# CRS-iEclat: A hybrid support measure using Critical Relative Support in iEclat model for infrequent itemset mining

**W. A. W. A. Bakar[1]*, M. Man[2]*, Z. Abdullah[3], L. C. Hwa[4], J. A. Jusoh[5]**
[1,5] Faculty od Informatics and Computing, Universiti Sultan Zainal Abidin, Besut Campus, 22200 Besut, Terengganu, Malaysia

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

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

[4] One Team Network Sdn. Bhd., 1-1B, Block 6, Jalan Pahat 15/H, Section 15, 40200 Shah Alam, Selangor, Malaysia

* Email: wanaezwani@unisza.edu.my, mustafaman@umt.edu.my

**Abstract.** Data mining and data analytic are the key components in the field of data science and applied research. Specifically, generating a rule or pattern in mining still suffer in the issue of high consumption of memory storage. Recent research is focusing on interestingness measure to define a significant threshold value that refers to minimum support or confidence level of certain items. Whereas the incremental or parallel approach to reduce memory consumption are some of the initiatives from active researchers. The research purpose is to develop a performance enhancement in Incremental Eclat (iEclat) model by embedding CRS measure 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 idea of association rule mining is to discover relationships among sets of items (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 improve of ET efficiency.

**Keywords:** Data mining, Eclat, Critical Relative Support (CRS)

## 1. Introduction

The main objectives of association rules mining (ARM) are to find the correlations, associations or casual structures among sets of items in the data repository. In other words, it allows non discovery of

implicative and interesting tendencies in databases. Frequent itemset and infrequent itemset mining are critical fields in association rule mining. The fields are widely used across a variety of domains such as market basket analysis, remedial, biology, banking or retail services [1], [2]. Frequent or infrequent itemsets may contribute to big data generation. Undoubtedly, the critical issues regarding memory space consumption and data storage capacity will significantly be affect prior to frequent or infrequent generation of itemsets [3], [4], [5]. The objective of frequent itemset is to find frequent grouping of items in database containing series of item transactions while the objective of infrequent itemset is contradict to frequent. All itemsets which has value that is greater than minimum support is called frequent itemsets. Infrequent itemset finds hidden association and correlation among rare itemsets. The rare consolidation of these itemsets may be interesting and gain more profit making. Rare cases have special concern since they represent significant difficulties for data mining algorithms. All itemsets which has the value that is lesser than minimum support is called infrequent itemsets. The idea of mining association rule originates from the analysis of market basket data [6]. Example of a simple rule is A customer who buys bread and butter will also tend to buy milk with probability s% and c%. The applicability of such rule to business problems makes the association rule to become a popular mining method.

The standard ARM searching mechanism are considered either horizontal database format [2], [7] or vertical database format. Horizontal format suffers in persistent issues of storage and memory, thus contemporary efforts to utilize on the vertical association rules mining algorithms can be seen in [8], [9], [10], [11]. The three basic models in frequent/infrequent itemset mining are Apriori [6], [11] that lies on horizontal format whereas Eclat [8] and FP-Growth [7] are based upon vertical format.

The research followers via depth first search (vertical data format) are [3], [4], [5], [6], [12], [14], [15], [16]. Among those efforts, Equivalent Class Transformation (Eclat) algorithm is known for its 'fast' intersection of its tidlist and the total tidlist respresents the support (frequency of itemset occurences) of itemset [8], [12]. The filtering (threshold) value of either minimum or maximum support (min_supp or max_supp) is depending upon user-specified. As one of association rule mining base models, Eclat algorithm has attracted few research followers such as in [9], [11], [17-18].

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 [19]. 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.

## 2. Eclat Design

### *2.1. Eclat main steps*

There are two main steps in Eclat: candidate generation and pruning. In candidate generation, each k-itemset candidate is generated from two frequent (k-1)-itemset and its support is counted, if its support is lower than the threshold, then it will be discarded, otherwise it is frequent itemsets and used to generate (k+1)-itemset. Since Eclat uses the vertical layout, counting support is trivial. Depth-first searching strategy is done where it starts with frequent items in the item base and then 2-itemset from 1-itemset, 3-itemset from 2-itemset and so on. The 4 algorithms underlying in i-Eclat model are tidset [1], diffset [9], sortdiffset [11] and postdiffset [19].

*2.2. Original Eclat (tidset)*

A k-itemset is generated by taking union of two (k-1)-itemset which have (k-2) items in common, the two (k-1)-itemsets are called parent itemsets of the k-itemset. Fox example, {}, {ab} and {ac} are parent of {abc}. To avoid generating duplicate itemsets, (k-1)-itemset are sorted in some order. To generate all possible k-itemsets from a set of (k-1)-itemset sharing (k-2)-itemset union operation is conducted of a (k-1)-itemset with the itemsets that stand behind it in the sorted order, and this process takes place for all (k-1)-itemsets except the last one. For example, from a set of {a,b,c,d,e} which share 0 item, then this could be sorted into alphabet order. To generate all 2-itemsets, the union of {a} with {b,c,d,e} will result into 2-itemsets {ab,ac,ad,ae} then for the union of {b} with {c,d,e} will result in {bc,bd,be}, similarly for {c} and {d}. Finally, all possible 2-itemsets {ab,ac,ad,ae,bc,bd,be,cd,ce,de} is generated to get all possible 3-itemsets until the rest of the number of possible itemsets.

Original Eclat (we named as tidset) starts with prefix {} and the search tree is actually the initial search tree. The prefix {a} generate the corresponding equivalence class and does frequent itemset mining in the sub tree of all itemsets containing {a}, in this sub tree it divides further into two sub trees by picking the prefix {ab}: the first sub tree consists of all itemset containing {ab}, the other consists of all itemsets containing {a} but not {b}, and this process is recursive until all itemsets in the initial search tree are visited. The search tree of an item base {a,b,c,d,e} is represented by the tree as shown in Figure 1.

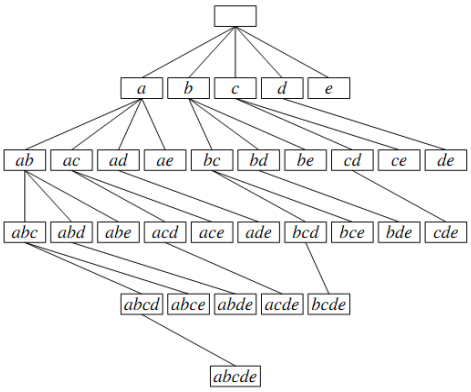


**Figure 1**. Example of candidate generation with {a,b,c,d,e}

*2.3. dEclat (diffset)*

dEclat (different set or diffset) as referred in [9] is the difference between two tidsets (tid of itemsets and its prefix). Through diffset, the cardinality of sets representing itemsets is reduced rigorously and that contributes in faster intersection and less memory usage. Consider an equivalence class with prefix P contains the itemsets X and Y [7]. Let t(X) denotes the tidset of X and d(X) denotes the diffset of X. In tidset, t(PX) and t(PY) available in the equivalence class and to obtain t(PXY) we check the cardinality of $t(PX) \cap t(PY) = t(PXY)$. In diffset format, we only have d(PX) instead of t(PX) and $d(PX) = t(P) - t(X)$, , the set of tids in t(P) but not in t(X). Similarly, we have $d(PY) = t(P) - t(Y)$. Thus, support of PX is not the size of its diffset. Refer to the illustration in Figure 2.

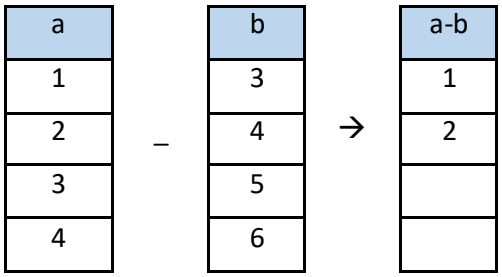


**Figure 2**. Diffset between itemset $a$ and $b$

*2.4. Sortdiffset Algorithm*

Sortdiffset [11] purposely to enhance dEclat during switching condition. When switching process takes place, there exist tidsets which do not satisfy the switching condition, thus these tidsets remain as tidsets instead of diffset format. The situation results in both tidsets and diffsets format of itemsets in particular equivalence class and the next intersection process will involve both formats. Conceptually, by given equivalence class with prefix P consisting of itemsets Xi in some order, intersection of PXi with all PXj with j>i is to be performed in order to obtain a new equivalence class with prefix PXi and frequent itemsets XiXj. PXi and PXj could be in either tidset or diffset format. If PXi is in diffset format and PXj is in tidset format, d(PXi)∩t(PXj) = d(PXjXi) which belongs to the equivalence class of prefix PXj , not PXi as expected. In other words, in order to do intersection between itemsets in diffset format and itemsets in tidset format to produce new equivalence classes properly, itemsets in tidset format must stand before itemsets in diffset format in the order of their equivalence class. That can be achieved by swapping (sorting) itemsets in diffset and tidset format, a process which has the complexity O(n) where n is the number of itemsets of the equivalence class.

*2.5. Postdiffset Algorithm*

Postdiffset [19,26] is designed prior to suggestion that is made in [11] to use tidset format at starting for sparse database and later switch to diffset format when switching condition is met. In postdiffset algorithm, the first level of looping is based on tidsets process, follows by the second level onwards of looping are getting the result of diffset (difference intersection set) between ith column and i+1th column and save to db. For the first level looping, Xi ∩ Xj is performed while in second level looping, only candidates itemsets that differ in Xi will be counted from the process of Xi – Xj. Referring to Fig. 3, the min_support threshold value is determined in terms of percentage where the user-specified min_support value will be divided by 100 and multiply with total rows (records) of each dataset. Then in each loop, starting with the first loop, if the support is greater than or equal (>=) to min_support, then, in postdiffset, the first level of looping is based on tidsets process, follows by the second level onwards of looping are getting the result of diffset (difference intersection set) between ith column and i+1th column and save to database.

From Eclat model, we have proposed the incremental approach of Eclat where we named as iEclat [20]. We embed the Critical Relative Support [21] as the filtering threshold instead of minimum support (min_supp). CRS 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 5.

## 3. iEclat Model

*3.1. Incremental Eclat (iEclat)*

To improve performance and accuracy of itemset mining, recent researches are focus towards parallel and incremental mining approach [22-25]. Incremental mining in a dynamic database is established with regards to the itemsets or records of transaction. Incremental in itemsets means an additional new item 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 sequence of transaction, T each of which is called itemset. For a database D and a sequence α, 1the support of in D is denoted by supportD(α) is the frequency of items in D. Suppose that 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, find all frequent itemsets in database (D + δ) with minimum possible recomputation and I/O overheads. For each, denotes the collection of frequent itemsets of length in the updated database (D + δ).

*3.2. Critical Relative Support in iEclat*

In this phase, an 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 model's prototype of 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 will be denoted as Least Items and

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

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 will be denoted as

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

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

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

CRS value is in a range of 0 and 1, and is determined by multiplying the highest value either supports of antecedent divide by consequence or in another way around with their Jaccard similarity coefficient. It is a measurement to show the level of CRS between combination of the both Least Items and Infrequent Items either as antecedent or consequence, respectively.

The architecture of CRS-iEclat is diagrammed in Figure 3. 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 2A or **G2A** takes place. Getaway G2A 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.

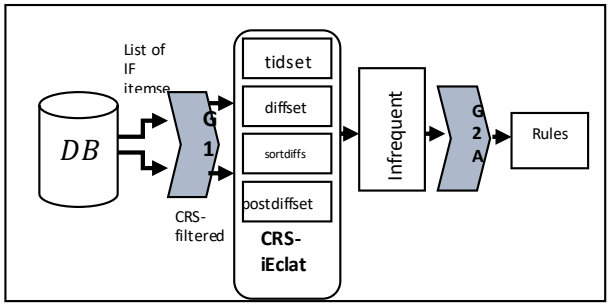


**Figure 3.** CRS-iEclat Architecture

## 4. Experimentation

*4.1. Setup*

In this phase, the proposed hybrid model is 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.

*4.2. Dataset*

The retrieval of benchmark datasets are from [15] 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 selected dense datasets are chess and mushroom. The datasets descriptions are illustrated in Table 1.

**Table 1.** 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 is occupied in every combination of dimensions). The overall characteristics of benchmark datasets is tabulated in Table 2.

**Table 2.** Database Characteristics

| Database | #Size (Kb) | #Length (Attribute) | #Item | #Records (Transaction) | #Category |
|---|---|---|---|---|---|
| Chess | 334 | 37 | 75 | 3196 | Dense |
| Mushroom | 557 | 23 | 119 | 8124 | Dense |

## 5. Result and Discussion

The snapshot screen of the login page of our CRS-iEclat tool is shown in Figure 4 whereas Figure 5 depicts the selection of datasets and algorithms in CRS-iEclat engine. The execution of the process is illustrated through command prompt window as in Figure 6.

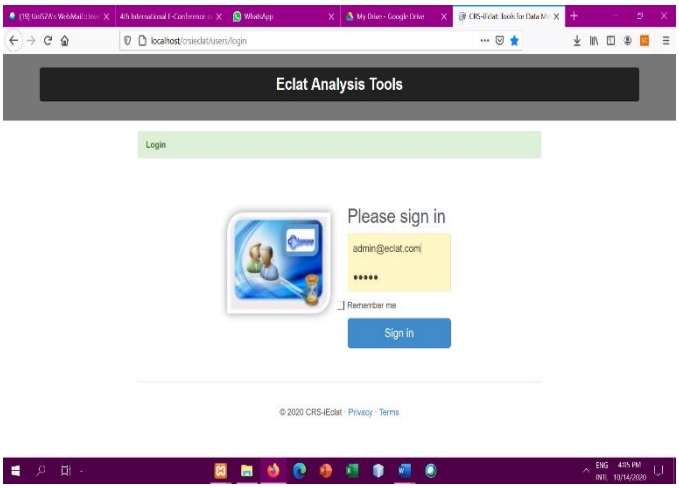


**Figure 4.** CRS-iEclat Login Page

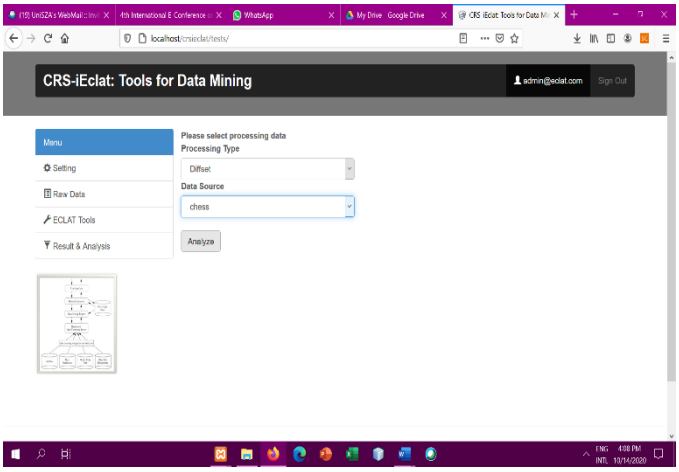


**Figure 5.** Processing and dataset selection to be analysed

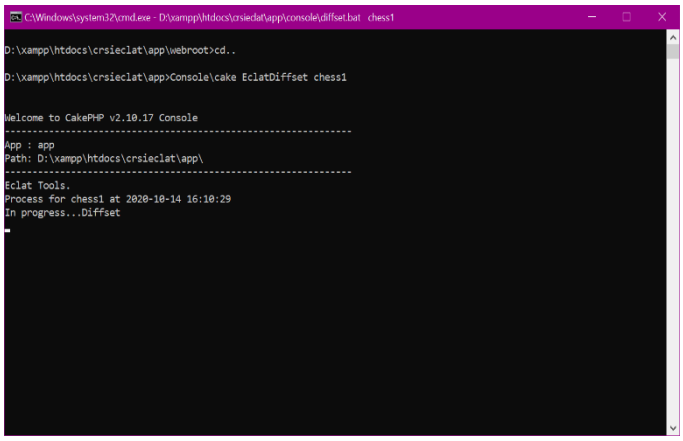


**Figure 6.** Diffset process is in progress

The performance of 2 dense datasets is 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 $B$.

$$\frac{(\text{ET } in\ A)-(\text{ET } in\ B)}{\text{ET } in\ A} \times 100\% \quad (1)$$

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. Figure 7 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.

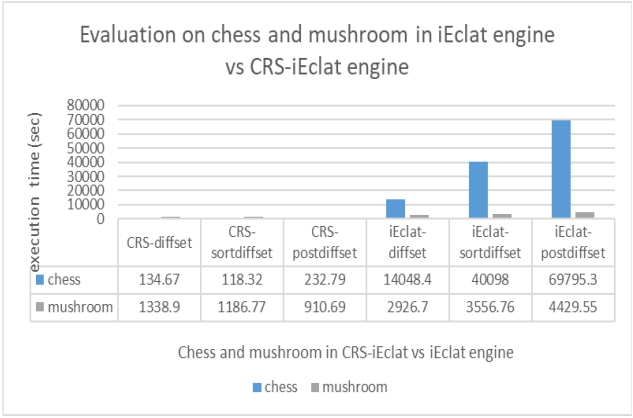


**Figure 7.** Evaluation of ET between CRS-iEclat Vs iEclat engine

## 6. 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 3 algorithms, postdiffset outperforms other 2 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, we would test this CRS-iEclat model for frequent itemset especially to cater for big data retrieval [27] and try to improve the issues of data extraction using method in [28] for the Aedes mosquito eggs, our academic-industry collaborator grant.

**Acknowledgments**
This project is funded by UniSZA with *UniSZA/2018/DPU/11* and Fundamental Research Grant Scheme of KPT with *FRGS/1/2018/ICT04/UMT/02*. A sincere gratitude goes to all faculty members, 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] Agrawal R and Srikant R 1994 Fast algorithms for mining association rules *Proceedings of 20th International Conference on Very Large Data Bases (VLDB)* 1215 pp 487–499
[2] Shrivastava S and Johari P K 2016 Analysis on high utility infrequent ItemSets mining over transactional database *Recent Trends in Electronics, Information & Communication Technology (RTEICT)* IEEE International Conference pp 897-902
[3] Thalor M A and Patil S 2016 Incremental Learning on Non-stationary Data Stream using Ensemble Approach *International Journal of Electrical and Computer Engineering* 6(4) pp 1811
[4] Bathla G et al 2018 A Novel Approach for Clustering Big Data based on MapReduce *International Journal of Electrical & Computer Engineering (2088-8708)* 8(3)
[5] Man M B et al Sep 1 2016 Mining Association Rules: A Case Study on Benchmark Dense Data *Indonesian Journal of Electrical Engineering and Computer Science* 3(3) pp 546-553
[6] Agrawal R et al 1993 Mining association rules between sets of items in large databases *ACM SIGMOD Record* 22(2) pp 207–216
[7] Han J et al 2000 Mining frequent patterns without candidate generation *ACM SIGMOD Record* 29(2) pp 1–12
[8] Zaki M J et al 1997 New algorithms for fast discovery of association rules *Proceedings of the ACM SIGKDD international conference on Knowledge Discovery and Data Mining (KDD'97)* pp 283–286
[9] Zaki M J and Gouda K 2003 Fast vertical mining using diffsets *Proceedings of the ninth ACM SIGKDD international conference on Knowledge Discovery and Data Mining* pp 326–335
[10] Shenoy P et al 2000 Turbo-charging vertical mining of large databases *ACM SIGMOD Record* 29(2) pp 22–33
[11] Trieu T A and Kunieda Y 2012 An improvement for declat algorithm *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication (ICUIMC'12)* 54 pp 1–6
[12] Hipp J et al 2000 Algorithms for association rule mininga general survey and comparison *ACM SIGKDD Explorations Newsletter* 2(1) pp 58–64
[13] Han J et al 2007 Frequent pattern mining: current status and future directions *Data Mining and Knowledge Discovery* 15(1) pp 55–86
[14] Borgelt C 2003 November Efficient implementations of apriori and eclat *FIMI'03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations* pp 90

[15] Goethals B and Zaki M J 2003 November FIMI'03: Workshop on frequent itemset mining implementations *Third IEEE International Conference on Data Mining Workshop on Frequent Itemset Mining Implementations* pp 1-13
[16] Savasere A et al 1995 An efficient algorithm for mining association rules in large databases *Proceeding of the 21th International Conference on Very Large Data Bases (VLDB '95)* pp 432–444
[17] Slimani T and Lazzez A 2014 Efficient analysis of pattern and association rule mining approaches *International Journal of Information Technology and Computer Science* 6(3) pp 70–81
[18] Toivonen H 1996 Sampling large databases for association rules *Proceeding of the 22nd International Conference on Very Large Data Bases (VLDB '96)* pp 134–145
[19] Man M et al 2018 Postdiffset Algorithm in Rare Pattern: An Implementation via Benchmark Case Study *International Journal of Electrical & Computer Engineering (2088-8708)* 8(6)
[20] Bakar W A W A et al 2020 i-Eclat: performance enhancement of Eclat via incremental approach in frequent itemset mining *Telkomnika* 18(1) pp 562-570
[21] Zailani A et al 2011 Mining significant association rules from educational data using critical relative support approach *Procedia-Social and Behavioral Sciences 28* pp 97-101
[22] Bakar W A B W A et al 2015 August Vertical Association Rule Mining: Case study implementation with relational DBMS *International Symposium on Technology Management and Emerging Technologies (ISTMET)* IEEE pp 279-284
[23] Yong Q 2007 August Integrating Frequent Itemsets Mining with Relational Database *8th International Conference on Electronic Measurement and Instruments* IEEE pp 2-543
[24] Ramesh G et al 2002 June Indexing and Data Access Methods for Database Mining *DMKD*
[25] Kung J et al 2017 November) IFIN+: a parallel incremental frequent itemsets mining in shared-memory environment *International Conference on Future Data and Security Engineering* Springer Cham pp 121-138
[26] Bakar W A W A et al 2018 Postdiffset: an Eclat- like algorithm for frequent itemset mining. *International Journal of Engineering & Technology* 7(2.28) pp 197-199
[27] Yusof M K & Man M 2017 Efficiency of JSON for data retrieval in big data *Indonesian Journal of Electrical Engineering and Computer Science* 7(1) pp 250-262
[28] Sabri I A A & Man M 2018 Improving performance of DOM in semi-structured data extraction using WEIDJ model *Indonesian Journal of Electrical Engineering and Computer Science* 9(3) pp 752-763