Journal of Theoretical and Applied Information Technology
15th June 2022. Vol.100. No 11

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

IR-ECLAT: A NEW ALGORITHM FOR INFREQUENT ITEMSET MINING

MUSTAFA MAN[*1], NUR AQILAH BT RUSLAN[*2] WAN AEZWANI WAN ABU BAKAR[*3], JULAILY AIDA JUSOH[4], MOHD. KAMIR YUSOF[5], NUR LAILA NAJWA BT JOSDI[6]

[1,2] Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu (UMT), 21030 Kuala Nerus, Terengganu, Malaysia

[3,4,5,6] Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA) Besut Campus, 22200 Besut, Terengganu, Malaysia

E-mail: mustafaman@umt.edu.my[*1], aqilah@umt.edu.my[*2], wanaezwani@unisza.edu.my[*3], julaily@unisza.edu.my[4], mohdkamir@unisza.edu.my[5], laila455659@gmail.com

ABSTRACT

The key challenge of association rule mining (ARM) is to discover and extract a valuable information from databases. Mining valuable information from database could be very challenging especially for decision-making process. This is because mining association rule may require repetitious scanning of large dataset in the databases that can lead to high memory usage and time processing. A few algorithms were introduced by researchers to handle these related problems in data mining. The Incremental Equivalence Class Transformation (I-Eclat), Rare Incremental Equivalence Class Transformation (R-Eclat) algorithm are example of algorithms in rule mining techniques using vertical format data repositories for frequent and infrequent pattern mining. The main operation in I-Eclat and R-Eclat are intersecting tidset. Since the size of tidsets would affect the memory usage and its execution time, more memory and time required for a bigger tidsets. This paper introduces a new incremental of rare pattern mining approach by adopting R-Eclat called Incremental Rare Equivalence Class Transformation (IR-Eclat). The IR-Eclat specifically designed for infrequent pattern mining, and it is beneficial for dynamic database as the data is increasing towards volume with linear proportionate into time. In conjunction with big data explosion, the end users are at an advantage for the use of this incremental approach. The experimental results on several benchmark datasets indicate that IR-Eclat outperforms compare with R-Eclat by reducing its processing time especially in dynamic database as the data is increasing in volume from time to time.

Keywords: *Data Mining, Equivalence Class Transformation (Eclat), I-Eclat Model, R-Eclat, IR-Eclat, Interestingness Measure*

1. INTRODUCTION

Mining in the scope of data dictates a process of unsolved and hidden pattern extraction in a database that is useful for crucial decision making. It involves integrative research field among database theory, information system, data science, and statistics. The objective(s) is to comprehend the past or/and to plan the future. Prediction is necessary for estimating future work based on existing data. Hence, various mining of data techniques focuses at discovering patterns and some of the most common pattern detected in databases are classes, common clusters, itemsets, trends, and unwanted outliers [1]. Data mining are divided into two major categories: predictive task and descriptive task [2][3]. The predictive task aims to perform a prediction value of a certain attribute that lies under the other attribute's value by implementing several techniques; classification, regression, time series analysis, and prediction. On the other hand, descriptive task focuses on deriving patterns (clusters, correlations, anomalies, and trends) to extract the underlying relationships in the database. The techniques used in descriptive task include clustering, summarizing, association rules pattern detecting, and sequence discovering. Data mining implicit knowledge, focusing on interesting association relationship among set of objects, lead to the association rule that may reveal useful information. This information undoubtedly will aid in financial forecasting, future decision-making, medical diagnostics, marketing policy, and a variety of other applications. This research aims to contribute to the incremental association rule mining for rare pattern analysis in the attempt to reduce execution time taken during the process of intersecting itemsets in generating candidate itemsets as well as minimizing the memory usage.

The rest of the sections are organized as follows. Section 2 presents the review the previous literatures, Section 3 denotes the Eclat and R-Eclat basic algorithm, Section 4 introduces the new enhanced design of IR-Eclat model. Next Section 5 discusses the experimentation and results achieved while Section 6 summarizes the conclusions as well as future recommendations.

2. RELATED WORKS

The idea of huge databases can be seen as data mines holding valuable information that can be discovered using association rules technique. Association rule mining (ARM) is introduced by [4][5]. The initial objective of ARM is to discover whether a common pattern or itemset (a collection of one or more items) can be found in a particular database. Other reason is to investigate whether any existence of interesting relationship between these itemsets that have frequent occurrences (an itemset whose support is greater than or equal to a minimum support threshold or also denotes a frequency). This may deduce to a new pattern discovery for crucial management decision-making. The rules of association are only the if-then-else statements that may yield some relationships between those unrelated data in relational database or other data repositories. It can be defined as $Itemsets = \{i_1, i_2, \ldots, i_n \; for \; |n| > 0\}$ is a set of items. A database of transactions defined as D. The transaction T simplifies a set of items where $T \subseteq Itemsets$. Tid is a unique identifier together with every transaction in D. For a given transaction database T, the rule of association exists in the form of $X \subseteq Y$ where the portion of antecedent is represented by X and Y represents the consequent portion of the rule where $X \subseteq I, Y \subseteq I$ and $X \cap Y = \emptyset$. The rule of association statement would be *"If a customer buys a sedan car, he would 80% likely to also buys the steering lock"*. Support, c with the confidence, c are the filtering criteria that used to mine association rules. The $X \Rightarrow Y$ rule in transaction set D with confidence c if c% of transaction in D containing X that also contain Y.

$$Confidence\ (X \Rightarrow Y) = \frac{(Support\ (X \cup Y))}{(Support\ (X))} \quad (1)$$

The $X \Rightarrow Y$ rule holds in transaction set D has support s if s% of the transaction in D contain $X \cup Y$. The support $(X \Rightarrow Y)$ equals to the fraction of both $X \cup Y$ into $|D|$ where $|D|$ is the summation of total transactions in database.

$$Support\ (X \Rightarrow Y) = \frac{(X \cup Y)}{|D|} \quad (2)$$

Itemset Mining is extensively used in data mining and knowledge discovery in big data. Two significant patterns that are trending in itemset mining are frequent and infrequent. Infrequent itemsets may indicate unpredicted or previously unidentified relationships, whereas frequent itemsets represent known and predictable facts. Certain domain sectors still find infrequent itemsets as useful and important as the frequent itemsets. It is possible that obtaining rare information or data has appeared as a vital component, particularly in terms of extracting an accelerating data into useful knowledge. Finding infrequent itemsets is an important part of producing fascinating data for a specific demand in a fast processing time. Furthermore, technological advancements are now leading to the adoption of DBMS platforms for data storage and a high demand for mining processing in a short period of time. However, mining association rule may require repetitious scanning of large databases [4] [5] that leads to the high memory usage and affects the running time.

3. ECLAT AND R-ECLAT ALGORITHM

Equivalence Class Transformation (ECLAT) algorithm is proposed by [6] for mining frequent pattern which deploys under depth-first (DF) searching and a vertical data layout to represent the transaction database. The ECLAT algorithm is the founder or initiator of a vertical data layout database implementation and working efficiently for mining large itemsets regardless of lesser efficiency for small number of itemsets. Work done by [6][7] show shows that vertical database layout has great advantages over horizontal database layout. Computing supports of itemsets is a lot easier and way faster because vertical data layout involves only intersections of tids and the number of tids automatically reindicates the support. The automatic "reduction" of the database prior each scan where only relevant itemsets are accessed from disk for next scan of the mining process. An important component of ECLAT is the fast intersecting of tids list that contributes to the main reason for reduction in most machine execution time and memory usage [1]. There are four (4) variation of this algorithm which is Tidset [6], Diffset [7], Sortdiffset [8], and the latest variant of ECLAT is Postdiffset [9][10]. Tidset is implemented by intersecting items within the itemsets while Diffset uses the difference of tidset to extract the rules. Sortdiffset is a process of combining Tidset and Diffset and then sorts the Diffset in the order of descending. Postdiffset approach is starting with Tidset process for the first level loop, then it switches to the Diffset for the second level ahead.

Using the same concept of ECLAT with some modification on the algorithm, R-Eclat (Rare Equivalence Class Transformation) [1][10] is introduced and the focus is mining infrequent itemsets in large database. The R-Eclat algorithm also applies depth-first matching of tids in a vertical database formatting to represent itemsets in the transaction database. A set of transaction IDs (called a tids) whose transactions contain the item. It represents dataset in a column format (vertically) instead of row (horizontal) format. The R-Eclat determines support of m-itemsets on the intersecting tid-list of its m-1 subsets. The traditional algorithm or called as Tidset, Diffset, Sortdiffset, and Postdiffset are all the variants of Tidset in such a way having a modification and alteration to make sure it is compatible to mine for rare itemset.

The support for each itemset is count by undergo all transactions in the transaction database while it is inspecting if the transaction contains the related itemsets. R-Eclat is the implementation of intersection Tidset, so the bigger the size of Tidsets, the higher the time consumed and memory usage of R-Eclat to complete the process. R-Eclat approach consist of two pre-processing stages. At first, all the algorithms in association rule mining related vertical mining are checked to determine the suitable algorithm for itemset mining and the last part of pre-processing, an algorithm modification process has occurred. All four variants of Eclat algorithms are transformed to assure it is compatible for mining among rare and peculiar pattern. Figure 1 shows the architecture model of R-Eclat.

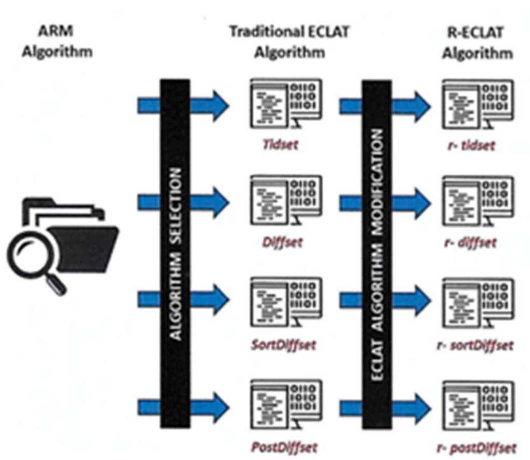

Figure 1: R-Eclat model

Since R-Eclat is designed based on previous Eclat algorithm that consist of four variants, it also comes with four varieties: R-Tidset, R-Diffset, R-Sortdiffset, and R-Postdiffset. The advantage of R-Tidset is the size of tidsets represents the support and R-Tidset perform vertical intersection of tidlist. In R-Diffset, it only keeps track of differences in tidsets, thus make the intersection faster and less memory usage. In R-Sortdiffset, it is a combination of R-Tidset and R-Diffset, then tidset is sorted in ascending order while Diffset is sorted in descending order. There is no need for switching condition, hence reduce its running time and memory usage. Last variant R-Postdiffset is the combination of R-Tidset and R-Diffset, which offer a better performance among the others. In this paper, authors will focus on R-Tidset variant only. This would make it easier to understand the basic concept of R-Eclat as the algorithm is more straightforward compared to the other variants. However, the main drawback of R-Tidset is difficult in pruning technique as the longer the tidset, the more time and memory is needed. Figure 2 illustrated the chronology of infrequent pattern mining.

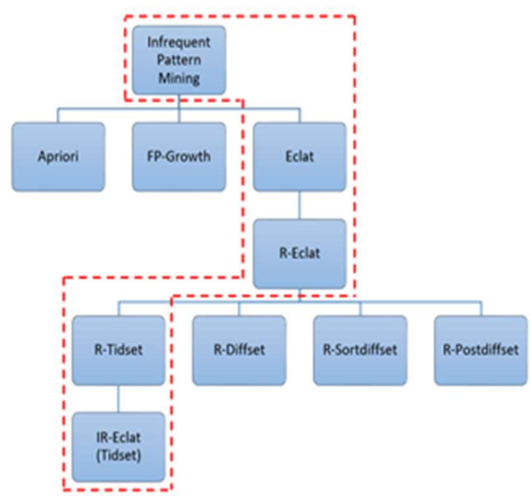

Figure 2: Infrequent Pattern Mining Chronology

4. NEW IR-ECLAT ALGORITHM

R-Eclat algorithm mostly operate on transaction databases that are static in nature [11]. Static database is a case where no addition or deletion of the items or transactions hence they are not updated on a regular basis. Mining infrequent itemset from dynamic database is more complex since the database are constantly changed as the transactions are added or removed. Former incremental updating strategy was called Fast Update (FUP) [12] and it is

Journal of Theoretical and Applied Information Technology
15th June 2022. Vol.100. No 11

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

focused in dealing with new additional transaction data [13]. Other strategies introduced for incremental mining are ZIGZAG [12], FIIU [13], DFIIU [13], CATS Tree [14], Parallel Incremental [15], INUP-Tree [16], CanTree [17] and Incremental Updating [18]. All these incremental update strategies are targeted for dynamic database mining purposes.

The incremental of R-Eclat algorithm (IR-Eclat) is a technique that specially generated for infrequent itemset mining. It is modified based on the traditional incremental Eclat in [19] and R-Eclat algorithm [20] by embedding the variance of Postdiffset algorithm in [21-22]. This initiative serves for enhancement in the incremental approach. The research design for the IR-Eclat is as follows: First, the itemsets are mined through four (4) variants i.e., R-Eclat-Tidset, R-Eclat Diffset, R-Eclat-Sortdiffset and R-Eclat-Postdiffset. Secondly, the resulting itemsets at each variant are then mined with the IR-Eclat engine respectively before the final pattern of rules are generated. Figure 3 shows the enhanced architecture model of IR-Eclat. This algorithm would be beneficial to mining dynamic database. The incremental approach is advantageous for dynamic database which subject to addition or deletion of items or record of transaction in the database.

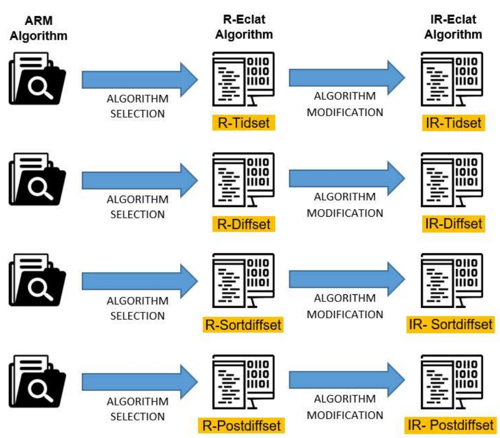

Figure 3: IR-Eclat Model

Generate and prune are 2 core steps in IR-Eclat strategy. First step is generation; the database is scanned to form $m - itemset$ candidates from the frequency of two $(m-1) - itemsets$ and its support is counted. Second step is pruning; checking mechanism where if the support is greater than minimum support threshold, then it will be pruned, otherwise if the support is lesser than minimum support threshold, then it is denoted as infrequent itemsets and used to generated $(m+1) - itemsets$. The first step will continue until no more candidate itemset can be generated. Here, the consideration of minimum support threshold value (MSTV) is labeled as benchmark threshold value to discover a low occurrence in each dataset. The value is determined in terms of percentage,

$$\frac{\alpha}{100} * \beta \tag{3}$$

where: α = Threshold value for minimum support
β = Number of transactions in database

```
IR-Tidset Pseudocode
Input: E((i_1, t_1), … (i_n, t_n))|P), s_min
Output: IF(E, s_min)

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 intersect 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 4: IR-Tidset Algorithm

```
IR-Diffset Pseudocode
Input: E((i_1, t_1), … (i_n, t_n))|P), s_min
Output: IF(E, s_min)

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 5: IR-Diffset Algorithm

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

start      //get minimum support
sort data by itemset in descending order
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 6: IR_Sortdiffset Algorithm

```
IR-Postdiffset Pseudocode
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)
        run tidset in first loop
                add data to the next process
                add data into db
        run diffset in next looping
                add data to the next process
                add data into db
                end looping
                end diffset
        write to text file the value for the current / last
        transaction data;
end
```

Figure 7: IR-Postdiffset Algorithm

Fig. 4 to Fig. 7 indicate the step-by-step actions taken in IR- Eclat algorithm for the following IR-Tidset, IR-Diffset, IR-Sortdiffset, and IR-Postdiffset algorithm. The main difference of IR-Eclat algorithm compared to previous R-Eclat algorithm is the database will be processed by batches. For example, 50000 records in a database. 5000 records will be processed first, then followed by next 5000 records, until all records have been processed. Other than that, the original process steps in each version are maintained. The MSTV is the benchmark to determine the occurrence of infrequent itemsets in the database hence the support counting for every process is constant. In each loop, begin with the first loop, when support, s is less than or equal to min_supp, ($s <= min\_supp$) then there are 4 conditions that follows such that,

i. *If in IR-Tidset, the candidate itemset is the result of the intersection between z^{th}column and z^{th}+1 column and save to the database.*

ii. *If in IR-Diffset, the candidate itemset is obtained from the result of Diffset (difference in the intersection process is denoted as candidate) between z^{th}column and z^{th}+1 column before saving to database*

iii. *If in IR-Sortdiffset, the itemsets are sorted in descending order in terms of its value from the highest to the lowest of equivalence class. The Diffset value between z^{th}column and z^{th}+1 column is identified prior to saving in the database*

iv. *If in IR-Postdiffset, Tidsets process is commenced in initial looping before the Diffset process in the forwarding level onwards between z^{th}column and z^{th}+1 column and save the IR-Postdiffset value into database.*

5. EXPERIMENTS AND DISCUSSIONS

This experiment performed on Acer, AMD A6-7310 APU with AMD Radeon R4 graphics @ 2.00GHz with 4GB RAM in Windows 10 64-bit platform OS. The experiment uses real datasets of disease database that retrieved from Korean Disease Control and Prevention Agencies (http://www.kdca.go.kr/) in Domestic Infectious Disease Occurrence archive to evaluate the performance of IR-Eclat algorithm. The dataset is converted to Structured Query Language (SQL) format and then pre-processed by removing the instances consist of incomplete data and attributes of only one categorical value for experimentation purposes. The disease dataset consists of 8 items with 17 attributes and contain 74 records of transactions. This dataset is categorized as dense dataset and the size of the file is 368 kilobytes. For experimental purpose, the minimum support threshold value is set at 3%.

The experiment includes all IR-Eclat algorithm variants comprises IR-Tidset, IR-Diffset, IR-Sortdiffset, and IR-Postdiffset and the same datasets used for previous R-Eclat algorithm in each variant. The performance of both algorithms is measured in terms of its runtime for disease dataset. Figure 8 shows the performance evaluation on execution time for disease dataset for both algorithms. Based on the observation of performance analysis for both algorithms as illustrated in Figure 3, the execution time for IR-Tidset reduced by 63.7% compared to R-Tidset. For IR-Diffset, the execution time dropped 76.3% to R-Diffset. Among all four variants, R-Sortdiffset consumes the highest execution time as compared to all R-Eclat variants, thus Sortdiffset variant depict the highest contrast which IR-

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

Sortdiffset reduced its execution time up to 84.9% compared to R-Tidset. The execution time for both R-Postdiffset and IR-Postdiffset consume shorter execution time among the other variants. However, the execution time of IR-Postdiffset is 56.7% faster than R-Postdiffset. The results illustrate that Postdiffset performs best in both algorithms. This may suggest that the Incremental approaches is relatively surpass the performance of traditional R-Eclat.

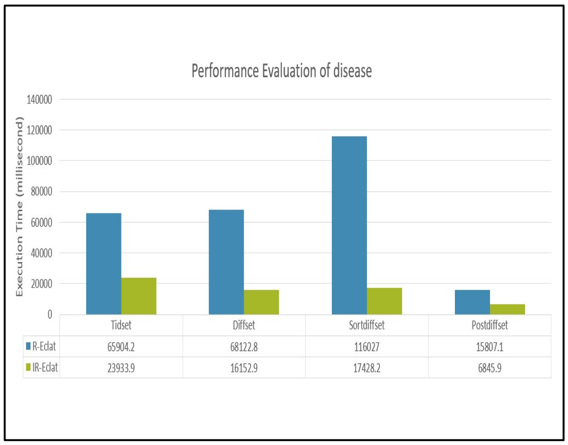

Figure 8: Performance Evaluation of Disease Database

6. CONCLUSION

The popularity of infrequent itemset mining has arisen and receive a great attention among researchers to establishing the most efficient way for mining infrequent patterns. In this paper, we discussed about R-Eclat algorithm and proposed a new incremental R-Eclat algorithm to improve the current R-Eclat algorithm that would benefit to mining dynamic database. The incremental approach is advantageous for dynamic database which subject to addition or deletion of items or record of transaction in the database. The implementation of incremental mining is to apply previously mined knowledge and scanning of a basic incremented database. The results obtained show that the performance of IR-Eclat algorithm is efficiently reduced in term of its execution time in contrast with the formal R-Eclat algorithm in all four variants. This can be seen in the minimum of 50% to 90% reduction of execution time in comparison of R-Eclat versus IR-Eclat in all four variants (Tisdet, Diffset, Sortdiffset and Postdiffset). However, further experiment and evaluation is needed to test for its compatibility to different type of databases. The results portrayed are subject to the selected database. This research may have limitation by having different and contradict results in time execution if testing in other datasets and with different machine specification.

ACKNOWLEDGMENT

We would like to express our gratitude to the Fundamental Research Grant Scheme (FRGS) for the financial support under Ministry of Education Malaysia for grant with VOT: FRGS/1/2018/ICT04/UMT/02/1(UMT: 59543). A sincere thanks to the corresponding authors (Assoc. Prof. Dr. Mustafa Man as the FRGS-UMT leader, Dr. Wan Aezwani Wan Abu Bakar as the editor and Miss Nur Aqilah Ruslan as the FRGS postgraduate student) and the grant members, Dr. Julaily Aida Jusoh and Dr. Mohd. Kamir Yusof. To all collaborators from UniSZA and UMT for networking and technical support in proofreading for spelling errors and synchronization inconsistencies.

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] P. -N. Tan, M. Steinbach, and V. Kumar, " *Introduction to Data Mining*," Pearson Addison-Wesley, pp. 25-70, 2016.

[3] J. Han, M. Kamber, and J. Pei, "Classification: Basic Concepts," in *Data Mining: Concepts and Techniques*, pp. 326-335, 2011.

[4] R. Agrawal, T. Imielinski, and A. Swami, "*Mining Association Rules Between Sets of Items in Large Databases," in* ACM SIGMOD International Conference on Management of Data, pp. 207-216, 1993.

[5] R. Agrawal, and R. Srikant, "*Fast Algorithms for Mining Association Rules in Large Databases," in* Proceedings of 20th International Conference on Very Large Data Bases (VLDB), pp. 487- 499, 1994.

[6] M. J. Zaki, S. Parthasarathy, M. Ogihara, & W. Li, "*New Algorithms for Fast Discovery of Association Rules," in* Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, pp. 283-286, 1997.

[7] M. J. Zaki and K. Gouda, "*Fast Vertical Mining Using Diffsets," in* ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.

[8] T. A. Trieu and Y. Kunieda, "An Improvement for Declat Algorithm," in 6th International Conference on Ubiquitous Information Management and Communication, New York, 2012.

[9] M. Man, J. A. Jusoh, M. A. Jalil, W. A. W. A. Bakar, and Z. Abdullah, "*Postdiffset: An Eclat-Like Algorithm For Frequent Itemset Mining,"* in International Journal of Engineering & Technology, pp. 197-199, 2018.

[10] 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.

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

[12] 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.

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

[14] A. Veloso, Jr. W. Meira, M. B. de Carvalho, B. Possas, S. Parthasarathy, M.J. Zaki, "*Mining Frequent Itemsets in Evolving Databases," in* Proceedings of the 2nd SIAM Intl. Conf. on Data Mining, 2002.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] J. A. Jusoh. "A New R-Eclat Algorithm for Infrequent Itemset Mining," Ph.D thesis, Universiti Malaysia Terengganu, 2019.

[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] W. A. W. A. Bakar, M. Man, M. Man, and Z. Abdullah, "i-Eclat: performance enhancement of Eclat via incremental approach in frequent itemset mining," *Telkomnika* 18, no. 1, pp. 562-570, 2020