CRS-iEclat: Implementation of Critical Relative Support in iEclat Model for Rare Pattern Mining

Wan Aezwani Wan Abu Bakar[1]
Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin (UniSZA)
Besut Campus, 22200 Besut, Terengganu, Malaysia

Zailani Abdullah[3]
Faculty of Entrepreneurship and Business (FEB)
Centre of Computing and Informatics (CCI), Universiti Malaysia Kelantan (UMK), City Campus, Malaysia

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

Mahadi B Man[4]
Faculty of Ocean Engineering Technology and Informatics
Universiti Malaysia Terengganu (UMT)
21030 Kuala Nerus, Terengganu, Malaysia

***Abstract*—The research purpose is to develop a performance enhancement in Incremental Eclat (iEclat) model by embedding Critical Relative Support (CRS) in mining of infrequent itemset. The CRS measure acts as an interestingness measure (filter) in iEclat model that comprises of i-Eclat-diffset algorithm, i-Eclat-sortdiffset algorithm and i-Eclat-postdiffset algorithm for infrequent (rare) itemset mining. The association rule is performed to reveal the relationships among itemsets in a transactional database. The task of association rule mining is to discover if there exist the frequent itemset or infrequent patterns in the database and if any, an interesting relationship between these frequent or infrequent itemsets can reveal a new pattern analysis for the future decision making. Regardless of frequent or infrequent itemsets, the persisting issues are deemed to execution time to display the rules and the highest memory consumption during mining process. CRS-iEclat engine is proposed to overcome the said issues. Prior to experimentation, results indicate that CRS-iEclat outperforms iEclat from 54% to 100% accuracy on execution time (ET) in selected database as to show the improvement of ET efficiency.**

Keywords*—Critical relative support; equivalence class transformation (Eclat); iEclat model; interestingness measure

I. Introduction

Association Rule is among the four (4) core domains in Data Mining. The rule or pattern generated determines the associations or similar structures among sets of items in the database transaction. Correlation or association allows the tendencies between one item and another item in one particular set of items in a typical dataset. The association rule implementation can be seen in market basket analysis to predict the potential item buying by customers, remedial medications for no vaccine disease, biological cells actions that constitutes to certain disease symptoms, offering banking or retail services [1-2]. There are two categories of item i.e. Frequent itemset (frequent occurring) and infrequent (rare occurring) itemset. Main contribution of frequent itemset is finding frequent correlation of items that constitutes to certain pattern in database transactions while infrequent itemset is finding the contradiction or peculiar or rare pattern. To determine either the itemset is frequent or infrequent, one threshold value must be set that is called minimum support (min_supp) or the maximum support (max_supp) where these values are pre-defined user settings. When the itemset is above min_supp, then it is considered as frequent itemset and vice versa. While frequent itemset discovers the normal operations i.e. buying types or disease occurrences, the rare itemset in contrast finds abnormal and peculiar association and correlation of abnormal itemsets. This abnormal consolidation may discover hidden or new findings that require for further attention by domain experts. Further investigation of the rare patterns generated would provide solutions for a significant difficulty through formulation in association rule mining algorithms. Setting of rare patterns depending upon certain predefined threshold value considering on lower than minimum occurrences of the itemsets from database transactions.

The rest of the sections are organized as follows. Section 2 describes the previous literatures, Section 3 illustrates the Eclat basic principles, Section 4 explains the design of iEclat model. Next Section 5 prescribes the experimentation settings while Section 6 discusses on the results achieved. Section 7 summarizes the conclusions as well as future recommendations.

II. Related Work

Regardless either mining data via frequent or infrequent association, the critical issues still remains on memory space consumption and data storage capacity [3-5]. To reduce memory and data consumption during mining process, the previous researches have made effort on the 2 searching strategy i.e. horizontal database record or breadth first searching [6] and vertical database record [7-8] or depth first searching. When the horizontal record drawback issues are subjected to storage and memory, thus contemporary works are then utilized on the vertical database for rules mining algorithms that are proposed in [8-10]. In ARM, the so-called state-of-the-art frequent/infrequent models are Apriori [1, 6] underlying on horizontal records. Meanwhile Eclat [9] and FP-Growth [14] are vertical database records practitioners.

To the best of our knowledge, Equivalent Class Transformation (Eclat) algorithm [8] outperforms because of its 'fast' intersection of its transaction-id-list to determine the minimum or maximum support threshold [9, 14]. The Eclat followers and the invariants are [9-13], [15-20], [22] and [26].

In response to its simple and quick method in finding the threshold value as the interestingness measure in mining, we have done an improvement in original Eclat where we have proposed Incremental Eclat (iEclat) model in our previous work [20], [22] and [26]. To continue, this research presents a deployment of Critical Relative Support (CRS) as the interestingness measure or filtering or pruning method in our Incremental Eclat (iEclat) model. Our proposed solution, CRS-iEclat algorithm is used in selected dense dataset to improve the performance of execution time.

III. Basic Principles of Eclat

Eclat works in two-steps i.e. first, generate candidate itemsets during intersecting and second is pruning. In step 1, each i-itemset candidate is generated by (i-1)-itemset and the number of frequency occurrences (support) are calculated. If the support<threshold, then pruning/removing it, if not, then is frequent itemsets later is set to generate (i+1)-itemset. Because of its depth first searching, start with the frequent items in the item base 1-itemset, then move to 2-itemset, next is 3-itemset and continues until all the depths of itemset trees are visited. The four algorithms underlying in i-Eclat model are tidset [9], diffset [9], sortdiffset [12] and postdiffset [20, 22, 26].

A. Original Eclat (tidset)

The i-itemset formulates when joining of (i-1)-itemset which have similar (i-2)-itemset, both (i-1)-itemsets are named as superclass itemsets of the i-itemset. Let {}, {ab} and {ac} are superclass of {abc}. To get rid of duplication, (i-1)-itemset are arranged in some order. For example, itemset {a, b, c, d, e} are arranged into alphabet order. Finding all 2-itemsets, items {a} is joined with {b,c,d,e} resulting into {ab, ac, ad, ae} then for the union of {b} with {c,d,e} resulting in {bc, bd, be}, similarly for {c} and {d}. Lastly, all candidate of 2-itemsets {ab, ac, ad, ae, bc, bd, be, cd, ce, de} are formulated that later used for formulation of 3-itemsets. The union process continues to the higher depths of itemset trees and finish when all items in the itemsets are visited. These operations are illustrated in Fig. 1.

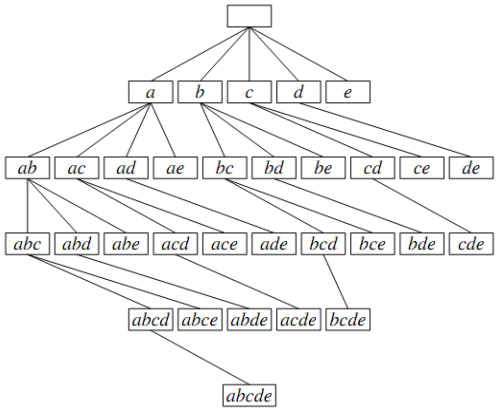

Fig. 1. Example of Candidate Generation with {a,b,c,d,e}.

B. Eclat (diffset)

The dEclat (where d stands for different set or diffset) as referred in [9] is the process of finding the different in prefix items among 2 tidsets (tid of itemsets and its prefix). When finding prefix items that differs, the matching correlation (cardinality) of itemsets is lesser and fasten the intersecting process and reduce memory consumption because candidate itemsets is vastly reduced. Let equivalence class with prefix F contains the itemsets X and Y [7]. Let t(X) to be the tidset of X while d(X) to be the diffset of X. In tidset, t(FX) and t(FY) are formed in the equivalence class and to obtain t(FXY). When we check the matching correlation of t(FX)∩t(FY) = t(FXY). Much simpler in diffset where we formulate d(FX) instead of t(FX) and d(FX) = t(F) – t(X), , the set of tids in t(F) but not in t(X). Then it results in d(FY) = t(F) – t(Y). Hence, the frequency occurrences (support) of FX does not constitute to diffset size. Refer to diffset process illustration in Fig. 2.

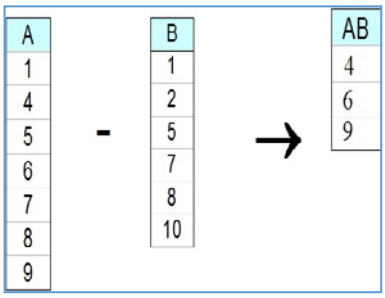

Fig. 2. Diffset between Itemset A and B.

C. Sortdiffset Algorithm

Diffset sorting [12] tries to improve dEclat [9] during intersecting process. Sorting takes place where switching is done to the itemsets. But during switching, it happens when certain tidsets are not eligible for the switching condition, thus instead of diffset, but these tidsets remains. For example, if the equivalence class with prefix E consisting of itemsets Xi, intersection of EXi with all EXj with j>i is processed to achieve a new prefix EXi class and itemsets XiXj. EXi and EXj potential to be found in either tidset format or diffset format. If EXi is in diffset format and EXj is in tidset format, the d(EXi)∩t(EXj) = d(EXjXi). Relatively for each itemset, tidset format must appear before diffset format in the order of their equivalence class according to Sortdiffset algorithm.

D. Postdiffset Algorithm

Postdiffset [22, 26-27] is proposed to answer the suggestion that is made in [12] to use tidset format in the first level of looping for sparse database and later switch to diffset format. The second level onwards of looping is done in diffset (difference intersection set) between i^{th} column and $i+1^{th}$ column before saving to database. For the first level looping, $Xi \cap Xj$ is performed while in second level looping, only candidates itemsets that differ in X_i is considered in differentiating process of $X_i - X_i$. From Fig. 3, the min_support value is given in percentage of min_support value over 100 and multiplies with total of transaction records of each dataset. If the min_supp is lower, then it is set to be rare itemsets and vice versa. Next, in the first loop, if the itemset support >= to min_support (that is set), then, tidset

takes place in first looping and follows by diffset process in the second looping onwards between i^{th} column and $i+1^{th}$ column before saving it to database.

```
Input: E((i1, t1), ... (in, tn))|P), smin

Output: F(E, smin)
Start
//get min_support
1.  min_supp=number_of_rows*percentage_min
    _support;
2.  run tidset for first loop;
3.  if(support<=min_support){
4.  add data to the next process;
5.  add data into db
6.  } end tidset
//for next loop
7.  start looping ;
8.  run diffset;
9.  if(support<=min_support){
10. add data to the next process;
11. add data into db
12. } end looping
13. end diffset;
14. flush value for current/last transaction data;
```

Fig. 3. Postdiffset Pseudocode.

Critical Relative Support (CRS) is designed by [20] is a measurement to mine the critical least association rules. The range of CRS is between 0 and 1. The value that is mostly reached 1 is considered to be the most significant and critical rule. CRS value plays around between 2 threshold value (i.e. lowest support, α and highest support, β). Detail explanation of CRS is given in Definition 13.

IV. Design of iEclat Model

A. Incremental Eclat (iEclat)

To improve performance and accuracy of itemset mining, recent researches are focus towards parallel and incremental mining approach [21-23]. Incremental mining in a dynamic database is established with regards to the itemsets or records of transaction [24-25]. Incremental in itemsets means an additional of new items being added or deleted to the existing itemsets in database whereas incremental in records of transaction means the additional transactions to the existing database transaction. The basic definitions of incremental mining concept are as follows:

Definition 1: (Incremental Database). Given a transaction itemset, T, and database, D and a sequence α. The support of D is denoted by supportD(α) is the frequency of items in D. When new data, δ is to be added to database D. Then D is said to be original database and δ is the incremental database. The updated database is denoted by (D + δ).

Definition 2: (Incremental Records and Itemsets Discovery Problem). Given an original database D and a new increment to the D which is δ, for all frequent itemsets in database (D + δ) with minimum possible recomputation and I/O overheads. The length of frequent itemsets in the updated database (D + δ) is called Incremental Records.

B. Critical Relative Support in i-Eclat

In this phase, a CRS-iEclat model is designed. First step is to design a base model in vertical approach of infrequent pattern models such as CRS in iEclat-diffset, CRS in iEclat-sortdiffset and CRS in iEclat-postdiffset. The enhancement of iEclat algorithm is required to suit for infrequent pattern mining. The completion of these steps produces an enhancement of iEclat model called as CRS-iEclat-diffset, CRS-iEclat-sortdiffset and CRS-iEclat-postdiffset format.

The outcomes are first, the embedded CRS definition in i-Eclat algorithm, second is the completion of incremental algorithm in CRS-iEclat-diffset, CRS-iEclat-sortdiffset and CRS-iEclat-postdiffset. Third, the completion of all artefact's compilation in the proposed hybrid algorithms.

Definition 3: (Least Items). An itemset X is called least item if $(a <= sup(X) <= b)$ where a and b is the lowest and highest support, respectively. The set of least item is denoted as.

$$Least\ Items = \{X \in I \mid a <= sup(X) <= b\}$$

Definition 4: (Infrequent Items). An itemset X is called infrequent item if $(sup(X) <= b)$ where b is the highest support. The set of infrequent item is denoted as.

$$Infrequent\ items = \{X \in I \mid sup\,(X) <= b\}$$

Definition 5: (Critical Relative Support). A CRS is a maximum of relative frequency among itemset and their Jaccard similarity coefficient. The value of Critical Relative Support denoted as CRS and.

$$CRS = max[(sup(A)/sup(B)), ((sup(A \longrightarrow B)/(sup(A) + sup(B) - sup(A \longrightarrow B)))]$$

The CRS value is ranging from 0 to 1, getting the results of multiplication of the highest value either antecedent support and divide by the consequence or otherwise with their Jaccard similarity coefficient. The measurement value refers to the level of CRS between combination of the both Least Items and Infrequent Items to be set as antecedent or consequence.

The architecture of CRS-iEclat is diagrammed in Fig. 4. From all infrequent items will be passed to the first pruning process, getaway G1. G1 is set with the CRS value. To set G1, total transaction records are scanned to be multiplied with the percentage of user-specified relative value of min_sup, max_sup and min_conf (minimum confidence) value. Once the value is obtained, only candidate of infrequent itemsets that passed the G1 value will be processed either through Eclat-tidset, Eclat-diffset, Eclat-sortdiffset or Eclat-postdiffset algorithms in Eclat engine. Second pruning process, getaway G2 takes place. Getaway G2 plays an important role in each itemset prior to generating frequent association rules where, filtered infrequent itemset is written to text file. Candidate itemsets are directed to hard disk storage, so that the resource of memory storage is automatically reduced to enable the processing and executing of full datasets.

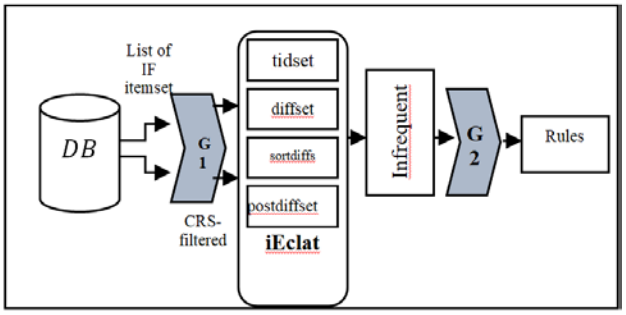

Fig. 4. CRS-iEclat Architecture.

V. Experimental Settings

A. Setup

In this phase, the proposed hybrid model will be implemented by converting all algorithms, data structures and measures into PHP-MySQL programming in a relational database management system (RDBMS) platform. The outcome is the completion of the workable prototype to mine infrequent AR.

B. Dataset

The retrieval of benchmark datasets is obtained from (Goethals, 2003) in a *.dat file format. The two (2) category of datasets are dense (i.e. a dimension with a high probability that one or more data points is occupied in every combination of dimensions) and sparse (i.e. a dimension with a low percentage of available data positions filled). The datasets descriptions are illustrated in Table I.

TABLE I. Database Source

Dataset	Description
Chess	lists the chess end game positions for king vs. King and rook
Mushroom	contains different attributes of 23 species of gilled mushrooms in the Agaricus and Lepiota family

The category of datasets is dense (i.e. a dimension with a high probability that one or more data points are occupied in every combination of dimensions). The overall characteristics of benchmark datasets is tabulated in Table II.

TABLE II. Database Characteristics

Database	#Size (KB)	#Length (attribute)	#Item	#Records (transaction)	Category
Chess	334	37	75	3196	Dense
Mushroom 557	23	119	8124	Dense	

VI. Result and Discussion

Performance of two dense datasets are measured based upon the formula in (1). The example of percentage of reduction ratio of execution time (ET) in B as compared to execution time (ET) in A is calculated based on (1) that determines the outperform percentage of B.

$$\frac{(ET\ in\ A)-(ET\ in\ B)}{2ET\ in\ A} x100 \quad (1)$$

We reveals the experimentation with only taking 50% min_supp threshold for iEclat engine whereas in CRS-iEclat, we take 30%, 40% and 50% of min_supp, min_conf and max_supp value respectively that we have tested for only 3 algorithms which are diffset, sortdiffset and postdiffset algorithms since tidset algorithms consistently to response in highest execution time both in iEclat as well as CRS-iEclat engine. Fig. 5 plots the graph of full chess dataset running in iEclat algorithm and the proposed CRS-iEclat algorithm. The CRS-iEclat outperforms iEclat engine in chess for diffset with 99% while in sortdiffset and postdiffset it shows 100% outperformance towards lesser execution time. Meanwhile, CRS-iEclat outperforms iEclat in diffset, sortdiffset and postdiffset with 54%, 66% and 79% respectively for mushroom dataset.

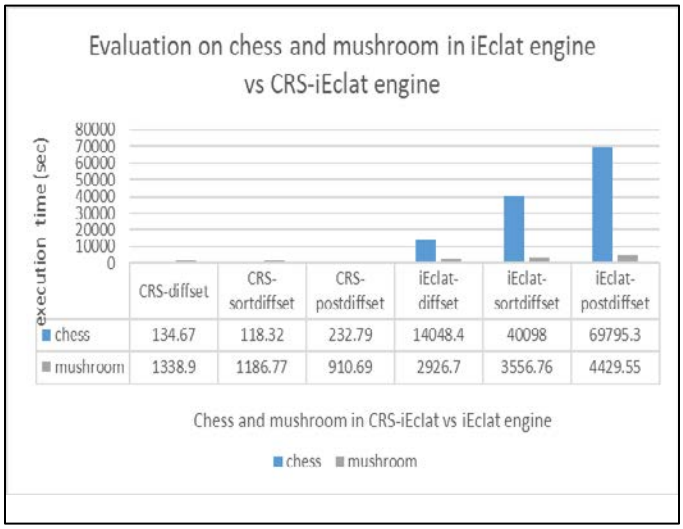

Fig. 5. Evaluation of ET between CRS-iEclat Vs iEclat Engine.

VII. Conclusion

The research proves that the more increment in itemset (column) resulting in the more usage of memory as compared to the increment of records of transaction. This is due to the increment of itemsets produces the higher cardinality of intersection between each item that needs to be conducted in vertical mining. That is why the much higher execution time can be seen in chess despite mushroom dataset. Our work also confirmed that when CRS measure is adopted in the filtering of support-confidence of our iEclat model, the execution time has drastically reduced. Either iEclat or CRS-iEclat engine, the performance of both engines is actually depending upon the nature of dataset itself when testing in diffset, sortdiffset and postdiffset algorithms. However, both engines conform that among these three algorithms, postdiffset outperforms other two algorithms by certain order of magnitude in all selected datasets. This research has proved that with CRS used as the value-added interestingness measure and filtering (pruning) in original iEclat engine, the performance is significantly improved in mining of infrequent itemsets. For our future work, the remaining test would undertake other FIMI dense datasets such as connect and pumbstar or sparse datasets such as retail or T10I4D100K to observe the performance of CRS-iEclat algorithm. The consistency of results obtained is important in determining the robustness of this model in mining process.

Acknowledgment

This project is funded by FRGS grant with reference code FRGS/1/2020/ICT06/UNISZA/03/1. A sincere gratitude goes to all faculty members of UniSZA and grant collaborators of UMT for supporting our work in reviewing for spelling errors and synchronization consistencies and also for the meaningful comments and suggestions.

References

[1] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in Proceedings of 20th International Conference on Very Large Data Bases (VLDB), 1215, pp. 487–499, 1994.

[2] S. Shrivastava and P.K. Johari, “Analysis on high utility infrequent ItemSets mining over transactional database,” InRecent Trends in

Electronics, Information & Communication Technology (RTEICT), IEEE International Conference on pp. 897-902, 2016.

[3] M.A. Thalor and S. Patil, “Incremental Learning on Non-stationary Data Stream using Ensemble Approach,” International Journal of Electrical and Computer Engineering, Aug 1;6(4):1811, 2016.

[4] G. Bathla, et al., “A Novel Approach for clustering Big Data based on MapReduce,” International Journal of Electrical and Computer Engineering (IJECE), Jun 1;8(3), 2018.

[5] M.B. Man, et al., “Mining Association Rules: A Case Study on Benchmark Dense Data,” Indonesian Journal of Electrical Engineering and Computer Science on pp. 546-553, Sep 1;3(3), 2016.

[6] R. Agrawal, et al., “Mining association rules between sets of items in large databases,” ACM SIGMOD Record, 22(2), pp. 207–216, 1993.

[7] J. Han, et al., “Mining frequent patterns without candidate generation,” ACM SIGMOD Record, 29(2), pp. 1–12, 2000.

[8] M. J. Zaki, et al., “New algorithms for fast discovery of association rules,” In Proceedings of the ACM SIGKDD international conference on Knowledge Discovery and Data Mining (KDD’97), pp. 283–286, 1997.

[9] M. J. Zaki and K. Gouda, “Fast vertical mining using diffsets,” In Proceedings of the ninth ACM SIGKDD international conference on Knowledge Discovery and Data Mining, pp. 326–335, 2003.

[10] P. Singh, S. Singh, P. K. Mishra, and R. Garg. "RDD-Eclat: Approaches to Parallelize Eclat Algorithm on Spark RDD Framework." In *International Conference on Computer Networks and Inventive Communication Technologies*, pp. 755-768. Springer, Cham, 2019.

[11] P. Shenoy, et al., “Turbo-charging vertical mining of large databases,” ACM SIGMOD Record, 29(2), pp. 22–33, 2000.

[12] T. A, Trieu and Y. Kunieda, “An improvement for declat algorithm,” In Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication (ICUIMC’12), 54, pp. 1–6, 2012.

[13] J. Hipp, et al., “Algorithms for association rule mininga general survey and comparison,” ACM SIGKDD Explorations Newsletter, 2(1), pp. 58–64, 2000.

[14] J. Han, et al., “Frequent pattern mining: current status and future directions,” Data Mining and Knowledge Discovery, 15(1), pp. 55–86, 2007.

[15] C. Borgelt, “Efficient implementations of apriori and eclat,” In Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI03), pp. 90, 2003.

[16] B. Goethals, and M. J. Zaki. "FIMI’03: Workshop on frequent itemset mining implementations." In *Third IEEE International Conference on Data Mining Workshop on Frequent Itemset Mining Implementations*, pp. 1-13. 2003.

[17] A. Savasere, et al., “An efficient algorithm for mining association rules in large databases,” In Proceeding of the 21th International Conference on Very Large Data Bases (VLDB '95), pp. 432–444, 1995.

[18] T. Slimani and A. Lazzez, “Efficient analysis of pattern and association rule mining approaches,” International Journal of Information Technology and Computer Science, 6(3), pp. 70–81, 2014.

[19] H. Toivonen, “Sampling large databases for association rules,” In Proceeding of the 22nd International Conference on Very Large Data Bases (VLDB '96), pp. 134–145, 1996.

[20] M. Man, W. A. W. A. Bakar, M. A. Jalil, & J. A. Jusoh, "Postdiffset Algorithm in Rare Pattern: An Implementation via Benchmark Case Study." International Journal of Electrical & Computer Engineering (2088-8708) 8, 2018.

[21] Z. Abdullah, T. Herawan, N. Ahmad, and M. M. Deris. "Mining significant association rules from educational data using critical relative support approach." Procedia-Social and Behavioral Sciences 28, pp. 97-101, 2011.

[22] W. A. W. A. Bakar, Z. Abdullah, M. Y. M. Saman, M. A. Jalil, M. Man, and T. Herawan. "Vertical Association Rule Mining: Case study implementation with relational DBMS." In *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*, IEEE, pp. 279-284, 2015.

[23] Q. Yong, "Integrating Frequent Itemsets Mining with Relational Database." In *2007 8th International Conference on Electronic Measurement and Instruments*, IEEE, pp. 2-543, 2007.

[24] G. Ramesh, M. William, and M. J. Zaki. "Indexing and Data Access Methods for Database Mining." In *DMKD*. 2002.

[25] J. Küng, J. Markus, and K. D. Tran, "IFIN+: a parallel incremental frequent itemsets mining in shared-memory environment." In *International Conference on Future Data and Security Engineering*, pp. 121-138, Springer, Cham, 2017.

[26] 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.

[27] 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.