Journal of Theoretical and Applied Information Technology

15th November 2023. Vol.101. No 21

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

IMPLEMENTATION OF INCREMENTAL APPROACH IN R-DIFFSET FOR INFREQUENT ITEMSET MINING

JULAILY AIDA JUSOH[1], SHARIFAH ZULAIKHA TENGKU HASSAN[2], WAN AEZWANI ABU BAKAR[3], MOHD KHALID AWANG[4], SYARILLA AHMAD SAANY[5], NORLINA UDIN @ KAMARUDDIN[6]

[123456]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Besut Campus, 22200 Besut,Terengganu, Malaysia.

E-mail: julaily@unisza.edu.my, sharifahzulaikha1992@gmail.com, wanaezwani@unisza.edu.my, khalid@unisza.edu.my, syarilla@unisza.edu.my, norlina@unisza.edu.my

ABSTRACT

Data mining is a well-established approach for extracting crucial information from databases that employs the Association Rule Mining (ARM) technique. It can unearth hidden information that can help with decision-making, financial forecasting, marketing policy, medical diagnostics, and other uses. ARM is the most widely used data mining approach for discovering exciting correlations and connection pattern among itemsets in transaction databases. This vital data can lead to the association rule, suggesting a positive trend. The advantageous itemset of the association's regulations is typically expressed as frequent and infrequent. There are two data formats in itemset mining that is horizontal and vertical. R-Eclat, or Rare Incremental Equivalence Class Transformation, is an example of a vertical data mining approach for an infrequent itemset. The R-Diffset variant, one of four R-Eclat algorithm variants, will be the focus of this study. Previous research has shown that the R-Diffset algorithm takes a long time to process data. Current research outcomes in infrequent mining techniques focus on vertical data formats. The experimental result this indicates that the comparison analysis for three (3) datasets that is mushroom, pumsb_star, and chess. The average performance in terms of the execution time of IR-Diffset is better than R-Diffset.

Keywords: *Data mining, Asscosiation Rule Mining (ARM), Infrequent itemset mining, R-Eclat algorithm, R-Diffset algorithm, IR-Diffset algorithm*

1. INTRODUCTION

Data Mining [2,39] is an established methodology for obtaining critical information from databases. It is one of the crucial roles in the oversized data approach for obtaining meaningful knowledge resulting in a complex system. The approach includes collaborative studies in statistics, machine learning, data science, and database theory. It aims to grasp the past or predict the future by studying the present data. In addition, this approach, also known as Knowledge Discovery in Databases (KDD), focuses on finding patterns in databases, resulting in an association rule that may disclose important information. Some common patterns are found in the databases, such as clusters, sets of items, trends, and outliers [1,29]. There are two major tasks in data mining that is predictive and descriptive. For the predictive task, it strives to compute the predicted value of one feature based on the value of another feature using techniques like statistics, categorization, regression, and forecasting. Then, descriptive task it generates patterns (clusters, correlations, anomalies, and trends) to extract the database's underlying relationships. The techniques used in descriptive tasks include clustering, summarizing, association rules, pattern detecting, and sequence discovery. There are two types of itemset mining in the database that is frequent and infrequent. From the records of the previous studies, in frequent itemset mining, there are three well-known algorithms: Apriori [2, 3], FP-Growth [4], and Eclat [5]. This research focuses on the vertical format by looking deeper into the equivalence class transformation (Eclat) algorithm [5]. Tidset, Diffset, Sortdiffset, and Postdiffset are four extension variations introduced in Eclat. In 2018, Jusoh et al. [6,7] introduced the R-Eclat algorithm, particularly useful for mining infrequent itemsets. This algorithm is based on the infrequent itemsets mining Eclat [8–10]. The current R-Eclat algorithm contains four variations [6,7,38], which result in poor sequential processing performance. This poor performance motivates the development of IR-Diffset, which will

employ an incremental approach for infrequent itemsets mining. Currently, Man et.al [37] has introduced IR-Eclat where experimentation is done on disease dataset. The result shows that there is a big contrast between the performance of R-Eclat and IR-Eclat. Experimentally, this research compared the runtime processing between R-Diffset and IR-Diffset. The arrangement of the remainder of the work is as follows: The second section of this study introduces infrequent itemset mining; the third section describes the Eclat and R-Eclat algorithms on the diffset structure; the fourth section demonstrates the IR-Diffset technique; the fifth section describes the incremental approach; and the sixth section concludes the analysis.

2. INFREQUENT ITEMSET MINING

The concepts usually come from massive databases, which are seen as data mines containing valuable information by using association rule techniques [40]. In 1994, Agrawal R. et al. [3] presented association rule mining (ARM) for the first time. Databases and data mining communities are essential components for data mining approaches. ARM aims to determine whether there are frequent patterns or itemsets (collections of one or more items) in databases. If any, a relationship between these frequent itemsets (itemsets with support higher than or equal to a min_supp criterion) can reveal a new pattern analysis for subsequent decision-making. Association rules are if-then-else expressions that establish specific associations between unrelated relational databases or other types of information storage. it is classified as a set of items if s $itemsets = \{i1, i2, \ldots, in\ for\ |n| > 0\}$. D refers to a transaction database. T ⊆ Itemsets is a set of items simplified by transaction, T. Tid is a distinct identifier issued to each D transaction. For the transaction, T, the association rule has the form of X ⊆ Y for the transaction database T, where X represents the preceding component, and the subsequent component of the rule represented by Y; X ⊆ I, Y ⊆ I, and X ∩ Y = ∅. The statement for association rule might be, "If a customer purchases a set of diapers, he is 80% likely also to purchase a bottle of milk.". If s% of the transactions in the transaction set D contains both X and Y, the rule X ⇒Y the transaction holds support s. If there are c% of D transactions containing both X and Y, then the rule X⇒Y has confidence c in the D transaction. The following is an illustration of the support-confidence framework:

$$Support\ (X \Rightarrow Y) = \frac{(XUY)}{|D|} \quad (1)$$

a) The X ⇒ Y rule is supported by the portion of D transactions containing both X and Y, where |D| is the count of entries in a database.

$$Confidence\ (X \Rightarrow Y) = \frac{supp(XUY)}{supp(x)} \quad (2)$$

b) The confidence of rule X ⇒ Y refers to the fraction of transactions in D which contains both X and Y.

The "Strong rules" are association rules that satisfy the minimal confidence (min_ conf) threshold, where min_conf and min_supp are user-specified values. [11]. It is deemed interesting if a rule meets both the min_supp and min_conf standards [12– 15]. A rule is considered as frequent if its support equals or exceeds a particular integer, known as the minimum support threshold (min_supp). However, the rule is considered as infrequent if its support is equal to or lower than the minimum support (min_supp) criterion and meets the maximum confidence (max_conf) criterion.

3. ECLAT AND R-ECLAT ALGORITHM

Zaki et al. introduced the Equivalence Class Transformation (Eclat) approach [8]. This approach includes four (4) variants, each with its own structure: Tidset [8], Diffset [19], Sortdiffset [20], and Postdiffset [21,22]. It represents databases using depth-first search (DFS) with a vertical layout; a collection of transaction IDs (referred to as a tidset) represents each item in the databases, including the item [16]. The Eclat algorithm's primary technique intersects the transactions-IDs (Tids) list.

As proposed by Apriori [17,18], the most commonly itemset mining algorithms that uses a breadth-first search (BFS) and the downward closure property. However, utilizing tidsets does not require counting support, as the size of the tidset is its support. The Eclat principal operation is intersecting tidsets; hence, the amount of tidsets is the crucial factor determining the Eclat operating time and memory use. The larger the tidsets, the more time and memory are required. Eclat utilizes bottom-up search as well as prefix-based equivalence relations. It lists out all of the frequent itemsets [8, 23]. The two most important phases are candidate creation

and trimming. During the candidate creation process, two frequent (k-1) itemsets generated one k-itemset candidate. The candidate's support will be deleted if it falls below the threshold. Otherwise, frequent itemsets will be used to produce (k+1) itemsets. Eclat vertical layout facilitates counting assistance [24]. depth-first searching (DFS) starts with frequent itemset in the item base, then 2-itemsets from 1-itemsets, 3-itemsets from two itemsets, and so on. R-Eclat (Rare Equivalence Class Transformation) [1,22,37] is offered as a result, with an emphasis on infrequent itemset mining in a vast database. It runs on the same principles as Eclat, with minor algorithm modifications. Support counting, which finds the support of each k-itemset by intersecting tid-lists of its k-1 subsets, is a crucial component of a rapid R-Eclat.

In a basic concept of this algorithm, let B be the universe of an item, where B = $\{i1, i2, \ldots, im\}$, and m > 0 signifies the number of m items in a collection of literals. If a set $X = \{i1, \ldots, ik\} \subseteq$ B has k-items, it is termed an itemset or a k-itemset. $Ti = (tid, I)$ is a transaction over B, where Tid represents a transaction identification, and I signify an itemset. A transaction $Ti = (tid, I)$ supports an itemset $X \subseteq$ B if $X \subseteq I$. A database for transactions T is a collection of transactions over B. A tidset of an itemset X in T is a set of transaction IDs in T that support X, where (support, X) = $\{tid \mid (tid, I) \in T, X \subseteq I\}$. The cardinality of an itemset X in T is the number of transactions that include T, where (support, X) = $|(X)|$. Figure 1 depicts the R-ECLAT structure model.

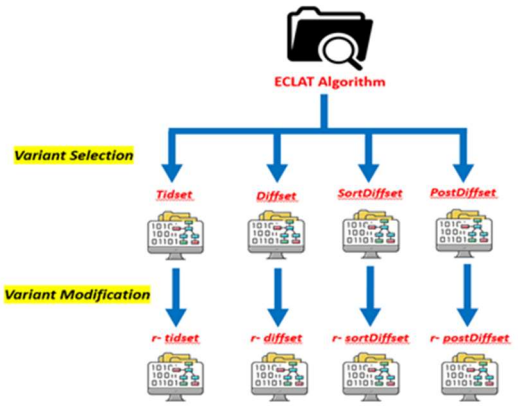


Figure 1: R-Eclat Model

Association Rules (ARs) are the most essential concepts in Data Mining. There are three (3) classic algorithms that are widely used in the main ARM namely Apriori, FP-Growth, and Eclat. So, for the first step, we solely look on the Eclat algorithm. The Eclat algorithm is only used for frequent itemset mining. The second step, R-Eclat is introduced in infrequent itemset mining. However, R-Eclat is only used in infrequent itemset mining. Meanwhile, in R-Eclat there are four (4) variants which are similar to Eclat which has four (4) variants also, but for R it signifies that it changes all its pseudocodes. The R-Eclat algorithm modifies the four preceding Eclat variants to ensure they are appropriate for mining an infrequent itemset. R-Diffset, R-Tidset, R-Sortdiffset, and R-Postdiffset [6,7,38] are the newly improve forms of the R-Eclat algorithm, where R signifies rare. R-Tidset indicates the size of tidsets and performs vertical intersection of tidlist. R-Diffset, on the other hand, maintains a record of differences between tidsets, making the intersection quicker and using fewer data. R-Sortdiffset is a hybrid between R-Diffset and R-Tidset. Diffset is arranged downward, whereas Tidset is arranged ascending. The last variation, R-Postdiffset, is a mixture of R-Diffset and R-Tidset that outperforms the others. In this study, we solely look at the R-Diffset variant, which R-Diffset is to modify its pseudocode with an element incremental approach in R-Diffset named as Incremental R-Diffset (IR-Diffset). This incremental approach is proposed to overcome the limitations in extracting infrequent itemsets via sequential processing. Figure 2 depicts the topology of keywords for this research.

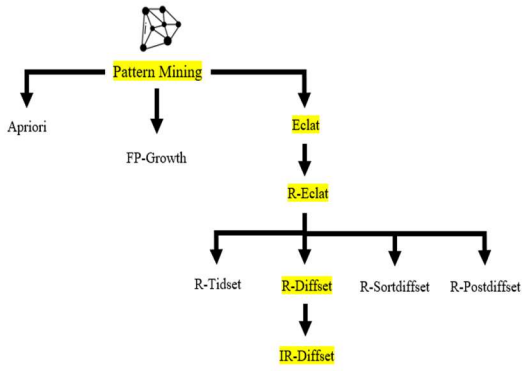


Figure 2: Topology of Research Background

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

3.1 R-Diffset Variant

As the R-Diffset variation records the tidset's differences, the intersection is faster and consumes lesser memory. The R-Diffset technique, a variation of the Eclat method, employs the 'diffset' structure rather than the 'tidset' structure. Zaki M. J. et al. [19] suggested an R-Diffset (different set or diffset), in which the authors describe an itemset by tids that exist in the tidset of its prefix but not in its tidsets. In other words, diffset is simply the difference between two tidsets, such as the itemsets tidset and its prefix. These distinctions are passed down from a node to its offspring, beginning at the root. The diffset minimizes the attributes of sets representing itemsets, leading to a quicker intersection and less memory consumption.

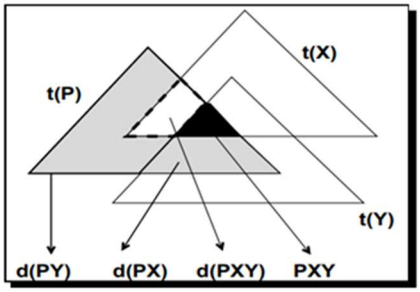


Figure 3: Diffsets Illustration

According to Trieu T. A. et al. [25], an equivalence class with the prefix P is expected to contain X and Y itemsets. Let (X) represent the tidset of X and $d(X)$ represent the diffset of X. When utilizing the tidset format, we will have $t(PX)$ and $t(PY)$ accessible in the equivalence class, and to obtain $t(PXY)$, we check the cardinality of $t(PX) \cap t(PY) = t(PXY)$. When we use the diffset variant, we get (PX) instead of $t(PX)$, and $d(PX) = t(P) - t(X)$, which is the set of tids in $t(P)$ but not in $t(X)$. Likewise, we have $(PY) = (P) - t(Y)$.

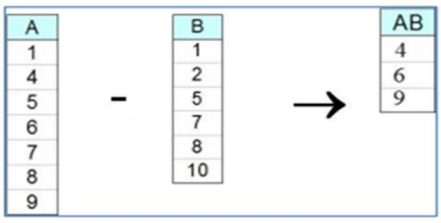


Figure 4: Diffset between itemset A and B

As a result, it is PX's support rather than the size of its diffset. According to the definition of (PX), $|(PX)| = |t(P)| - |t(P) - t(X)| = |t(P)| - |d(PX)|$. To put it another way, $\sup(PX) = sup(P) - |d(PX)|$. Figure 3 shows how Trieu et al. [9] diffsets explain this formula. As a result, the frequency of occurrences (support) of PX does not constitute to diffset in size as in Figure 4.

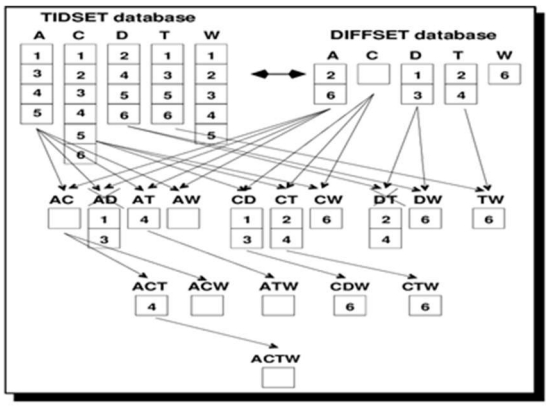
To utilize the diffset variation, first convert the initial transaction database in a vertical arrangement to the diffset variant, where the diffset of items are sets of tids for transactions that do not include in these items. This method is deduced from the definition of diffset; the initial transaction database in the vertical layout is equivalent to the prefix P = {}. Hence, the tidset of P includes all tids, all transactions contain, and the diffset of an item is $d(i) = t(P) - t(i)$, which is a set of tids whose transactions do not include items. We created all itemsets with associated diffsets and supported from the initial equivalence class. Figure 5 show pseudocode for R-Diffset that used in data infrequent itemset mining.

```
Pseudocode R-Diffset
Input: E((i1, t1),...(i1, t1))|P), smin
Output: F(E, smin )

Begin // get minimum support
1. Arrange data by itemset
2. Looping = numberOfColumns;
3. min_supp = number_of_rows * percentage_min_support
4. Run tidset;
5. for (i=0; i<=min_support)
6. if (support <= min_support){
7.      get diffset data for column [i] with column [i+1];
8.      save to DB;}
9. Set next transaction data;
10. Write to text file the value for the current / last transaction data;}
End;
```

Figure 5: Pseudocode for R-Diffset

The approach that we apply to minimizes the attributes of sets representing itemsets, leading to a quicker intersection and less memory consumption in R-Diffset. Line number 6 show that (support<=min_support) where support determines how often a rule is applicable to a given dataset. A rule is frequent if its support is equal to or greater than the given integer called minimum support threshold (min_supp). The rules which satisfy the minimum confidence (min_conf) threshold is called strong rule and both min_supp and min_conf are user-specified values [41]. An association rule is considered interesting if it satisfies both min_supp and min_conf thresholds. In contrast situation, a rule can be infrequent if its support is equal to or lower than the minimum support (min_supp) threshold and satisfies maximum confidence (max_conf). Support and confidence are the basic evaluation measure of interestingness in association rule mining. This step



is different from others pseudocode variants where mostly applied by R-Diffset.

Figure 6: Diffsets for itemset counting

The R-Diffset variation outperforms the tidset option concerning speed and memory utilization, particularly in dense databases. If the database is sparse, the benefits of tidsets are lost. In 2003, Zaki et al. [9] suggest to use the tidset variant at the start of the sparse database and later switching to the diffset variant when the switching condition is met. In the case of dense datasets, it is better to start with the diffset variant. However, because diffset is typically an order of magnitude smaller than tidsets, starting with the tidset variation when dealing with sparse datasets before shifting to the diffset variant for subsequent phases is preferable. Consider the itemset PXY in a new class, PX, which may be stored in either tidset (PXY)) or diffset (PXY). The reduction ratio, r = (PXY) / (PXY). To be advantageous, diffsets must have a reduction ratio of at least one. That is $r \geq 1$ or (PXY) / (PXY) ≥ 1 Since (PX) – (PY) = $t(PX) - t(PXY)$, we obtain $t(PXY)$ / ($t(PX) - t(PXY)$) ≥ 1. If we divide by (PXY), so 1 / ((PX) / $t(PXY) - 1$) ≥ 1. After simplification, the results will be (PX) / (PXY) ≤ 2. In other words, the authors discover that switching to a diffset variation is preferable if PXY support is at least half that of PX. Empirically it is advisable to utilize diffset from length 2-itemsets onwards [9], [25]. However, if the reduction ratio is less than 1, switch to the diffset starting at three itemsets. As more itemsets are discovered, the diffset data structure compresses the database rapidly. In comparison to other approaches, the diffsets method is flexible. Figure 6 indicates that the tidset database requires more space in memory to hold 23 Tids than the diffset database, which only requires 7 Tids.

4. IR-DIFFSET ALGORITHM

This research will introduce an incremental approach to improve the current R-Diffset, it named as Incremental R-Diffset (IR-Diffset). This approach is modified by using the standard Eclat [34] and R-Eclat [35] additional algorithms. This initiative serves for enhancement in the incremental approach. This technique is also used in the variance of the R-Diffset algorithm [36, 37]. The IR-Diffset algorithm is developed to solve the drawback of time consumption in infrequent itemset mining. The incremental approach is introduced as complementary to R-Diffset in order to ensure this process becomes more efficient in lessen time.

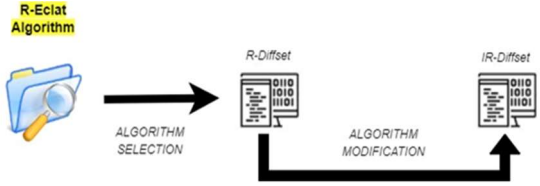


Figure 7 shows the IR-Diffset model.

Figure 7: IR-Diffset Model

Our approach is to increment the itemset from R-Eclat as our based model. The main steps in R-Eclat over the dataset are listed as follows:

Support counting, which finds the support of each k-itemset by intersecting tid-lists of its k-1 subsets, is a crucial component of a rapid R-ECLAT. The first step is repeated until no candidate itemsets can be generated. In second step is the IR-Diffset

$$\frac{\alpha}{100} * \beta \qquad (3)$$

differs from ECLAT in that a new diffset is developed instead of a new tidset. After that, the minimum support threshold value (MSTV) was considered as a benchmark to detect low occurrences in each data set. In [40], MSTV is determined in terms of percentage,

where: α = User specified minimum support value
β = Total of records in datasets.

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

```
Pseudocode IR-Diffset
Input: E((i1, t1), ... (in, tn)| P), smin
Output: IF(E, smin)

start          //get minimum support
sort data by itemset
looping = num_of_column
r = record_of_transactions
min_supp = num_of_row * percentage_min_supp
run tidset
while r ≠ 0 do          //process itemset by batch
for (i=0; i<looping; i++)
      if (support<=min_support)
            get diffset data for column [i] with column [i+1];
            save to db
      add next transaction data;
      write to text file the value for the current / last transaction data;
end
```

Figure 8: Pseudocode for IR-Diffset

Figure 8 depicts the step-by-step action for the pseudocode of IR-Diffset. In the pseudocode, the term "min_supp" refers to the minimum support threshold value, which is expressed as a percentage after the user-specified value is divided by 100 and multiplied by the total number of rows (records) in each dataset. While, the MSTV anyway is the benchmark to determine the occurrence of infrequent itemsets in the database hence the support counting for every process is constant. Begin with the first loop in each loop, and if the support is less than or equal to min_supp, (support <= *min_supp*) and then getting the diffset intersection in incremetal approach R-Diffset (IR-Diffset). Then, the diffset value between k^{th} column and k^{th} + *1* column will be encountered and save to db. Then specific criteria are met, such as:

i. Rather than utilizing intersection, IR-Diffset obtains the result of Diffset (difference intersection set) between the k^2 and k^{th}+1 columns and stores it in the database.

5. THE INCREMENTAL APPROACH

An incremental approach is used to calculate the execution time of each process in the transaction with (adding row) or in a number of itemsets with (adding column). The incremental approach is advantageous for dynamic database which subject to addition or deletion of items or record of transaction in the database. The process of this model is executed in solely sequential order. Sequential processing implies that one process must be completed before the next starts. In this study, the benchmark dataset will be processed using sequential processing. Assume the dataset has 1000 rows of records; this model will analyze all records once before moving on to the subsequent datasets. Nonetheless, the issue is the amount of time required to complete the mining process on a large dataset, as well as the size of the memory space required. In these instances, an incremental process is a promising option for improving execution time and memory utilization on a vast dataset. An algorithm is also deemed efficient if it requires little execution time and memory. The different implementations of the variants and may differ in their effectiveness. The Association Rule Mining (ARM) approach may be used to assess the algorithm's performance across all R-Eclat variants. Figure 9 depicts the physical design of the incremental approach

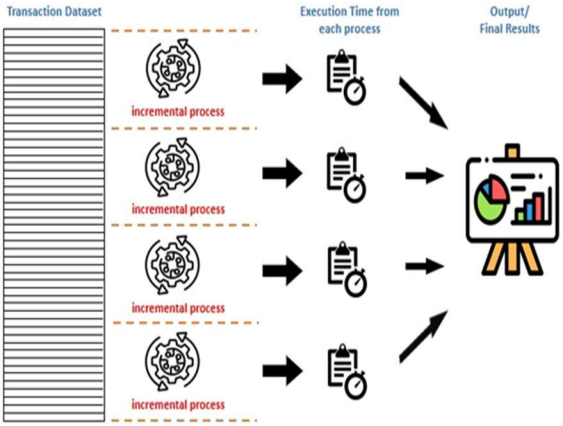


Figure 9: Physical Design of Incremental Approach

6. RESULT AND DISCUSSION

All studies are carried out on an HP Notepad with an Intel ® Core TM ® i7-3520M CPU running at 2.90 GHz and 8GB of RAM running Windows 10 64-bit. Open-source software is used to implement the algorithm development software standard. MySQL (version 5.6.25 - MySQL community server (GPL)) for our database server, Apache/2.4.16 (Win32) OpenSSL/1.0.1i PHP/5.6.11 for our web server, PHP as a programming language, and phpMyAdmin with (version 4.8.4) to manage MySQL via the Web are all included in the package. This study employed four datasets from the dense categories; Mushroom, Pumsb_star, Chess. The benchmark datasets were collected in their raw form from the Frequent Itemset Mining Dataset Repository (http://fimi.ua.ac.be/data/). The benchmark is also converted into Structured Query Language (SQL)
format to make things easier. Table 1 summarises the attributes of each dataset.

Table 1: Dataset Attributes.

Dataset	No. of Transactions	Length (Attributes)	Size (KB)	Category
Mush-room	8125	43	558	Dense
Pumsb star	49046	57	11526	Dense
Chess	3196	37	335	Dense

To facilitate and accelerate research, we limited datasets to a thousand rows of randomly processed item sets for mining purposes. Our research is focused on R-Diffset and IR-Diffset. Figure 10 depicts the performance assessment graph in terms of execution time (in seconds) for three (3) datasets: mushroom, pumsb_star, and chess. For the R-Diffset, the chess dataset displayed a very high performance in terms of execution time that is (419.08 second), compared to the mushroom dataset (229.26 second), the pumsb_star dataset displayed a performance in terms of the execution time of the which very lowest that is (219.96 second). After that, for the IR-Diffset anyway, the performance in terms of execution time for the chess dataset is very high that is (127.69 second), compared to another dataset that is the mushroom dataset (74.36 second), pumsb_star dataset displayed a performance in terms of the execution time of the which very lowest that is (71.15 second). Thus, among three (3) dataset that is mushroom, pumsb_star, and chess, the average performance in terms of the execution time of IR – Diffset is better than R – Diffset.

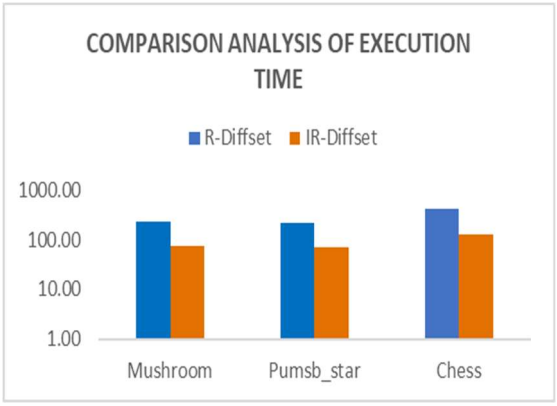


Figure 10: Performance on R-Diffset and IR-Diffset in mushroom, pumsb_star, and chess

7. CONCLUSION

In this research, we studied the variants associated with R-Diffset that employs infrequent itemsets mining. According to the previous study, the execution time of data processing in the R-Diffset method is time-consuming. This study presents a modify algorithm that is an incremental approach in R-Diffset named as Incremental R-Diffset (IR-Diffset) which is an additional strategy to improve the current R-Diffset to reduce the disadvantage of time consumption in infrequent itemsets mining. This strategy also offers a novel alternative for speeding up processing time in infrequent itemsets mining, particularly in large datasets.

ACKNOWLEDGEMENTS

We would like to express our gratitude to the CREIM of UniSZA for providing financial support under UniSZA internal grant code (UniSZA/2021/GOT/02). Thanks to the corresponding authors, Dr. Julaily Aida Jusoh as the CREIM-UniSZA research project leader, Sharifah Zulaikha Tengku Hassan as the research assistant, and the grant participants. We also like to thank all of the faculty members who helped us evaluate spelling problems and synchronization consistency and for their thoughtful remarks and recommendations.

REFERENCES

[1] M. Man, J. A. Jusoh, S. I. A. Saany, W. A. W. A. Bakar, and M. H. Ibrahim, "Analysis Study on R-ECLAT Algorithm in Infrequent Itemsets Mining", International Journal of Electrical and Computer Engineering (IJECE), pp. 5446-5453, 2019.

[2] Agrawal R, Imieliński T, Swami A., "Mining association rules between sets of items in large databases". Proceedings Acm sigmod record, vol. 22, no. 2, pp. 207-216, 1993.

[3] Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

[4] Han J, Pei J, Yin Y. "Mining frequent patterns without candidate generation," Proceedings ACM sigmod record, vol. 29, No. 2, pp. 1-12, 2000.

[5] Ogihara Z. P., Zaki M. J., Parthasarathy S., Ogihara M., Li W., "New algorithms for fast discovery of association rules," 3 rd Intl. Conf. on Knowledge Discovery and Data Mining, 1997.

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

[6] Jusoh, J. A., Man, M., & Bakar, W. (2018). Mining infrequent patterns using R-ECLAT algorithms. *J Fundam Appl Sci*, *24*.

[7] Jusoh, J. A., & Man, M. (2018). Modifying iECLAT Algorithm for Infrequent Patterns Mining. *Advanced Science Letters*, *24*(3), 1876-1880.

[8] Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997, August). New algorithms for fast discovery of association rules. In *KDD* (Vol. 97, pp. 283-286).

[9] Zaki, M. J., & Gouda, K. (2003, August). Fast vertical mining using diffsets. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 326-335).

[10] Trieu, T. A., & Kunieda, Y. (2012, February). An improvement for dECLAT algorithm. In *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication* (pp. 1-6).

[11] Ghonge, M., & Rane, M. N. (2018). Mining Rare Patterns by Using Automated Threshold Support. International Journal of Engineering & Technology, 7(3.8), 77-81.

[12] Li, Z. F., Liu X. F., & Cao, L. (2011). A Study on Improved ECLAT Data Mining Algorithm. Advanced Materials Research. 328-330. 1896-1899.

[13] Bakariya, B., Thakur, G., & Chaturvedi, K. (2019). An Efficient Algorithm for Extracting Infrequent Itemsets from Weblog. International Arab Journal of Information Technology, 16(2), 275-280.

[14] Borah, A., & Nath, B. (2017). Rare Association Rule Mining: A Systematic Review. International Journal of Knowledge Engineering and Data Mining. 4(3-4). 204-258. ACM.

[15] Han J., Pei J., Yin Y., Mao R. (2004). Mining Frequent Patterns Without Candidate Generation: A Frequent Pattern Tree Approach. Data Mining and Knowledge Discovery, 8(1), 53–87.

[16] Szathmary, L., Valtchev, P., Napoli, A., & Godin, R. (2012, October). Efficient Vertical Mining of Minimal Rare Itemsets. In *CLA* (pp. 269-280).

[17] Szathmary, L., Valtchev, P., Napoli, A., Godin, R., Boc, A., & Makarenkov, V. (2014). A fast compound algorithm for mining generators, closed itemsets, and computing links between equivalence classes. *Annals of Mathematics and Artificial Intelligence*, *70*(1), 81-105.

[18] Darrab, S., & Ergenc, B. (2017). Vertical pattern mining algorithm for multiple support thresholds. *Procedia computer science*, *112*, 417-426.

[19] Zaki M. J., Gouda K., "Fast vertical mining using diffsets," Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 326-335, August 2003.

[20] Trieu, T. A., & Kunieda, Y. (2012, February). An improvement for dECLAT algorithm. In *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication* (pp. 1-6).

[21] W. A. W. A. Bakar, M. A. Jalil, M. Man, Z. Abdullah, and F. Mohd., "*Postdiffset: An ECLAT-like algorithm for frequent itemset mining,*" International Journal of Engineering & Technology 7, no. 2.28, pp. 197-199, 2018.

[22] M. Man, W. A. W. A. Bakar, M. M. A. Jalil, and J. A. Jusoh, "*Postdiffset Algorithm in Rare Pattern: An Implementation via Benchmark Case Study,"*, International Journal of Electrical and Computer Engineering (IJECE), pp. 4477-4485, 2018.

[23] Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE transactions on knowledge and data engineering*, *12*(3), 372-390.

[24] Li, Z. F., Liu, X. F., & Cao, X. (2011). A study on improved ECLAT data mining algorithm. In *Advanced Materials Research* (Vol. 328, pp. 1896-1899). Trans Tech Publications Ltd.

[25] Trieu T. A., Kunieda Y., "An improvement for dECLAT algorithm," Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, no. 54, pp. 1-6, February 2012.

[26] A. Borah and B. Nath, "*Rare Pattern Mining: Challenge and Future Perspectives,*" Springer International Publishing, 2018.

[27] D. Cheung, J. Han, V. Ng, C.Y. Wong, *Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique,*" in Proceeding of the 12th Intl. Conf. on Data Engineering, 1996.

[28] R. Hernandez, J. hernandez, J. A., J. Fco, "*A Novel Incremental Algorithm for Frequent Itemsets Mining in Dynamic Datasets,*", pp. 145-152, 2008.

[29] M. K. Yusof and M. Man, "Efficiency of JSON for Data Retrieval in Big Data," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 7, no. 1, pp. 250–

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

262, Jul. 2017, doi: 10.11591/IJEECS.V7.I1.PP250-262.

[30] A. Veloso, W. Gusm˜ao, Jr. W. Meira Jr., W., de Carvalho, S. Parthasarathy, M. J. Zaki, "*Parallel, Incremental and Interactive Mining for Frequent Itemsets in Evolving Databases,*" in Intl. Workshop on High Performance Data Mining: Pervasive and Data Stream Mining, 2003.

[31] W. Cheung, O. R. Zaiane, "*Incremental mining of frequent patterns without candidate generation or support constraint,*" in Proceedings of the Seventh IEEE International Database Engineering and Applications Symposium, pp. 111–116, 2003.

[32] C. K. Leung, I. K. Quamrul, T. Hoque, "*CanTree: A Tree Structure for Efficient Incremental Mining of Frequent Patterns,*" in Proceedings of the Fifth IEEE International Conference on Data Mining, 2005.

[33] T. H. Hai, L. Z. Shi, "*A New Method for Incremental Updating Frequent Patterns Mining,*" in Proceedings of the Second International Conference on Innovative Computing, Information and Control, pp. 561, 2007.

[34] W. A. W. A. Bakar. "*An Enhanced ECLAT Algorithm Based on Incremental Approach for Frequent Itemset Mining*," Ph.D thesis, Universiti Malaysia Terengganu, 2015.

[35] Bakar, W. A. W. A., Jalil, M. A., Man, M., Abdullah, Z., & Mohd, F. (2018). Postdiffset: An ECLAT-like algorithm for frequent itemset mining. *International Journal of Engineering & Technology*, *7*(2.28), 197-199.

[36] Bakar, W. A. W. A., Man, M., Man, M., & Abdullah, Z. (2020). I-ECLAT: Performance enhancement of ECLAT via incremental approach in frequent itemset mining. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *18*(1), 562-570.

[37] Man, M., Ruslan, N. A. B., Bakar, W., Jusoh, J. A., Yusof, M. K., & Josdi, N. L. N. B. (2022). IR-ECLAT: A NEW ALGORITHM FOR INFREQUENT ITEMSET MINING. *Journal of Theoretical and Applied Information Technology*, *100*(11), 3545-3551.

[38] Jusoh, J. A.. Man, M., Bakar, W., Rahman, M. N. A., & Hassan, S. Z. T. (2022). PARALLEL APPROACH IN R-DIFFSET ALGORITHM FOR INFREQUENT ITEMSET MINING. *Journal of Theoretical and Applied Information Technology*, *100*(4), 4761-4770.

[39] A. Aziz, N. H. Ismail, F. Ahmad, Z. Abidin, K. G. Badak, and M. Candidate, "MINING STUDENTS' ACADEMIC PERFORMANCE," *undefined*, 2013.

[40] W. A. B. W. A. Bakar, et al., "Incremental-Eclat Model: An Implementation via Benchmark Case Study," Springer International Publishing Switzerland, P.J. Soh et al. (eds.), Advances in Machine Learning and Signal Processing, Lecture Notes in Electrical Engineering, vol. 387, pp. 35-46, 2016.