Journal of Theoretical and Applied Information Technology
31st December 2022. Vol.100. No 24

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

PARALLEL APPROACH IN R-DIFFSET ALGORITHM FOR INFREQUENT ITEMSET MINING

JULAILY AIDA JUSOH[1], MUSTAFA MAN[2], WAN AEZWANI ABU BAKAR[3], MOHD NORDIN ABDUL RAHMAN[4], SHARIFAH ZULAIKHA TENGKU HASSAN[5]

[1,3,4,5]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA) Besut Campus, 22200 Besut, Terengganu, Malaysia

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

E-mail: julaily@unisza.edu.my, mustafaman@umt.edu.my, wanaezwani@unisza.edu.my, mohdnabd@unisza.edu.my, sharifahzulaikha1992@gmail.com

ABSTRACT

Data Mining is an established methodology for obtaining critical information from databases utilizing the Association Rule Mining (ARM) technique. This vital information can lead to the association rule, which may reveal a good trend. Association rules' beneficial pattern is frequently stated as frequent and infrequent. To perform Itemset mining, the data formats that are necessary are horizontal format and vertical format. The vertical data format is focused on current research trends in infrequent mining techniques. An example of a vertical data mining technique for an infrequent pattern is called Rare Incremental Equivalence Class Transformation or shorten as R-Eclat. Out of the four variations of the R-Eclat approach, this study will only focus on the R-Diffset form. Prior studies have shown that the R-Diffset algorithm's data processing execution time is time-consuming. The R-Diffset technique is a solution to load imbalance problems that employ numerous nodes and cluster power to provide a novel parallel approach. In response to the positive findings of mining in terms of faster processing time and less memory consumption, R-Diffset will be supplemented with a parallel technique. Finally, a novel parallel strategy is provided to overcome the restrictions of sequential processing in terms of speed.

Keywords: *Data mining, Association rule mining (ARM), Infrequent mining, Rare Incremental Equivalence Class Transformation (R-Eclat), R-Diffset*

1. INTRODUCTION

The rise of "big data" has been introduced in new technology [34]. Big data will require more complex technology and innovative algorithms due to its scale [1, 2]. Its scale is closely related to both big data and Data Mining (DM). It typically comprises three distinct formats: structured, unstructured, and semi-structured. One of the major theoretical issues in data mining that have been dominated big data concerns for many years is how to automatically uncover valuable data so that it will become useful information [3, 4, 5]. Thus, the computer science community is responsible for discovering and solving the problems residing in big data. Task of grouping a set of objects so that those to be in the same group called cluster analysis or clustering [6]. Obtaining specific information or data is critical, especially in decision-making. In a rapidly growing area of information and communication technology, extracting useful and meaningful information is exceptionally challenging and entirely subjective. Hence, the Data Mining (DM) technique is utilized. This technique is essential for large-scale data processing and applications. The used also in various methods to extract patterns as well as to discover them from stored data [7]. The association rule analyses the data using several measurement parameters such as support and confidence in order to identify how the data is associated [8], [9]. Besides, Knowledge Discovery in Databases (KDD) is a vital part of the process of data mining. The improvement of methods and techniques to discover the patterns in KDD datasets. DM is an established methodology for obtaining critical information from databases [33] utilizing the Association Rule Mining (ARM) technique. The key challenge of association rule mining (ARM) is to discover and extract a valuable information from databases [18]. 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 time processing. A few algorithms were introduced by researchers to handle these related problems in data mining. The Apriori, FP-Growth, Eclat and R-Eclat algorithm are example of algorithms in rule mining techniques for frequent and infrequent mining.

Support indicates how frequently the pattern appears in the database, while confidence indicates the frequency of the pattern's occurrences. Two significant patterns that can be seen in ARs are Frequent Itemset Mining (FIM) and Infrequent Itemset Mining (IIM) [10] – [12]. Frequent patterns are concerned with patterns that occur frequently, whereas infrequent patterns are concerned with patterns that occur infrequently. Both frequent and infrequent patterns offer distinct data points that are critical in prediction methods. With that, prediction is the essential value in DM that leads to future work through an analysis of the existing data.

In the previous research, the R-Diffset is an extension from Diffset algorithm has been developed and executed in sequential processing for infrequent itemset mining. However, the current execution of data processing often faces the constraint of slowness, especially when dealed with a large size of dataset. A parallel approach is a new approach that has been explored and introduced in order to enhance the processing time of infrequent mining. This parallel approach will complement the R-Diffset algorithm to ensure that it can mining the infrequent itemsets faster. This new approach is introduced as PR-Diffset, where PR represents parallel-rare. As a part of the R-Diffset algorithm, this research is introducing a parallel approach to handling the slowness processing time issues. After that, the remaining components are organised as follows. Section 2 covers the fundamentals of infrequent itemset mining. The R-Eclat algorithm is presented in Section 3. The R-Diffset variation is covered in Section 4. Section 5 describes the parallel technique in R-Diffset. Section 6 describes the experimental observation. Section 7 concludes the research.

2. INFREQUENT ITEMSET MINING

The research community has adopted and widely accepted this IIM. However, to solve the IIM problem efficiently, the current method's algorithm, which has some flaws and shortcomings, must be improved [13-16]. Events that happen regularly might not be as intriguing as those that happen infrequently. IPs are unexpected or previously unknown associations, whereas FP benefits domain specialists since it reflects the known or anticipated. Even though a large fraction of IPs is uninteresting, some of them may be beneficial in the investigation, particularly those that correspond to negative data correlations. Selling smartphones and tablets at the same time is very difficult because many customers have already tended not to buy tablets, and so on. To identify competing patterns, a negatively correlated pattern is essential so that each pattern can also be substituted for each other. The objective of IIM is to discover unusual but informative relationships between entries in a dataset. This is in contrast to FIM, where the concern is in discovering the relationships that are common within a dataset. Encouraged by the vital information that may be obtained from infrequent patterns, there is a compelling need for more research in this field. Itemsets, sub-sequences, or substructures that exist in the dataset are referred to as input IPs. A pattern that is rarely occur in a transaction dataset is called an infrequent pattern.

Each algorithm is created using a unique set of techniques and data structures. The result for sets of association rules is the same, but the execution time (computational efficiency) and memory use for each approach varies. For example, the set of association rules included in T is uniquely determined in execution time and different memory usage, after which given the transaction data set T and the minimum support threshold to see the computational efficiency.

The process of creating association rules is divided into two (2) phases. The first phase, minimum support is used to locate the infrequent patterns in the database. These patterns are then used to form rules. Finding all possible combinations of patterns requires more computing work in the first phase than in the second, which is the straightforward stage. The set of all patterns may be represented as a power set of size 2^n - 1. Even if the number of n patterns in I increases exponentially, an efficient search in the downward closure property (anti-monotonic) of support is achievable. It assures that all subsets of a collection of infrequent patterns are also infrequent.

ARM, which was previously identified as FIM, is the most extensively used pattern-finding tool in data mining [17-21]. ARs are a type of association between patterns. The following is an official ARM statement of a transaction database. It is described as s $Itemsets = \{i_1, i_2, \ldots, i_n for\ |n| > 0\}$ is a collection of patterns. D is identified as a transaction database. A collection of patterns where $T \subseteq Itemsets$ is simplified as the transaction T . Transaction identifier or Tid is a one-of-a-kind identifier

associated with each transaction D. For each transaction database T, ARM takes the form of $X \subseteq Y$ where X presents parts of the rule's antecedent component while Y represents parts of the rule's consequent component, where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$. For instance, if a consumer buys a dozen diapers, he is 80% likely to also buy milk. If c% of transactions in D include both X & Y, the $X \Rightarrow Y$ rule is implemented in the transaction.

Definition 2.1 (*Support Rule*). The support of rule $X \Rightarrow Y$ is the percentage of transactions in D which

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

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

include both X and Y, where |D| is the total number of data entries.

Definition 2.2 (*Confidence Rule*). The confidence of rule $X \Rightarrow Y$ is the fraction of transactions in set D that contain both X and Y.

Using the transaction database D, a minimum support (*min_supp*) threshold, and a minimum confidence (*min_conf*) threshold, association mining seeks to produce all association rules which support below and confidence above the user-specified thresholds. This is referred to as the "confidence support framework" [22]. However, this study only looks at one essential parameter: support. It is because identifying all patterns with support less than the minimum support value is the most intriguing thing in IIM. The next session will review on rare itemset mining using R-Eclat algorithm.

3. R-ECLAT ALGORITHM

R-Eclat (Rare Incremental Equivalence Class Transformation) [1, 35] is introduced to process the infrequent itemsets mining in large database [36]. It is 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. To complete phase 1, the first step that must be reviewed and considered is the basic concept of infrequent patterns. This basic concept will then be mapped into the Eclat algorithm ahead of time, including some modified components to make it compatible with infrequent itemset mining. The completion of a modified algorithm is named the R-Eclat algorithm, where R represents rare. Figure 1 illustrates a conceptual model of R-Eclat.

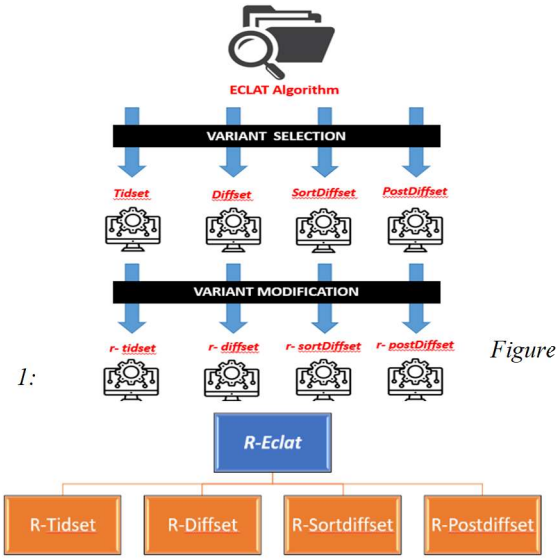

Figure 1: Conceptual Model of R-Eclat

Figure 2: R-Eclat Algorithm and its Variant

Previously, Eclat had four variants, namely Tidset, Diffset, Sortdiffset, and Postdiffset. Then, R-Eclat improved its algorithm to accommodate mining infrequent patterns, and it generated newly variations of its algorithm, including R-Tidset, R-Diffset, R-Sortdiffset, and R-Postdiffset. The R in each of the variant modification stands for rare or infrequent. This diagram is shown in Figure 2.

The basic idea behind this approach is as follows: consider B to be the universe of patterns, where B = $\{i_1, i_2, ..., i_m\}$, for $m > 0$ refers to the set of literals known as a set of m patterns. If a set $X = \{i_1, ..., i_k\} \subseteq$ B contains k-patterns, it is referred as a pattern or a k-Itemset . A transaction over B is made up of a pair of $T_i = (tid, I)$ where tid is a transaction "identifier" and I is a pattern. a transaction $T_i = (tid, I)$ is said to support a pattern, $X \subseteq$ B if $X \subseteq I$. A transaction database, T, is a collection of transaction over B. A tidset of a pattern X in T is a collection of transaction identifiers (tids) in T which support X. $(support, X) = \{tid \mid (tid, I) \in T, X \subseteq I\}$ backs up this statement. The cardinality of the transactions that comprise a pattern X tidset determines its support in a transaction (T), where ($support$, X) =

Journal of Theoretical and Applied Information Technology
31st December 2022. Vol.100. No 24

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

|(X)|. The illustration of the B transaction is given in Figure 3. Based on this figure, the following definitions are explained with its given example.

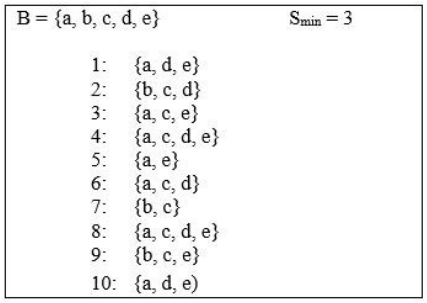

Figure 3: Itemsets in Dataset B

Definition 1: By providing a transaction database T over a pattern base B and a minimal support threshold, s_{min}, the set of all infrequent patterns is denoted.

$$IF\,(T, s_{min}) = \{X \subseteq B |\, (X) \leq s_{min}\} \quad (3)$$

Definition 2 (Infrequent Itemset Mining): The infrequent itemset mining problem is expressed as IF (T, s_{min}), with a transaction database T and a minimal support threshold, s_{min}.

Definition 3 (Search Tree): P(B) signifies all potential patterns over B, and the infrequent patterns mining problem's search tree includes precisely $2^{|B|}$ distinct pattern. Figure 4 depicts one example.

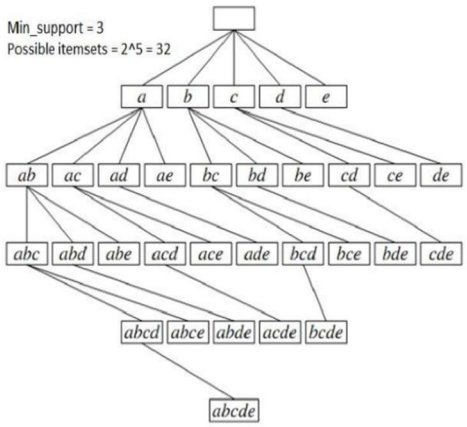

Figure 4: Search Tree of Dataset B

Definition 4 (Candidate Itemset): Given a transaction dataset T with a minimal support threshold, s_{min}, and an IIM of *IF* method (T, s_{min}). A candidate pattern is one in which the algorithm determines if a pattern set X is infrequent or not. Assuming that the search tree has 2^5 Itemsets and the patterns in dataset B are {a, b, c, d, e}, if s_{min} is set to 3, any pattern of at least three transactions or less is considered infrequent.

Definition 5 (Downward Closure Property/Support Monotonicity): Assuming X, $Y \subseteq$ A are the two patterns in a transaction database T over a pattern base A, if a pattern is infrequent, all of its supersets are infrequent as well. Furthermore, in algorithms that employ this attribute, only candidates from infrequent subsets are generated and tallied for this support. Itemsets, on the other hand, are frequently

$$X \subseteq Y \Rightarrow sup(Y) \leq sup(X) \quad (4)$$

closed upwards.

Definition 6 (Intersection): Assume that A and B are one set. The intersection of sets A and B is denoted as $A \cap B$. It is the set in which same patterns presents in both A and B such that $A \cap B = \{x \mid x \in A \wedge x \in B\}$

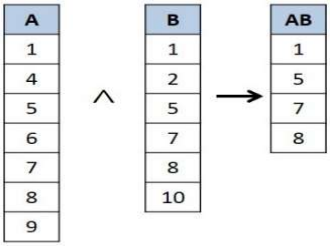

Figure 5: Intersection of Itemset A and B

Definition 7 (Difference Set): Assume A and B are two distinct sets. A-B is used to expressed the difference between A and B. It is the set in which the patterns are present only in A. The difference between A and B is also referred to as B's complement to A, thus $A-B = \{x \mid x \in A \wedge x \notin B\}$

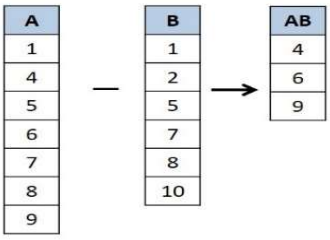

Figure 6: Difference of Itemset A and B

This section has discussed the R-Eclat's theoretical background in detail. The next section will focus on one of the available variants in R-Eclat which known as R-Diffset. The advantage of R-Diffset is it only keeps track of differences in tidsets, thus make the intersection faster and less memory usage.

Journal of Theoretical and Applied Information Technology
31st December 2022. Vol.100. No 24

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

4. R – DIFFSET VARIANT

In the R-Diffset variant, the changes in tidsets will be analyzed. This situation makes the mining process becomes more effective (faster intersection process and lesser memory usage). The R-Diffset is a variant of R-Eclat algorithm that uses the "diffset" structure rather than the "tidset" structure to implement it. The authors of [23] developed the R-Diffset (different set or diffset), which represented a pattern by Tids by utilizing Tids that occurred in the tidset of its prefix but did not appear in their own tidsets. To put it another way, diffset only considers changes between two (2) tidsets, such as a class member's tidset and the prefix tidset. Starting at the root, these distinctions are passed down to a node's offspring. Using diffset reduces the cardinality of sets that represent the presents significantly, allowing for quicker intersection and reduced memory consumption. The patterns X and Y are deemed to be included in equivalence class with the prefix P. Consider (X) to be the tidset of Y and d(X) to be the diffset of X, as described in [24]. When we use the tidset option, we will have $t(PX)$ and $t(PY)$ in the equivalence class, and to get $t(PXY)$, we verify the cardinality of $t(PX) \cap t(PY) = t(PXY)$.

When we use the diffset version, we get (PX) which is $(PX) = (P) - t(X)$, the set of tids in $t(P)$ but not in $t(X)$. Likewise, we have $(PY) = (P) - (Y)$. 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 in another way, sup$(PX) = sup(P) - |d(PX)|$. Figure 7 depicts the Diffsets' formula developed by Trieu et al. [24] and [12]. As a result, the frequency of occurrences (support) of PX does not equal the diffset size. Refer to Figure 8 for an illustration of the diffset method.

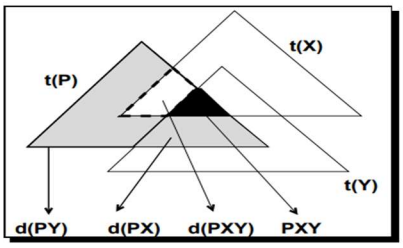

Figure 7: Diffsets Illustration

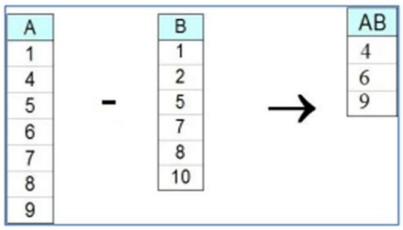

Figure 8: Diffset between Itemset A and B

The basic transaction database in the vertical layout must first be modified to the diffset variation, where the diffset of patterns is a set of tids that do not include patterns. In the vertical layout, the initial transaction database is comparable to the prefix $P = \{\}$, therefore the tidset of P includes all tidsets, and all transactions contain P.

The diffset of a pattern i is $d(i) = \{t(P) - t(i)\}$, this is a set of tids which transactions do not contain i. This can be inferred from the definition of diffset. Using the basic equivalence class, we were able to create all patterns along with the necessary diffsets and supports. Step 7 is where the R-Diffset is different from Eclat, whereby instead of generating an intersection data, a new diffset data will generate.

Pseudocode R-Diffset

Input: $E((i_1, t_1),\ldots(i_1, t_1))|\ P),\ s_{min}$

Output: $F\ (E,\ s_{min})$

```
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 9: Pseudocode for R-Diffset [14].

In the dense database, R-Diffset has shown a process to achieve significant improvements in execution time over the tidset variant. It will be losing its advantages over tidsets if the databases are sparse. In 2003, Zaki et al. [14] recommended that the tidset variant be used first and a transition to the diffset variant be made once the switching condition has been satisfied. However, initializing the diffset variant first is better when dealing with compact datasets. In contrast, because a diffset is often an order of magnitude smaller than a tidset, it is recommended to begin with the tidset variation when working with sparse datasets and move to the diffset variant later on.

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

Considering the Itemset PXY in a new class, PX can either be stored in tidset (PXY) or as diffset (PXY). The reduction ratio, $r = (PXY) / (PXY)$. Diffsets must have a reduction ratio of at least 1 in order to be beneficial. That is $r \geq 1$ or $(PXY) / (PXY) \geq 1$. Since $(PX) - (PY) = (PX) - (PXY)$, so we have $t(PXY) / (t(PX) - t(PXY)) \geq 1$. If we divide by (PXY), so $1 / ((PX) / (PXY) - 1) \geq 1$. After simplification, the results will be $(PX) / (PXY) \leq 2$. In other words, the authors conclude that switching to the diffset option is preferable if the support of PXY is at least half that of PX. From length 2-Itemsets on, it is advisable to utilise diffset [14], [24]. If the reduction ratio is less than one, the diffset should be used, beginning with three patterns.

Diffset data structure exponentially compresses the database when longer patterns are found. Because of this, the diffsets method is far more scalable than previous methods. Figure 10 shows that the tidset database, which has to store 23 Tids, uses more memory resources than the diffset database, which needs to store only 7 tids.

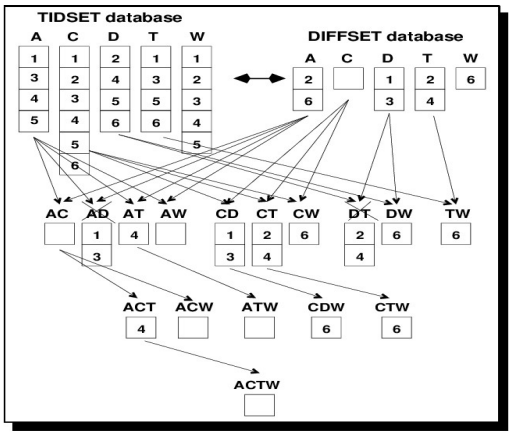

Figure 10: Diffsets for pattern Counting

As already known, R-Diffset executes the infrequent itemsets mining process in sequential form. However, in order to deal with the issue of large datasets and long processing times, the sequential processing method is seen as no longer appropriate. Therefore, the parallel processing approach is one of the alternatives that can be explored to ensure either it is suitable to be embedded in the R-Diffset variant. The next session will discuss in detail regarding parallel implementation in R-Diffset variant.

5. PARALLEL APPROACH IN R-DIFFSET

Nowadays, with a large number of records and attributes, itemset mining has become computationally intensive. This section discusses in detail our innovative parallel method for mining infrequent patterns based on the MapReduce (MR) technique. The MapReduce technique is used to execute the R-Diffset algorithm in parallel. This section highlighted a design that enables the proposed R-Diffset method to be executed in parallel. It also includes the goals, architecture, and algorithm specifications.

A parallel approach is essential to parallelize R-Diffset algorithms in order to gain better performance and scalability across massive datasets. First and foremost, it is vital to design an effective dataset organising and decomposition method. As a result, the effort may be divided into minimum data reliance. Second, reducing synchronization and communication overhead is essential. It is to make sure that the parallel algorithm will smoothly operate as the number of processes increases. In this research, the parallel approach will address the issue of execution time during processing mining.

MR is a technique for efficiently processing and analyzing large amounts of data [26] – [29]. It is simple and abstracts the details of running a distributed program, such as parallelization, fault tolerance, data distribution, and load balancing. In this technique, the original data is split and mapped into several subsets called "mapper" and "input," which are represented as <key, value> pairs. The reduced task is then in charge of merging the partial output generated by each mapper.

The conventional technique employed in the distributed processing situation is called MR execution. The method is split into two primary parts using Map and Reduce. While Reduce function will gather and aggregate the results, the Maping function is dedicated to dividing the data for processing. In general, the MR approach addresses the <key, value> pair as a basic data structure. The final findings, intermediate results, and processed data all work in terms of <key, value> pairs. Figure 11 depicts a typical map reduction technique, including the map and reduction stages, to summary its process. The following is a description of the MR technique:

- The mapping function first reads the data and then turns it into <key, value> pairs. This stage of the transformation can utilize any order of operations on each record prior to transmitting the tuples over the network.

Journal of Theoretical and Applied Information Technology
31st December 2022. Vol.100. No 24

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

- Next, the output keys are combined and sorted by <key, value>, such that the coincident keys are arranged by the relevant value list. The keys are then split and transmitted to the Reducer in accordance with the previously determined key-based scheme.
- Finally, the Reducer combines lists of several types to generate a single value for each pair. Reducer is also used to merge the map output and for extra optimization. By merging each word generated from the Map phase into a pair, this enhancement reduces the total amount of data sent across the network.

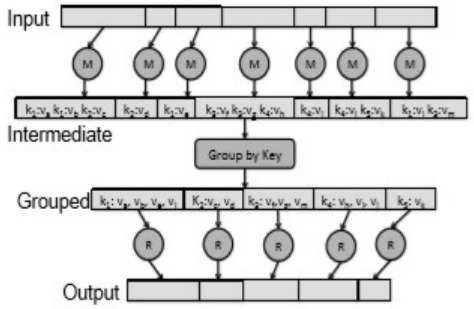

Figure 11: The MapReduce Technique's Layout

Figure 12 depicts the parallel approach pseudocode. Assume *D* be a transaction database with *T* transaction. *T* is mapped into parallel processing in line with *D* divisions. Each partition is assigned to a mapper, which is a sub-database. Then there are *M* mappers, which automatically split *D* into $M = \{m_1, m_2, m_3 \ldots\ldots m_n\}$. $I = \{i_1, i_2, ..., i_m\}$ is the set of all patterns, and each $X \in P(I)$ of I is termed a pattern, as discussed in Chapter 4. Assume that *X* is an Itemset in *M* and σ(pattern), or σ(*X*) is X support count. For a given minimum support threshold s_{min}, X is infrequent if:

$$\sigma(pattern) \leq s_{min}$$

The essential task in distributing *D* to *M* offering a better performance to accelerate the process of mining infrequent itemsets in a shorter time.

Pseudocode PR – Diffset

Input: $E((i_1, t_1),\ldots(i_1, t_1))|\ P),\ s_{min}$
Output: $F\ (E,\ s_{min})$
Begin
Map (key = item, value = list-of-tids)
1. **foreach** itemset X ∈ M **do**
2. M. σ(item) = count (X, M.tids)
3. output ⟨item, tids⟩;
4. **end foreach**

Reduce (key = item, value = list-of-tids), *where key is an element of infrequent itemsets and value is its occurrence in each mappers*
1. **foreach** item I **do**
2. **foreach** value t **do**
3. σ(item) + = t; // *initially σ(item)=0 in each mapper*
4. **if** σ(item) ≤ s_{min} **then**
5. get all infrequent itemsets in each mapper
6. output ⟨item,t.tids⟩;
7. **end if**
8. **end foreach**
9. Merge into a single file;
10. **if** ⟨item,t.tids⟩ ≤ s_{min} **then**
11. write the value to text file;
End;

Figure 12: Pseudocode of Parallel Approach in R-Diffset [14].

The implementation of parallel approach in R-Diffset algorithm invokes MR function to find all infrequent patterns and results in a list of infrequent patterns. The input is the original transaction database *D*, which has been partitioned into *m* partitions. Following the representation of input partitioning in each *m*, the content of each *m* reflects all partitioned patterns designated as:

$$\{im_1, im_2, im_3,\ldots,im_n\} \in m$$

"*im*" represented the patterns in *m*. Each m will be processed by any chosen R-Diffset algorithm, carrying out the map duties. A set of patterns is mined in each m throughout this phase. The output of the m consists of a list of <key, value> pairs, where the key is a pattern element and the value is the support count of each Itemset. The map function is invoked as many times as the number of transactions in the map phase. Every time the map function is used, it reads a single transaction and returns a <pattern, tid> pair for each pattern in the transaction. The start of the reduction task occurs after all map tasks have been completed. The reduction function is used as many times throughout reduce phase as there are distinct patterns in each m. The reduction function is given a unique

pattern each time, as well as the transaction identifiers (Tids) of the transactions containing the patterns. If an itemset's support count is $\sigma(pattern) \leq s_{min}$, it will output the infrequent itemsets represented by $m_{res}=\{im_1, im_2, im_3,\ldots\ldots im_n\}$, where "$m_{res}$" represents the list of infrequent patterns in each mapper. The final stage of parallel R-Diffset (PR-Diffset) technique is the outcome of output integration and filtering. To obtain the final result of infrequent Itemset mining, all results from each mapper, <pattern, t.tids> are grouped or merged into a single file represented as $\{m_{res1}, m_{res2}, m_{res3},\ldots\ldots m_{resn}\} \in I_{res,}$ where I_{res} indicates the integrated result. The filtering step filters <pattern, t.tids> to ensure that the patterns supported are less than or equal to S_{min}.

As a complementary for R-Diffset algorithm, a new algorithm is produced based on the parallel MapReduce technique. This algorithm efficiently mines the infrequent itemsets in an order of magnitude less execution time as compared to R-Diffset algorithm.

6. EXPERIMENTAL OBSERVATION

This experiment also highlights the benefit of parallel approaches, since patterns are mined in large quantities and processed quicker through sequential processing. It proves that, rather than using hardware parallelization to solve the problem, the parallel approach in software provides an efficient speed-up to address the running time issue. In a parallel approach, faster processing effectively decreases the percentage of total running time that is spent on sequential processing. This approach is not limited to R-Diffset only; it could be used to solve running time problems in other variants of R-Eclat.

The implementation parallel approach in R-Diffset is experimentally tested to assure its parallel performance. In summary, Figure 13 depicts the attainment of 72% quicker execution time in parallel processing than in sequential processing. The achievement of execution time demonstrates that using the parallel technique as a supplement and improvement to R-Diffset was the correct option.

Figure 13: The general comparison analysis of execution time in R-Diffset between sequential and parallel processing [30].

7. THE CONCLUSION

In this research, we studied a parallel MapReduce (MR) technique applied to a vertical data format that depends on an R-Diffset. It effectively reduces the mining time for infrequent itemsets in sequential processing. The implementation of the MapReduce (MR) technique in the proposed parallel approach efficiently "mines" infrequent patterns in order of magnitude less execution time in comparison to the R-Diffset algorithm in sequential processing. The result of the total execution time for PR-Diffset is less compared to the total execution time for R-Diffset. After that, the performance of both algorithms depends on the nature of the datasets when testing (i.e., mushroom, pumsb_star, chess, retails, and T10I4D100k). However, both algorithms confirm that among these five (5) datasets, chess outperforms between other datasets in execution time. Looking at the effectiveness of this result, it can be applied to the real dataset from various fields such as fisheries. It might can assist in identifying the extinction of fish species and recommendation for the reproduction of the species.

ACKNOWLEDGEMENT

We would like to express our gratitude to all UniSZA and UMT colleagues for networking and technical assistance in proofreading and synchronization issues, as well as for their constructive remarks and recommendations.

REFERENCES

[1] M. Man, et. al, "Analysis study on R-Eclat algorithm in infrequent Itemsets mining," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 6, pp. 5446–5453, Dec. 2019, doi:

10.11591/IJECE.V9I6.PP5446-5453.

[2] I. Yaqoob *et al.*, "Big data: From beginning to future," *Int. J. Inf. Manage.*, vol. 36, pp. 1231–1247, 2016, doi: 10.1016/j.ijinfomgt.2016.07.009.

[3] S. Kok and P. Domingos, "Statistical Predicate Invention."

[4] G. Piatetsky-Shapiro, "Data mining and knowledge discovery 1996 to 2005: overcoming the hype and moving from 'university' to 'business' and 'analytics,'" *Data Min. Knowl. Discov. 2007 151*, vol. 15, no. 1, pp. 99–105, Jan. 2007, doi: 10.1007/S10618-006-0058-2.

[5] Y. Djenouri, D. Djenouri, Z. Habbas, and A. Belhadi, "How to exploit high performance computing in population-based metaheuristics for solving association rule mining problem," *Distrib. Parallel Databases*, vol. 36, no. 2, pp. 369–397, Jun. 2018, doi: 10.1007/S10619-018-7218-4.

[6] S. A. S. S. M. A. B. Safei and K. Yusof, "Targeted Ranking-Based Clustering Using AHP K-Means," *Int. J. Advance Soft Compu. Appl*, 2015. https://www.academia.edu/70818561/Targeted_Ranking_Based_Clustering_Using_AHP_K_Means.

[7] A. A. Aziz, N. H. Ismail, and F. Ahmad, "Mining students' academic performance," 2013. https://www.researchgate.net/publication/258124336_Mining_students'_academic_performance.

[8] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," 1995.

[9] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1057 LNCS, pp. 3–17, 1996, doi: 10.1007/BFB0014140/COVER.

[10] R. Lakshmi, C. S. Hemalatha, and V. Vaidehi, "Mining infrequent patterns in data stream," *2014 Int. Conf. Recent Trends Inf. Technol. ICRTIT 2014*, Dec. 2014, doi: 10.1109/ICRTIT.2014.6996199.

[11] C. Sweetlin Hemalatha, V. Vaidehi, and R. Lakshmi, "Minimal infrequent pattern based approach for mining outliers in data streams," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1998–2012, Mar. 2015, doi: 10.1016/J.ESWA.2014.09.053.

[12] P. Fournier-Viger, J. C. W. Lin, B. Vo, T. T. Chi, J. Zhang, and H. B. Le, "A survey of Itemset mining," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 7, no. 4, p. e1207, Jul. 2017, doi: 10.1002/WIDM.1207.

[13] Seema Vaidya* and Deshmukh PK, "Predicting Rare Disease of Patient by Using Infrequent Weighted Itemset," *J. Comput. Sci. Syst. Biol.*, vol. 8, no. 4, pp. 233–238, 2015, Accessed: Oct. 12, 2022. [Online]. Available: https://www.hilarispublisher.com/open-access/predicting-rare-disease-of-patient-by-using-infrequent-weighted-Itemset-jcsb-1000194.pdf.

[14] M. J. Zaki and K. Gouda, "Fast vertical mining using diffsets," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 326–335, 2003, doi: 10.1145/956750.956788.

[15] J. Pillai and O. P. Vyas, "High Utility Rare Itemset Mining (HURI): An approach for extracting high-utility rare Itemsets," *i-manager's J. Futur. Eng. Technol.*, vol. 7, no. 1, pp. 25–33, 2011, Accessed: Oct. 12, 2022. [Online]. Available: https://www.academia.edu/63891579/High_Utility_Rare_Itemset_Mining_Huri_An_Approach_for_Extracting_High_Utility_Rare_pattern_Sets.

[16] Y. Ji, H. Ying, J. Tran, P. Dews, A. Mansour, and R. Michael Massanari, "A method for mining infrequent causal associations and its application in finding adverse drug reaction signal pairs," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 721–733, 2013, doi: 10.1109/TKDE.2012.28.

[17] Y. Lu, F. Richter, and T. Seidl, "Efficient infrequent Itemset mining using depth-first and top-down lattice traversal," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10827 LNCS, pp. 908–915, 2018, doi: 10.1007/978-3-319-91452-7_58/COVER.

[18] 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.

[19] B. Bakariya and G. S. Thakur, "Mining Rare Itemsets from Weblog Data," *Natl. Acad. Sci. Lett.*, vol. 39, no. 5, pp. 359–363, Oct. 2016, doi: 10.1007/S40009-016-0465-X.

[20] M. Manikrao Ghonge, N. Pradipkumar Rane, and A. D. Potgantwar, "A Review on Infrequent (Rare) pattern patterns Analysis in Data Mining," *2018 Int. Conf. Adv. Commun. Comput. Technol. ICACCT 2018*, pp. 388–391, Nov. 2018, doi: 10.1109/ICACCT.2018.8529629.

[21] A. Rahman, Y. Xu, K. Radke, and E. Foo, "Finding anomalies in SCADA logs using rare sequential pattern mining," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9955 LNCS, pp. 499–506, 2016, doi: 10.1007/978-3-319-46298-1_32.

[22] L. Xiang, "Interesting Association Rules Mining Based on Improved Rarity Algorithm," 2018, Accessed: Oct. 12, 2022. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-35320.

[23] M. Ghonge and M. N. Rane, "Mining Rare patterns by Using Automated Threshold Support," *Int. J. Eng. Technol.*, vol. 7, no. 3.8, pp. 77–81, Jul. 2018, doi: 10.14419/ijet.v7i3.8.15225.

[24] T. A. Trieu and Y. Kunieda, "An improvement for dEclat algorithm," *Proc. 6th Int. Conf. Ubiquitous Inf. Manag. Commun. ICUIMC'12*, 2012, doi: 10.1145/2184751.2184818.

[25] J. A. Jusoh, J. A. Jusoh, M. Man, and W. A. W. A. Bakar, "Performance of IF-Postdiffset and R-Eclat Variants in Large Dataset," *Int. J. Eng. Technol.*, vol. 7, no. 4.1, pp. 134–137, Sep. 2018, doi: 10.14419/ijet.v7i4.1.28241.

[26] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters."

[27] Z. Farzanyar and N. Cercone, "Efficient mining of frequent Itemsets in social network data based on MapReduce framework," *Proc. 2013 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2013*, pp. 1183–1188, 2013, doi: 10.1145/2492517.2500301.

[28] J. Liu, Y. Wu, Q. Zhou, B. C. M. Fung, F. Chen, and B. Yu, "Parallel eclat for opportunistic mining of frequent Itemsets," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9261, pp. 401–415, 2015, doi: 10.1007/978-3-319-22849-5_27/COVER.

[29] S. Maitrey and C. K. Jha, "MapReduce: Simplified Data Analysis of Big Data," *Procedia Comput. Sci.*, vol. 57, pp. 563–571, Jan. 2015, doi: 10.1016/J.PROCS.2015.07.392.

[30] 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.

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

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

[33] R. Rosly, M. Makhtar, M. K. Awang, M. I. Awang, M. N. A. Rahman, and H. Mahdin, "Comprehensive study on ensemble classification for medical applications," *Int. J. Eng. Technol.*, vol. 7, no. 2.14, pp. 186–190, Apr. 2018, doi: 10.14419/ijet.v7i2.14.12822.

[34] 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. 250–262, Jul. 2017, doi: 10.11591/IJEECS.V7.I1.PP250-262.

[35] 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.

[36] Jusoh, J. A. (2019). *A New R-Eclat Algorithm for Infrequent Itemset Mining* (Doctoral dissertation, PhD Thesis. Malaysia: Universiti Malaysia Terengganu).