. Fig. 7 compares the execution times of all comparative algorithms. The results show that the RUPM algorithm performance outperforms other algorithms in processing time except for the Fruithut dataset, where the vised nodes and execution time is high compared to other algorithms. This issue is due to the Fruithut dataset, which contains transactions from grocery stores, and the average number of items in a transaction is very small. This dataset includes items that may be products or categories [34]. However, the categories cannot express the consumption behaviors of its items; it is just a general view of the market. Consequently, the proposed pruning strategies cannot discover the unreproducible features such as irregular, significant downtrend, or non-periodical behavior of items to decrease the number of the candidate 1-itemsets. At the same time, it provides a significant competitive advantage and outperforms the comparative algorithms in terms of prediction accuracy.

**TABLE 5. Comparison of maximum memory usage (MB).**

| Dataset   | PHM  | FCHM all-confidence | RUPM |
|-----------|------|----------------------|------|
| Retail    | 417  | 346                  | 324  |
| Ecommerce | 2489 | 2482                 | 2302 |
| Mushroom  | 1992 | 1678                 | 2555 |
| Fruithut  | 967  | 1112                 | 1162 |

Memory consumption was also compared to investigate the efficiency of the proposed pruning strategies for each dataset. Memory measurements were performed using the Java API. According to Table 5, the results show that RUPM can efficiently reduce memory consumption, especially in Retail and Ecommerce. However, the FCHM all-confidence algorithm outperforms the others for datasets with a small number of transactions and a high average number of items in a transaction such as Mushroom. Because the all-confidence measurement is anti-monotonic and all single items with high utility are directly output since their all-confidence is 1. In comparison, the PHM algorithm outperforms the others for datasets with a high number of transactions, such as Fruithut. Because a considerable amount of non-periodic items is pruned directly depending on how MaxPer threshold values are set. Finally, although RUPM produces the maximum predictive accuracy rates, it can efficiently reduce memory consumption, especially in datasets with a large number of items such as Retail and Ecommerce. This is because the RUPM pruning strategies ignore worthless items, which have a poor chance of reproducibility. At the same time, it tries to compute the maximum number of nodes to avoid losing satisfied patterns.
C. RESULTS AND DISCUSSION

Firstly, the reliability value depends on how the user adjusts the minUtil threshold. For example, in the dataset Ecommerce, when minUtil = 0.01% and maxPer = 3000, the reliability proportion of the traditional HUIM method = 0.43%. Thus it is significantly difficult to rely on resultant patterns in decision making. However, using the RUPM approach with parameters minIcc = 0.7 and minUcc = 0.5, the reliability proportion, in this case, will be 90%.

Secondly, experimental results show that RUPM pruning strategies can discard the search space consisting of unpromising patterns and save up to 99.9%, 98.5%, 98.2%, and 99.8% of search space for dataset Retail, Ecommerce, Mushroom, and Fruithut, respectively.

Finally, experimental results show that the average reliability proportion in the resultant patterns obtained from the RUPM algorithm is up to 19.45%, 42.64%, 44.02%, and 15.39% more reproducible than comparative algorithms in datasets Retail, Ecommerce, Mushroom, and Fruithut, respectively. As a result, experiments demonstrated that the resultant patterns obtained from the proposed approach are more reliable than using the comparative approaches in mining RUPs, and the proposed measurements provide appropriate criteria to estimate the reliability in utility-based patterns.

VI. CONCLUSION

High utility pattern mining is an emerging research area that aims to discover patterns with high utility. However, the vast majority of the discovered patterns are unreliable. The process of mining reliable patterns with high utility values has significant limitations. The existing pattern mining literature is very imprecise as the reliability domain consists of free from error or bias, internal reliability, and reliability across time. This paper proposed a novel approach of mining reliable patterns with high utility values by adapting the concept of reliability to mine a significant type of pattern called reliable high utility patterns, an approach is named RUPM (Reliable Utility-based Pattern Mining). The proposed approach introduces new pruning strategies for dramatically reducing the search space and providing three measurements to avoid drawbacks of lack of reliability in traditional utility-based pattern mining approaches. An experimental study on several databases has been performed to demonstrate the effectiveness and efficiency of the RUPM algorithm. The results reveal that the patterns discovered by different utility-based pattern mining algorithms are often unreliable, especially with dense datasets. However, extensive experiments on sparse and dense, synthetic and real-world data suggest that the proposed measurements provide appropriate criteria to estimate the reliability in utility-based patterns. Therefore, the decision-makers can extract extremely reliable high utility itemsets by adjusting the minimum internal consistency and minimum utility consistency thresholds. Furthermore, the experimental results show that the discovered patterns obtained from the RUPM algorithm are up to 41.89% and 47.60% more reproducible than the periodic high utility itemset mining algorithm and correlated high utility itemset mining algorithm.
mining algorithm, respectively. In addition, the proposed pruning strategies can discard the search space consisting of unpromising patterns and save at last 98.2% of search space for experimental datasets.

REFERENCES

[1] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and P. S. Yu, “HUOPM: High-utility occupancy pattern mining,” IEEE Trans. Cybern., vol. 50, no. 3, pp. 1195–1208, Mar. 2020.

[2] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, V. S. Tseng, and P. S. Yu, “A survey of utility-oriented pattern mining,” IEEE Trans. Knowl. Data Eng., vol. 33, no. 4, pp. 1306–1327, Apr. 2021.

[3] P. Fournier-Viger, J. Chun-Wei Lin, T. Tuong-Chi, and R. Nkambou, “A survey of high utility itemset mining,” in High-Utility Pattern Mining, Studies in Big Data, vol. 51, Cham, Switzerland: Springer, 2019.

[4] Q. Zhang, W. Fang, J. Sun, and Q. Wang, “Improved genetic algorithm for high-utility itemset mining,” IEEE Access, vol. 7, pp. 176797–176813, 2019.

[5] W. Gan, J. C.-W. Lin, J. Zhang, H. Yin, P. Fournier-Viger, H.-C. Chao, and P. S. Yu, “Utility mining across multi-dimensional sequences,” 2019, arXiv:1902.09582.

[6] T. Wei, B. Wang, Y. Zhang, K. Hu, Y. Yao, and H. Liu, “FCHUM: Efficient frequent and closed high-utility itemsets mining,” IEEE Access, vol. 8, pp. 109928–109930, 2020, doi: 10.1109/ACCESS.2020.3001975.

[7] C. Zhang, M. Han, R. Sun, S. Du, and M. Shen, “A survey of key technologies for high utility patterns mining,” IEEE Access, vol. 8, pp. 55798–55814, 2020, doi: 10.1109/ACCESS.2020.3081962.

[8] S. Krishnamoorthy, “Pruning strategies for mining high utility itemsets,” Expert Syst. Appl., vol. 42, no. 5, pp. 2371–2381, 2015, doi: 10.1016/j.eswa.2014.11.001.

[9] L. Zhang, G. Fu, F. Cheng, J. Qiu, and Y. Su, “A multi-objective evolutionary algorithm for mining frequent and high utility itemsets,” Appl. Soft Comput., vol. 62, pp. 974–986, Jan. 2018.

[10] P. Fournier-Viger, Y. Wu, D. Dinh, W. Song, and J. C. Lin, “Discovering periodic high utility itemsets in a discrete sequence,” in Periodic Pattern Mining. Singapore: Springer, 2021, pp. 133–151.

[11] P. Fournier-Viger, Y. Zhang, J. C.-W. Lin, D.-T. Dinh, and H. B. Le, “Mining correlated high-utility itemsets using various measures,” Log. J. IGPL, vol. 28, no. 1, pp. 19–32, Jan. 2020.

[12] P. Fournier-Viger, P. Yang, W. Yang, J. C.-W. Lin, U. Yun, and R. Uday, “TSPIN: Mining top-k stable periodic patterns,” Appl. Intell., vol. 104, pp. 1–22, Feb. 2021.

[13] H. C. W. de Vet, L. B. Mokkink, D. G. Mosmuller, and C. B. Terwee, “Spearman–Brown prophecy formula and Cronbach’s alpha: Different faces of reliability and opportunities for new applications,” J. Clin. Epidemiol., vol. 85, pp. 45–49, May 2017.

[14] H.-J. Shyur, C. Jou, and K. Chang, “A data mining approach to discovering reliable sequential patterns,” J. Syst. Softw., vol. 86, no. 8, pp. 2196–2203, Aug. 2013.

[15] M. Liu and J. Qiu, “Mining high utility itemsets without candidate generation,” in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage., pp. 55–64, 2012.

[16] S. Zida, P. Fournier-Viger, J.-C. W. Lin, C.-W. Wu, and V. S. Tseng, “EEMF: A fast and memory efficient algorithm for high-utility itemset mining,” Knowl. Inf. Syst., vol. 51, no. 5, pp. 595–625, 2017.

[17] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and P. S. Yu, “Beyond frequency: Utility mining with varied item-specific minimum utility,” ACM Trans. Internet Technol., vol. 21, no. 1, pp. 1–32, Feb. 2021.

[18] Y. Djenouri, J. C.-W. Lin, K. Nørvang, H. Ramampiaro, and P. S. Yu, “Exploring decomposition for solving pattern mining problems,” ACM Trans. Manage. Inf. Syst., vol. 12, no. 2, pp. 1–36, Jun. 2021, doi: 10.1145/3439771.

[19] U. Ahmed, J. C.-W. Lin, G. Srivastava, R. Yasin, and Y. Djenouri, “An evolutionary model to mine high expected utility patterns from uncertain databases,” IEEE Trans. Emerg. Topics Comput. Intell., vol. 5, no. 1, pp. 19–28, Feb. 2021, doi: 10.1109/ETTCI.2020.3000224.

[20] L. Geng and H. J. Hamilton, “Interestness measures for data mining: A survey,” ACM Comput. Surv., vol. 48, no. 3, p. 3, 2006.

[21] D. J. Prajapati, S. Garg, and N. C. Chauhan, “Interesting association rule mining with consistent and inconsistent rule detection from big sales data in distributed environment,” Future Comput. Inform. J., vol. 2, no. 1, pp. 19–30, Jun. 2017.
SHERINE RADY received the B.Sc. degree in electrical engineering (computer and systems) and the M.Sc. degree in computer and information sciences from Ain Shams University, Cairo, Egypt, and the Ph.D. degree from the University of Mannheim, Germany. She is an Associate Professor at the Faculty of Computer and Information Sciences, Ain Shams University. Her research interests include artificial intelligence, data mining, and data science.

PHILIP S. YU received the B.S. degree in EE from the National Taiwan University and the M.S. and Ph.D. degrees in EE from Stanford University. He is a Distinguished Professor of computer science with the University of Illinois Chicago, Chicago, and also currently the Wexler Chair of information technology. He was a recipient of ACM SIGKDD 2016 Innovation Award for his influential research and scientific contributions on mining, fusion, and anonymization of big data and the IEEE Computer Society’s 2013 Technical Achievement Award for pioneering and fundamentally innovative contributions to the scalable indexing, querying, searching, mining, and anonymization of big data. He was the Editor-in-Chief of ACM Transactions on Knowledge Discovery From Data, from 2011 to 2017, and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, from 2001 to 2004.

TAREK F. Gharib received the Ph.D. degree in theoretical physics from Ain Shams University, Cairo, Egypt, in 1994. He is currently a Full Professor of information systems at Ain Shams University, where he is also the Head of the Information Systems Department, Faculty of Computer and Information Sciences. He has over 70 publications. His research interests include developing novel data mining and machine learning techniques, especially for applications in text mining, social networks, bioinformatics, and data analytics. He received the National Science Foundation Award, in 2001.

* * *

VOLUME 10, 2022