Journal of Applied Research and Technology

www.jart.icat.unam.mx

Journal of Applied Research and Technology 23(2025) 8-21

Original

Leveraging LSTM for precision inventory management by future demand forecasting

A. Tiwari[a,b*] • J. Pillai[c] • R. R. Janghel[d]

[a]*Department of Computer Science and Information technology Engineering, Bhilai Institute of Technology, Durg, Chhattisgarh, India*
[b]*Department of Computer Science Engineering, Rungta College of Engineering and Technology, Bhilai, Chhattisgarh, India*
[c]*Department of Computer Application, Bhilai Institute of Technology, Durg, Chhattisgarh, India*
[d]*Department of Information technology, National Institute of Technology, Raipur, Chhattisgarh, India*

Received 04 04 2024; accepted 06 28 2024
Available 02 28 2025

Abstract: The extraction of high utility data from massive datasets is one of the most well-known areas of research in data mining. The goal of high utility itemset mining is to identify the inventory's most lucrative items that users tend to favor. An LSTM-based approach is suggested to determine what consumers buy most frequently. Based on these purchases, high utility items that are expected to be in demand in the future are then identified based on client buying patterns. The development of the design, which will be utilized to manage inventories going forward and identify each consumer's item set, is the focus rather than just the price or amount of the item purchased. Additionally, related consumer groups can be put together, and consumers of similar commodities can be located by expanding the use case of the model. Lastly, an empirical comparison of the algorithms examines the accuracy of the approach and the number of valuable item sets and consumers that have been discovered via the use of the algorithms. The LSTM-based approach is the most effective, as seen by its 98% accuracy rate. It is especially useful for predicting future consumer purchases, identifying the most lucrative items, and effectively managing inventories.

Keywords: High utility itemset mining, LSTM, machine learning, deep learning, neural networks.

*Corresponding author.
E-mail address: adityarise0609@gmail.com (A. Tiwari).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

In the domain of knowledge discovery and data mining, high utility pattern mining (HUIM) is an essential research issue. High utility pattern mining (HUIM) solves the age-old problem that exists in traditional frequent pattern mining (FIM) by evaluating the frequency of recurrence of patterns and uncovering the occurrence of ways that lead to greater profits by assessing both the demand and supply of the utility (which depends both on amount and profit) for each item (Lin, Gan et al., 2016). Additionally, the extended notion of high utility can provide greater flexibility to different decisions made by consumers and suppliers (Lin, Zhang, Fournier-Viger et al., 2017). Itemset utility depends on both internal and external conceptions that are not restricted to only quantity or profit. Thus, the item can give a more adaptable interpretation based on the application area, which justifies its requirement. There are many literature surveys on high utility item mining techniques, which may be classified primarily as candidate generation and testing (Duong et al., 2018), analyzing the growing pattern based (Kannimuthu & Premalatha, 2014), usability and importance-based (Gan, Lin, Zhang, Yin, et al., 2021; Lin, Zhang & Fournier-Viger, 2017) and clustering-based methods. Pattern mining using the item's high utility data has piqued academics' interest recently. Numerous real-world application scenarios, such as website clickstreams, retail store transactions, and sensor data using IoT devices, generate real-time information about transactions, which are extensively used to perform different analyses.

Information analytic systems flourish because they can unearth actionable insights hidden within data. HUIM aims to extract user-important patterns that significantly impact sellers' profit or consumers' usability (Lin, Gan et al., 2016). A variety of criteria may determine the function that defines utility. Still, it was initially introduced to measure the profit made from the sale of a product, either by the seller or by the consumer (Lin, Zhang, Fournier-Viger et al., 2017). HUIM has primarily been viewed as a more general form of FIM, in which each item in the dataset is assigned a value, signifying its relevance to consumers and sellers. Another critical component of HUIM is that any object can appear several times in a single transaction, unlike FIM, which only displays whether each item occurs in every transaction. Early in the 21st century, the first HUIM techniques were published to provide more practical and valuable insights. Since then, numerous application disciplines, such as market basket analysis (Lin, Gan et al., 2016) and sentiment analysis, have explored the task.

However, HUIM algorithms may have extremely lengthy runtimes in huge search areas. Exact approaches decelerate as the amount of input data rises in terms of the transaction made by the customer and the sum of multiple components (Fournier-Viger et al., 2014). Thus, the performance is no longer suitable for those who want instant results and cannot wait for some time to get the results. Because of the performance constraints of accurate HUIM approaches, bio-inspired optimization methods such as genetic algorithms (Kannimuthu & Premalatha, 2014) and particle swarm optimization are used (Lin, Gan et al., 2016).

Most accurate or bio-inspired HUIM algorithms require a predetermined level of the utility threshold value, which determines the item's importance. Determining the actual quantity, on the other hand, is complex and usually necessitates a thorough grasp of the application domain. Both beginners and experienced users need to experiment with various thresholds by guessing and repeating the algorithms until the results are satisfactory (Tseng et al., 2010). We have shown that a slight adjustment in the threshold value can result in either a little (or even zero) or a substantial change (requiring a filtering step) set of solutions, as well as significantly longer execution durations. If this is the case, it might be best to just focus on mining the things that will be needed in the future. This is where LSTM (Long Short-Term Memory) (Hochreiter & Schmidhuber, 1997) comes in. It is a deep learning model that can determine what products customers will need in the future by knowing what they will need by looking at the patterns of their past purchases. It is easier to understand than setting a threshold and does not depend on database attributes.

The suggested technique would swiftly identify things that will be required shortly, efficiently search high utility itemset over various transactions, and predict future items required. Time series data is used to learn the pattern in which consumers shop, and then our model can identify the repeating pattern in the future. Knowing the anticipated demand will make it simpler to pinpoint the goods that various consumer types will need the most, which could directly affect sales. Our model can also identify recurring customers and cater to their needs, which will help retain the customer. Another issue that will be solved with the model is the time taken to know the result, as the inference time of the trained model is incredibly low and can learn up to 10,000 future transactions in the future, implying the items needed in the future can be stocked. The contributions of this paper are as follows:

- A new LSTM-based strategy for HUIM is suggested to forecast item data in a time series manner, significantly decreasing the search space and boosting time efficiency. This will maintain valuable information related to utility patterns. Thus, high utility itemset as well as the buying patterns of the consumer, are then identified.

- The approach is also enhanced to identify the items the customers will buy based on their previous purchase patterns.
- The model can also identify recurring customers and cater to their needs. By implementing the model, strategies are used to estimate the utilities of consumers, which can be reduced by discarding the utilities of items that are impossible to be high utility or are not involved in the search space.
- Up to 10,000 future items needed can be predicted, which can help significantly in stock management.

The advantage of discovering the relationship between consumers and what they purchase using LSTM is that it allows us to predict the products that will be required in the future. There is already work on high utility itemset mining. However, these products pay attention to the items they profit from and ignore the inventory management issue that can deal with itemset needed in the future. Our suggested approach tries to find a balance between the consumer, what they want and need in the future, and what item will make the most money or be the most popular in the future so that\ its stock can be maintained. The proposed approach has turned the problem into a time series, which lets us predict what will happen in the future and keep track of the stock. First, we have the dataset, which includes what users have purchased, their prices, how frequently they visit, etc. Then, the LSTM-based method is described and compared in terms of dependence strategy, comparison algorithms, dataset, parameter settings, benefits, and drawbacks, among other things. Finally, an experimental comparison of the algorithms looks at how accurate the method is and how many valuable sets of items and customers have been found using the algorithms. The results of the experimental research demonstrate that the LSTM-based process is the best one, particularly for forecasting items people would purchase in the future, finding the most profitable things, and managing the inventory accordingly (Hochreiter & Schmidhuber, 1997). The experimental comparison examination of the algorithm is conducted in terms of its efficiency while operating in real-time and the accuracy of its prediction of high utility itemset, which the customer will need in the future.

Furthermore, Section 2 of the article provides a brief background on the work performed in this experiment and provides the work related to this article. Section 3 explains the methodology adopted to carry through the task. The outcomes are regressively analyzed in the results section. The work is concluded in Section 5.

2. Background

2.1. LSTM

Long Short-Term Memory (LSTM) is one of the multiple forms of recurrent neural network (RNN) that is used for capturing the long-term dependencies relationship in sequential data and using it to make predictions about the future. An artificial neural network (ANN) typically consists of three layers:

- Input layer
- Hidden layers
- Output layer

In a NN with a single hidden layer, the number of input layer nodes is always proportionate to the size of the data and the input layer nodes are synaptically linked to the hidden layer. The weight coefficient between each pair of nodes from (the input to the hidden layer) determines the node's importance and whether the information will be passed on to the next layer. While learning the latent weight update process, the artificial NN will learn the appropriate weights for each individual synapse after learning is finished. The input and bias amounts, along with each importance, are added up and multiplied. This information is then sent to the hidden layer nodes using the SoftMax function to a sigmoid or tangent hyperbolic (tanh) function, which is known as the activation function. The numbers produced by this transformation need to have the lowest feasible error in the train after training, which can be verified in the test set. The values acquired due to this modification make up the NN's output. In this case, to get the most out of the model, the output layer and hidden layer are connected using a back-propagation approach, and the back-propagation process will be finished by delivering a signal that conforms with the ideal weight and ideal error for the chosen number of epochs. This approach will be used again to improve our forecasts and reduce mistakes. After this stage is finished, the model will have been trained. Recurrent neural networks (RNN) are a family of NNs that anticipate future value based on a previous sequence of observations; this kind of NN uses earlier stages to learn data and forecast patterns. To estimate and forecast forthcoming values, it is necessary to remember the data past locations. In this instance, the hidden layer stores the previous data from the sequential sequence. The approach of predicting future data based on the characteristics of earlier lines is referred to as "recurrent" in this context. Because RNN cannot store long-term data, using Long Short-Term Memory (LSTM) based on a "memory line" has proven to be quite beneficial in correctly predicting future scenarios involving long-term sequential

data. Prior-stage memory may be stored in an LSTM employing gates with an integrated memory line. The diagram below demonstrates the structure of LSTM nodes.

In contrast to other RNNs, LSTM is capable of recognizing data sequences and making predictions accordingly. The top line of each cell serves as a transport line, carrying data from the previous cell state to the present, and cell autonomy enables the model to filter or add values from the last cell to the current cell. Each LSTM node must comprise a collection of cells accountable for storing the sequence from the sequential data. The sigmoidal neural network layer, including the gates, optimizes the cell's value by deleting or permitting input to pass. A binary value (0 or 1) is assigned to each sigmoid layer, where 0 denotes "let nothing through" and one indicates "let everything through." To control the state of each cell, the gates are managed as follows:

- The output of the forget gate is a value between 0 and 1, with 1 denoting "totally keep this" and 0 denoting "absolutely disregard this."
- Which new data will be saved in the cell is determined by the memory gate. A sigmoid layer's initial "input door layer" determines the values to be modified. The next step is to create a vector of potential new candidate values that could be added to the state using a tanh layer.
- The output gate determines the output of each cell. The cell state and the current filtered and added data will both affect the output result.

2.2. Related works

Much work has been done in HUIm; the most relevant results related to the proposed approach are discussed. When exploring the domain of e-commerce/ retail for HUIM, N. Guo et al. devised a technique for extracting high-utility item sets from online retail data; the authors suggested a brand-new approach dubbed HUIF (Lin, Zhang & Fournier-Viger, 2017). To narrow the search field and expedite the mining process, HUIF employs a heuristic methodology. A real-world dataset from an online retailer was used to evaluate the algorithm, and the findings revealed that HUIF outperformed existing algorithms in terms of mining time and memory use. This is explored in a distributed fashion by García-Sánchez et al. (2005) where online e-commerce data is presented as an incremental and distributed approach. The algorithm can manage both the considerable volume of transactions and the dynamic nature of the data. The authors evaluated the method on a sizable dataset from an online retailer, and the findings demonstrated that the system could manage the mining operation well. X. He and colleagues suggested utilizing Apache Spark to implement a parallel algorithm for mining high-utility item sets from massive e-commerce data (Gan, Lin, Zhang, Yin, et al., 2021). The algorithm can be deployed over numerous nodes for parallel processing and managing high volumes and velocities of data. The results of the algorithm's evaluation on a dataset collected from a Chinese e-commerce website demonstrated that the algorithm could manage the mining task successfully and efficiently. A parallel technique named PHUI-Miner was developed by Yin et al., (2021) to mine high-utility items in e-commerce datasets. This approach is on par with many innovative algorithms using several sizable e-commerce datasets. According to their findings, PHUI-Miner performed superiorly in execution speed and scalability. For mining utility itemset in e-commerce datasets, Li et al. suggested a hybrid technique dubbed UIMiner-H. The method is evaluated against several other algorithms using various real-world data mining sets to determine its efficacy. Their findings demonstrated that UIMiner-H performed better in terms of efficiency and efficacy (Lin, Zhang & Fournier-Viger, 2017).

Similarly, innovative approaches such as Apriori and genetic algorithms were also used. Zhang et al. (2019) used a genetic algorithm for mining high-utility item sets from e-commerce data (Gan et al., 2019). The algorithm improves the efficiency and efficacy of the mining process by combining the benefits of genetic algorithms with pattern growth algorithms. A dataset from a Chinese e-commerce platform was used to evaluate the algorithm, and the findings indicated that it could find high-utility item groups that are beneficial for business applications. (Dong et al., 2020) used weighted negative utility and correlation-based pruning; the authors proposed a novel technique for mining high-utility item sets from online transaction data (Lin, Zhang, Fournier-Viger et al., 2017). The system can manage the uneven distribution of the data and can exclude unfavorable candidate item sets using correlation-based metrics. The technique was evaluated on several real-world datasets by the authors, and the results revealed that it can find high utility item sets beneficial for business applications. (He et al., 2022) suggested a unique two-step technique for mining high utility itemset from online e-commerce datasets. The Apriori method was used in the first stage to mine frequent item sets, and a utility-based pruning strategy was used in the second step to discover utility item sets (Duong et al., 2020). (Ghadekar & Dombe, 2019; Pillai & Vyas, 2015) suggested that the unique HUIM-Miner method be used in another work to mine high-utility items from online e-commerce datasets. The suggested algorithm employed a hybrid strategy to find high-value itemset that coupled the Apriori algorithm with a utility-based pruning mechanism. In subsequent research, (Rahman et al., 2022). Sra and Chand, (2024) proposed an improved technique for mining high value itemset from online e-commerce datasets. The efficient depth-first search method utilized by the suggested algorithm, HUIM-EDR, decreased the search space and increased algorithm efficiency (Nam et al., 2020). Chen et al. used a practical HUI-Max approach for mining high utility itemset in transactional datasets. The approach can be compared

against current state-of-the-art algorithms using a variety of real-world datasets, including e-commerce datasets. According to their findings, HUI-Max performed better than the other algorithms regarding memory use and execution time (Xiong et al., 2018) expanded this by introducing a unique HUI-Miner technique to find high-value itemset in e-commerce datasets. This approach employs a hybrid search strategy (Lin, Yang et al., 2016). The result shows reliable performance against several innovative algorithms using synthetic and real-world datasets. According to their findings, HUI-Miner performed better regarding memory usage and time taken for execution (Duong et al., 2018). A novel HUI-TS approach was introduced in a different work by Liu and Qu, (2012) for mining high-utility itemset in time-sensitive e-commerce datasets, making their model performance at par with several existing algorithms using various real-world datasets. According to their findings, HUI-TS performed better in efficiency and efficacy (Li et al., 2023). A brand-new plan named UIMiner was put out by Li et al. (2023) for mining utility item sets in e-commerce datasets. The evaluation of the algorithm was compared against a variety of real-world datasets and with a variety of innovative algorithms. Their results indicated that UIMiner outperformed other systems in terms of efficiency and efficacy. UIM-Span is a practical approach that (Shankar et al., 2009) suggested for mining utility item sets in e-commerce datasets. Using several real-world datasets, they assessed their method and contrasted it with several other algorithms (Song et al., 2014). Their findings demonstrated that UIM-Span performed better than the other algorithms in terms of execution speed and memory use. An approach named UIMiner-T was put out by Liu and Qu (2012) in different research for mining utility item sets from time-sensitive e-commerce datasets. The algorithm was tested on several real-world datasets and contrasted it with other innovative algorithms. Their findings demonstrated that UIMiner-T performed better in terms of efficiency and efficacy (Li et al., 2023) For mining utility itemset in e-commerce datasets with limitations (Wu et al., 2021) suggested a unique technique dubbed UIMiner-C. Using several real-world datasets, they assessed their method and contrasted it with several other algorithms. According to their findings, UIMiner-C fared better than the different algorithms in terms of efficacy and efficiency.

Overall, the literature suggests several efficient and effective algorithms are available for mining utility itemset in e-commerce datasets. Researchers have proposed various techniques, including time-sensitive and constraint-based mining, to manage the dynamic nature, high velocity, and volume of online transaction data and hybrid mining to improve the efficiency and effectiveness of these algorithms. The algorithms can be parallelized and distributed over multiple nodes for efficient processing and can manage the imbalanced nature of the data using negative utility and correlation-based measures. Further research is needed to develop more efficient and effective algorithms to address the increasing volume and velocity of online e-commerce data (Sohrabi, 2020).

The main problem is that high utility itemset mining aims to find the items that make more profit. This concept applies where yields are based on individual articles, which offers enormous profits. The main issue with this approach is that it is still being determined when the item will be sold. To overcome this issue, a model is developed to predict what will be sold. This will solve two problems. The first is inventory management, as the items predicted by the proposed model will be sold. The second is identifying similar groups of people. This will ensure a more appropriate categorization of the customers coming. This approach to finding high-utility items based on upcoming events is new and has yet to be explored (Song et al., 2014). Sukanya and Thangaiah (2023) proposed a HUI mining algorithm technique based on differential evolution and particle swarm optimization on voluminous transactional databases. Loukili et al. (2023) have successfully created an algorithm that utilizes association rules and the Frequent Pattern-Growth algorithm to provide personalized recommendations to consumers. and yielded favorable outcomes, including a high average probability of purchasing the subsequent product recommended by the recommendation system. Xue et al. (2017) propose a novel matrix factorization model with neural network architecture by creating user-item matrix for item prediction. Umayaparvathi and Iyakutti (2017) proposed a CNN-based solution by identifying customer attributes. Premanand G. et al proposed a CNN-based technique that uses the product history of the customer and the linkage between the same products bought by other customers to find high-utility products. Pazhaniraja et al. (2021) have used the random forest to predict the next order in an online grocery store depending on the transactions. Siva and Chaudhari (2024) has proposed a convolutional sequential semantic embedding technique for predicting top n recommendations and high utility itemset over sequential transactions.

3. Methodology

3.1. Proposed approach

In Figure 2 this study aims to construct and develop a deep learning-based system for anticipating and examining online grocery orders. A Long Short-Term Memory (LSTM) focuses on feature extraction and pattern detection in the analysis to predict how many groceries will be needed in the inventory. This saves time and effort and lets you find the most popular products, helpful items, etc. Time series is an integral part of statistics and machine learning that is often overlooked in data mining. This way of predicting grocery sales will help

companies meet customer demand for goods and group customers with similar tastes. This will also boost client happiness and create opportunities (Pillai & Vyas, 2015). Although time series analysis and forecasting have been well studied, only some research publications apply LSTM to real-time, live data in this domain. By pulling out time series patterns like seasonality or trends, grocery stores will better manage their logistics. An architecture was created using internal data that collects customer details and the items they bought from the shop over time to offer the best solution. The architecture does data preprocessing, which also finds their features and patterns (Gan, Lin, Zhang, Fournier-Viger, et al., 2021). Finally, an LSTM was created to solve the problem of anticipating the grocery requirements for the specified period using all the information learned during the analysis step. The proposed approach compares the data produced by the model with the actual demand for the products to validate the architecture and the deep learning-based artificial intelligence model.

The proposed approach has used Instacart, a grocery ordering and delivery app, as the dataset to develop the proposed model. There are over 300 million transaction data from more than 200,000 Instacart users. Here, it helps to identify all the patterns and habits of what consumers buy and what they will need in the future. The proposed model predicts which previously purchased products will be in need in future orders and keeps track of items needed if a previous customer comes again, which is done by finding the reorder pattern of the existing customer. Once the model is trained, it can identify future needs.

The input sequence in itemset mining uses the short-term memory from the preceding cell and the long-term memory from the previous cell state at a particular time t. Thus, LSTM will use two types of information to make it accurate. The forget gate will find and only keep the relevancy of the information passed from the earlier cell state after the sigmoid layer converts the translated information from the current input and the previous cell into a value between 0 and 1. Non-important pieces of information are then deleted. When the value is 1, all data is saved in the cell; however, when the value is 0, all data from the earlier state is erased. Like the output gate, the input gate specifies which data the change gate should update. Which data from the present cell state should be the output gate determines output (Fournier-Viger et al., 2019).

The aim of developing this model is to predict the grocery requirement in the future from the analysis of current and past orders. To achieve this, LSTM architecture is proposed, which takes input from the existing orders and predicts future orders, as it is evident that around 60% of the items are reordered. This prediction of future grocery items will help to maintain the required stock (Luna et al., 2023). By predicting future orders, it is evident that the high-utility item can be restocked.

The data needed to predict future demands which are finding the user (use_id), the product's the user buys (product_id), the day in the week when the order is placed, the hour of the day when the order was placed, and the time since the last order, which is used to predict the product_id to be purchased in the upcoming days. With the use of a multivariate sequence of input variables and supplied multivariate grocery time series data collected from multiple sources, the suggested model aims to anticipate future demand (Liu et al., 2005). The following LSTM implementation strategies are being assessed to fulfil this assignment from the original dataset modified. Moreover, a continuous sequence is used to build an input sequence that includes the crucial dimension of the LSTM design. Figure 1 depicts the recommended model framework.

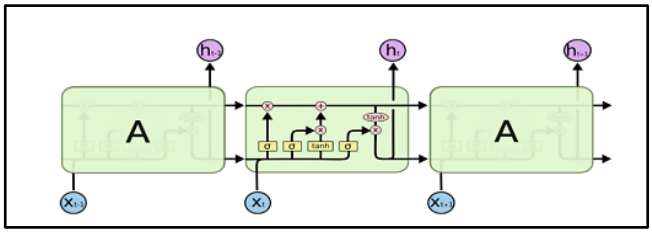

Figure 1. The internal structure of an LSTM (Liu & Qu, 2012).

The input sequence consists of k = 5 (user (user_id), the product's user buys (product_id), the day in the week when the order is placed, the hour of the day when the order was placed, the time since the last order) features with a time step of m, as seen in the above section of the image. The LSTM is supplied with the input Xt1 at time t1, a 5 m matrix correlated with ht2 and ct2. The second stage's LSTM takes as inputs the output ht1 of the earlier stage, the input sequence Xt, and the cell memory ct1. This procedure is conducted four times until the final input sequence Xf and corresponding output hf, a vector of length equal to the number of neurons in the final LSTM layer, is obtained. hf is finally sent to a fully linked layer, as seen in the last part of Figure 3, where a linear activation function is used to estimate the closing products. The output will be a multilabel classification where every item in the inventory for the future is predicted. If the label output is 1, then the item will be needed in future order, and if the output is 0, the item is not needed in future order for the current time step. The average frequency of each item that is regularly purchased by the customer is fixed as the quantity of that item. For example, we can have only 5 items in our itemset that are: Milk, oil, banana, strawberry, and tea. So, if our predicted output is 01100, this means only oil and bananas are purchased in the next order. The quantity is the average amount of the items the customer buys, which is 1ltr for oil and 6 bananas (Lin, Yang et al., 2017).

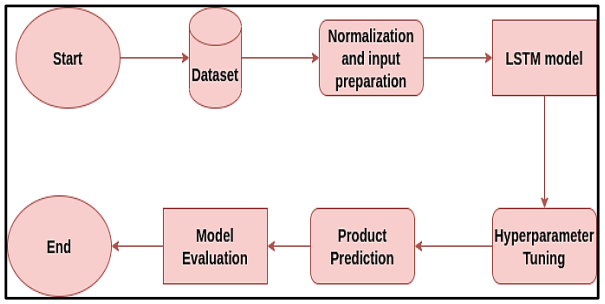

Figure 2. Proposed framework.

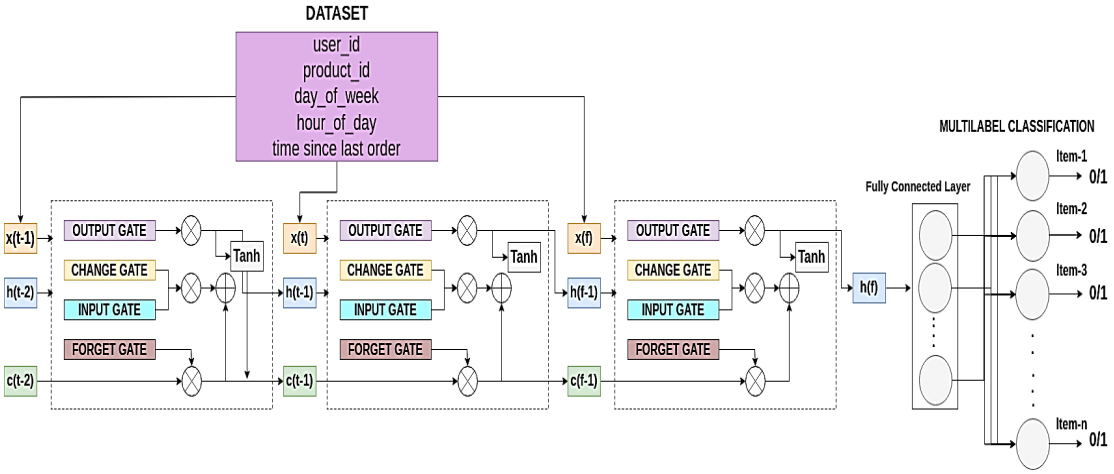

Figure 3. The proposed model architecture for product prediction using LSTM and multilabel classification.

The architecture of the LSTM is a 4-layered LSTM stacked upon each other; a linear layer follows this. The input to the network is the data discussed above. Then, for each time stamp, the hidden and cell states are found for each layer, and the last hidden state is given to the linear layer, which predicts the product_id, which will be purchased in the future. While training, 10,000 future items are expected, and the loss is calculated and backpropagated by comparing all the future predictions. The proposed approach will use an LBFGS (Limited-Memory Broyden-Fletcher-Goldfarb-Shanno) optimizer to train the model and an MSE (Mean squared error) loss to measure the accuracy of our model. At the same time, training the model, the future prediction made by the model, and the failure to finetune the model (Nam et al., 2020). The training was done on an Rtx-5000 GPU, and it took around 32 hours to converge and give a reasonable accuracy. After successful movement, the model predicts up to 10,000 future orders successfully.

The architecture of the LSTM is a 4-layered LSTM stacked upon each other. This is followed by a linear layer. The input to the network is the data discussed above. Then, for each time stamp, the hidden and cell state is found for each layer, and the last hidden state is given to the linear layer, which predicts the product_id that will be purchased in the future. While training, 10,000 future items are predicted, and the loss is calculated and backpropagated by comparing all the future predictions(*(PDF) a predictive analytics model for maximizing profit in e-commerce companies*, n.d.). The proposed approach will be used using LBFGS (Limited-Memory Broyden-Fletcher-Goldfarb-Shanno) optimizer to train the model and use MSE (Mean squared error) loss to measure the accuracy of our model. While training the model, the future prediction is made by the model and the loss to finetune the model (Lai et al., 2023). The training was done on an Rtx-5000 GPU, and it took around 32 hours to converge and achieve good accuracy. After successful training, the model predicts up to 10,000 future orders successfully (Nam et al., 2020).

When the input is embedded in the hidden space, every recently purchased item can be clubbed together. Similarly, the cell state will contain more frequent item embedding but for long sequences. Hence, the combination of recent and the whole sequence finds every type of small and large product

pattern, making the model perfectly accurate for future prediction. The model also considers the recurring customers and caters for their needs. Implementing this strategy to estimate the item adds extra dimension, such that there can be a reduction in item sets by discarding items that are impossible to be of high utility or not involved in the search space.

3.2. Data description

The Instacart data from the app is the dataset that is analyzed. This data consists of a relational set of transactional data that tracks the orders of clients over time. To determine HUI, the goal is to predict which items will be in a user's future order (high utility Itemset). Around 200,000 Instacart members contributed over 3 million food orders to the dataset, which is anonymous. Values between 4 and 100 of each user's orders and a list of the things they bought when.

The description of the dataset is as follows:

- Aisles: There are 134 distinct aisles in this file, each of which is different.
- Departments: A total of 21 distinct departments are represented in this file.
- Orders: This file includes all the orders that various users have placed. The following may be inferred from the analysis below:
  - 206209 users have placed 3421083 orders.
  - The three sets are the prior, train, and test sets of orders. While order distributions in the train and test sets are comparable, order distributions in the prior set are different.
  - The range of a customer's total orders is 0 to 100.
  - Based on the plot of "orders vs day of week," thus map 0 and 1 as Saturday and Sunday, respectively, on the assumption that most people buy food on the weekends.
  - Most orders are placed during the day.
  - The 7, 14, 21 and 30 peaks in the "orders vs days since prior order" graph indicate that customers only place one order weekly.
  - Based on the relationship between "day of week" and "hour of day," it is inferred that Saturday afternoons and Sunday mornings are the busiest periods for orders.
- Products: This file includes a list of all 49688 items, together with information about each one's aisle and department. Different aisles and sections have varying numbers of merchandise.
- Order_products_prior: This file contains details on the products that were ordered and the order in which they were put in the shopping cart. It also discloses whether the goods were reordered.
  - This file contains details on a total of 3214874 orders that resulted in 49677 assorted products being ordered.
  - According to the "count vs things in cart" plot, most customers only order 1 to 15 items, with orders including a maximum of 145 items.
  - In this set, 58.97% of the elements are repeats.
- Order_products_train: This file contains details on the items that were ordered and the order in which they were put in the shopping cart. It also discloses whether the goods were reordered.
  - This file contains details about a total of 131209 orders that resulted in orders for 39123 assorted products.
  - According to the "count vs things in cart" plot, most customers only order 1 to 15 items, with orders including a maximum of 145 items.
  - In this set, 59.86% of the elements are repeats.

Here is some visualization in Figure 4, 5 and 6 on the dataset, which can be seen here: The hour from 9 to 5 is busy, and that is the time when a customer orders the most. The list of most popular items here can also be seen, which changes over time depending on the demand. Here is the pattern of re-ordering pattern of existing customers.

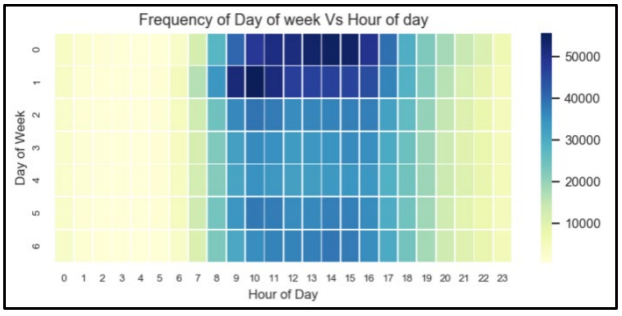

Figure 4. Orders frequency for every day of the week.

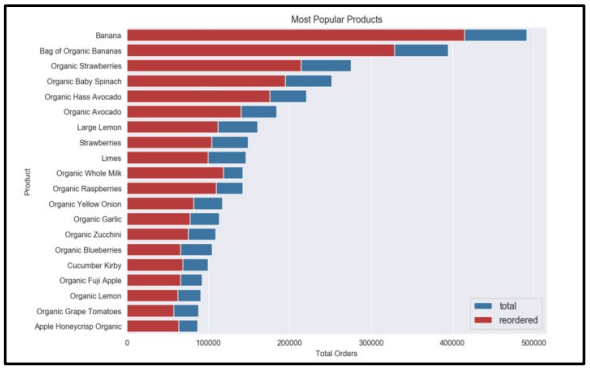

Figure 5. Popular product from the itemset.

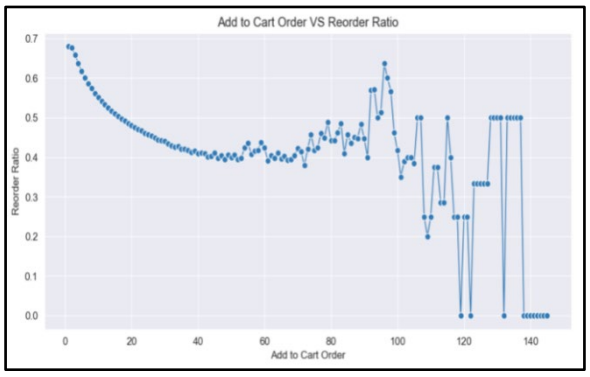

Figure 6. Pattern of reordering of products by the existing customers.

4. Results

4.1. Data preprocessing

Data preprocessing is a crucial step in our analysis, where raw data is cleaned, transformed, and organized to prepare it for further analysis. The first step is identifying missing values in the dataset and replacing missing values with estimated ones. Then we dealt with outliers and standardizing formats. This is followed by normalization or scaling of data to ensure that all features have a similar scale. It may also involve encoding categorical variables into a numerical format. At last, normalization is performed by scaling numeric data to a standard range, often between 0 and 1, to prevent certain features from dominating others. and transforming continuous data into categorical data. The dataset is also reorganized and split into train and test sets. The objective is to precisely predict the product that will be bought in the future and exhibit complex, distracting, and volatile behavior.

4.2. Model performance metrics

The model's accuracy is around 98%, even though the repeated orders are about 60 %. Still, accuracy was high due to finding large and small patterns, which not only see the products that consumers purchase every time but also keep track of the products purchased occasionally. This gives the original advantage of LSTM over other regular mining models, which is proven by the accuracy of predicting 10,000 items in advance. The accuracy was calculated by finding the several types of products ordered, the quantity of each product, the user who requested the product, and the amount of the product called by each user. The model is trained for 250 epochs. We have used the Adam optimizer with a learning rate of 0.0001. We have employed early stopping so that the model does not overfit.

A four-layered LSTM architecture is used to anticipate the product. Several options with varying numbers of neurons are considered within each of these models. These models' prediction accuracy and dependability are evaluated using three separate performance metrics: RMSE, MAPE, and R. These measures are defined in their analytical form as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \quad (1)$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N} \left|\frac{y_i - \hat{y}_i}{y_i}\right|, \quad (2)$$

$$R = \frac{\sum_{i=1}^{N} (y_i - \underline{y_i})(y_i - \underline{\hat{y}_i})}{\sqrt{\sum_{i=1}^{N} (y_i - \underline{y_i})^2 (y_i - \underline{\hat{y}_i})^2}} \quad (3)$$

Where,

- yi: Original sequential data,
- yi: The average value in the original sequential data,
- yi: Predicted sequential data calculated by the model,
- yi: Average value of the predicted sequential data,
- N: Number of observations.

The corresponding precision between the absolute and predicted values is calculated by the relative average of the error (or MAPE) metric, the fundamental and predicted values are determined by the R metric and the square root of the mean square error of the actual and estimated values is measured by the RMSE metric. When the RMSE and MAPE values are lower, the model has achieved good accuracy and precision. On the other hand, a higher R-value suggests that the expected and actual sequences are comparable. The model's forecast for a particular phase was not directly compared to the actual output; instead, the number of different product types and the quantity of each product were measured, and this information was utilized to determine all the matrices. Suppose it is tried to compare each transaction. In that case, the error will be high. Also, it needs to make sense to give the correct criteria as the focus is on finding various products that consumers will require, and the order of finding the products is not an important feature. Also, the predictions calculated using the normalized data are transformed in the opposite direction to provide performance metrics. The model is independently run several times to address and eliminate stochastic behavior. The typical MAPE and R scores are displayed in Table 1. The proposed approach used four different models and found that the four-layer LSTM-based model was optimal.

Table 1. RMSE, MAPE, and R values for predicting 10,000 items to be purchased.

		3 Layer LSTM	5 Layer LSTM	7 Layer LSTM	4 Layer LSTM
RMSE	MIN	34.73	43.82	38.55	37.27
	AVERAGE	49.95	57.07	47.19	42.7
	MAX	77.48	72.16	60.74	49.49
	STD	9.77	8.08	4.96	2.95
MAPE	MIN	0.75	0.89	0.76	0.72
	AVERAGE	1.12	1.23	0.97	0.86
	MAX	1.61	1.5	1.24	1.1
	STD	0.25	0.17	0.099	0.0912
R	MIN	0.99	0.99	0.99	0.99
	AVERAGE	0.99	0.99	0.99	0.99
	MAX	0.99	0.99	0.99	0.99
	STD	0.0006	0.0007	0.0003	0.0002

The primary model selection criteria are the average RMSE score produced from these several repetitions. A model with the lowest RMSE, MAPE, and highest R values are considered good.

The graph Figure 7, 8, 9, 10 and 11 for training AUC (Area Under Curve) is given, which clearly shows the model correctly predicts future orders. Thus, AUC is almost one for training and 0.9 for validation. The value of loss and log loss while training is 0.05, boosting the accuracy to 98%.

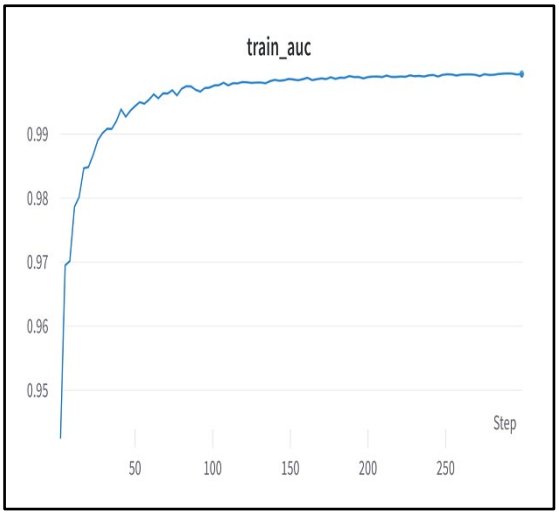

Figure 7. Training AUC (area under the curve) of LSTM mode.

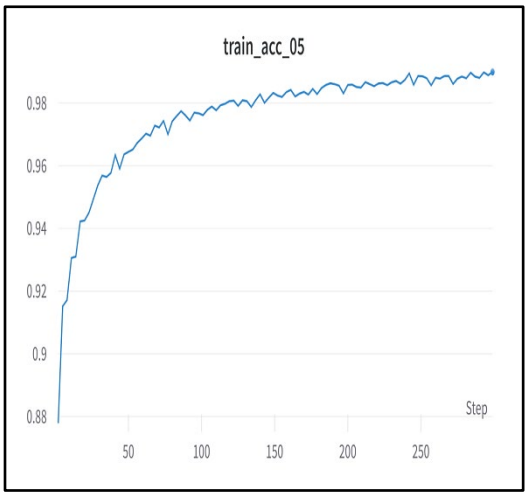

Figure 8. Training accuracy of LSTM mode.

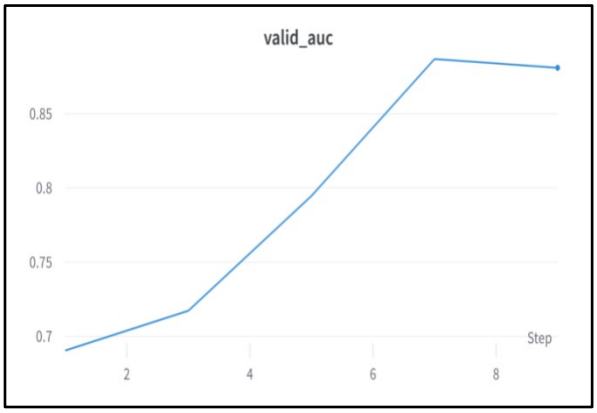

Figure 9. Validation AUC of LSTM mode.

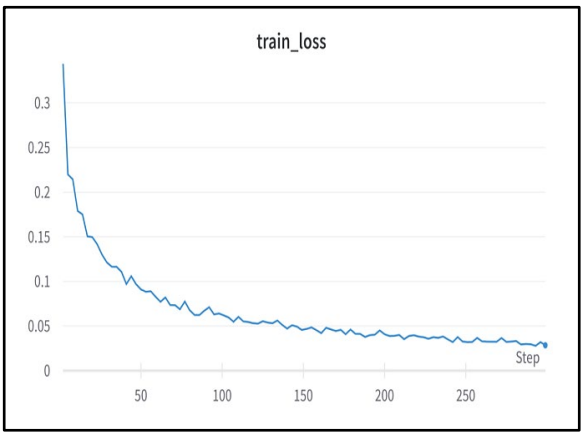

Figure 10. Training RMSE loss of LSTM mode.

Figure 11. Training log loss of LSTM mode

4.3. Discussion

The main aim of this approach is to provide inventory management to maximize profit. The higher accuracy of the model suggests that the item that will be required in the future is correctly predicted using LSTM. This includes both the items that are demanded in huge quantity and have less profit per item as well as less demanded items that have higher profit per item. This is achieved using LSTM as the global objective is to maximize the profit and due to the architecture of LSTM, both cases are covered. Once the LSTM is trained, we have evaluated our model and have an accuracy of 98%. We used RMSE, MAPE, and R values to validate the accuracy and fix the best architecture suitable to us. To calculate the loss between the predicted and absolute value, RMSE uses a simple distance between the items in the 2-D plane while MAPE measure the distance in the 1-D plane and the R score identifies the relation between dependent and independent items purchased. This way, our approach not only finds the best architecture for prediction but also uses different metrics so that the predicted and absolute items chosen are the same, as well the sequence of choosing the items can be closer. The focus is also given to the item, often taken in pairs, so the prediction can be as accurate as possible.

4.4. Relationship between effectiveness vs. average transaction size of customer

Various customers have different tractions, which include other items. Traditional item-set mining considers individual items. Still, our model focuses on multiple things simultaneously, maximizing the total profit by keeping everything in the inventory beforehand. Another observation is that the number of transaction-weighted utilizations of any itemset X is defined as the sum of the transaction utilities of all the transactions containing X. This overestimation worsens as transactions become more extended because more significant transactions tend to involve more unrelated things. Despite the overestimation, Phase I's efficiency is clear. As a result, our suggested approach performs better, especially in dense databases.

4.5. Comparison with existing techniques

In this section, we have compared our approach with some existing approaches that use machine learning and deep learning for the prediction of high utility itemset. It can be seen in Table 2 and the accuracy of our approach outperforms the existing techniques.

Table 2. Comparison with other techniques.

	Technique	Accuracy
Xue Hong-Jian	Federated NN and CNN	71
Umayaparvathi V.	Random Forest, and XGboost	70
Premanand G.	CNN	88
Pazhaniraja N.	Random Forest	91
Siva S.	CNN with semantic Embedding	99
Ours	LSTM	98

5. Conclusion

The region of most excellent attention is itemset mining, which is what will come next. However, accurate and consistent product prediction is challenging because of its chaotic and nonlinear behavior. Market data, macroeconomic data, and other factors impact output predictions. This study focuses on building LSTM-based models to anticipate the product that will be in demand in the future so that customers can get the things they desire, merchants can know what to provide, logistics management is simplified, and profit is increased. The LSTM model was implemented, and its performance was assessed using various assessment criteria to verify the model. A four-layer LSTM model may offer the best fit and excellent accuracy for prediction, as per the experimental data. The suggested method can be used in other key market segments where the data indicates an equivalent tendency. The development of predictive model's hybrid using the LSTM or other sequence models, making more sophisticated neural networks. This can be used with existing data mining architectures and is another intriguing field for future study. In addition, the proposed method trains the model parameters using hybrid optimization methodologies such as genetic and particle swarm optimization algorithms to improve prediction accuracy even more. The only shortcoming of this approach is that the addition of the latest items in the inventory requires retraining of the model.

Conflict of interest

The authors have no conflict of interest to declare.

Funding

The authors received no specific funding for this work.

References

Duong, H., Truong, T., Tran, A., & Le, B. (2020). Fast generation of sequential patterns with item constraints from concise representations. *Knowledge and Information Systems*, *62*(6), 2191-2223. https://doi.org/10.1007/s10115-019-01418-2

Duong, Q. H., Fournier-Viger, P., Ramampiaro, H., Nørvåg, K., & Dam, T. L. (2018). Efficient high utility itemset mining using buffered utility-lists. *Applied Intelligence*, *48*(7). https://doi.org/10.1007/s10489-017-1057-2

Dong, X., Hao, F., Zhao, L., & Xu, T. (2020). An efficient method for pruning redundant negative and positive association rules. *Neurocomputing*, *393*, 245-258. https://doi.org/10.1016/j.neucom.2018.09.108

Fournier-Viger, P., Wu, CW., Zida, S., Tseng, V.S. (2014). FHM: Faster High-Utility Itemset Mining Using Estimated Utility Co-occurrence Pruning. In: Andreasen, T., Christiansen, H., Cubero, JC., Raś, Z.W. (eds) Foundations of Intelligent Systems. ISMIS 2014. *Lecture Notes in Computer Science*, vol 8502. Springer, Cham. https://doi.org/10.1007/978-3-319-08326-1_9

Fournier-Viger, P., Zhang, Y., Lin, J. C. W., Fujita, H., & Koh, Y. S. (2019). Mining local and peak high utility itemsets. *Information Sciences*, *481*, 344-367. https://doi.org/10.1016/j.ins.2018.12.070

García-Sánchez, F., Valencia-García, R., & Martínez-Béjar, R. (2005). An integrated approach for developing e-commerce applications. *Expert Systems with applications*, *28*(2), 223-235. https://doi.org/10.1016/j.eswa.2004.10.004

Gan, W., Lin, J. C. W., Chao, H. C., Fujita, H., & Philip, S. Y. (2019). Correlated utility-based pattern mining. *Information Sciences*, *504*, 470-486. https://doi.org/10.1016/j.ins.2019.07.005

Gan, W., Lin, J. C. W., Zhang, J., Fournier-Viger, P., Chao, H. C., & Philip, S. Y. (2021). Fast utility mining on sequence data. *IEEE transactions on cybernetics*, *51*(2), 487-500. https://doi.org/10.1109/TCYB.2020.2970176

Ghadekar, P., & Dombe, A. (2019). Image-Based Product Recommendations Using Market Basket Analysis. In *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICCUBEA47591.2019.9128524

Gan, W., Lin, J. C. W., Zhang, J., Yin, H., Fournier-Viger, P., Chao, H. C., & Yu, P. S. (2021). Utility mining across multi-dimensional sequences. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *15*(5), 1-24. https://doi.org/10.1145/3446938

He, J., Han, X., Wang, J., & Zhang, K. (2022). Efficient high-utility occupancy itemset mining algorithm on massive data. *Expert Systems with Applications*, *210*, 118329. https://doi.org/10.1016/j.eswa.2022.118329

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kannimuthu, S., & Premalatha, K. (2014). Discovery of high utility itemsets using genetic algorithm with ranked mutation. *Applied Artificial Intelligence*, *28*(4). https://doi.org/10.1080/08839514.2014.891839

Lai, F., Zhang, X., Chen, G., & Gan, W. (2023). Mining periodic high-utility itemsets with both positive and negative utilities. *Engineering Applications of Artificial Intelligence*, *123*, 106182. https://doi.org/10.1016/j.engappai.2023.106182

Li, Z., Li, G., Zhao, L., & Shang, T. (2023). List-based mining top-k average-utility itemsets with effective pruning and threshold raising strategies. *Applied Intelligence*, *53*(21), 25678-25696. https://doi.org/10.1007/s10489-023-04864-2

Lin, J. C. W., Gan, W., Fournier-Viger, P., Hong, T. P., & Tseng, V. S. (2016). Efficient algorithms for mining high-utility itemsets in uncertain databases. *Knowledge-Based Systems*, *96*, 171-187. https://doi.org/10.1016/j.knosys.2015.12.019

Lin, J. C. W., Yang, L., Fournier-Viger, P., Wu, J. M. T., Hong, T. P., Wang, L. S. L., & Zhan, J. (2016). Mining high-utility itemsets based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, *55*, 320-330. https://doi.org/10.1016/j.engappai.2016.07.006

Lin, J. C. W., Yang, L., Fournier-Viger, P., Hong, T. P., & Voznak, M. (2017). A binary PSO approach to mine high-utility itemsets. *Soft Computing*, *21*, 5103-5121. https://doi.org/10.1007/s00500-016-2106-1

Lin, J.CW., Zhang, J., Fournier-Viger, P. (2017). High-Utility Sequential Pattern Mining with Multiple Minimum Utility Thresholds. In: Chen, L., Jensen, C., Shahabi, C., Yang, X., Lian, X. (eds) Web and Big Data. APWeb-WAIM 2017. Lecture Notes in Computer Science(), vol 10366. Springer, Cham. https://doi.org/10.1007/978-3-319-63579-8_17

Lin, J. C. W., Zhang, J., Fournier-Viger, P., Hong, T. P., & Zhang, J. (2017). A two-phase approach to mine short-period high-utility itemsets in transactional databases. *Advanced Engineering Informatics*, *33*, 29-43. https://doi.org/10.1016/j.aei.2017.04.007

Liu, Y., Liao, Wk., Choudhary, A. (2005). A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets. In: Ho, T.B., Cheung, D., Liu, H. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2005. Lecture Notes in Computer Science(), vol 3518. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11430919_79

Liu, M., & Qu, J. (2012). Mining high utility itemsets without candidate generation. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 55-64). https://doi.org/10.1145/2396761.2396773

Luna, J. M., Kiran, R. U., Fournier-Viger, P., & Ventura, S. (2023). Efficient mining of top-k high utility itemsets through genetic algorithms. *Information Sciences*, *624*, 529-553. https://doi.org/10.1016/j.ins.2022.12.092

Loukili, M., Messaoudi, F., & El Ghazi, M. (2023). Machine learning based recommender system for e-commerce. *IAES International Journal of Artificial Intelligence*, *12*(4), 1803-1811.

Nam, H., Yun, U., Yoon, E., & Lin, J. C. W. (2020). Efficient approach of recent high utility stream pattern mining with indexed list structure and pruning strategy considering arrival times of transactions. *Information Sciences*, *529*, 1-27. https://doi.org/10.1016/j.ins.2020.03.030

Pazhaniraja, N., Sountharrajan, S., Suganya, E., & Karthiga, M. (2021). Top 'N'Variant Random Forest Model for High Utility Itemsets Recommendation. *EAI Endorsed Transactions on Energy Web*, *8*(35), e7-e7. https://doi.org/10.4108/eai.25-1-2021.168225

Pillai, J. & Vyas, O. P. (2015). Exploration of Soft Computing Approaches in Itemset Mining. In I. Management Association (Ed.), *Business Intelligence: Concepts, Methodologies, Tools, and Applications* (pp. 1830-1856). IGI Global. https://doi.org/10.4018/978-1-4666-9562-7.ch091

Rahman, M. R., Arefin, M. S., Rahman, S., Ahmed, A., Islam, T., Dhar, P. K., & Kwon, O. J. (2022). A Comprehensive Survey on Affinity Analysis, Bibliomining, and Technology Mining: Past, Present, and Future Research. *Applied Sciences*, *12*(10), 5227. https://doi.org/10.3390/app12105227

Sra, P., & Chand, S. (2024). A Reinduction-Based Approach for Efficient High Utility Itemset Mining from Incremental Datasets. *Data Science and Engineering*, *9*(1), 73-87. https://doi.org/10.1007/s41019-023-00229-4

Shankar, S., Babu, N., Purusothaman, T., & Jayanthi, S. (2009). A fast algorithm for mining high utility itemsets. In *2009 IEEE International Advance Computing Conference* (pp. 1459-1464). IEEE. https://doi.org/10.1109/IADCC.2009.4809232

Sohrabi, M. K. (2020). An efficient projection-based method for high utility itemset mining using a novel pruning approach on the utility matrix. *Knowledge and Information Systems*, *62*, 4141-4167. https://doi.org/10.1007/s10115-020-01485-w

Song, W., Liu, Y., & Li, J. (2014). Mining high utility itemsets by dynamically pruning the tree structure. *Applied intelligence*, *40*, 29-43. https://doi.org/10.1007/s10489-013-0443-7

Tseng, V. S., Wu, C. W., Shie, B. E., & Yu, P. S. (2010). UP-Growth: an efficient algorithm for high utility itemset mining. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 253-262). https://doi.org/10.1145/1835804.1835839

Sukanya, N. S., & Thangaiah, P. R. J. (2023). Enhanced differential evolution and particle swarm optimization approaches for discovering high utility itemsets. *International Journal of Computational Intelligence and Applications*, *22*(01), 2341005. https://doi.org/10.1142/S1469026823410055

Siva, S., & Chaudhari, S. (2024). Deep Learning Framework with Convolutional Sequential Semantic Embedding for Mining High-Utility Itemsets and Top-N Recommendations. *22*, 44-55. https://doi.org/10.56977/jicce.2024.22.1.44

Umayaparvathi, V., & Iyakutti, K. (2017). Automated feature selection and churn prediction using deep learning models. *International Research Journal of Engineering and Technology (IRJET)*, *4*(3), 1846-1854.

Wu, J. M. T., Srivastava, G., Wei, M., Yun, U., & Lin, J. C. W. (2021). Fuzzy high-utility pattern mining in parallel and distributed Hadoop framework. *Information Sciences*, *553*, 31-48. https://doi.org/10.1016/j.ins.2020.12.004

Xiong, X., Chen, F., Huang, P., Tian, M., Hu, X., Chen, B., & Qin, J. (2018). Frequent itemsets mining with differential privacy over large-scale data. *IEEE access*, *6*, 28877-28889. https://doi.org/10.1109/ACCESS.2018.2839752

Xue, H. J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017). Deep matrix factorization models for recommender systems. In *IJCAI* (Vol. 17, pp. 3203-3209). https://www.ijcai.org/Proceedings/2017/0447.pdf

Zhang, Q., Fang, W., Sun, J., & Wang, Q. (2019). Improved genetic algorithm for high-utility itemset mining. *IEEE Access*, *7*, 176799-176813. https://doi.org/10.1109/ACCESS.2019.2958150