Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

UTILITY LIST-BASED MINING AND RECURRENT NEURAL NETWORK FOR UTILITY ITEMSET MINING BASED TRANSACTIONAL DATA

*1G.N. SOWJANYA, 2M. BABU REDDY
1Research Scholar, Krishna University Machilipatnam,
2Assistant Professor, Krishna University Machilipatnam
E-mail:1sowjanyagn2024@outlook.com,2babureddy21221@outlook.com
*Corresponding Author: G.N. Sowjanya

ABSTRACT

The study of Frequent Itemset Mining (FIM) and High-Utility Itemset Mining (HUIM) is essential because it provides practical insights to improve company outcomes and explains customer behavior. To detect high-utility itemsets, including those with negative utilities, HUIM algorithms have been efficiently developed. Utility Itemset Mining (UIM) has changed into an important area of research aimed at identifying high-utility patterns in transactional databases, where item utility is measured using criteria such as profit, frequency, or relevance. This research introduces a new framework that associations Utility List-Based Mining (ULBM) with Recurrent Neural Networks (RNNs) for effective and scalable utility itemset mining in transactional data. The utility list organization stores significant utility data for each itemset, enabling direct and effective calculation of utility values without the need for candidate generation. This considerably reduces recollection usage and processing costs. To enhance prediction abilities and capture temporal relationships amongst transactions, we use an RNN, specifically a Long Short-Term Memory (LSTM) network, to detect sequential patterns in transactional data. This hybrid technique leverages the utility list's efficiency in performing utility computations while the RNN learns temporal correlations between itemsets and predicts high-utility itemsets across transaction sequences. Our experiments validate that the proposed method, RNN-UBLM, not only outperforms existing utility mining algorithms in terms of accuracy and effectiveness but also excels at capturing dynamic, time-varying utility patterns. This approach is particularly well-suited for requests in e-commerce, retail analytics, and other sectors where utility-based decision-making is critical.

Keywords: *Frequent Itemset Mining, High-Utility Itemset Mining, Utility List-Based Mining, Transactional data, Recurrent Neural Networks.*

1. INTRODUCTION

Itemset mining is a crucial aspect of data mining, aiming to determine itemsets in large datasets that are valuable to users based on support and other criteria. Recently, HUIM has gained significant attention. In this work, we extend the FIM problem [1] by employing a utility function, rather than the traditional support function, to identify interesting itemsets. The input consists of a quantitative database (QDB) containing valuable quantitative data and a minimum utility threshold. The output is high utility itemsets (HUIs). HUIM has numerous beneficial applications, such as analyzing biological data, cross-marketing, detecting patterns in online store user behavior, and identifying consumer groups associated with items that generate substantial profits [2].

Therefore, while it is helpful to detect HUIs, developing efficient algorithms for this task is difficult. Another issue is that utility functions usually fail to satisfy the anti-monotonic (AM) criterion, in addition to the incredibly huge search space of itemsets that must be analyzed. Consequently, an itemset's utility might be the same as, greater than, or less than that of its subsets. In frequent itemset mining (FIM), on the other hand, the frequency function fulfills the AM requirement, allowing the application of very efficient search space reduction techniques. By establishing upper limits (UBs) on utility functions that satisfy the AM condition, researchers have narrowed the search field in HUIM. Low utility itemsets (LUIs) were then filtered out using pruning techniques based on these UBs. Furthermore, a number of modifications and unique data structures have been developed to improve the effectiveness of HUIM algorithm. These developments and structures make it easier to powerfully store candidate patterns and intermediate data required for HUI mining, while also minimalizing the computational cost of developing utility purposes and UBs.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HUIM methods use a minU threshold, as opposed to FIM algorithms, which search over all itemsets where an item occurs at least once. This value is selected by the user to be entered into the HUIM algorithms [4]. Itemsets classified as high-utility are those whose utilities, as established by the database, are extra than or equal to minU. A quantity is linked to each instance of an item, and each item's unit value is quantified by an extra number. An item may appear more than once in a single transaction. Compounding an item's cost and advantage yields its utility [5]. HUIM is considerably more challenging than FIM since utility does not meet the downward closure criteria. The supersets of a particular set may be more, equal to, or less useful than the set itself. To decrease the search area of an itemset's supersets, HUIM employments a novel anti-monotonic upper limit [6].

This research on Utility List-Based Mining and RNNs presents a novel approach to utility itemset mining since transactional data. Utility List-Based Mining focuses on identifying itemsets that deliver high utility values, such as profit or customer satisfaction, slightly than simply their frequency of occurrence. This method efficiently handles large datasets by maintaining a utility list that ranks itemsets based on their total utility, enabling quicker access to valuable information. By combining these two methods, the proposed framework not only improves the accuracy of utility itemset identification but also leverages the sequential nature of transactions to uncover patterns and trends over time. This synergy allows businesses to make informed decisions by focusing on the most advantageous itemsets, ultimately leading to optimized advertising strategies and enhanced customer experiences.

1.1 The main contribution

- Develop a new background that combines RNN-UBLM for operative and scalable utility itemset mining in transactional data.
- To enhance prediction capabilities and represent temporal connections between transactions, we employ an RNN-LSTM network, to detect sequential patterns in the data.
- This hybrid approach leverages the efficiency of the utility list in performing utility computations, while the RNN learns temporal correlations between itemsets and predicts high-utility itemsets across transaction sequences.

2. RELATED WORK

Sra, P., and Chand, S. [7] suggested a reinduction-based technique for obtaining high utility itemsets from incremental databases. To improve mining efficiency, they proposed a threshold-raising technique and introduced the concept of reinduction counters. In terms of memory use and execution time, our proposed strategy outperforms previous approaches. They also changed the window size to evaluate its scalability. Its suggested SUM approach addresses numerous difficulties with existing high utility itemset mining strategies for incremental datasets.

The notion of residual usefulness is used by Sra, P., and Chand, S. [8] to develop residue-maps and master-maps, two new data structures. R-Miner utilizes these two data formats to provide a novel approach to mining high-utility objects. To enhance the mining process, additional trimming criteria are investigated, and the cumulative efficacy value is employed as a threshold. To compare R-Miner's performance with alternative list-based techniques, a number of experiments are performed on both synthetic and actual datasets.

Oguz, D. [9] suggested that HUIM-WOIU identifies itemsets with predetermined profits, independent of the capacity sold in each transaction. More logically, it selects fewer groups of high-utility items and reduces execution time. According to this study, the external utility value significantly influences the selection of high-utility itemsets. Our future plans include generating controlled synthetic datasets to explore the effect of external efficacy values on HUIM. Additionally, they will examine opposing perspectives and investigate the effect of internal utility values on HUIM.

Tung, N. T. et al. [10] developed techniques to reduce the number of potential applications while improving database scanning performance. They also demonstrate that various types of database items can be pruned using the proposed upper bounds. Based on the solutions provided, they devise a specific approach to efficiently address this issue. The procedures are evaluated against state-of-the-art HUIM algorithms on a range of datasets with different sizes and characteristics, including volatile negative earnings, to demonstrate their effectiveness.

This issue was addressed by Vu, V. V. et al. [11], and since then, researchers have proposed several useful methods. This research uses a new technique called FTKHUIM (Fast top-k HUI

Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Mining) to analyze the top-k HUIs. The study also shows that a new threshold-raising method based on transaction utility (TU) termed RTU significantly improves the speed of top-k HUIM. Additionally, a global background for storing utility values and improving these methods using threshold-raising approaches is providing in the study.

The P-EFIM technique, proposed by Cheng, Z., et al. [12], breaks the problem into several distinct subtasks. It achieves load balancing by distributing these subtasks evenly among all nodes using the recommended S-style distribution approach. Additionally, P-EFIM accelerates the Reduce step by applying the EFIM approach to mine each subtask dataset. Experimental results on eight datasets show that P-EFIM outclasses the PHUI-Growth HUIM approach, which is based on the Hadoop framework.

Boukhalat, A. et al. [14] proposed evolutionary techniques as a solution to the HUI mining challenge. This research investigates the application of evolutionary approaches in HUI mining. Furthermore, they present a classification of evolutionary technique-based HUI mining that employs single, multi-objective, and hybrid optimization strategies. This article compares various approaches and explores the theoretical aspects of a wide range of nature-inspired algorithms.

2.2 Research Gap

- Transactional data often exhibits sparsity and imbalance, where certain items or itemsets are significantly more frequent or contribute disproportionately to the utility. Addressing this issue and designing algorithms that can handle such uneven distributions of data without bias is an under-explored area
- In many real-world productions, transactional data container be incomplete or uncertain. However, traditional UIM approaches often assume perfect, fully-observed datasets. A gap exists in utility itemset mining techniques that can account for and operate on uncertain or imprecise data.
- As data privacy concerns continue to rise, developing UIM techniques that respect user privacy, especially when mining sensitive transactional data, remains an important but underexplored area. The integration of privacy-preserving mechanisms, such as differential privacy, into utility itemset mining models is still in its infancy.

3. PROPOSED SYSTEM

This proposed model presents a novel framework for effective and scalable utility itemset mining in transactional data by combining Utility List-Based Mining (ULBM) with Recurrent Neural Networks (RNNs). The utility list construction stores crucial utility statistics for each itemset, enabling the direct and efficient calculation of utility values without the need for candidate generation. This approach significantly reduces memory consumption and processing costs. To enhance prediction capabilities and capture temporal relationships between transactions, we employ an RNN, specifically a LSTM network, to discover sequential patterns in transactional data. This hybrid technique leverages the utility list's efficiency for utility computations, while the RNN learns temporal correlations between itemsets and predicts high utility itemsets across transaction sequences. Figure 1 illustrates the block diagram of the RNN-UBLM model.

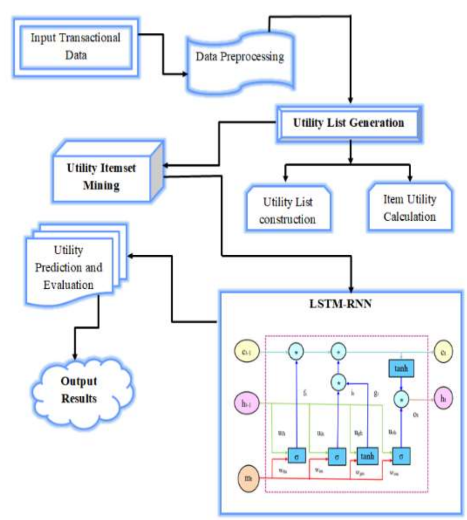

Figure 1: Overall block diagram of RNN-UBLM

3.1 Dataset

The dataset for this competition is a relational set of files that monitors the orders placed by customers over time. The competition's objective is to forecast the next order a user will place. More than 200,000 Instacart users have placed over 3 million

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

grocery orders, and the data has been anonymized [14]. Each user is provided with between four and one hundred orders, along with the sequence in which the products were purchased. Additionally, the dataset includes the relative time between orders, as well as the day, week, and time of day the orders were placed. For further information, please refer to the blog post published alongside the press release.

3.2 Data Preprocessing

In this competition, the dataset consists of multiple relational files describing customer orders, including product sequences, order times, and user information. The objective is to forecast which products will be included in a user's subsequent order.

To begin, the following pre-processing steps can be applied:

Combine the relational files (orders, products, users, etc.) into a unified dataset using the user and order IDs as keys.

$$D_{merged} = D_{orders} \times D_{poducts} \times D_{users} \tag{1}$$

Where × denotes a join operation.

Extract temporal features from the order time (week, hour of day, days since prior order). Create lag features to represent the time gaps and frequency between consecutive orders for each user.

Example lag feature:

$$Lag_n = t_n - t_{n-1} \tag{2}$$

Where t_n and t_{n-1} are the times of the current and previous orders, respectively.

For each user, identify the products in the current order as the target to predict for the next order.

3.3 Utility List Generation

Utility List Generation is a critical step in utility itemset mining, consisting of two main components: Item Utility Calculation and Utility List Construction. In the Item Utility Calculation phase, the utility of each item is considered depending on the profit, cost, or customer preference, which gives an overall evaluation of the importance of each item within the transactional dataset. Following this computation, the Utility List Construction phase groups these computed utility values into a well-defined list where each item is linked to its utility measures. This utility list is useful for the next mining procedure as it helps in the discovery of high utility itemsets that can be useful in decision making and preparation. When arranged in order of relevance, the items help industries to get a better understanding of clienteles and their needs in relation to products.

3.4 Utility Itemset Mining

UIM is disturbed with the discovery of high utility itemsets in transactional databases using algorithms that are specifically developed for this perseverance. The first step involves the mining algorithm in which methods such as UPGrowth (Utility Pattern Growth) or UCM (Utility Constraint Mining) are used to efficiently search for item sets that have utility greater than a quantified utility value. These algorithms are exactly designed to search through the utility list, trying to minimize the computational overhead when mining, but at the same time, trying to cover all possible itemsets. Then, the Frequent Itemset Extraction phase includes the identification of itemsets that are not only useful but also persistent within the dataset. This kind of extraction process is made conceivable by applying predefined utility thresholds to identify patterns that are both valuable and relevant to the business, which in turn can help businesses make better conclusions about marketing, managing their inventories, and other aspects of their operations. By so doing, organizations can quintessence on the high utility itemsets that will be of high benefit to the achievement of their goals.

3.5 Utility List-Based Mining (ULBM)

ULBM is an effective technique for HUIM that utilizes utility lists to decrease the search space and increase mining performance. ULBM is built on two fundamental components: utility lists and pruning techniques [15].

1. Utility List Construction:

For each itemset, a utility list is constructed, which contains detailed information for each transaction in which the itemset appears. Each utility list entry for a transaction contains three key values:

- T(i, t): The transaction utility, which represents the utility contribution of item i in transaction t.
- R(i, t): The remaining utility, which signifies the utility of items that appear after i in the lexicographic order in transaction t.
- S(i, t): The support utility, representing the sum of the utilities of the itemset in all transactions it performs in.

Thus, the utility list U(i) for an item i can be expressed as:

$$U(i) = \{(T(i,t), R(i,t), S(i,t)) \mid t \in Transaction containing i\} \tag{3}$$

Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. Pruning Strategies:

ULBM employs pruning strategies that use utility list-based bounds to eliminate unpromising itemsets early in the mining process.

Utility Upper Bound (UUB): The maximum utility that an itemset can achieve in any extension can be computed by summing the transaction utility T(i,t) and the remaining utility R(i,t) for each transaction. For an itemset I:

$$U(I) = \sum_{t \in T(I)} [T(I,t) + R(I,t)] \tag{4}$$

If the upper bound of an itemset is less than the user-defined minimum utility threshold δ\deltaδ, the itemset can be safely pruned:

$$U(I) < \delta \Rightarrow I \, is \, unpromi \sin g \tag{5}$$

Exact Utility Calculation: For promising itemsets, their actual utility is computed by summing the transaction utility T(i,t) across all relevant transactions:

$$U(I) \sum_{t \in T(I)} T(I,t) \tag{6}$$

If U(I)≥δ, the itemset I is considered a HUI.

3. Efficiency in ULBM:

By maintaining and using utility lists, ULBM drastically reduces the need to scan the database repeatedly, as the utility information is directly accessible from the lists. This approach not only accelerates the mining process but also minimizes memory usage since unpromising itemsets are pruned early.

3.6 RNNs – LSTM

This is the most fundamental form of a RNN. The internal structure of a Vanilla RNN is shown in Figure 2. Equations (7) and (8) mathematically illustrate its temporal sequencing behavior. While the beginnings from the earlier time step, ht-1, are linked to the recurring weight matrix Uhh , the input feature vector mt is linked to the non-recurrent weight matrix Whh . The appropriate weight matrix, Wnh , is determined by the output vector, nt. F1 is a nonlinear purpose, like the sigmoid or tanh nonlinearity, while f2 may represent a nonlinearity in the softmax output [16].

$$h_t = f1 \left[u_{hh} h_{t-1} + w_{hm} m_t + b_h \right] \tag{7}$$

$$n_t = f2 \left[w_{yh} h_t + b_n \right] \tag{8}$$

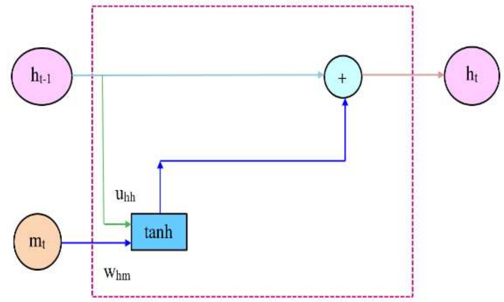

Figure 2: Structure of a Vanilla RNN

The recurrent cell in an LSTM-RNN is transformed into the applicant recollection cell for Equation (12), the forget gate for Equation (10), the reset gate for Equation (9), and the output gate for Equation (13). The inner structure of an LSTM cell is exposed in Figure 3.

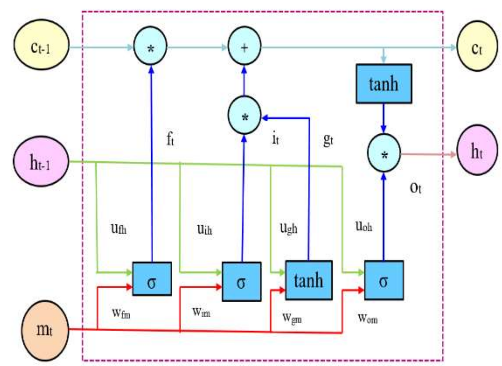

Figure 3: Structure of LSTM-RNN

$$input\ gate,\ i_t = \sigma \left[u_{ih}^l \, \mathrm{h}_{t-1} + w_{im}^l m_t^l + diag\left[p_i^l \right] c_{t-1} + b_i \right] \tag{9}$$

$$\text{forget } gate,\ \mathrm{f}_t = \sigma \left[u_{fh}^l \, \mathrm{h}_{t-1}^l + w_{fm}^l m_t^l + diag\left[p_f^l \right] c_{t-1} + b_f \right] \tag{10}$$

$$g_t = tanh \left[u_c^l h_{t-1}^l + w_{cm}^l \, \mathrm{m}_t^l + b_c^l \right] \tag{11}$$

$$memory\ cell,\ c_t = f_t^l \odot c_{t-1}^l + i_t^l \odot g_t^l \tag{12}$$

$$output\ gate,\ o_t = \sigma \left[u_{oh}^l \, \mathrm{h}_{t-1}^l + w_{om}^l \mathrm{m}_t^l + diag\left[p_o^l \right] c_{t-1} + b_o^l \right] \tag{13}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$recurrent\ hidden\ state, h_t = o_t^l \odot \tanh c_t^l \quad (14)$$

$$where, m_t^l = \begin{cases} h_t^{l-1}(hidden\ temporal\ features), & l>1 \\ input\ features, & l=1 \end{cases} \quad (15)$$

$$and\ final\ output\ at\ the\ last\ layer\ will\ be \quad r_t = \phi[w_{nt} h_t + b_n] \quad (16)$$

ϕ denotes an output activation function, such as softmax activation, whereas ⊙ signifies element-wise multiplication. To obtain the recurrent weight matrix (uhh) of an LSTM, the four-gate recurring weight matrices [uih,uoh,ufh,uch]T are stacked vertically. To create the non-recurrent weight matrix (whm), the four-gate non-recurrent weight matrices [wim,wom,wfm,wcm]T are stacked vertically.

4. RESULT

4.1 Experimental setup

The Windows 10 computer used for the studies had an Intel Core i5TM processor with 8 GB of RAM and a clock speed of 3.1 GHz.

4.2 Performance Metrics

Accuracy: Accuracy describes how frequently a model's predictions are right. It is determined by dividing the number of right guesses by the total number of forecasts made.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

Precision: In classification issues, precision is a metric that measures the accuracy of affirmative predictions. It indicates the proportion of actual positive outcomes out of all positive predictions provided by the model.

$$Precision = \frac{TP}{TP+FP} \quad (18)$$

Scalability: Scalability refers to a system's capacity to holder improved workloads or expand to meet demand. A scalable system can grow in size or capacity while maintaining or improving performance in business and computing.

$$S(n) = \frac{T(1)}{T(n)} \quad (19)$$

- $S(n)$ = scalability (speedup) for nnn units (e.g., processors, resources)
- $T(1)$ = time to complete a task with 1 unit
- $T(n)$ = time to complete the task with nnn units

The closer $S(n)$ is to n, the better the scalability.

Memory usage: Memory usage refers to the amount of computer memory (RAM) used by programs or processes while they are running. It measures how much space in the memory is occupied by active tasks and data, affecting system performance. High memory usage can slow down the system if RAM is insufficient, leading to the use of slower storage, such as swap.

$$Memory\ Usage = \sum_{i=1}^{n} Memory\ used\ by\ process\ i \quad (20)$$

Where n is the number of processes running.

F-Score: The F-score, commonly known as the F1-score, measures a model's accuracy in binary classification while balancing precision and recall. It is especially effective when the class distribution is skewed.

$$F-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (21)$$

4.3 Comparative Methods

The current systems utilized in this study include TNT-HUI (Tail-Node Tree-based High-Utility Itemset) [20], FEACP (Fast and Efficient Algorithm for Cross-level High-Utility Pattern Mining) [18], H-FHAUI (Hiding Frequent High Average-Utility Itemsets) [19], and FOTH (Fast sOrted iTemset search) [17].

1. Precision Analysis

Table 1: Precision Analysis for proposed RNN-UBLM model

Number of Transactions	FOTH	FEACP	H-FHAUI	TNT-HUI	RNN-UBLM
100	67.34	61.19	62.24	72.29	93.45
200	71.34	74.45	78.74	79.19	94.57
300	87.56	79.91	88.81	88.34	96.19

400	91.56	69.91	90.45	81.23	95.67
500	88.61	70.33	66.43	89.91	96.76

Table1 and Figure 4 presents the precision analysis of the RNN-UBLM method compared to four existing systems: FOTH, FEACP, H-FHAUI, and TNT-HUI, for different number of transactions. The RNN-UBLM is again found to be superior to the other methods in terms of precision percentages across all the transactions. For instance, at 100 transactions, RNN-UBLM has 93.45%, and the highest among others is TNT-HUI which has 72.29%. When the total number of transactions is 500, the precision of RNN-UBLM is 96.76% which shows that the suggested model is more accurate and effective for large datasets.

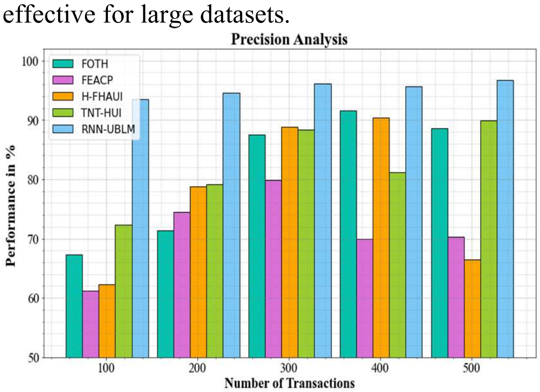

Figure 4: Precision Analysis for proposed RNN-UBLM model

2. Accuracy Analysis

Table 2: Accuracy Analysis for proposed RNN-UBLM model

Number of Transactions	FOTH	FEACP	H-FHAUI	TNT-HUI	RNN-UBLM
100	73.45	86.34	77.12	89.91	93.55
200	79.91	81.77	91.56	90.91	94.67
300	84.56	71.18	83.45	79.33	96.19
400	80.91	70.56	84.56	79.23	97.77
500	82.34	85.56	87.91	92.56	97.91

Table 2 and Figure 5 presents the accuracy analysis of the RNN-UBLM method compared to four existing systems: The simulation was conducted with FOTH, FEACP, H-FHAUI, and TNT-HUI across different numbers of transactions. The results indicate that the suggested RNN-UBLM technique is superior to the other approaches in terms of accuracy rates. For 100 transactions, the proposed RNN-UBLM achieves 93.55%, while the highest competitor, namely TNT-HUI, achieves 89.91%. RNN-UBLM also increases with the number of transactions and reaches the maximum of 97.91% for 500 transactions, which indicates better presentation of the suggested systems in comparison with the existing ones in all cases.

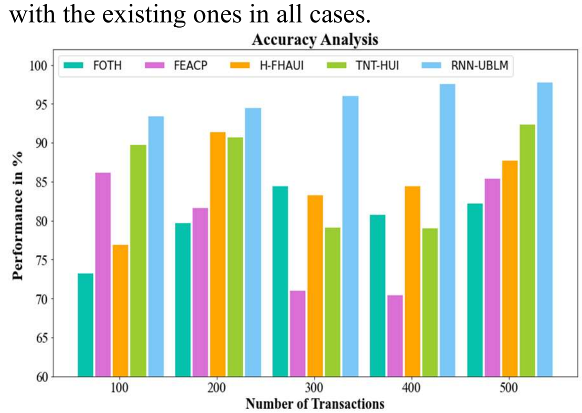

Figure 5: Accuracy Analysis for proposed RNN-UBLM model

3. Scalability Analysis

Table 3: Scalability Analysis for proposed RNN-UBLM model

Number of Transactions	FOTH	FEACP	H-FHAUI	TNT-HUI	RNN-UBLM
100	61.19	60.22	87.23	89.93	91.11
200	65.45	67.43	77.34	89.91	93.45
300	89.45	75.12	72.34	77.34	92.34
400	81.23	79.46	89.91	79.22	94.66
500	80.55	81.11	80.33	90.16	95.71

Table 3 and Figure 6 shows the scalability analysis of the proposed RNN-UBLM method against FOTH, FEACP, H-FHAUI, and TNT-HUI systems in terms of the number of transactions. The RNN-UBLM method is seen to have higher percentages of transactions processed than the other methods. For example, at 100 transactions, the proposed RNN-UBLM reaches 91.11%, while the FOTH and FEACP are lower at 61.19% and 60.22%

Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

respectively. With the increase in the number of transactions, RNN-UBLM still outperforms all the other models and reaches 95.71% at 500 transactions, while the second best model, TNT-HUI, reaches 90.16%. This shows the advantage of the RNN-UBLM method in terms of scale and speed.

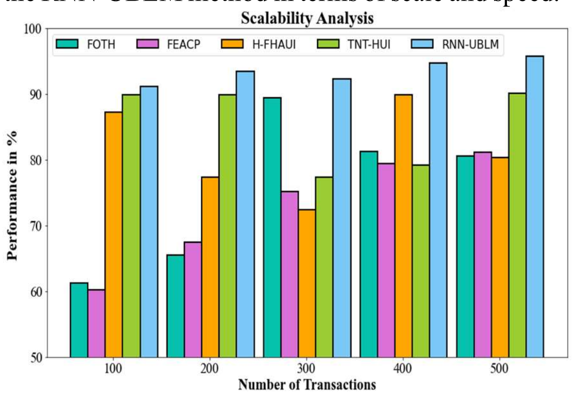

Figure 6: Scalability Analysis for proposed RNN-UBLM model

4. F-Score Analysis

Table 4: F-Score Analysis for proposed RNN-UBLM model

Number of Transactions	FOTH	FEACP	H-FHAUI	TNT-HUI	RNN-UBLM
100	87.56	80.44	71.23	90.45	92.54
200	80.33	83.45	70.56	89.22	94.56
300	81.13	89.56	85.33	85.45	95.11
400	75.45	90.66	75.67	81.23	95.67
500	79.33	73.37	71.91	87.56	96.12

Table 4 and Figure 7 presents the F-Score analysis for the RNN-UBLM technique compared to four existing systems: FOTH, FEACP, H-FHAUI, and TNT-HUI, for different numbers of transactions (100 to 500). The proposed RNN-UBLM is always superior to other methods in terms of accuracy and indicates a substantial improvement. For instance, at 100 transactions, the proposed RNN-UBLM has an F-score of 92.54% which is higher than that of TNT-HUI (90.45%) and the other methods. This tendency persists as the number of transactions rises, with RNN-UBLM achieving 96.12% at 500 transactions and outcompeting all others in terms of accuracy and efficiency with large datasets.

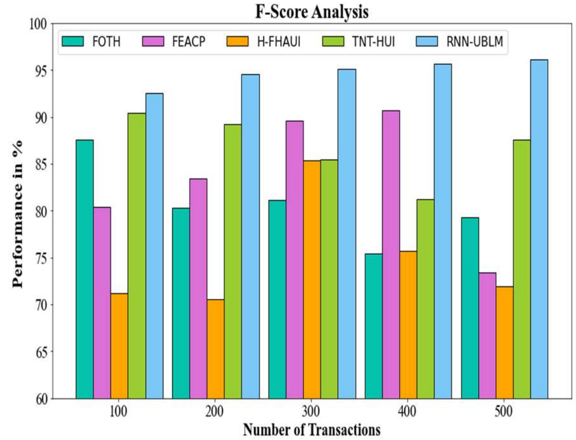

Figure 7: F-Score Analysis for proposed RNN-UBLM model

5. Memory Usage Analysis

Table 5: Memory Usage Analysis for proposed RNN-UBLM model

Number of Transactions	FOTH	FEACP	H-FHAUI	TNT-HUI	RNN-UBLM
100	65.45	71.56	77.56	81.45	86.23
200	61.45	70.45	74.35	78.87	91.45
300	66.34	77.27	83.55	85.45	87.77
400	63.56	70.23	78.18	81.19	92.91
500	64.11	76.11	79.56	82.45	94.66

Table 5 and Figure 8 show the memory usage analysis of the proposed RNN-UBLM technique in comparison with the FOTH, FEACP, H-FHAUI, and TNT-HUI systems for changed numbers of transactions. The memory efficiency of the RNN-UBLM method is the highest in all cases and is higher than the other approaches for all the transaction levels. For example, with 100 transactions, memory usage of RNN-UBLM is 86.23%, which is still advanced than the other approaches. The accuracy of RNN-UBLM increases with the number of transactions, which reaches 94.66% with 500 transactions, which confirms the efficiency of the suggested technique in terms of scalability and optimization of large datasets.

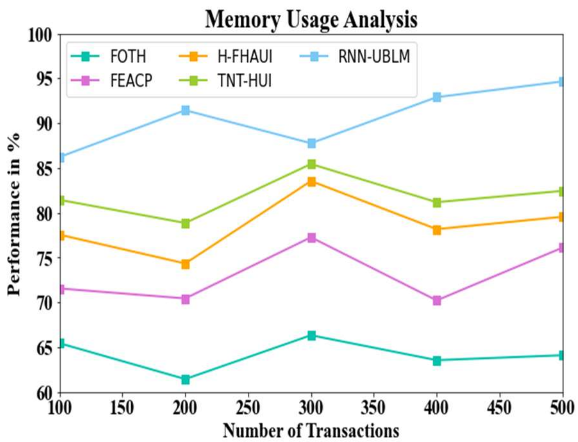

Figure 8: Memory Usage Analysis for proposed RNN-UBLM model

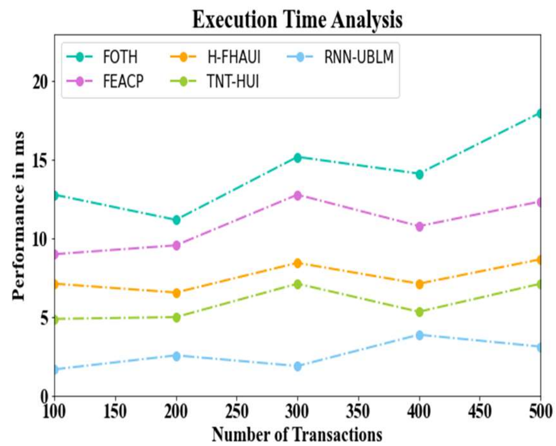

Figure 9: Execution Time Analysis for proposed RNN-U BLM model

6. Execution Time Analysis

Table 6: Execution Time Analysis for proposed RNN-UBLM model

Number of Transactions	FOTH	FEACP	H-FHAUI	TNT-HUI	RNN-UBLM
100	12.776	8.991	7.116	4.876	1.667
200	11.176	9.556	6.554	4.991	2.559
300	15.181	12.778	8.441	7.115	1.811
400	14.118	10.776	7.118	5.331	3.873
500	17.991	12.345	8.667	7.115	3.115

Table 6 and Figure 9 shows the execution time (Milli seconds (ms)) of the RNN-UBLM method against four benchmark systems (FOTH, FEACP, H-FHAUI, and TNT-HUI) for different numbers of transactions. The findings show that RNN-UBLM yields a higher accuracy than the other methods in all the cases of transaction size. For example, when 100 transactions are managed, RNN-UBLM takes only 1.667 ms, and FOTH takes 12.776 ms. When the number of transactions rises, RNN-UBLM remains operative, and the execution time is significantly lower than that of the other methods, which verifies the efficiency of the proposed method in processing a large number of transactions

4.4 Discussion

The integration of Utility List-Based Mining and Recurrent Neural Networks (RNNs) for utility itemset mining in transactional data is a promising method for extracting high-utility patterns. Utility List-Based Mining is able to track both the incidence and utility of itemsets and this means that the patterns of interest are not only frequent but also useful. This is done by integrating RNNs which helps in capturing temporal dependencies and sequential patterns in the data and hence successful on the predictive accuracy. This is especially the case when the item utility and transaction patterns are changing over time, which is helpful for creation better decisions in areas such as market study and recommendation systems.

4.5 Ablation Study

An ablation study of "Utility List-Based Mining and Recurrent Neural Network for Utility Itemset Mining Based on Transactional Data" would involve persistently eliminating or changing some of its parts, counting but not limited to the utility list-based mining algorithm, or the recurrent neural network (RNN) model, to assess the detriment in its presentation. This approach is also useful in identifying the role of each of the components in identifying high utility itemsets with clarification on how the list structure utilities mining, the role of RNN in pattern identification and accuracy prediction.

4.5.1 Influence of UBLM

Utility List-Based Mining (ULBM) plays a major role in enhancing the impact of data mining by enhancing the efficiency of identifying high-utility itemsets in massive datasets. Unlike other methods, ULBM employs utility lists to hold valuable information to minimize the number of times databases are scanned. This approach not only reveals frequent but also valuable patterns, which makes it extremely useful in various requests such as retail, market analysis, and resource management.

4.5.2 Limitations

- The approach may face experiments in climbing to very large datasets due to the computational complexity of utility list construction and recurrent neural network training.
- High-dimensional and sparse transactional data could reduce the performance of utility itemset mining, as relevant utility patterns may be hard to extract.
- The model might struggle to generalize well to transactional data with significantly different patterns or structures than those used in training, limiting its application scope.

5. CONCLUSION

Therefore, this paper has provided a strong foundation for the mining of utility itemsets in transactional databases using Utility List-Based Mining and RNN-LSTM. This approach integrates utility mining that targets high-utility patterns with the sequence learning capability of LSTM to overcome the challenges of utility itemset mining and the temporal nature of many transactional databases. The proposed method refines the pattern discovery by correctly identifying high utility itemsets and incorporates temporal dependency to increase the predictive accuracy. This hybrid model appears to be promising for different types of problems, including market basket analysis, recommendation systems, and financial forecasting, where both usefulness and temporal patterns are significant. Future work can extend the model to other domains and larger data sets to demonstrate its applicability in real-world problems.

DECLARATION

Data Availability: Data will be made available on request.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

Ethical Approval: The declaration is "Not Applicable".

REFERENCE

[1]. P. Fournier-Viger, Wu CW, S. Zida, V.S. Tseng. “FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning, InFoundations of Intelligent Systems: 21st International Symposium, ISMIS 2014, Roskilde, Denmark, June 25-27, 2014,” *Proceedings,* vol. 21, pp. 83-92, 2014. https://doi.org/10.1007/978-3-319-08326-1_9

[2]. Y. Liu, L. Wang, L. Feng x, B. Jin, “Mining high utility itemsets based on pattern growth without candidate generation,” *Mathematics*, vol. 25, No.91, pp.35, 2020, https://doi.org/10.3390/math9010035

[3]. T. Truong, H. Duong, B. Le, P. Fournier-Viger, U. Yun, H. Fujita. “Efficient algorithms for mining frequent high utility sequences with constraints,” *Information Sciences.* vol. 1, pp.568:239-64, Aug 2021. https://doi.org/10.1016/j.ins.2021.01.060

[4]. H. Kim, T. Ryu, C. Lee, H. Kim, E. Yoon, B. Vo, J.C. Lin, U. Yun. “EHMIN: Efficient approach of list based high-utility pattern mining with negative unit profits,” *Expert Systems with Applications,* vol. 15, pp. 209:118214, Dec. 2022 https://doi.org/10.1016/j.eswa.2022.118214

[5]. S. salman Qasim, L.M. Hasan. “Mining Utilities Itemsets based on social network,” *Babylonian Journal of Networking,* vol. 3, pp. 2024:25-30, Mar 2024. https://doi.org/10.58496/BJN/2024/004

[6]. N. T. Tung, L. T. Nguyen, T. D. Nguyen, B. Vo. “An efficient method for mining multi-level high utility itemsets,” *Applied Intelligence.* vol. 52, No. (5), pp. 5475-96, Mar. 2022, https://doi.org/10.1007/s10489-021-02681-z

[7]. P. Sra, S. Chand. “A Reinduction-Based Approach for Efficient High Utility Itemset Mining from Incremental Datasets,” *Data Science and Engineering.* vol. 9, No.1, pp. 73-87, Mar 2024. https://doi.org/10.1007/s41019-023-00229-4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[8]. S. Chand. "An efficient mining scheme for high utility itemsets," *arXiv preprint arXiv,* pp.2112.13485, 2021 Dec 27.

[9]. D. Oguz. "Ignoring internal utilities in high-utility itemset mining," *Symmetry,* vol. 7, No. 1411, pp. 2339, Nov 2022. https://doi.org/10.3390/sym14112339

[10]. N.T. Tung, T.D. Nguyen, L.T. Nguyen, B. Vo. "An efficient method for mining High-Utility itemsets from unstable negative profit databases," *Expert Systems with Applications.* vol. 1, No.237, pp.121489, Mar. 2024. https://doi.org/10.1016/j.eswa.2023.121489

[11]. V.V. Vu, M.T. Lam, T.T. Duong, L.T. Manh, T.T. Nguyen, L.V. Nguyen, U. Yun, V. Snasel, B. Vo. "FTKHUIM: a fast and efficient method for mining top-K high-utility itemsets," *IEEE Access.* vol.13, Sep 2023 https://doi.org/10.1109/ACCESS.2023.3314984

[12]. Z. Cheng, W. Shen, W. Fang, J.C. Lin. "A parallel high-utility itemset mining algorithm based on Hadoop," *Complex System Modeling and Simulation.* vol. 9, No.31, pp.47-58, Mar 2023. https://doi.org/10.23919/CSMS.2022.0023

[13]. A. Boukhalat, K. Heraguemi, M. Benouis, S. Akhrouf, B. Bouderah. "A Survey on Using Evolutionary Approaches-based high-utility itemsets Mining," *InArtificial Intelligence Doctoral Symposium,* vol.18, pp. 43-57, Sep 2022. Singapore: Springer Naturvole Singapore. https://doi.org/10.1007/978-981-99-4484-2_4

[14]. https://www.kaggle.com/code/msp48731/frequent-itemsets-and-association-rules/input

[15]. Z. Cheng, W. Fang, W. Shen, J.C. Lin, B. Yuan. "An efficient utility-list based high-utility itemset mining algorithm*," Applied Intelligence,* vol. 53, no. 6, pp. 6992-7006, Mar 2023. https://doi.org/10.1007/s10489-022-03850-4

[16]. V.S. Lalapura, V.R. Bhimavarapu, J. Amudha, H.S. Satheesh. "A systematic evaluation of recurrent neural network models for edge intelligence and human activity recognition applications," *Algorithms,* vol. 28, No.173, pp.104, Feb. 2024. https://doi.org/10.3390/a17030104

[17]. Y. Yan, X. Niu, Z. Zhang, P. Fournier-Viger, L. Ye, F. Min. "Efficient high utility itemset mining without the join operation," *Information Sciences. 2024 Oct* vol.1, pp. 681:121218. https://doi.org/10.1016/j.ins.2024.121218

[18]. N.T. Tung, L.T. Nguyen, T.D. Nguyen, P. B. "Efficient mining of cross-level high-utility itemsets in taxonomy quantitative databases," *Information Sciences.* vol. 1, pp. 587:41-62, Mar2022. https://doi.org/10.1016/j.ins.2021.12.017

[19]. B. Le, T. Truong, P. Duong Fournier-Viger, H. Fujita. "H-FHAUI: Hiding frequent high average utility itemsets," *Information Sciences.* vol. 1, pp. 611:408-31, Sep2022. https://doi.org/10.1016/j.ins.2022.07.027

[20]. Y. Liu, L. Wang, L. Feng, B. Jin. "Mining high utility itemsets based on pattern growth without candidate generation," *Mathematics, vol.* 25, No.91, pp.35, Dec.2020 https://doi.org/10.3390/math9010035