Journal of Theoretical and Applied Information Technology
15th October 2024. Vol.102. No. 19

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

LEVERAGING DEEP LEARNING MODEL WITH OPTIMIZATION ALGORITHM FOR HIGH UTILITY ITEMSET MINING IN TRANSACTIONAL DATA

G.N. SOWJANYA[1], M. Babu Reddy[2]

[1]Research Scholar, Krishna University Machilipatnam, Andhra Pradesh, India.

[2]Asst.Professor, Krishna University Machilipatnam, Andhra Pradesh, India.

E-mail: sowjanyagn2024@outlook.com

ID 55443 Submission	Editorial Screening	Conditional Acceptance	Final Revision Acceptance
28-08-2024	30-08-2024	07-09-2024	03-10-2024

ABSTRACT

Mining patterns with higher utilization (or higher-utility itemset mining, HUIM) are believed the main problem in the past few decades particularly in the market (For example, supermarkets) engineering later exposes useful products or information for decision-making. Most of the current works concentrated on mining higher-utility itemsets from datasets that showed a very large number of patterns. These processes can't make accurate decisions in a limited time, for example, real and online decision-making methods, since it is not an unimportant task to discover useful and appropriate information from an enormous quantity of the revealed information in a short time. Mining closed patterns with higher utilization (or termed closed higher-utility pattern mining) is another method to expose concise and smaller patterns with higher utilization in market engineering. Nevertheless, various previous works considered the comprehensive mining growth of all HUIs and they do not reflect the connection between transactions hence once the transactions are not very relevant, the trained method cannot be fully employed for the prediction, which indicates unsuitable results in machine learning (ML) tasks. This manuscript develops a new Deep Learning Based Mining High Utility Itemsets Using the Kepler Optimization Algorithm (DLMHUI-KOA) method. The major objective of the DLMHUI-KOA technique lies in the DL model and optimization parameter using mining high utility itemsets. Initially, the DLMHUI-KOA approach takes place when the design of the recurrent neural network (RNN) model is exploited. Next, the Gated Recurrent Unit (GRU) technique has been applied for predicting the MHUI. At last, the Kepler optimization algorithm (KOA) can be employed for the parameter tuning of the DL model. To determine the higher performance of the DLMHUI-KOA approach, a broad variety of experimentations occurs and the outcomes are examined under several measures such as number of candidates and high utility itemsets, memory usage, and runtime. The comparative analysis reported the improvement of the DLMHUI-KOA approach with recent methods.

Keywords: *High Utility Itemsets; Recurrent Neural Network; Kepler Optimization Algorithm; Gated Recurrent Unit; Memory Usage*

1. INTRODUCTION

Data analytics for larger data sets has developed as a new study field in existing years. With the exponential growth in data, it is most important to effectively mine this enormous data to obtain different perceptions of it [1]. Several methods such as association rule analysis, clustering, trend analysis, classification, and regression are applied to mining the data depending on the knowledge type we need for mining [2]. Association Rule Mining (ARM) is a significant data mining technology that can be applied to discover the association between various items that take place at the same time in a dataset. The rules are enclosed and then data can be mined according to them [3]. Frequent Itemset Mining (FIM) was the principal model for ARM. These patterns take place regularly within the dataset, and guidelines are made based on them. Nevertheless, it only comprises the patterns or associations that depend on the occurrence frequency [4]. It may find the itemsets have the least significance or utility and lose some of the least frequent itemsets but with a higher utilization feature.

The utility now is captured by the presence or absence of the item which is not an ample reason to imitate its significance [5]. For instance, selling a refrigerator is considerably more profitable than selling a glass bottle however a sequence having a refrigerator is much less common than the one taking a bottle. An added limitation of FIM is it believes an item once for all transactions. To discover several patterns in a database it is vital to mind another essential factor also along with the frequency like interestingness, weight, profit, and so forth [6]. These problems are directed by Weighted Association Rule Mining (WARM), which discovers the dataset by having notes of several types of weight or profit of an item [7]. Because of this, utility mining appears as a main subject in the data mining areas. Mining higher utility itemsets from datasets refers to the discovery of itemsets with higher gains [8]. The item's utility in a transacted dataset contains dual features: 1) the significance of different items that can be termed external utility, and 2) the significance of items within transactions that can be named internal utilities [9]. The item set utility is distinct by the product of their internal and external utility. This item can be designated as a higher-utility item when its utility cannot be lower than a user-specific minimal utility threshold; then, it is termed a lower-utility item [10].

This manuscript develops a new Deep Learning Based Mining High Utility Itemsets Using Kepler Optimization Algorithm (DLMHUI-KOA) model. The main objective of the DLMHUI-KOA technique lies in the DL model and optimization parameter using mining high utility itemsets. Initially, the DLMHUI-KOA approach takes place when the design of recurrent neural network (RNN) model is exploited. Next, the gated recurrent unit (GRU) technique is applied for predicting the MHUI. At last, the Kepler optimization algorithm (KOA) is applied for the hyperparameter tuning of the DL model. To depict the higher performance of the DLMHUI-KOA system, a extensive range of simulations take place and the outcomes are inspected under several measures such as number of candidates and high utility itemsets, memory usage, and runtime.

This study remaining sections are organized as follows: Section 2 discusses related work. Section 3 provides a detailed design and description of the proposed methodology. The experimental results and a detailed analysis are presented in Section 4. Finally, Section 5 concludes the study.

2. RELATED WORKS

Chen et al. [11] present a complete overview of the advanced iHAUIM methods, studying their distinctive advantages and features. Initially, the theory of iHAUIM has been described as giving formulations and real instances for a complete understanding. Consequently, the author discusses and classifies the important technologies utilized by changing kinds of iHAUIM methods, including Utility-list-based, Apriori-based, and Tree-based methods. Additionally, a crucial study of every mining technique's disadvantages and advantages is conducted. Finally, the author discovers possible upcoming directions, study offers, and iHAUIM method expansions. Sra and Chand [12] present the residual utility concepts for designing 2 novel data frameworks, named master-map and residue-map. Various experimentations have been executed on either synthetic or real datasets for equating the performance of R-Miner by the present list-based methods. Huynh et al. [13] introduce an effective techniques for MHUI-MUT. The paper utilizes upper limits and pruning strategy, consequently decreasing dataset scanning, and presents cut-off thresholds for reducing the time of mining. This method is presented to equate the system to create the highest performance of multi-core computers.

Bhikhabhai [14] presented a SCAO-based search space exploration method to decrease the number of similarities, which is essential to detect usual proceedings among the utility lists. Thus, it will reduce the joint operations cost. Furthermore, present advanced methods make excessive utility list join operations of the item sets, which are not HUIs. To tackle the issue, presented PUCP utilizes PUCS to remove excessive join operations. The presented technique utilizing the PUCP is named PUCP Miner. The performance of the presented methods has been assessed with the current methods such as mHUI-Miner, ULB miner, and HUI-Miner on some of the usual real-time datasets. Carstensen and Chun-Wei Lin [15] propose EC-based PSO methods for top-k HUIM which is named TKU-PSO. Furthermore, the author has advanced various tactics to sack the computing difficulty in the method. Initially, unnecessary and redundant candidate commutations have been evaded by using discovered solutions and computing item set utilities. Next, the discouraging items were pruned in the performance based on a threshold-raising conception called minimal solution fitness. Lastly, the conventional population

Journal of Theoretical and Applied Information Technology
15th October 2024. Vol.102. No. 19

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

initializing technique can be studied to enhance the method's capability to identify optimum solutions in enormous search spaces.

Tharini [16] develops a GA-combined BFA based on the HUIM method that is effective owing to its exploration and exploitation capability. This combination aims to tackle the problems of initial conjunction in item set mining, and CE performs as a genetic operator that improves the mining operation efficacy. Cheng et al. [17] proposed a P-EFIM method that depends on Hadoop platform to resolve this issue. In P-EFIM, the transaction weight employment values have been ordered and calculated for the item sets using the MapReduce method. Later the ordered item sets have been compensated, and the lower utility item sets have been pruned to enhance the dataset utility. It utilizes the presented S-style distribution tactic to dispense the subtasks consistently over every node to assure load balancing.

3. MATERIALS AND METHODS

In this manuscript, we have developed a new DLMHUI-KOA model. The main objective of the DLMHUI-KOA technique lies in the DL model and optimization parameter using mining high utility itemsets. Figure 1 shows the workflow of DLMHUI-KOA model.

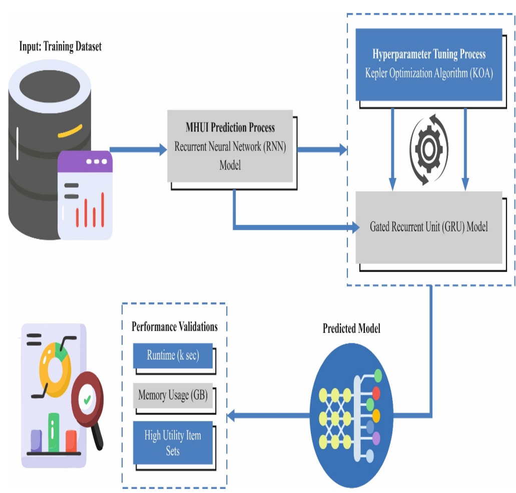


Figure 1: Workflow of DLMHUI-KOA model

3.1 Dataset

- Accidents: This dataset can be applied to improve traffic safety, predict accident hotspots, or develop machine learning models for accident severity prediction. In research, it helps in evaluating the effectiveness of public safety measures or automated driving systems.
- Chess: Used for training AI algorithms in strategic decision-making, this dataset aids in the development of reinforcement learning models for game theory or decision analysis. It also serves as a benchmark for complex problem-solving in AI research.
- Chain-store: This dataset provides insights into consumer behavior, inventory management, and sales forecasting, making it useful for developing algorithms that optimize supply chain operations or demand prediction models in retail research.
- Foodmart: Focused on retail transactions, this dataset supports market basket analysis, customer segmentation, and sales trend forecasting. It is ideal for building recommendation systems, optimizing pricing strategies, or evaluating promotional impacts in a retail context.

3.2. Design of RNN

Initially, the DLMHUI-KOA approach takes place when the design of RNN model is exploited. The RNN concept was born in the 1980s, inspired by the ANN requirement to manage sequential data [18]. Certainly, RNN has addressed dual main challenges intrinsic in these kinds of data: input information of changing lengths, and the fact is that preceding inputs can impact the forthcoming outputs and inputs. The forward pass expression for the utmost simple RNN method can be displayed in Eq. (1) (reminder that $tanh$ and sigmoid have been substitutable by any other activation function, and b and c represents the bias parameters).

$$a_t = b + Wh_{t-1} + Ux_t$$

$$h_t = tanh(a_t) \qquad (1)$$

$$o_t = c + Vh_t$$

$$\hat{y}_t = sigmoid(o_t)$$

Various forms of RNN are initiated in the article. Depending on the relationship of output and input existing in the data, here obtain various pattern designs: one to many, one to one, many to many, or many to one. Text classification or Sentiment analysis are the best instances of many-to-one forms, while machine transformation from many to many. The value of variables will also identify when the method can be multivariate or univariate. Additionally, the RNN original concept has been developed into more complex and sophisticated methods, like LSTM, Bi-directional RNN, or GRU. Despite the huge achievement attained by RNN in different applications without any drawbacks. They generally depend on backpropagation method that becomes more complicated in RNN and the issue sources like exploding and vanishing gradients. Furthermore, RNNs do not function well while creating the predictions outside the training data area, as other ANNs do. This issue can be intensified in individual multi-step fast predicting, wherever the predicted recent time step value can be utilized to identify the next time step value. Whereas that problem is alleviated in changing degrees, such as utilizing LSTM for gradient associate difficulties.

3.3. GRU Model

Next, the GRU technique is implemented to predict the MHUI. The GRU is a sophisticated RNN method. It is the simplest procedure of LSTM [19]. It is a gating approach applied to handle serial data and control the flow of data over a gating framework. Either GRU or LSTM are utilized to find longer-term dependence in sequence data. It is equivalent to LSTM with smaller gates, including the update gate (joins forget and input gates) and the reset gate, while LSTM contains 3 gates: the input, output, and forget gates. The update gate defines how prior data is needed to move to the following gate. The reset gate can be employed to define the crucial data from the preceding condition and decide several previous pieces of data to forget. Then, this method can recollect valuable data preventing the forgetting threat important details. This method is applied in various domains that perform classification of sequence tasks, for sample, spam detection, sentiment analysis, and subject classification. This method has smaller parameters and gates in comparison to the LSTM approach, so it is normally faster and easier to train. This model applied in these studies involves one input and output layers, and dual hidden layers (HL). This process can be measured as an RNN type, and the input layer size is based on dual factors series length and feature counts. In this regard, the length of the sequence is equivalent to 10 and the feature count is 5. The length of the sequence denotes the previous examination numbers (data points) that the method can be applied for predicting at some assumed time. The mathematical problem of the function of SoftMax is presented in Eq. (2) whereas e denotes a continual integer that is nearly equivalent to 2.71828 and z_i represent logit, and N stands for the overall class count.

$$Softmax(z_i) = \frac{e^{zi}}{\sum_{j=1}^{N} e^{zj}} \tag{2}$$

x_t indicates the input χ by present time t and h_{t-1} denotes the output of the preceding HL at time $t-1$. These reset gates have been mathematically demonstrated in Eq. (3):

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{3}$$

Here r_t signifies the variable of resetting gates at $t\ th$ time. A gate of rest can be employed to decide the previous HL number that is required to be forgotten. σ stands for the sigmoid functions, which normalized the input to range between $zero$ and one. W_r refers to the weight combined using the gate of rest, and b_r denotes the biased term of the gate of reset.

The updated gate can be offered in Eq. (4) in the following:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{4}$$

Now z_t mention the variable of updated gates at $t\ th$ time, σ denotes sigmoid functions, h_{t-1} represents the output of the preceding HL at time t-1, and x_t stands for the input of the present time, W_z symbolizes weight related to the updated gate, and b_z characteristics biased term of the updated gates.

The candidate HL can be considered in Eq. (5):

$$\overline{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h) \tag{5}$$

The above-mentioned equations compute a candidate for the novel HL whereas W_h denotes the weight. Once r_t is 0, the outcome of $r_t * h_{t-1}$ is 0, which specifies that the preceding condition should be neglected, and this method will be concentrated on the unique position. b_h denotes the biased term of the candidate's HL. Lastly, Eq. (6) computes the end activation. Figure 2 represents the infrastructure of GRU.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \overline{h}_t \tag{6}$$

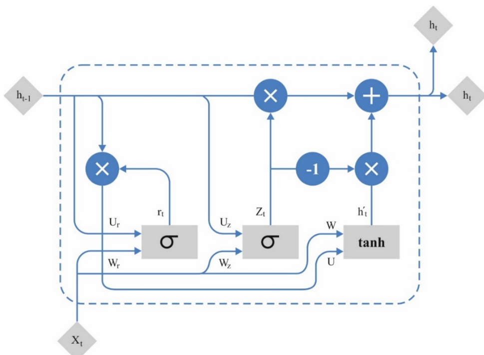


Figure 2: Architecture of GRU

3.4. KOA-based Parameter Tuning

At last, the KOA is applied for the hyperparameter tuning of the DL model. In recent times, a novel meta-heuristic-based method named KOA has been developed to challenge nonstop optimizer issues [20]. This kind of algorithm was enthused by Kepler's law of planetary motion. This law has displayed that 4 factors can able to manage the track of the planet. These factors have been signified in the gravitational force, location, mass, and orbit speed. KOA reflects every planet as a solution for the candidate. Throughout the planet's motion, it might evaluate a novel result for the optimizer issue. Those planets near the sun enlarge their exploitation abilities, whereas others improve their exploration abilities. In the short term, similar to the other meta-heuristic techniques, KOA is dependent upon dual operators such as exploration and exploitation. The KOA mathematical method is discussed below.

Step 1: Initialize stage

The KOA allocates N planets within the space of search, whereas every planet is included of d sizes. The formulation is employed to allocate the planets is expressed as below:

$$X_i^j = X_{i,low}^j + r \times (X_{i,up}^j - X_{i,low}^j), \begin{cases} i = 1,2, \dots, N \\ j = 1,2, \dots, d' \end{cases} \quad (7)$$

Here, X_j denotes to the ith planet; $X_{i,low}^j$ and $X_{i,up}^j$ specifies the search bounds. r is an amount nominated at random in the range of $(0 \; and \; 1)$. Furthermore, KOA has few parameters such as orbital period (T) and orbital eccentricity (e), which want to be set before beginning the optimizer procedure. The orbital eccentricity (e) can be set by:

$$e_i = r, i = 1, \dots, N \quad (8)$$

Lastly, T for every $it\,h$ planet can be initialized by:

$$T_i = |rn|, i = 1, \dots, N, \quad (9)$$

Whereas, rn denotes the number nominated at random as per the normal distribution.

Step 2: Define F

The planet's indications around the Sun have been directed by gravity which is measured as the essential force.

$$F_{gj}(t) = e_i \times \mu(t) \times \frac{\overline{M}_s \times \overline{m}_i}{\overline{R}_i^{\,2} + \varepsilon} + r_1, \quad (10)$$

Whereas, ε covers the smallest value to evade division by 0, r_1 is a variable with values produced at random among 0 and 1. $\overline{M}_s \; and \; \overline{m}_i$ specifies the mass regularized values of both $X_S \; and \; X_i$, correspondingly; the mass M_s and m_i of $X_S \; and \; X_i$ are correspondingly calculated as per Eqs. (13) and (14); e_i specifies the eccentricity of the orbit of a planet; μ denotes the stable value to signify the complete gravitational constant; $\overline{R}_i$ denotes the regularized value of Euclidian distance amid X_i and X_S, which can be calculated by:

$$R_i(t) = \|X_S(t) - X_j(t)\|_2 = \sqrt{\sum_{j=1}^{d} (\vec{X}_{Sj}(t) - \vec{X}_{ij}(t))^2}, \quad (11)$$

$$\overline{R}_i = \frac{R_i(t) - \min(R(t))}{\max(R(t)) - \min(R(t))'} \quad (12)$$

$$M_s = r_2 \frac{it_s(t) - worst(t)}{\sum_{k=1}^{N}(fit_k(t) - worst(t))'} \quad (13)$$

$$m_i = \frac{fit_i(t) - worst(t)}{\sum_{k=1}^{N}(fit_k(t) - worst(t))'} \quad (14)$$

where

$$t_s(t) = best(t) = \min_{k\in\{1,2,\dots N\}} fit_k(t), \tag{15}$$

$$worst(t) = \max_{k\in\{1,2,\dots N\}} fit_k(t), \tag{16}$$

Here, M_s signifies the mass of X_S; m_i signifies the mass of X_j; r_2 denotes the number selected randomly in the range $(0 \; and \; 1)$; Where $\mu(t)$ is expressed by:

$$\mu(t) = \mu_0 \times \exp\left(-\gamma \frac{t}{T_{\max}}\right), \tag{17}$$

Whereas γ is a stable value; μ_0 contains an initial value; and t and $T_{\max}$ signify the existing and maximum function estimations, correspondingly.

Step 3: Computing an object's speed

A planet's velocity is intended by its distance from the Sun. It is simulated utilizing the below-mentioned calculated method:

$$V_i(t) =$$

$$\begin{cases} \ell \times \left(2r_4\vec{X} - \vec{X}\right) + \ell \times \left(\vec{X}a - \vec{X}\right) + \left(1 - R_j\text{-}norm(t)\right) \times \mathcal{F} \times \vec{U} \times \vec{r_5} \times \left(\overrightarrow{X_{i,up}} - \vec{X}_{i,low}\right) \\ \quad if\ R_{i-norm}(t) \le 0.5, Else \\ r_4 \times L \times \left(\vec{X}a - \vec{X}\right) + \left(1 - R_{i-norm}(t)\right) \times \mathcal{F} \times U_2 \times \vec{r_5} \times \left(r_3\vec{X} - \vec{X}\right) \end{cases} \tag{18}$$

$$\ell = \vec{U} \times \mathcal{M} \times \mathcal{L}, \tag{19}$$

$$\mathcal{L} = \left[\mu(t) \times (M_S + m_i)\left(\frac{2}{R_i(t) + \varepsilon} - \frac{1}{a_{i(t)} + \varepsilon}\right)\right]^{\frac{1}{2}}, \tag{20}$$

$$\mathcal{M} = (r_3 \times (1 - r_4) + r_4), \tag{21}$$

$$\vec{U} = \begin{cases} 0 \, \vec{r_5} \le \vec{r_6} \\ 1 \; Else \end{cases}, \tag{22}$$

$$\mathcal{F} = \begin{cases} 1 \; if \; r_4 \le 0.5 \\ -1 \; Else \end{cases} \tag{23}$$

$$\ddot{\ell} = (1 - \vec{U}) \times \vec{M} \times \mathcal{L} \tag{24}$$

$$\vec{M} = (r_3 \times (1 - \vec{r_5}) + \vec{r_5}), \tag{25}$$

$$\vec{U}_1 = \begin{cases} 0 \; \vec{r_5} \le r_4 \\ 1 \; Else \end{cases}, \tag{26}$$

$$U_2 = \begin{cases} 0 \; r_3 \le r_4 \\ 1 \; Else \end{cases}, \tag{27}$$

Here, $\vec{V_i}(t)$ denotes the ith velocity of objects, r_3 and r_4 specifies the dual numbers nominated randomly in the range of $(0 \; and \; 1)$; and $\vec{r_5}$ and $\vec{r_6}$ are dual vectors, with decimal numbers designated randomly in the range $(0 \; and \; 1)$. $\vec{X}_a$ and $\vec{X}_b$ are dual solutions chosen arbitrarily from the existing solutions; M_s and m_j denotes the mass of $\vec{X}_s \; and \; \vec{X}_i$, correspondingly; and a_i refers to the semi-major axis of the ith object's elliptical orbit. $\mathcal{F}$ contains any 1 or -1, designated randomly to modify the search way. a_i is formulated by:

$$a_i(t) = r_3 \times \left[T_i^2 \times \frac{\mu(t) \times (M_s + m_i)}{4\pi^2}\right]^{\frac{1}{3}} \tag{28}$$

Step 4: Escape from the local optimal

In the solar system, many objects switch on their axes and orbit anti-clockwise around the Sun; but, there are some exclusions. This behavior has been used by KOA to flight local optimum areas by substituting the search way at consistent ranges with the aid of flag F.

Step 5: Upgrade the objects' locations

The planets move in their elliptical paths around the Sun. KOA pretends this behavior in dual phases namely exploitation and exploration. The operator of exploration is pretend in KOA if the planets are far away, whereas the operator of utilization is attained when the planets are nearer.

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \mathcal{F} \times \vec{V}_i(t) + \left(F_{g_i}(t) + |r|\right) \times \vec{U} \times \left(\vec{X}_s(t) - \vec{X}_i(t)\right). \tag{29}$$

Eq. (29) pretends the Sun's gravitational force, whereas this calculation utilizes an extra step size dependent upon computing the reserve between the Sun and the existing planet which is increased by the gravitational force to help KOA in discovering the areas nearby the effective result and discovering superior solution in less task estimations.

Step 6: Update the distance with the Sun

In a further determination to increase the KOA's exploration abilities, the usual variant in the reserve

Journal of Theoretical and Applied Information Technology
15th October 2024. Vol.102. No. 19

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

to the Sun and planets over time has been pretend. When planets are nearer to the Sun, the operator of exploitation is initiated to improve the rate of convergence, while the Sun is distant, then the operative of exploration is motivated to reduce receiving fixed in local goals.

$$\vec{X}_i(t+1) = \vec{X}_i(t) \times \vec{U}_i + \left(1 - \vec{U}_1\right)$$
$$\times \left(\frac{\vec{X}_i(t) + \vec{X}_s + \vec{X}_a(t)}{3.0} + h \times \left(\frac{\vec{X}_i(t) + \vec{X}_s + \vec{X}_a(t)}{3.0} - \vec{X}_b(t) \right) \right), \quad (30)$$

$$h = \frac{1}{e^{\eta r}}, \quad (31)$$

Where r denotes the number nominated at random in the interval $(0 \ and \ 1)$, while η is defined as:

$$\eta = (a_2 - 1) \times r_4 + 1, \quad (32)$$

Here, a_2 signifies a cyclic control parameter defined as:

$$a_2 = -1 - \left(\frac{t \% \frac{T_{\max}}{TC}}{\frac{T_{\max}}{TC}} \right), \quad (33)$$

Whereas TC denotes the number of cycles, and $\%$ refers to the modulo operator.

Step 7: Elitism

This stage performs a superior plan to guarantee that the Sun and planets have been constantly in the local finest locations. The mathematical formulation is given below

$$\vec{X}_{i,new}(t+1) = \begin{cases} \vec{X}_i(t+1), & if \ f(\vec{X}_i(t+1)) \le f\left(\vec{X}_i(t)\right) \\ \vec{X}_i(t), & Else \end{cases}. \quad (34)$$

In this article, the KOA can be applied to define the hyperparameter included in the GRU technique. The candidate counts and HUIs, memory usage, and runtime are considered the objective function.

4. RESULT ANALYSIS AND DISCUSSION

In this work, the performance validation analysis of the DLMHUI-KOA model can be examined using dataset. Table 1 represents the detailed description of the dataset.

Table 1: Details of dataset.

Parameters	Dataset Details			
	Accidents	Chess	Chain-store	Food mart
Total Number of Transaction (|D|)	340183	3196	1112949	4141
Average Transaction Length T	33.8	37	7.2	4.4
Number of Distinct items (|I|)	468	75	46086	1559
Type	Dense	Dense	Sparse	Sparse

The dataset details for the Accidents [21], Chess [22], Chain-store [23], and Foodmart [24] datasets provide key insights into their structure and scale. The total number of transactions (|D|) varies significantly across these datasets, with Chain-store being the largest at 1,112,949 transactions, followed by Accidents with 340,183 transactions, Foodmart with 4,141, and Chess with 3,196. The average transaction length (T) is notably longer in Chess at 37, followed closely by Accidents at 33.8. Chain-store has a moderate transaction length of 7.2, while Foodmart's transactions are the shortest with an average of 4.4. In terms of distinct items (|I|), Chain-store has an overwhelmingly high count of 46,086 distinct items, whereas Accidents, Foodmart, and Chess feature 468, 1,559, and 75 distinct items, respectively. These variations highlight the diverse nature of the datasets, with differences in transaction sizes, lengths, and item diversity.

4.1 Applications of the dataset :

Accidents Dataset: This dataset can be applied to identify frequent patterns in accident occurrences, high-risk factors, and correlations between certain conditions (e.g., weather, time, or location) and accident rates. Using DLMHUI-KOA, it could efficiently mine critical high-utility patterns that impact accident rates, offering insights to improve traffic safety policies.

Chess Dataset: In this scenario, the utility might be defined by the probability of certain moves leading to victory. Mining high-utility itemsets could reveal common move sequences or strategies

Journal of Theoretical and Applied Information Technology
15th October 2024. Vol.102. No. 19

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

that consistently offer high chances of success, providing competitive insights for strategy optimization.

Chain-store Dataset: This dataset could be used to find high-utility itemsets that represent combinations of items frequently purchased together in a retail environment, but with a focus on the profitability of the combinations. DLMHUI-KOA would identify high-profit patterns, optimizing inventory and marketing strategies for store chains.

Foodmart Dataset: In the context of grocery or retail data, the model can be used to mine high-utility itemsets that reflect combinations of products purchased by consumers that yield the highest profits. Such insights would help improve product placement, promotions, and inventory management, enhancing store profitability.

Table 2 defines the overall comparative runtime analysis of the DLMHUI-KOA model with existing approaches under Accidents (Dense) and Chess (Dense) datasets [25].

Figure 3 illustrates the runtime outcome of DLMHUI-KOA model under Accidents (Dense) dataset with various Min_util. The results inferred that the DLMHUI-KOA approach has exhibited better performances compared to other approaches. For instance, on 30% of Min_util, the DLMHUI-KOA model has lower runtime of 8.07k sec while the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ approaches have higher runtime of 11.66k sec, 10.88k sec, 10.36k sec, and 9.6k sec, respectively. In addition, on 70% of Min_util, the DLMHUI-KOA method has minimum runtime of 8.13k sec whereas the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ techniques have greater runtime of 11.73k sec, 11.21k sec, 10.56k sec, and 9.79k sec, correspondingly.

Figure 4 indicates the runtime result of DLMHUI-KOA approach under Chess (Dense) dataset with several Min_util. The outcomes gathered that the DLMHUI-KOA model has shown superior performances in comparison with other techniques. For example, on 30% of Min_util, the DLMHUI-KOA system has least runtime of 9.51k sec but the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ methods have better runtime of 13.54k sec, 12.75k sec, 11.94k sec, and 11.15k sec, correspondingly. Additionally, on 70% of Min_util, the DLMHUI-KOA model has lesser runtime of 8.90k sec however the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ methodology have maximum runtime of 12.72k sec, 12.01k sec, 11.31k sec, and 10.47k sec, individually.

Table 2: Runtime outcome of the DLMHUI-KOA model with existing approaches under two datasets.

Runtime (k sec)					
Accidents (Dense) Dataset					
Min_util (%)	IHUPT-FPG	UPT-FPG	UPT-UPG	UPT-UPG+	DLMHUI-KOA
30	11.66	10.88	10.36	9.6	8.07
40	10.45	9.76	9.25	8.47	6.87
50	8.32	7.81	7.1	6.51	4.77
60	9.64	8.92	8.27	7.77	5.97
70	11.73	11.21	10.56	9.79	8.13
Chess (Dense) Dataset					
30	13.54	12.75	11.94	11.15	9.51
40	12.31	11.46	10.62	9.83	8.09
50	12.55	11.75	11.03	10.18	8.52
60	13.02	12.24	11.34	10.55	8.88
70	12.72	12.01	11.31	10.47	8.90

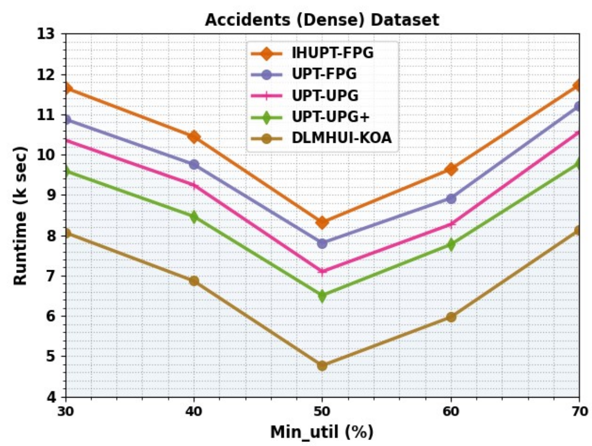
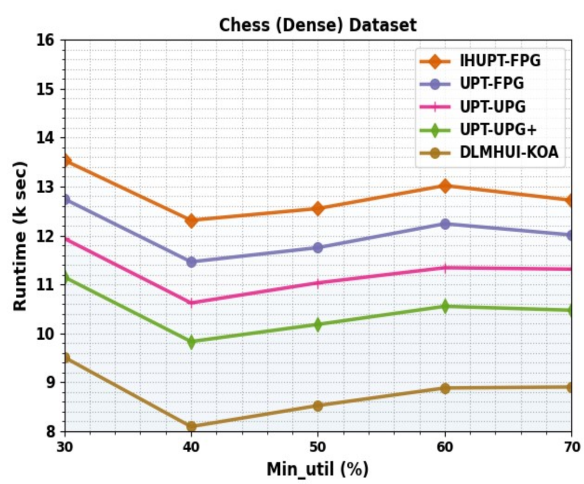


Figure 3: Runtime outcome of the DLMHUI-KOA model under Accidents (Dense) dataset



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

Figure 4: Runtime outcome of the DLMHUI-KOA model under Chess (Dense) dataset

Table 3 describes the complete comparison runtime study of the DLMHUI-KOA process with recent techniques under Chain-store (Sparse) and Foodmart (Sparse) datasets. Figure 5 demonstrates the runtime result of DLMHUI-KOA technique under Chain-store (Sparse) dataset with different Min_util. The outcomes implied that the DLMHUI-KOA model has presented superior performances in comparison with another methodology. For sample, on 30% of Min_util, the DLMHUI-KOA system has minimum runtime of 11.25k sec whereas the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ methods have greater runtime of 19.22k sec, 17.61k sec, 16.06k sec, and 14.55k sec, appropriately. Moreover, on 70% of Min_util, the DLMHUI-KOA system has least runtime of 10.93k sec however the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ methodology have better runtime of 18.82k sec, 17.04k sec, 15.37k sec, and 13.72k sec, correspondingly.

Table 3: Runtime outcome of the DLMHUI-KOA model with existing approaches under two datasets.

Runtime (k sec)					
Chain-store (Sparse) Dataset					
Min_util (%)	IHUPT-FPG	UPT-FPG	UPT-UPG	UPT-UPG+	DLMHUI-KOA
30	19.22	17.61	16.06	14.55	11.25
40	18.48	16.69	14.90	13.35	10.64
50	16.47	14.79	13.00	11.44	8.24
60	16.05	14.49	12.77	11.20	8.13
70	18.82	17.04	15.37	13.72	10.93
Foodmart (Sparse) Dataset					
30	25.40	23.86	22.30	20.74	17.08
40	29.75	28.16	26.63	25.11	22.47
50	28.80	27.27	25.49	23.96	21.42
60	28.99	27.49	25.88	24.35	21.04
70	27.38	25.76	24.15	22.42	19.62

Figure 6 portrayed the runtime result of DLMHUI-KOA technique under Foodmart (Sparse) dataset with different Min_util. The outcomes imply that the DLMHUI-KOA model has demonstrated greater performances compared with alternate methods. For example, on 30% of Min_util, the DLMHUI-KOA system has reduced runtime of 17.08k sec but the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ methodology have supreme runtime of 25.40k sec, 23.86k sec, 22.30k sec, and 20.74k sec, individually. Furthermore, on 70% of Min_util, the DLMHUI-KOA system has least runtime of 19.62k sec however the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ process have superior runtime of 27.38k sec, 25.76k sec, 24.15k sec, and 22.42k sec, correspondingly.

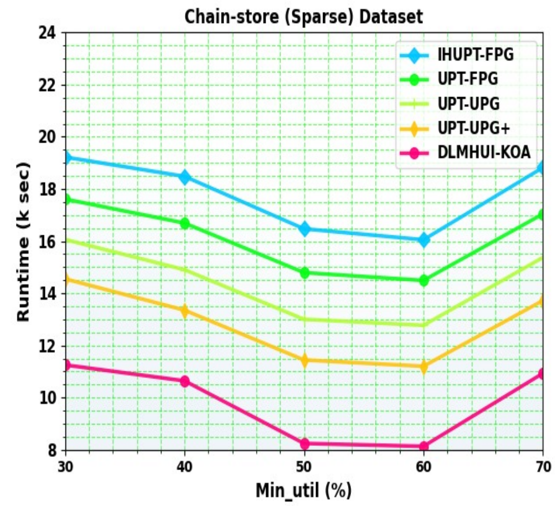


Figure 5: Runtime outcome of the DLMHUI-KOA model under Chain-store (Sparse) dataset

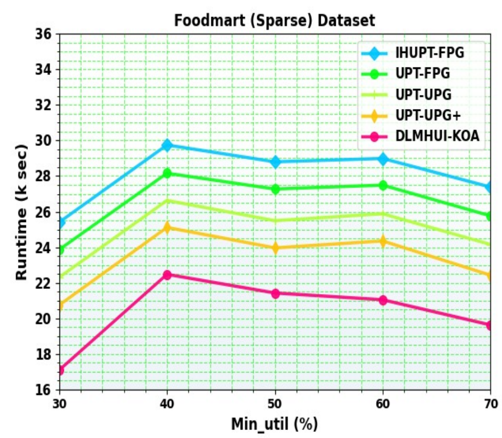


Figure 6: Runtime outcome of the DLMHUI-KOA model under Foodmart (Sparse) dataset

In Table 4 and Figure 7, the high utility itemset (HUI) examination of the DLMHUI-KOA model with existing methods under various databases. The table values stated that the DLMHUI-KOA algorithm has optimum solution compared to existing methods. For instance, on 200k database with an HUI is 8175, the DLMHUI-KOA model obtained HUI of 34171 while the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ approaches gained HUI of 70976, 60254, 53844, and 36292, respectively. Followed by, on 1000k database and HUI is 7144, the DLMHUI-KOA technique gained HUI of 32236 whereas the IHUPT-FPG, UPT-FPG, UPT-UPG, and UPT-UPG+ methodology obtained

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

HUI of 68920, 58160, 52073, and 36104, correspondingly.

Table 4: HUI analysis of DLMHUI-KOA model with existing methods under various databases.

High Utility Item sets						
Database	IHUPT-FPG	UPT-FPG	UPT-UPG	UPT-UPG+	DLMHUI-KOA	#HUIs
200k	70976	60254	53844	36292	34171	8175
400k	74551	64560	58050	40556	37415	8976
600k	69015	59918	53829	37169	35081	7324
800k	68088	59528	53491	38887	35009	8188
1000k	68920	58160	52073	36104	32236	7144

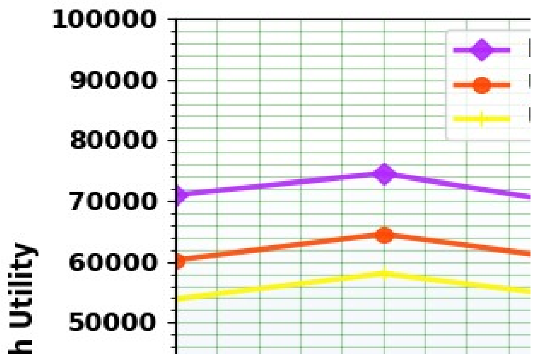


Figure 7: HUI analysis of DLMHUI-KOA model under various databases

In Table 5 and Figure 8, the memory usage (MU) analysis of the DLMHUI-KOA method with recent models under numerous databases. The table values indicated that the DLMHUI-KOA approach has optimal solution in comparison to recent techniques. For cases, on 200k database, the DLMHUI-KOA methodology attained least MU of 0.927GB whereas the IHUPT&FPG, UPT&UPG, and UPT&UPG+ models reached greater MU of 1.324GB, 1.128GB, and 1.079GB, correspondingly. Accompanied by, on 1000k database, the DLMHUI-KOA system acquired less MU of 1.569GB however the IHUPT&FPG, UPT&UPG, and UPT&UPG+ methodologies obtained better MU of 1.693GB, 1.648GB, and 1.837GB, correspondingly.

Table 5: MU analysis of DLMHUI-KOA model with existing methods under various databases.

Memory Usage (GB)				
Database	IHUPT&FPG	UPT&UPG	UPT&UPG+	DLMHUI-KOA
200k	1.324	1.128	1.079	0.927
400k	1.400	1.279	1.285	1.108
600k	1.490	1.404	1.452	1.274
800k	1.585	1.522	1.626	1.465
1000k	1.693	1.648	1.837	1.569

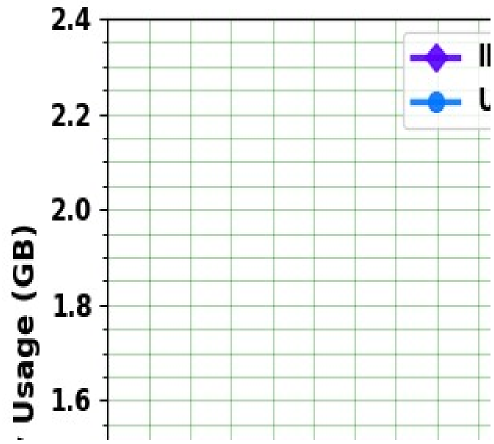


Figure 8: MU analysis of DLMHUI-KOA model under various databases

5. CONCLUSION

In this manuscript, we have advanced a novel DLMHUI-KOA technique. The major goal of the DLMHUI-KOA technique lies in the DL model and optimization parameter using mining high utility itemsets. Initially, the DLMHUI-KOA approach takes place when the design of the RNN model is exploited. Next, the GRU technique is implemented to predict the MHUI. At last, the KOA has been utilized for the hyperparameter tuning of the DL approach. To determine the better performance of the DLMHUI-KOA method, a broad variety of experimentations occurs and the outcomes are examined under several measures such as several candidates and higher utility itemsets, memory usage, and runtime. The comparative analysis reported the improvement of the DLMHUI-KOA approach over recent methods.

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".

REFERENCES:

[1] M. Y. Eltabakh, M. Ouzzani, M. A. Khalil, W. G. Aref, & A. K. Elmagarmid, "Incremental

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

mining for frequent patterns in evolving time series datatabases", 2008.

[2] A. Erwin, R. P. Gopalan, & N. R. Achuthan, "Efficient mining of high utility itemsets from large datasets", *In Advances in Knowledge Discovery and Data Mining: 12th Pacific-Asia Conference, PAKDD 2008 Osaka, Japan, May 20-23, 2008 Proceedings,* vol. 12, pp. 554-561, 2008. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-68125-0_50

[3] E. Georgii, L. Richter, U. Rückert, & S. Kramer, "Analyzing microarray data using quantitative association rules", *Bioinformatics*, vol. 21, 2005. https://doi.org/10.1093/bioinformatics/bti1121

[4] V. S. Tseng, C. W. Wu, B. E. Shie, & P. S. Yu, "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, July 2010. https://doi.org/10.1145/1835804.1835839

[5] C. X. Meng, "An efficient algorithm for mining frequent patterns over high speed data streams", *In 2009 WRI World Congress on Software Engineering IEEE,* vol. 1, pp. 319-323, May 2009. https://doi.org/10.1109/WCSE.2009.153

[6] J. D. Ren, H. L. He, C. Z. Hu, L. N. Xu, & L. B. Wang, "Mining frequent pattern based on fading factor in data streams", *In 2009 International Conference on Machine Learning and Cybernetics IEEE,* vol. 4, pp. 2250-2254, July 2009. https://doi.org/10.1109/ICMLC.2009.5212115

[7] B. E. Shie, V. S. Tseng, & P. S. Yu, "Online mining of temporal maximal utility itemsets from data streams", *In Proceedings of the 2010 ACM Symposium on Applied Computing,* pp. 1622-1626, March 2010. https://doi.org/10.1145/1774088.1774436

[8] K. Sun, & F. Bai, "Mining weighted association rules without preassigned weights", *IEEE transactions on knowledge and data engineering*, vol. 20, no. 4, pp. 489-495, 2008. https://doi.org/10.1109/TKDE.2007.190723

[9] D. Paulraj, T. Sethukarasi, & E. Baburaj, "An Efficient Hybrid Job Scheduling Optimization (EHJSO) approach to enhance resource search using Cuckoo and Grey Wolf Job Optimization for cloud environment". *Plos one*, vol. *18, no.* 3, pp. 1-20, 2023. https://doi.org/10.1371/journal.pone.0282600

[10] M. Xie, & L. Tan, "An efficient algorithm for frequent pattern mining over uncertain data stream", *In 2019 12th International Symposium on Computational Intelligence and Design (ISCID), IEEE,* vol. 1, pp. 84-88, December 2019. https://doi.org/10.1109/ISCID.2019.00026

[11] J. Chen, S. Yang, W. Ding, P. Li, A. Liu, H. Zhang, & T. Li, "Incremental high average-utility itemset mining: survey and challenges," *Scientific Reports*, vol. 14, no. 1, pp. 9924, 2024. https://doi.org/10.1038/s41598-024-60279-0

[12] P. Sra, & S. Chand, "A residual utility-based concept for high-utility itemset mining., *Knowledge and Information Systems*, vol. 66, no. 1, pp. 211-235, 2024. https://doi.org/10.1007/s10115-023-01948-w

[13] B. Huynh, N. T. Tung, T. D. Nguyen, C. Trinh, V. Snasel, & L. Nguyen, "New approaches for mining high utility itemsets with multiple utility thresholds," *Applied Intelligence*, vol. 54, no. 1, pp. 767-790, 2024. https://doi.org/10.1007/s10489-023-05145-8

[14] P. S. Bhikhabhai, "An Efficient Approach for High Utility Itemset Mining,"

[15] S. Carstensen, & J. Chun-Wei Lin, "Tku-pso: an efficient particle swarm optimization model for top-k high-utility itemset mining", 2024. https://doi.org/10.9781/ijimai.2024.01.002

[16] V. J. Tharini, "Cross-Entropy Assisted Optimization Technique for High Utility Itemset Mining from the Transactional Database", *Communications on Applied Nonlinear Analysis*, vol. 31, no. 3s, pp. 90-104, 2024. https://doi.org/10.52783/cana.v31.734

[17] Z. Cheng, W. Shen, W. Fang, & J. C. W. Lin, "A parallel high-utility itemset mining algorithm based on Hadoop", *Complex System Modeling and Simulation*, vol. 3, no. 1, pp. 47-58, 2023. https://doi.org/10.23919/CSMS.2022.0023

[18] J. Fernández, J. Chiachío, J. Barros, M. Chiachío, & C. S. Kulkarni, "Physics-guided recurrent neural network trained with approximate Bayesian computation: A case study on structural response prognostics", *Reliability Engineering & System Safety*, vol. 243, pp. 109822, 2024. https://doi.org/10.1016/j.ress.2023.109822

[19] A. A. Alsulami, Q. Abu Al-Haija, B. Alturki, A. Alqahtani, F. Binzagr, B. Alghamdi, & R. A. Alsemmeari, "Exploring the efficacy of GRU model in classifying the signal to noise ratio of microgrid model," *Scientific Reports*, vol. 14, no. 1, pp. 15591, 2024. https://doi.org/10.1038/s41598-024-66387-1

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

[20] R. Mohamed, M. Abdel-Basset, K. M. Sallam, I. M. Hezam, A. M. Alshamrani, & I. A. Hameed, "Novel hybrid kepler optimization algorithm for parameter estimation of photovoltaic modules", *Scientific Reports*, vol. 14, no. 1, pp. 3453, 2024. https://doi.org/10.1038/s41598-024-52416-6

[21] https://www.kaggle.com/datasets/saurabhshahane/road-traffic-accidents

[22] https://www.kaggle.com/code/gabrielhaselhurst/chess-dataset

[23] https://www.kaggle.com/datasets/aishwaryadeb/chain-store-dataset

[24] https://www.kaggle.com/datasets/dermisfit/foodmart-dataset

[25] V. S. Tseng, B. E. Shie, C. W. Wu, & S. Y. Philip, "Efficient algorithms for mining high utility itemsets from transactional databases", *IEEE transactions on knowledge and data engineering*, vol. 25, no. 8, pp. 1772-1786, 2012. https://doi.org/10.1109/TKDE.2012.59