Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.5, May : 2024

BUCKLIN BOOTSTRAP AGGREGATED CANONICAL CORRELATIVE ASSOCIATION RULE CLASSIFICATION FOR EXTRACTING HIGH FREQUENT AND UTILITY ITEMSETS

R. Savitha Research Scholar, Government Arts College (Autonomous), Karur, savithabca@gmail.com
Dr V. BabyDeepa Research Advisor, Government Arts College (Autonomous), Karur, deepamct @gmail.com

ABSTRACT

This paper presents a Bucklin Bootstrap Aggregated Canonical Correlative Association Rule Classification (BBACCARC) model in order to raise the performance of mining the high frequent and utility itemsets in larger transactional dataset. On the contrary to conventional system, BBACCARC model is bagging classifier where it fits base classifiers each on arbitrary subsets of the original dataset and subsequently combined their individual predictions with the help of Bucklin voting concept to acquire a final prediction results. The proposed BBACCARC model is an ensemble method which supports to resolve over fitting problem during the classification process. First, BBACCARC model constructs 'n' number of base (i.e., Canonical Correlative Association Rule Based Classification (CCAR-C) result. Subsequently, BBACCARC model aggregates all base CCAR-C output and then applies Bucklin majority voting concepts and thus obtained the strong classifier result. The strong classifier in BBACCARC model precisely classifies each itemsets in big dataset as HFI and HUI with minimal computational time. From the classified outcomes, BBACCARC model efficiently mines the itemsets which are the more users' interested and maximum profitable from larger dataset with better accuracy. The BBACCARC model is implemented in java language using metrics such as accuracy, time complexity, and error rate.

Keywords:
Bucklin Vote, Canonical Association Rule, Correlation, Frequent Itemsets, Strong Classifier, Utility Itemsets

I. INTRODUCTION

Data mining is a procedure of finding valuable information. Frequent itemsets mining predicts the repeated item combinations. High-utility itemsets mining detect repetitive item combinations which include a high-profit value. The state-of-the-art techniques were implemented for extracting the HFI and HUI was not provided better accuracy results while taking a larger number of input itemsets.

For instance an improved binary PSO was designed in [1] with the target of discovering the high utility itemsets. Though, accuracy was observed to find the high utility itemsets in big dataset was deficient. An improved LHUI concept named HLHUI (Hminer-based Local HUI mining) was presented in [2] to identify the useful itemsets with better accuracy. But, the top most user interest itemsets was not extracted in HLHUI with minimal complexity.

In [3], various algorithms were implemented for predicting the high utility patterns was analyzed with their merits and demerits. GPU-based efficient parallel heuristic algorithm for HUIM (PHA-HUIM) was designed in [4] with the aiming at providing the solutions to find high-utility itemset. The conventional PHA-HUIM obtained better speedup performance, runtime, and mining quality. However, the false rate of extracting the HUI while getting the bigger size of transactional database was higher.

A Multi-Core Approach was introduced in [5] to significantly discover the high-utility itemsets in dynamic databases with minimal time consumption. But, the mining efficiency of this existing approach was not better when considering larger database. A Full Compression Frequent Pattern Tree

(FCFP-Tree) was constructed in [6] for presenting a solution for incremental frequent itemsets extraction. However, predicting the high-utility itemsets with lesser time complexity was remained open issue.

Fuzzy association rule-based classification method was developed in [7] to acquire an accurate and better computational cost result. Though, the accuracy using this classification algorithm was unsatisfactory. Parallel efficient high-utility itemset mining (P-EFIM) algorithm was implemented in [8] by using Hadoop platform to handle big database. But, computational complexity using P-EFIM was more.

To handle the above said issues during the discovery and extraction of the maximum user's interested and profited itemsets while taking a big transactional database as input, a new BBACCARC model is implemented.

The key roles of proposed BBACCARC model is discussed as follows,

- ✓ To reach an improved accuracy to discover the HFI and HUI through classification while getting larger number of items as input, Bucklin Bootstrap Aggregation (BBA) concepts is applied in BBACCARC model.
- ✓ To achieve the computational complexity during the accurate extraction of the most users' favorite and profitable itemsets in big database, Canonical Association Rule (CAR) is developed in BBACCARC model. On the contrary to state-of-the-art research works, CAR supports for BBACCARC model to determine the relation between multiple independent and dependent itemsets in given dataset.
- ✓ To gain minimal error rate during the mining of HFI and HUI from the enormous size of database, the majority votes of all base CCAR-C outcomes are taken as strong classifier in BBACCARC model using Bucklin voting concept.

The rest of paper is formulated as follows. Section II describes the literature survey. In Section III, the proposed BBACCARC model is detailed explained with aid of an architecture illustration. Section IV illustrates the experimental settings and result of proposed BBACCARC model. Finally, the paper concluded in section V.

II. LITERATURE STUDY

Single scan approach was planned in [9] to perform frequent itemset discovery process with minimal runtime for large databases. However, the extraction of high-profitable items was not focused. C-HUIM and MaxC-HUIM was constructed in [10] to detect the closed and maximal high utility itemsets. Though, error rate was observed during the mining task was higher. A new evolutionary algorithm was constructed in [11] for predicting the HFI and HUI. Though, the precision was poor while getting the huge number of transactional itemsets.

Minimum threshold determination technique was presented in [12] to locate the interesting itemsets with a utilization of association rule mining. However, the error rate was not minimal. A rule ranking method was planned in [13] to get better utility itemsets extraction results. However, mining accuracy was not sufficient with larger input itemsets. A Logic Design-based Approach was implemented in [14] with the objective of carried outing the Frequent Itemsets extraction task. But, HUI mining was remained an open problem.

The Closed Candidates-based Frequent Itemset mining was done in [15] with the idea of increasing the precision of determining the support count. However, association rules were not constructed to significantly carried outs the mining process with lesser time. The pattern-growth based HUI mining algorithm was intended in [16] to decrease the execution time and memory usage. Though, top most users favorite itemsets was not considered. The non-redundant high-utility association rules were utilized in [17] to detect the high-utility itemsets. A novel association rules was designed in [18] for predicting the frequent generator itemsets via classification. Though, highly profited itemsets

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.5, May : 2024

discovery was not focused in this work. With an inspiration of addressing the above analyzed problems in conventional research work, a new BBACCARC model is proposed.

III PROPOSED WORK

The proposed BBACCARC model is proposed with the aiming at boosting the accuracy and performance of association rule generation to effectively classify each itemsets in big transactional dataset into a two classes (i.e. highly frequent or not, highly utility or not) with better time complexity. In BBACCARC model, Canonical Correlative Association Rule Based Classification (CCAR-C) is implemented as a novel and base classifier. The implemented BBACCARC model generates ‘n’ number of base classifier and then utilizes Bucklin voting to present the strong classifier outcomes to accurately find or extract the frequent and utility itemsets. The architecture of BBACCARC model is demonstrated in Figure 1.

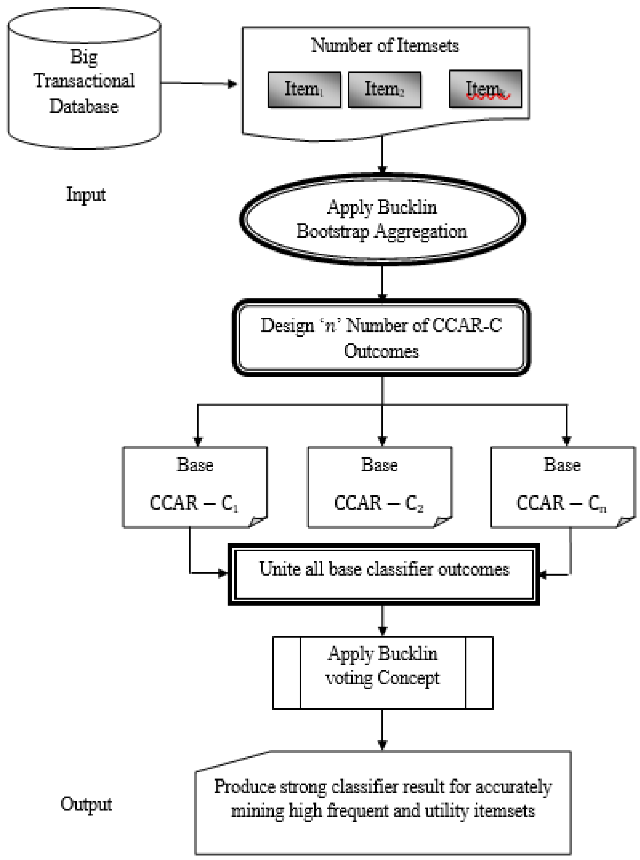

Figure 1 Architecture of BBACCARC model

Figure 1 Architecture of BBACCARC model

Figure 1 presents the processing diagram of BBACCARC model to attain better accuracy for predicting the top most users’ interested and profited itemsets. As illustrated in the above architecture, BBACCARC model at first gets huge volume of transactional database ‘D’ as input where it comprises of more number of items and itemsets. Consequently, BBACCARC model applies novel Bucklin Bootstrap Aggregation (BBA) concept. On the contrary to existing system, BBA is designed by applying Bucklin voting concept in bootstrap aggregation (i.e. bagging). With an employment of BBA, BBACCARC model creates the new training sets and thereby carried outs the bootstrap sampling procedure. Followed by, BBACCARC model develops ‘n’ number of base classifier (i.e., Canonical Correlative Association Rule Based Classification (CCAR-C) result. Then, BBACCARC model combines all base CCAR-C output and utilizes Bucklin majority voting concepts to generate the strong

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.5, May : 2024
classifier result. The discovered strong classifier in BBACCARC model exactly classifies each itemsets in big dataset. With the help of classified results, BBACCARC model significantly extracts the itemsets that are the top most users' interested and profitable in given dataset with minimal time complexity.

In BBACCARC model, CCAR-C algorithm is introduced as base classifier by using the Canonical Association Rule on the contrary to state-of-the-art works. Because, Canonical Association Rule is employed in this research work supports to accurately discover and measure the associations between two sets of items in larger transaction database with a lesser amount of time requirement. Besides to that, the Canonical Association Rule is applied in base CCAR-C algorithm focuses on the correlation among a linear combination of the itemsets. Let consider two items '$Item_i$' and '$Item_j$'. The Canonical Association Rule is utilized in base CCAR-C algorithm discover maximum correlation with each other. The Canonical Association Rule is designed in BBACCARC model using canonical correlation analysis (CCA). The From that, CCA is mathematically performed as,

$$Cr \rightarrow \sum Item_i \, Item_j = Cov(ItemS_i) \quad (1)$$

In mathematical formulation (1), covariance matrix 'Cov' is calculated based on information of itemsets '$ItemS_i$' i.e. support or utility value. The support value in BBACCARG model is determined based on the specific itemsets frequently in an input database using below,

$$S(ItemS_i) = \left|\frac{a}{b}\right| \quad (2)$$

In equation (2), '$S(Items_i)$' defines support value of itemset in given dataset and 'a' describes a number of occurrence of specific itemset '$ItemS_i$' and 'b' point outs a total number of an itemsets taken from a given dataset. Subsequently, utility value of item '$u(Item_i, T_x)$' in transaction is mathematically obtained as,

$$u(Item_i, T_x) = x(Item, T_x) \times \varphi(Item_i) \quad (3)$$

In formula (3), '$\varphi(Item_i)$' illustrates a profit value of an item. The profit value is estimated depends on the variation between the purchase price and selling price. Afterward, utility value of itemset '$ItemS_i$' in a transaction is mathematically obtained as,

$$u(ItemS_i, T_x) = \sum_{item_i \in D \wedge ItemS_i \subseteq T_x} u(Item_i, T_x) \quad (4)$$

Accordingly, utility value of itemset in a database '$u(ItemS_i)$' is mathematically acquired as,

$$u(ItemS_i) = \sum_{\tau_x \in D \wedge ItemS_i \subseteq T_x} u(ItemS_i, T_x) \quad (5)$$

Thus, base CCAR-C algorithm significantly finds the support and utility value of each itemsets in input dataset. Based on that, then base CCAR-C algorithm determine the support correlation among input itemsets '$Item_i$' and '$Item_j$' using,

$$Sup_{Cr}(ItemS_i) \leftarrow SupportCov(ItemS_i) = \frac{\sum(SItem_i - \overline{SItem_\iota})(SItem_j - \overline{SItem_J})}{n} \quad (6)$$

In equation (6), '$SItem_i$' represents the support value of '$Item_i$' and '$SItem_j$' denotes the support value of $Item_j$. Here, '$\overline{SItem_\iota}$' indicates the mean support value of '$Item_i$' and '$\overline{SItem_J}$'refers the mean support value $Item_j$ and 'n' represents a total number of the items in a given big dataset. Subsequently, base CCAR-C algorithm finds the utility correlation among input itemsets '$Item_i$' and '$Item_j$' using,

$$u_{Cr}(ItemS_i) \leftarrow UtilityCov(ItemS_i) = \frac{\sum(uItem_i - \overline{uItem_\iota})(uItem_j - \overline{uItem_J})}{n} \quad (7)$$

In expression (6), '$uItem_i$' represents the utility value of '$Item_i$' and '$\overline{uItem_\iota}$' denotes the utility value of $Item_j$. Here, '$\overline{Item_\iota}$' indicates the mean utility value of '$Item_i$' and '$\overline{Item_J}$'refers the mean utility value $Item_j$ and 'n' represents a total number of the items in a given big dataset. With the aid of above mathematical equations (6) and (7), base CCAR-C algorithm predicts the relationship between the set of itemsets in given transactional dataset. From that, Canonical Association Rule (CAS) for classification is mathematically described as,

$$CAR = \begin{cases} if\ Sup_{Cr}(ItemS_i) > \mathrm{T}_{Sup_{Cr}}, then\ ItemS_i\ is\ HFI \\ if\, u_{Cr}(ItemS_i) > \mathrm{T}_{u_{Cr}}, then\ ItemS_i is\ HUI \end{cases} \quad (8)$$

In the mathematical formula (8), '$\mathrm{T}_{Sup_{Cr}}$' depicts the threshold support correlation value and '$\mathrm{T}_{u_{Cr}}$' illustrates a threshold utility correlation value. Accordingly, base CCAR-C algorithm classifies the itemsets which includes a higher support correlation value as high frequent itemsets (HFI) and classifies the itemsets which includes a higher utility correlation value as high utility itemsets (HUI). By using the above canonical association rule, base CCAR-C algorithm in BBACCARC model returns classification results for effectively extracting the top most user interested and highly profitable itemsets. Though, classification accuracy using base CCAR-C algorithm was not adequate for precisely extracting the high frequent and utility itemsets in larger transactional dataset. Hence, the BBA is applied in this work where it primarily creates a number of bootstrap samples by considering the itemsets in input dataset. Next, BBACCARC model develops 'n' base CCAR-C outcomes for all itemsets in bootstrap samples. Subsequently, BBACCARG model units all base CCAR-C outcomes to achieve higher classification accuracy using,

$$\mathrm{CCAR-C}\ (ItemS_i) = \mathrm{CCAR-C}_{\,1}(ItemS_i) + \mathrm{CCAR-C}_{\,2}(ItemS_i) + \cdots + \mathrm{CCAR-C}_{\,n}(ItemS_i) \quad (9)$$

After completing the aggregation, BBACCARG model use bucklin vote 'v_i' for each base CCAR-C outcomes '$\omega(\$_i)$' using,

$$v_i \rightarrow \sum_{i=1}^{n} \mathrm{CCAR-C}(ItemS_i) \quad (10)$$

From that, the majority votes of all base CCAR-C outcomes are assumed as strong classifier. This strong classifier correctly classifies the itemsets in a given database with a minimal amount of time usage. In BBACCARC model, Bucklin Voting is a single-winner voting method. Here, it ranks the candidates (i.e. base CCAR-C outcomes) in order of preference (first, second, third, etc.). If one candidate gets a majority votes, that candidate wins. The class label which gets the majority votes is selected as the final prediction i.e. strong classifier. Thus, strong strong classifier outcomes to exactly discovery the high frequent and utility itemsets in big dataset is mathematically expressed,

$$\delta\ (ItemS_i) = arg \max_{n} v(CCAR - C(ItemS_i)) \quad (11)$$

In equation (11), '$\delta\ (ItemS_i)$' presents the final strong classifier result of BBACCARC model to efficiently find and mine the top most user interested and highly profitable itemsets with a minimal error rate. In BBACCARC model, '$arg \max_{n} v$' help to discover the majority votes of base classifier results. As a consequence, BBACCARC model gives enhanced classification performance to discover the highly frequent and most profitable itemsets.

The pseudocode description of BBACCARC model is illustrated as,

// Bucklin Bootstrap Aggregated Canonical Correlative Association Rule Classification Algorithm

Input: Number of itemsets '$ItemS_i = ItemS_1, ItemS_2, ItemS_3, \ldots, ItemS_m$

Output: Obtain better classification performance for mining top most user interested and highly profitable itemsets

Step 1: **Begin**
Step 2: **For** each input itemsets '$ItemS_i$'
Step 3: Apply BBA concept
Step 4: Develop the bootstrap samples using itemsets '$ItemS_i$'
Step 5: **For** each itemsets '$ItemS_i$' in bootstrap samples
Step 6: Measure support and utility value using (2) and (5)
Step 7: Find the support correlation among the itemsets using (6)
Step 8: Identify the utility correlation among the itemsets using (7)

Step 9:	Design canonical association rule using (8)
Step 10:	Acquire 'n' number of base CCAR-C outcomes using equation (1) to (8)
Step 11:	Combine all the base CCAR-C outcomes using (9)
Step 12:	Use Bucklin votes for each base CCAR-C outcomes using (10)
Step 13:	Strong classifier correctly classifies itemsets '$ItemS_i$' as HFI or HUI using (11)
Step 14:	**End For**
Step 15:	**End For**
Step 16:	**End**

Algorithm 2 Bucklin Bootstrap Aggregated Canonical Correlative Association Rule Classification

IV. PERFORMANCE RESULTS

In order to test the performance, proposed BBACCARC model and existing two techniques are implemented using dual core processor with 4GB RAM at 2GHz in Java language. During the experimental evaluation, big size of transaction dataset is taken as input from http://fimi.cs.helsinki.fi/data/ [19]. By using this dataset, different experimental process are carried out by considering input items in the range of 100-1000. The experimental performance of BBACCARC model is measured using below parameters.

- Accuracy
- Time Complexity
- Error Rate

The testing outcomes of BBACCARC model is compared against with traditional improved binary PSO algorithm [1] and HLHUI [2].

Test case a: Accuracy

Accuracy is calculated based on the ratio of itemsets that are correctly mined as HFI and HUI through classification to the total items taken as input. From the above description, accuracy of extracting the HFI and HUI is estimated using below,

$$Accuracy = \frac{M_{CE}}{m} * 100 \qquad (12)$$

In equation (12), 'M_{CE}' defines a number of accurately extracted itemsets HFI and HUI and 'm' shows the total items taken as input. The accuracy of discovering the top most users' interested and profited itemsets is obtained in percentages (%).

Table 1 Comparative Tabulation Results of Accuracy for Extracting HFI and HUI

Number of Items	Accuracy (%)		
	Improved Binary PSO	HLHUI	BBACCARC
100	83	87.12	94.5
200	84.05	87.88	95.11
300	84.52	87.96	95.56
400	84.96	88	96
500	85	88.88	96.21
600	86.11	89.75	96.45
700	87	90.45	96.95
800	88	91.02	98
900	88.02	91.30	98.20
1000	88.41	92	98.75

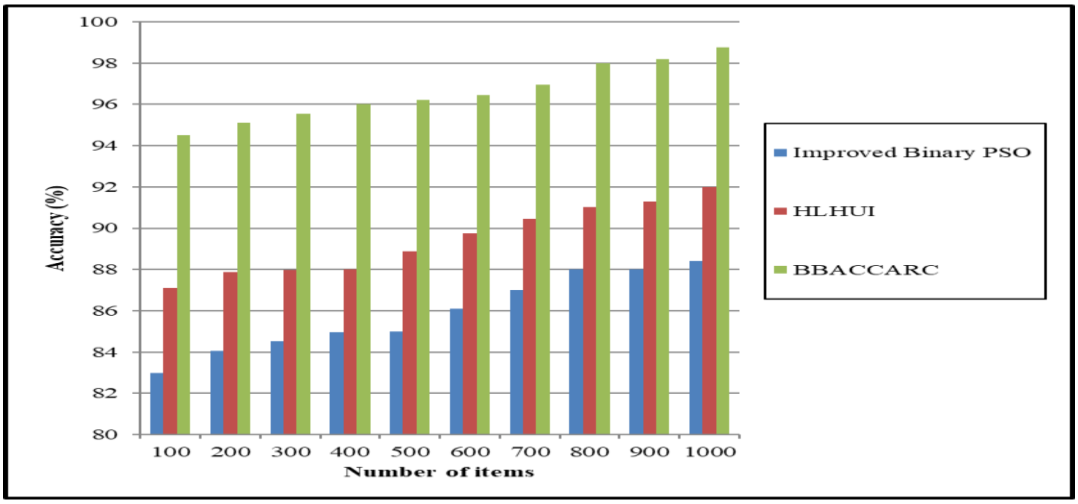

Figure 3 Graphical Simulation Performance of Accuracy versus Number of items

Table 1 and Figure 2 describes the comparative testing results of accuracy versus diverse number of items considered in the range of 100-1000 from large transactional database for proposed BBACCARC model and state-of-the-art improved binary PSO algorithm [1] and HLHUI [2]. As presented in the above graphcical performance diagram, proposed BBACCARC model achieves higher accuracy for significantly extracting the top most users interested and higher profitable itemsets while increasing number of items as input when compared to conventional improved binary PSO algorithm [1] and HLHUI [2]. The better mining accuracy is gained in proposed BBACCARC model due to employment of BBA concept in base CCAR-C algorithm to generate the strong classifier for finding the HFI and HUI for extraction. As an outcome, the proposed BBACCARC model improves the ratio of itemsets that are rightly mined as HFI and HUI through classification when compared to existing algorithms. When conducting a experimental process by taking 1000 items as an input, proposed BBACCARC model attained 98.75 % accuracy where conventional improved binary PSO algorithm [1] and HLHUI [2] acquired 88.41 %, 92 % respectively. Therefore, the proposed BBACCARC model gives enhanced accuracy for extraction of highly frequent and profitable itemsets when compared to existing improved binary PSO algorithm [1] and HLHUI [2].

Test case b: Time Complexity

Time complexity is calculated based on the total time required to significantly extract the HFI and HUI from a large dataset. By considering the above definition, time complexity is observed as,

$$TC = m * time(ESI) \qquad (13)$$

In equation (13), '$time(ESI)$' shows the time utilized to precisely mine the single itemsets as HFI and HUI and 'm' indicates to the total itemsets. The time complexity is obtained in milliseconds (ms).

Table 2 Comparative Tabulation Results of Time Complexity for predicting HFI and HUI

Number of Items	Time Complexity (ms)		
	Improved Binary PSO	HLHUI	BBACCARC
100	18.1	15.4	10.2
200	20.9	18.3	11.9
300	25	21.6	13.4
400	28.4	24	15.7
500	32.8	27.2	18
600	36.5	30	20.2
700	38.7	32.8	23.7
800	40.1	36.4	25

900	43	38	27.4
1000	45.6	40.5	30

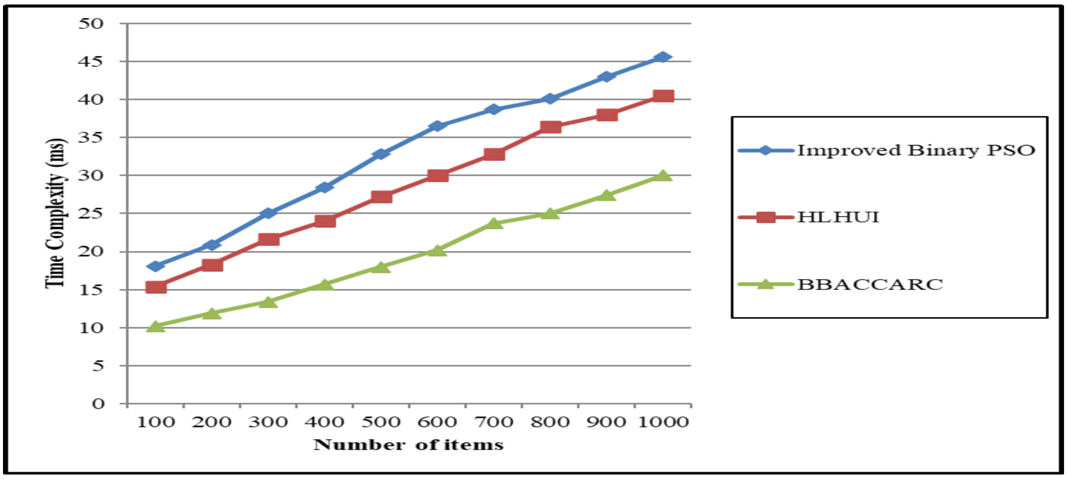

Figure 4 Graphical Simulation Performance of Time Complexity versus Number of items

Table 2 and Figure 4 presents the results evaluation of time complexity is observed during the identification and extraction of HFI and HUI based on dissimilar number of input items taken in the range of 100-1000 from big database for proposed BBACCARC model and state-of-the-art improved binary PSO algorithm [1] and HLHUI [2]. As displayed in the above comparative testing results analysis, proposed BBACCARC model obtained lower amount time complexity for accurately detecting the top most user interested itemsets and utility itemsets while getting more number of testing items as input while compared to state-of-the-art improved binary PSO algorithm [1] and HLHUI [2]. The better computational complexity is attained in proposed BBACCARC model because of application of CAR to predict the support and utility correlation among the itemsets in big dataset with lesser time requirement. From that, the proposed BBACCARC model utilizes lesser time to identify and mine the HFI and HUI when compared to existing algorithms. When considering 900 items as an input to perform experimental process, proposed BBACCARC model employs 30 ms time for efficiently extracting the HFI and HUI whereas traditional improved binary PSO algorithm [1] and HLHUI [2] takes 45.6 ms, 40.5 ms respectively. Hence, the proposed BBACCARC model presents better complexity for mining of HFI and HUI when compared to conventional improved binary PSO algorithm [1] and HLHUI [2].

Test case c: Error Rate

The error rate is calculated based on the ratio of itemsets that are mistakenly mined as HFI and HUI to the total items get as input. Thus, the error rate is computed using below,

$$Error\ rate\ = \frac{M_{IE}}{m} * 100 \qquad (14)$$

In the equation (14), 'M_{IE}' depicts a number of incorrectly extracted itemsets as HFI and HUI and 'm' portrays the total items obtained as input. The misclassification rate of mining the top most users' interested and profited itemsets is acquired in percentages (%).

Table 3 Comparative Tabulation Results of Error Rate for mining HFI and HUI

Number of Items	Error Rate (%)		
	Improved Binary PSO	HLHUI	BBACCARC
100	17	12.88	5.5
200	15.95	12.12	4.89
300	15.48	12.04	4.44
400	15.04	12	4
500	15	11.12	3.79

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.5, May : 2024

600	13.89	10.25	3.55
700	13	9.55	3.05
800	12	8.98	2
900	11.98	8.7	1.8
1000	11.59	8	1.25

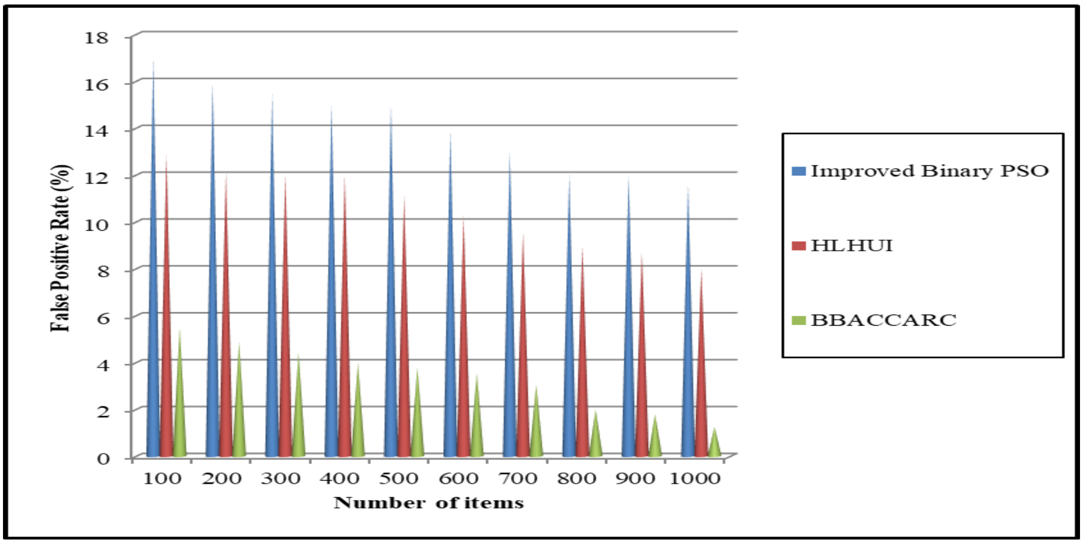

Figure 5 Graphical Simulation Performance of Error Rate versus Number of items

Table 3 and Figure 5 reveals the experimental performance results measurement of error rate is estimated during the discovery of most user favorite and profited itemsets along with different number of input items using proposed BBACCARC model and existing improved binary PSO algorithm [1] and HLHUI [2]. As discussed in the above table and graphcal performance examination, proposed BBACCARC model gets lesser false rate for extraction of HFI and HUI while taking more volume of testing items as input when compared to conventional improved binary PSO algorithm [1] and HLHUI [2]. The better error rate result is achieved in proposed BBACCARC model owing to utilization of BBA concepts where it finds the strong classifier which provides better mining results with misclassification error and thereby boosts the classification performance. Thus, the proposed BBACCARC model decreases ratio of number of itemsets that are wrongly mined as HFI and HUI through classification when compared to state-of-the-art works. When assuming 800 items as an input, proposed BBACCARC model attained 2 % error rate to extract the highly frequent and profitable itemsets in larger database whereas traditional improved binary PSO algorithm [1] and HLHUI [2] obtained 12 %, 8.98 % respectively. For this reason, the proposed BBACCARC model provides lower error for extracting the HFI and HUI when compared to conventional improved binary PSO algorithm [1] and HLHUI [2].

V. CONCLUSION

In this article, a BBACCARC model is presented with the plan of boosting the extraction performance of HFI and HUI via classification with minimal complexity when getting the larger size of input transactional dataset. The purpose of BBACCARC model is attained by applying BBA in base CCAR-C on the contrary to conventional classification algorithms. The designed BBACCARC model helps to predict the consumer preferences i.e. buying behavior of consumers in a given market. Moreover that, the BBACCARC model significantly discovers the top most users interested and profited itemsets which assists for marketing companies to produce the right products or services, making them available to consumers at the right time to maximize sales and profits. Besides to that, the proposed BBACCARC model reduced the computational complexity during the extraction of the

HFI and HUI through generating the canonical association rule. Furthermore, proposed BBACCARC model lessened the error rate of mining process through designing a strong CCAR-C result. The testing result confirmed that the proposed BBACCARC model acquired better mining performance with an enhancement of accuracy and minimization of complexity and error rate.

REFERENCES

[1] Wei Fang, Qiang Zhang, Hengyang Lu, Jerry Chun-Wei Lin, " High-utility itemsets mining based on binary particle swarm optimization with multiple adjustment strategies", Applied Soft Computing, Volume 124, 2022,

[2] Mohammad Sedghi, Mohammad Karim Sohrabi, "HLHUI: An improved version of local high utility itemset mining", Procedia Computer Science, Elsevier, Volume 220, 2023, Pages 639-644

[3] C. Zhang, M. Han, R. Sun, S. Du and M. Shen, "A Survey of Key Technologies for High Utility Patterns Mining," in IEEE Access, vol. 8, pp. 55798-55814, 2020

[4] W. Fang, H. Jiang, H. Lu, J. Sun, X. Wu and J. C. -W. Lin, "GPU-Based Efficient Parallel Heuristic Algorithm for High-Utility Itemset Mining in Large Transaction Datasets," in IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 2, pp. 652-667, Feb. 2024

[5] B. VO, L. T. T. Nguyen, T. D. D. Nguyen, P. Fournier-Viger and U. Yun, "A Multi-Core Approach to Efficiently Mining High-Utility Itemsets in Dynamic Profit Databases," in IEEE Access, vol. 8, pp. 85890-85899, 2020

[6] J. Sun, Y. Xun, J. Zhang and J. Li, "Incremental Frequent Itemsets Mining With FCFP Tree," in IEEE Access, vol. 7, pp. 136511-136524, 2019

[7] J. Alcala-Fdez, R. Alcala and F. Herrera, "A Fuzzy Association Rule-Based Classification Model for High-Dimensional Problems With Genetic Rule Selection and Lateral Tuning," in IEEE Transactions on Fuzzy Systems, vol. 19, no. 5, pp. 857-872, Oct. 2011

[8] Z. Cheng, W. Shen, W. Fang and J. C. -W. Lin, "A Parallel High-Utility Itemset Mining Algorithm Based on Hadoop," in Complex System Modeling and Simulation, vol. 3, no. 1, pp. 47-58, March 2023

[9] Y. Djenouri, D. Djenouri, J. C. -W. Lin and A. Belhadi, "Frequent Itemset Mining in Big Data With Effective Single Scan Algorithms," in IEEE Access, vol. 6, pp. 68013-68026, 2018

[10] Hai Duong, Tien Hoang, Thong Tran, Tin Truong, Bac Le, Philippe Fournier-Viger, Efficient algorithms for mining closed and maximal high utility itemsets, Knowledge-Based Systems, Elsevier, Volume 257, 2022

[11] Wei Fang, Chongyang Li, Qiang Zhang, Xin Zhang, Jerry Chun-Wei Lin, "An efficient biobjective evolutionary algorithm for mining frequent and high utility itemsets", Applied Soft Computing, Volume 140, 2023

[12] Hikmawati, E., Maulidevi, N.U. & Surendro, K, "Minimum threshold determination method based on dataset characteristics in association rule mining", Journal of Big Data, Springer, Volume 8, 146 (2021)

[13] Hikmawati E, Maulidevi NU, Surendro K, "Rule-ranking method based on item utility in adaptive rule model", Peer Journal of Computer Science, 2022

[14] Oday Ahmed Al-Ghanimi1, Hussein K. Khafaji, "A Logic Design-based Approach for Frequent Itemsets Mining Using LCO Algorithm", International Journal of Intelligent Engineering and Systems, Vol.16, No.2, 2023

[15] Magdy, M., Ghaleb, F.F.M., Mohamed, D.A.E.A. et al. CC-IFIM: an efficient approach for incremental frequent itemset mining based on closed candidates, Springer, The Journal of Supercomputing, Volume 79, 7877–7899, 2023

[16] Yiwei Liu, Le Wang, Lin Feng and Bo Jin, "Mining High Utility Itemsets Based on Pattern Growth without Candidate Generation", Mathematics 2021, 9(1), 35

[17] Thang Mai, Loan T.T. Nguyen, Bay Vo, Unil Yun and Tzung-Pei Hong, "Efficient Algorithm for Mining Non-Redundant High-Utility Association Rules", Sensors 2020, 20(4), 1078

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.5, May : 2024

[18] Makhlouf Ledmi, Mohammed El Habib Souidi, Michael Hahsler, Abdeldjalil Ledmi and Chafia Kara-Mohamed, "Mining association rules for classification using frequent generator itemsets in arules package", International Journal of Data Mining, Modelling and ManagementVol. 15, No. 2, 2023, pp 203-221
[19] http://fimi.cs.helsinki.fi/data/.