International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Significance of Data Structures and Data Retrieval Techniques on Sequence Rule Mining Efficacy

[1]Nayanjyoti Mazumdar, [2]Pankaj Kumar Deva Sarma
[1&2] Department of Computer Science
[1&2] Assam University, Silchar
[1&2] Cachar, Assam, India-788011
[1]nayan.mazumdar@gmail.com , [2]pankajgr@rediffmail.com

Abstract— Sequence mining intends to discover rules from diverse datasets by implementing Rule Mining Algorithms with efficient data structures and data retrieval techniques. Traditional algorithms struggle in handling variable support measures which may involve repeated reconstruction of the underlying data structures with changing thresholds. To address these issues the premiere Sequence Mining Algorithm, AprioriAll is implemented against an Educational and a Financial Dataset, using the HASH and the TRIE data structures with scan reduction techniques. Primary idea is to study the impact of data structures and retrieval techniques on the rule mining process in handling diverse datasets. Performance Evaluation Matrices- Support, Confidence and Lifts are considered for testing the efficacies of the algorithm in terms of memory requirements and execution time complexities. Results unveil the excellence of Hashing in tree construction time and memory overhead for fixed sets of pre-defined support thresholds. Whereas, TRIE may avoid reconstruction and is capable of handling dynamic support thresholds, leading to shorter rule discovery time but higher memory consumption. This study highlights the effectiveness of Hash and TRIE data structures considering the dataset characteristics during rule mining. It underscores the importance of appropriate data structures based on dataset features, scanning techniques, and user-defined parameters.

Keywords- Sequence Mining, AprioriAll, Data Structure, Scan Reduction, Educational Mining, Financial Prediction.

I. INTRODUCTION

In recent times, the impact of sequential pattern mining has grown significantly across various domains, leading to the introduction of numerous sequence mining frameworks and algorithms which can be seen in articles [1] and [2]. These algorithms cater to diverse user needs and are designed to run on different platforms. Some algorithms prioritize the discovery of sequential rules [3] from the datasets, even at the cost of increased memory requirements and processing overheads. Some other algorithms are focused on exploring sequential rules with optimum memory usage with shorter processing time. Due to the availability of limited primary memory, efficient data layout is required for manipulating datasets in memory. However, same data structure used with an algorithm may not be suitable for mining all datasets. They might present inefficiencies in discovering mining rules based on user requirements and varying dataset characteristics. Similarly, different structures can be applicable to the same algorithm for mining different dataset scenarios. Each data structure may exhibit particular strengths when handling a specific dataset. Therefore, selection of the most appropriate data structure for a mining paradigm for manipulating datasets in main memory is a very challenging task.

One of the crucial and unavoidable factors in generating sequential rules is to meet the users' needs by considering the user-defined constraints. Algorithms often necessitate frequent restructuring of the underlying data structures when these support constraints change. Accommodating dynamism in data structures that align with algorithmic requirements can be a highly expensive endeavor. To address these tradeoffs, efficient data structures are essential for algorithms to manipulate data in main memory during mining. These data structures not only alleviate the burden of extensive memory requirements but also enhance the processing capabilities. Another crucial observation from previous works is the vital role played by the database scanning techniques during runtime in mining bigger datasets. Minimizing database scans of secondary memory can significantly improve the efficiency and effectiveness of the rule mining process.

This study implements the AprioriAll algorithm [4] using two datasets: an "*Educational Dataset*" of student grades in their undergraduate courses and a "*NIFTY 50 Index Dataset*" covering the daily price movement data spanning last 25 years from 1997 to 2022. The reason behind considering these datasets is that, substantial research have been delivered towards these domains which can be seen in [5 - 9]. HASH tree and TRIE structures are used as the underlying data structures for the algorithms to investigate the memory consumption, processing efficiencies, database scanning techniques, data-restructuring policies, and user-defined support threshold provisioning etc. This research presents a comprehensive in-depth analysis of algorithmic performances. It highlights the importance of selecting appropriate data structures based on dataset characteristics and desired outcomes, providing insights into the efficiency and effectiveness of the algorithms in sequence mining.

This paper is organized as follows: In Section - II, a brief survey of the related works is presented. In Section - III, a roadmap for the process flows followed for the implementation works and analysis is depicted. In Section - IV, a detailed process for the entire experimental works along with the results obtained from the investigations is illustrated. In Section - V, a detailed discussion on the results obtained from the experimental works is provided. This paper is concluded in Section - VI, providing

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

implementation remarks, followed by a glimpse of the future works.

II. Related Works

Since the inception of the term Sequential Pattern Mining (SPM) by Agrawal et al. [3], it has got immense applications in diversified domains. Market-Basket Analysis, Decision Support Systems, Financial Predictions, Medical Data Analysis, Survival predictions, Product Quality Testing, DNA Sequencing, Drug-Discovery, Recommender Systems, Sentiment Analysis, IoT based Applications are some of the premier application areas of sequence mining. Considerable research is delivered in the fields of Educational Mining [10] and Bioinformatics [11] in recent years. To fulfill the demands of various domain specific needs and user requirements, a number of sequence mining algorithms [12] have evolved in last three decades. Majority of these algorithms are developed on the basis of structural design of the data for efficiently manipulating them into the main memory. Some algorithms are based on the formatting of the data for minimizing the data size while conversion from one format to another e.g. bitmap formats and regular expressions. Sequential algorithms are developed for mining bigger datasets by employing the concepts of data fragmentation [13], projections [14], incremental [15] & pattern growth techniques [16], task & data parallelization [17], distributed processing [18] etc. Algorithms are also designed to handle temporal, weighted as well as the user defined constraints [19]. Based on the way how the database to be scanned from the secondary memory during mining is taken care off by many algorithms strengthening the sequence mining process.

The AprioriAll is a sequence mining algorithm [4] which have been thoroughly and effectively implemented to different applications over the years. Due to its structural flexibility and capable of being executed in normal computing environments, it is suitable for testing against variable dataset sizes by considering different user defined measures of interestingness. It uses efficient database scanning techniques for search space pruning. In last few years, considerable work has been delivered towards sequence mining that uses the AprioriAll and its related algorithms [20] and [21]. The performance of these algorithms is highly affected due to the fact that a large number of candidate sequences and rules got generated during the mining process [22]. The manipulation of these huge candidates in main memory demands efficient data structures. Maintaining the available computing space, timely rule generation and minimization of database scans to an optimum elevation is a prevalent and open research challenge.

The AprioriAll algorithm uses the Hash Tree as the underlying data structure for handling the datasets [23]. Hash Tree can considerably minimize the size of the data sequences and the searching time as because it converts the data into (key, value) pairs instead of storing the actual datasets. Hashing is a very robust technique which have numerous applications to support the sequence mining [24]. Advanced hashing techniques such as the hierarchical hashing, Merkle hashing, perfect hashing, collaborative hashing etc. have evolved to meet various application specific requirements [25]. But, Hashing is not always the right structure for mining any type of datasets [26]. Common (key, value) pairs got generated for similar sequences due to which the collision domain increases and the management of these sequences in the hash table become a complex task. Moreover it is disadvantageous to maintain a huge number of link pointers for the Hashing. To overcome some of these difficulties the Prefix Tree structures have evolved [27]. It is a strong layout which is used with different sequence mining algorithms as the underlying data structure. It has also got numerous practical implications in recent times [28] and [29]. When there is more probability of occurring similar sequences in the datasets considered for mining, prefix tree could be a convenient choice.

Sequence mining in the field of Educational Domain has attracted many researchers' views. Analyzing the teaching learning correlations, Students' Learning behavior study, Course Structure Design, Educational Resource Planning, Book Recommendation System design, Students Performance Predictions are some of the open areas where interesting articles could be seen [30]. Due to availability and immense importance of this domain, an Educational dataset consisting of the marks obtained in different papers for each semester by some students in their undergraduate courses is considered for studying the effect of certain paper grades on their overall passing grade and the performance of the students. Market basket analysis and Financial Predictions are some of the other open research areas having continuous upward shifts. The Nifty 50 Index dataset consisting the daily price movements is considered to study the correlational effect of different participating features like daily high, daily low, shares traded, daily turnover, previous closing price etc. on the expected future price movements [31]. Many works related to market prediction and financial analysis using the sequence mining paradigms could be found in the literatures [32] and [33].

In summary, The AprioriAll algorithm is an ideal choice for analyzing the performance of different data structures due to its robustness and adaptability to support variable datasets e.g. the Educational dataset and the Nifty 50 Index dataset by considering the user specified measures of interestingness. The Hash Tree and TRIE are the widely implemented data layouts which could be considered as the underlying data structures for implementation with the algorithm for investigating the performances with respect to scan reductions, space and execution time requirements.

III. Methodology

Initially, a detailed survey of the related works is carried out for finding the advantages and drawbacks of some premier sequence mining algorithms and related aspects in order to formalize the research gaps. Then the data modelling activities such as data collection and pre-processing are performed. A description of the software and hardware requirement specifications are provided that suits the experimental needs. Different data structures used for handling the datasets during the sequence mining process is also presented. A generic sequence mining algorithm, the AprioriAll used in this work along with some variations in their underlying data structures and database scan reduction techniques. Performance evaluation matrices used in the experiments is also provided. Finally, a detailed analysis of the algorithms such as execution time during the mining process, effect of the performance evaluation matrices and database scans, space complexity etc. are discussed

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

in the result analysis and rule discovery sections. The following Fig. 1 depicts the overall workflow of the experimental works performed in this research work.

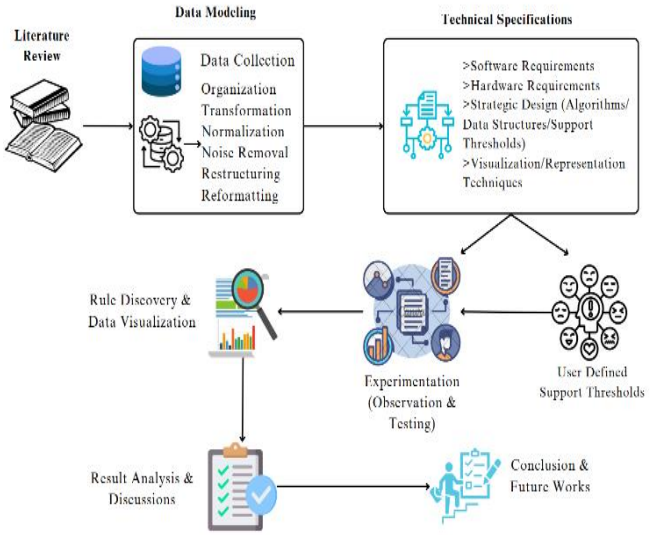

Figure 1. Process Flow Diagram

A. Data Collection and Pre-processing

For this experimental work two datasets are considered viz. (i) *Educational Dataset* – This is a highly dense dataset containing last 20 years raw data of passing grades obtained by 1000 students in their end-semester exam-papers of undergraduate studies. It has an attribute size of 31 (6 - Semesters having 5 - papers each with a passing grade and a final qualifying grade e.g. 101A, 102E, 103B, …….., 601B, 602E, 604A, 604A, 605A, Grade-A etc.). *Data Source* – Undergraduate Students' Examination Records, Department of Computer Science, Assam University, Silchar, Assam, India-788011. *Objective* - To generate positive and negative sequential rules from the above dataset describing the possibility of the cause and effect of a particular grade of some paper(s) on the overall passing grade of a students' performance. A sample segment of the Dataset is given in Table-I.

TABLE I. SEGMENT OF THE EDUCATIONAL DATASET

ID	Semester-I	.	Semester-VI	Gr
1	101A,103V,104B,102C,105V	.	601B,602A,603C,604C,605E	A
2	101C,103E,105E,104A,102C	.	601A,602C,603C,604A,605V	A
3	101C,103A,104C,105A,102C	.	601C,602C,603B,604B,605C	B
4	101A,102B,103V,104A,105E	.	601A,602V,603E,604V,605V	A
5	101B,102V,103O,104A,105V	.	601C,602B,603E,604B,605E	V
6	101C,103E,105E,104A,102C	.	601B,602A,603A,604C,605A	B
7	102A,103C,105V,101C,104C	.	601A,602A,603A,604A,605V	A
8	101A,103V,104B,102C,105V	.	601C,602B,603C,604C,605A	B
9	101B,103E,105E,102C,104C	.	601C,602B,603C,604C,605A	B
10	101C,104C,105E,102C,103B	.	601C,602B,603C,604C,605A	B

(ii) *NIFTY 50 Index Dataset* – This dataset contains last 25 years daily market movement data of NIFTY 50 Index price having 6221 records. It contains 7 attributes e.g. Previous Days' Closing price, Opening Price, Days' High, Days' Low, Today's Closing Price, Total Shares Traded, and Days' Total Turnover as described in Table 2 below. *Data Source*- https://www.nseindia.com/products-services/indices-nifty50-index. *Objective* - Generating Sequential Rules for supporting the prediction of future price index movements based on the other participating features of the dataset.

TABLE II. A SEGMENT OF THE NIFTY 50 INDEX DATASET

Days	Opening	High	Low	Closing	Shares Traded	Turnover in Cr
1	17244.5	17400.8	17238.5	17354.05	167025720	14588.54
2	17201.45	17264.05	17146.35	17203.95	320831676	21929.19
3	17220.1	17285.95	17176.65	17213.6	161679423	14320.79
4	17177.6	17250.25	17161.15	17233.25	176026100	14553.76
5	16937.75	17112.05	16833.2	17086.25	144777457	12567.03
6	937.35	953.1	937.35	946.7	49517340	1222.85
7	926.35	935.15	922.25	928.8	44473415	1050.05
8	932.45	937.95	925.65	927.8	26323660	665.51
9	924.3	932.6	919.55	931.65	35263845	866.74
10	941.95	944	925.05	927.05	49118380	1150.42

The raw datasets needs to be pre-processed before proceeding for the implementations. It may involve the activities like data organization, Transformation, Normalization, noise removal, restructuring, and formatting [34]. Data needs to be organized in such a way that the information in the data remains intact. If the data is not properly structured, important information may always be hidden even if efficient mining paradigms are imposed. Data uniformity plays a trivial role in mining process. Attribute values must be transformed into uniform formats throughout the transactions for unbiased rule discovery. Sometimes duplicate and Noisy data remains in the datasets which may lead to erroneous result and analysis. Therefore, these data needs proper normalization and cleaning activities. Sometimes, a little restructuring or reformatting of the data may considerably improve space efficiency. Initially the Educational dataset and the NIFTY 50 Index datasets considered here are collected as Jason formatted files and MS Excel formatted files respectively. Necessary pre-processing and cleaning activities are performed on both the datasets and transformed them into Parquet formatted files which have considerably improved the space requirements. These Parquet dataset files are then allowed for the sequence mining process to be carried out further.

B. Environmental Setup

For this experimentation, a number of hardware and software specifications needed to be setup. Most of the hardware and software support is provided by - Advanced Research Laboratory, Department of Computer Science, Assam University, Silchar, Assam, India-788011. Different other hardware and open source software supports have been incorporated into this work.

Hardware Specifications- For implementation and cross environment performance analysis, the following computing devices with networking abilities are employed – PC/Workstation with Core i3/i5/i7 processors, 10th/12th Gen, 4GB/8GB/16GB RAMs, without GPU/with GPU support, with

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Windows/Linux Platforms. For the algorithmic performance analysis, the Workstation is considered as the standard computing environment.

Software Specifications – Compiler/Debugger – Anaconda Library with python, Jupyter Notebook, pi-spark, JavaScript, Parquet viewer, IDE/Editor: JSON data –interchange formats, VS Code, notepad++, MS- Office Suite, Canva graphic designer tool, draw.io, Google-Colaboratory etc. Jupyter Notebook is used as the core programming tool for data preparation as well as for algorithmic implementations.

C. Algorithms and Data Structures

Algorithms - The association rule mining ignores the order of occurrences of items in a transaction. But the item occurrence order may carry vital information (e.g. AB ≠ BA). Sequence mining considers these ordering of items for generating sequential rules from the datasets.

AprioriAll is one such generic algorithm designed for mining rules from sequenced datasets which have got a wide range of application domains. Fig. 9 (*Appendix-A*) presents the working of the AprioriAll algorithm [4]. This algorithm requires that the database is to be scanned at least N-number of times for finding all the possible sequential rules where N- is the number of distinct attributes present in the dataset. In each database pass, sequences that do not satisfy the user defined minimum support threshold are pruned out resulting in search space minimization. It uses the Hash Tree as the underlying data structure for manipulating the data in main memory.

The Fig. 10 (*Appendix-B*) describes a modified version of the AprioriAll algorithm called as the *SinglePassAprioriAll* algorithm. It is implemented with a variations in both data layout and data retrieval techniques. It uses the TRIE as underlying data structure and a single database scan policy. Experimentations for the processing time and space requirements in accommodating different segments of the datasets in main memory using HASH and TRIE data structures with variable performance evaluation matrices are carried out. A detailed comparison of both the schemes- AprioriAll with HASH and AprioriAll with TRIE is presented. Interesting observations are noted while the database scan reduction and pruning techniques are incorporated in the algorithm during the implementation.

Data Structures – Sequence mining process involves generation of a huge volume of sub-sequences. Accommodating these subsequences in primary memory during the mining activities is a very tedious job. Hashing is a very popular technique for mapping such arbitrary number of records into a fixed sized records using a non-linear array. It uses the message digest function and stores records as (key, value) pairs in smaller tables called the Hash Tables. These smaller tables can also be arranged hierarchically to ensure multilevel hashing. Data is searched from the HASH tree using the (key, value) pairs instead of matching the original data. It has the search time complexity of O(N) while finding a data from the Hash Table [35]. Though Hashing has been widely used in manipulating records in main memory, it has some serious drawbacks of collision handling and pre-assigned memory requirements.

In practical sequence mining scenarios, repetition of similar sub-sequences and sub-sequence lattice growth are very common affairs. Due to this, the efficiency of the Hashing is drastically reduced. To overcome such difficulties, the prefix tree or the TRIE have come into existence that suits most of the sequence mining needs. In scenarios where there are chances of repeated occurrence of similar records, TRIE data structure for manipulating the data in main memory become a good choice. Though it has also the search time complexity of O (N), the reconstruction time for the prefix tree is less.

D. Performance Evaluation Matrices

During the sequence mining process, a large number of sub-sequences are generated. These sub-sequences form a huge search space lattice posing the difficulty of handling them in main memory. Moreover, not all sub-sequences generated out of the mining process have equal importance. In order to minimize the search space or to find the important sequences out of the huge sequence pool, the pruning or sequence-removal techniques are carried out. Support, Confidence and Lifts are some of the performance evaluation matrices and pruning measures that are incorporated in this work for dealing with the datasets for generating meaningful sequential rules [36]. *Support* is the ratio of occurrence of two or more sequences together to the total number of transactions in the dataset (Equation-1 below). *Confidence* is the ratio of occurrence of two sequences to the first sequence (Equation-2 below). *Lift* defines the ratio of occurrence of two sequences together to the product of their individual occurrence in the dataset (Equation-3 below).

$$Support(p,q) = \frac{\sigma(p \cap q)}{\mathrm{N}} \quad (1)$$

$$Confidence(p,q) = \frac{\sigma(p \cap q)}{\sigma(p)} \quad (2)$$

$$Lift(p \Rightarrow q) = \frac{Support(p \cap q)}{Support(p) \times Support(q)} \quad (3)$$

-where p and q are some sample sequences in a sequence dataset having N-number of transactions.

When the support and confidence of some sequence(s) is high, they qualify for generating strong sequential rules. Conversely, they generate weak sequential rules. The above measures are used as the user defined support thresholds for implementing the algorithms against the datasets.

IV. EXPERIMENTATION AND RESULTS

The following investigation framework depicted in Fig. 2 is used for the experimentation as well as testing purposes. The generic AprioriAll algorithm is implemented separately using the Hash and TRIE data layouts respectively against the two densely populated datasets for finding the sequential rules. The algorithmic performance is analyzed for primary memory requirements and execution time against variable user defined support thresholds and database scan minimization techniques. The performances show comparable results between the schemes during the exploration.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

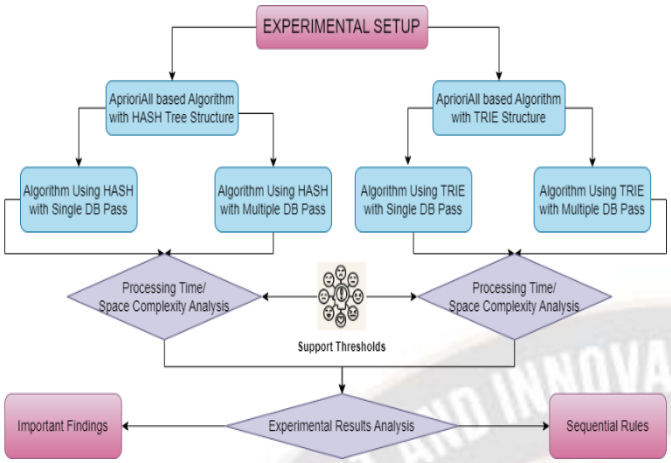

Figure 2. Experimental Work Flow Diagram

A. Experimental Result-1

(i) *Sequential Rules (Educational Dataset)* – The Educational dataset consists of the grades obtained by the undergraduate students in their end semester examinations. The paper-grades obtained by the students in each of the six semesters may have some associations among themselves which may have some impact on the overall passing grades of the students. The main objective of the sequence mining process in this context is to find any relation among the intermediate semester papers effecting the final passing grade of the students. The support and confidence constraints are used for minimizing the search spaces and for finding the qualifying results.

TABLE III. SAMPLE SUBSEQUENCES GENERATED FOR THE EDUCATIONAL DATASET

SL	Sequence(s)	Freq	Sup	Conf	Lift
1	601A	421	42.1	100	100
2	602B	341	34.1	100	100
3	605E	320	32	100	100
4	601A,605E	200	20	4.75	1.48
5	601C,602B	181	18.1	5.66	1.34
6	604C,605A	180	18	5.63	0.03
7	601C,604C,605A	160	16	5.00	0.03
8	601C,603C,604C	120	12	3.75	0.03
9	601C,603C,605A	120	12	3.75	0.03
10	601C,603C,604C,605A	120	12	3.75	0.03
11	601C,602B,604C,605A	100	10	3.13	0.03
12	602B,603C,604C,605A	80	8	2.35	0.02
13	601C,602B,603C,604C,605A	80	8	2.5	0.03
14	601C,602B,603B,604B,605V	41	4.1	1.29	0.03
15	601C,602C,603C,604C,605A	40	4	1.25	0.03

The above Table-III represents a segment of the sub-sequences generated out of the Educational dataset for a sample run of the Sixth semester paper grades. Based on the user specified support, confidence or lifts, the search space is narrowed down to get the most effective sequential rules. Higher the qualifying ratio of performance matrices of the sequences, higher is the chance of positive impact of it on the final results.

(ii) *Sequential Rules (NIFTY 50 Index Dataset)* – This dataset contains daily market movement records of 6221 days. Seven features of the dataset - Previous Days' Closing price(PC), Days' Opening Price (OP), Days' High (HI), Days' Low (LO), Today's Closing Price (CL), Total Shares Traded (SHTRD), and Days' Total Turnover (TURNO) along with their instances – very high (VH), little high (LH), average high (AH), very low (VL), little low (LL), average low (AL) etc. are selected as the participating factors. These instances are computed upon the percentage of market movements towards the higher or lower side in a day. This dataset is considered with a view point to predict future market movements based on these factors. A sample of the test run is provided in the following Table-IV that describes possibility of next days' market movement based on the previous days' closing price and number of total shares traded. Here also, the Support and confidence measures are used for search space pruning and sequential rule discovery.

TABLE IV. A SAMPLE OF SUBSEQUENCES GENERATED FOR THE NIFTY50 DATASET

SL	Sequence(s)	Freq	Sup	Conf	Lift
1	PCAH	1387	22.3	100	100
2	PCAL	1286	20.67	100	100
3	PCVH	1229	19.76	100	100
4	PCVH,OPVH	1040	16.72	84.62	0.07
5	PCVL,OPVL	870	13.98	81.69	0.08
6	PCAH,OPAH	853	13.71	80.09	0.05
7	PCVL,OPVL,CLVL	187	3.01	17.56	0.09
8	PCVL,OPVL,CLAL	130	2.09	12.21	0.07
9	PCVL,OPVL,LOAH	124	1.99	11.64	0.09
10	PCVH,OPVH,HIVH,LOVH	379	6.09	30.84	0.07
11	PCVL,OPVL,HIVL,LOVL	326	5.24	30.61	15.3
12	PCVH,OPVH,HIVH,CLVH	269	4.32	21.89	0.07
13	PCVH,OPVH,HIVH,LOVH,CLVH	255	4.1	100	0.07
14	PCVL,OPVL,HIVL,LOVL,CLVL	219	3.52	20.56	0.08
15	PCAH,OPAH,HIAH,LOAH,CLAH	89	1.43	6.42	0.05

B. Experimental Results-2

(i) Formation of Large Candidate Sequence Lattice - In this module, the AprioriAll algorithm is implemented on the datasets for finding sequential rules using the Hash Tree as the underlying data structure for manipulating the generated subsequences. The AprioriAll algorithm requires multiple passes over the dataset for generating the subsequences of different lengths. In first pass Length-1 sequences are generated, in second pass Length-2 sequences are generated and so on.

For a complete mining phase of a dataset having N-attributes, at least N-passes over the datasets is needed. For these N-distinct attributes, 2N-1 subsequences are generated. If a dataset contains T-number of transactions with N-number of attribute values, in worst case scenario, a total of (2N-1) ×T uncommon subsequences are generated. The algorithm makes use of efficient pruning techniques for deleting unimportant subsequences based on the user defined support threshold and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

confidence measures. The pruning minimizes the search space up to manifolds in subsequent passes during the mining process. The following Tables (Table V & Table VI) describes how the subsequences got generated during different phases of the mining process.

TABLE V. GENERATION OF CANDIDATE SEQUENCES FOR EDUCATIONAL DATASETS

Epoch	Records	L-1	L-2	L-3	L-4	L-5	Total
1	100	26	237	537	364	79	1243
2	200	28	303	802	600	134	1867
3	300	28	326	932	733	169	2188
4	400	29	344	1000	796	184	2353
5	500	29	355	1093	895	209	2581
6	600	29	357	1133	950	226	2695
7	700	29	361	1162	990	239	2781
8	800	29	365	1186	1016	246	2842
9	900	29	367	1121	1046	254	2817
10	1000	29	371	1242	1073	261	2976

TABLE VI. GENERATION OF CANDIDATE SEQUENCES FOR NIFTY50 DATASETS

Epoch	Records	L-1	L-2	L-3	L-4	L-5	Total
1	623	30	349	1268	1302	381	3330
2	1245	30	352	1414	1653	542	3991
3	1867	30	355	1514	1938	676	4513
4	2489	30	359	1620	2243	833	5058
5	3111	30	360	1715	2499	965	5569
6	3733	30	360	1841	2780	1098	6109
7	4355	30	360	1861	2866	1159	6276
8	4977	30	360	1887	2950	1208	6435
9	5599	30	360	1946	3128	1296	6760
10	6221	30	360	1978	3277	1383	7028

Both the datasets show exponential growth of candidate sequences during the sequence mining process. When no or minimal support threshold is provided, the sub-sequence lattice expands to its maximum size posing serious difficulty in loading them into the main memory as depicted in Fig. 3 below. Also the underlying Hash Tree structure used with the algorithm has to be upgraded to multilevel hashing when more common sequences got generated.

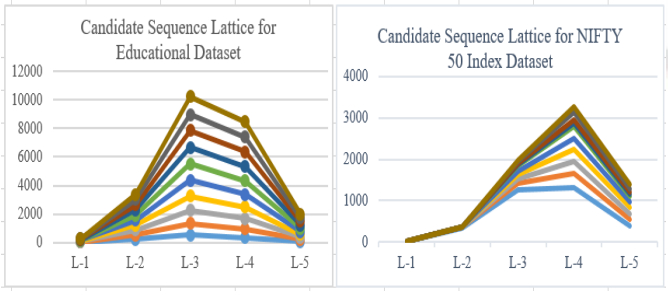

Figure 3. Candidate Sequence Lattices for the Datasets

(ii) Main Memory Requirement - The following Fig. 4 explains the additional memory required for holding the subsequences lattice in the main memory. It is seen that, at least three times of the original datasets memory is required for loading the subsequences into the primary memory. Moreover, accommodating the sub-sequences with multi-level hashing techniques integrates serious implementation difficulties both in logical as well as physical levels.

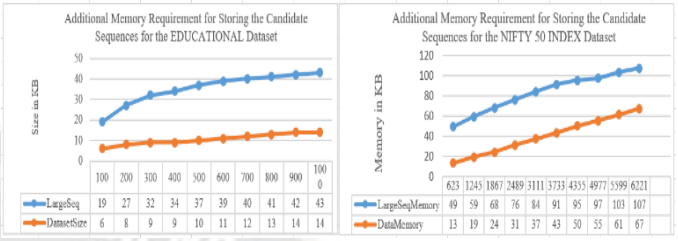

Figure 4. Additional memory required for storing candidate sequences of the datasets (In worst case scenarios, when Min_Sup=0)

The prefix tree or the TRIE is implemented in the subsequent experiments for investigating its suitability in different sequence mining contexts keeping in view the above difficulties. Achievements of the algorithms are also tested while the database scan minimization strategies are assimilated with the algorithm.

C. Experimental Result -3

The conventional AprioriAll algorithm uses multiple database passes and employs the Hash Tree as the underlying data structure. The number of database passes is equal to the number of distinct attributes present in the dataset. The algorithm suggests reconstruction of the Hash tree again and again each time the sequence-length or the support thresholds changes. To overcome these difficulties of multiple database passes and tree reconstruction issues, a new AprioriAll framework is implemented with the TRIE data structure and a single DB pass. All the possible sub-sequences out of the whole dataset transactions are generated on a single go. These are then stored using a single large prefix tree. Each time the user defined support thresholds change, there is no need for reconstruction of the tree, instead branches of the large prefix tree is traversed for manipulating the targeted-sequences. The burden of maintaining a large prefix tree in comparison to repeated reconstruction is more time-efficient.

(i) Execution Time (Educational Dataset) – The following Table-VII describes the Execution time required in each epoch of the educational dataset for the AprioriAll algorithm in finding the sequential rules. A comparison of the algorithm while using the "Hash tree structure with multiple database scans" and "TRIE data structure with single database scans" are implemented. This implementation is investigated in best case as well as in worst case scenarios when no user defined minimum support threshold is provided.

TABLE VII. EXECUTION-TIME COMPARISON FOR THE EDUCATIONAL DATASET

Epoch	Test Case	MinSup	AprioriAll with Hash	AprioriAll with TRIE
1	I	0	380.285	75.238
2	II	0	298.313	38.128
3	III	0	299.334	29.467
4	IV	0	311.959	30.759
5	V	0	302.029	32.158

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

6	VI	0	305.267	30.923

(ii) Execution Time (NIFTY 50 Index Dataset) – The similar experiments as given in the above case are implemented on the Nifty 50 Index Dataset for investigating the validity of the results obtained. Here also, a comparison of the AprioriAll algorithm while using the "Hash tree structure with multiple database scans" and "TRIE data structure with single database scans" is presented. This implementation is also investigated in best case as well as in worst case scenarios when no user defined minimum support threshold is provided. The Table-VIII describes the execution-time taken by both the algorithmic frameworks for a complete test run on variable size datasets where user defined minimum support is set to zero.

TABLE VIII. EXECUTION-TIME COMPARISON FOR THE NIFTY50 DATASET

Epoch	Test Case	MinSup	AprioriAll with Hash	AprioriAll with TRIE
1	623	0	814.457	69.093
2	1245	0	537.364	102.233
3	1867	0	758.089	152.361
4	2489	0	1080.188	159.926
5	3111	0	1234.026	195.476
6	3733	0	1235.129	224.572

The following Fig. 5 describes the execution time difference between the two schemes of the algorithms against the Educational Dataset. The AprioriAll algorithm using the TRIE as the underlying data structure is much more time-efficient than that of the algorithm using the Hash tree structure with variable support thresholds. Also the database scanning and pruning strategies plays a very vital role in the sequence mining process. When the data has to be repeatedly scanned and loaded from the secondary to the primary memory, a considerable time is consumed as described by the line graph in the following figure. Lower is the database scan, higher is the efficiency of the algorithm in respect of execution time.

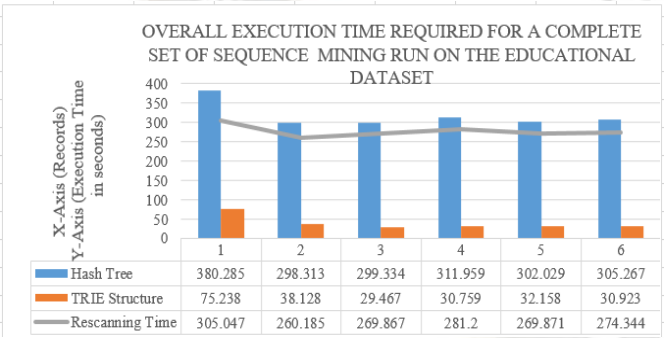

Figure 5. Candidate Sequence Lattices for the Educational Dataset

In case of the Nifty 50 Index dataset also, similar results could be found as shown in Fig. 6 below. The algorithmic paradigms are tested by varying the record size of the datasets. In every occasions, the similar type of performances are observed. Efficient data structure not only improves storage efficiency but also enhances the data access with great ease. Therefore, it may be stated that the data layout and the data retrieval techniques have direct impact on the sequence mining process.

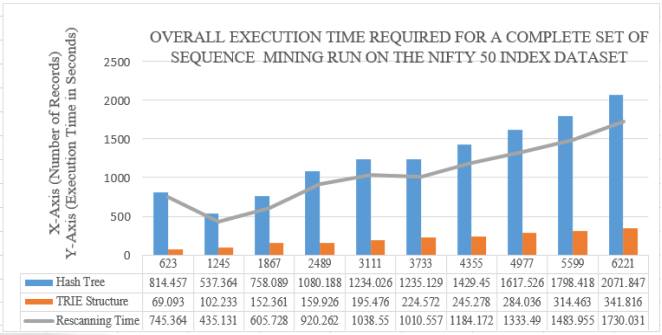

Figure 6. Comparison of Execution Time for Nifty50 Dataset

(iii) Tree Construction time – Datasets need to be analyzed using variable user defined support measures. This process requires repeated allocation & reallocation of memory in subsequent phases of sequence mining. While using the AprioriAll algorithm with the Hash Tree and multiple database scans, these allocation-reallocation process increases manifold thereby taking considerable time for the tree constructions. In case of the algorithm using Trie and Single database scan also the complete tree must be constructed before the actual mining process starts.

The following Fig. 7 and Fig. 8 depicts the time required for initial construction of both the Hash tree and Trie structures for the Educational datasets and Nifty 50 Index datasets respectively. It shows that, initially Trie takes some higher execution time than that of the Hashing due to the management of node pointers for individual sequences. Hash tree though takes somewhat less time for initial tree construction but it is repetitively reconstructed again and again taking comparatively longer time for a complete mining task. Trie with single database scan outperforms the hashing for completing the same mining task.

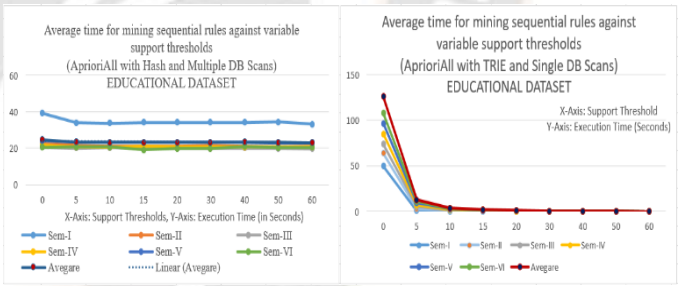

Figure 7. Tree Construction Time (Educational Dataset)

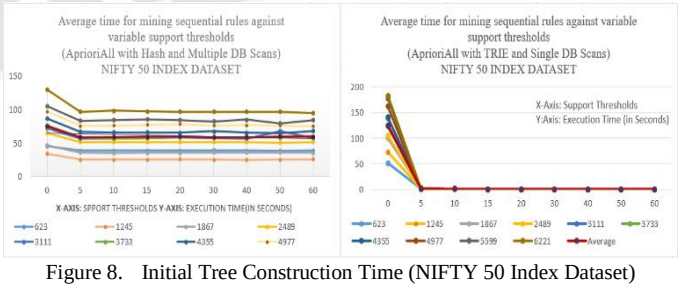

Figure 8. Initial Tree Construction Time (NIFTY 50 Index Dataset)

Trie structure with single database scan is a good option for sequence mining where there is sufficient abundance of main memory to hold the whole prefix tree of the implemented dataset. Hashing with multiple database scan is a good choice

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

where there is limitation of main memory and no fixed upper bound for execution time.

V. Results Analysis and Discussions

Due to the information explosion and huge data accumulation in real life applications, even terabytes of primary memory become insufficient for the high computing devices to load them into main memory. Data fragmentation is one such solution through which data are broken into slices for the ease of computation and memory allocation. But fragmentation poses serious threats of information loss especially when sequence mining scenarios are concerned. Horizontal fragmentation may lead to wrong interpretations e.g. if some attributes are occurring in contiguous transactions and they satisfy some user defined minimum support to qualify as heavyweight attributes but just after the first segmentation, it never or least occur in further transactions leading to biased rule generation. Likewise, vertical fragmentation may drop such an important attribute which may have influential impact on the others. But in sequence mining, every attribute and records are very crucial of its occurrence. An attribute occurrence may not have direct impact on other but, it may indirectly affect the meaning of occurrence of other attribute(s). Also, most sequence mining scenarios demand loading of the whole dataset into the main memory for generating the overall sequential rules out of the whole dataset instead of a segment of the same. But there is always a limitation on the available primary memory. To overcome and balance such tradeoff, efficient data layouts and retrieving policies are required for manipulating bigger datasets into the limited main memory during the mining process. Hashing and TRIE are the examples of popular data layouts which have been rigorously investigated in this study for their efficacies in terms of memory requirements and processing capabilities in terms of time and space especially in sequence mining scenarios.

In some situations it is likely that the dataset is highly dense where every attribute has a contribution to each of its records (e.g. the Educational and Nifty 50 Index datasets). The similar tuples may occur again and again thereby increasing the number of common sequences. When the Hash Tree structure is used to handle such sequences, same (key, value) pairs got generated which leads to more collision domains. Collisions in Hashing can be handled with techniques such as separate chaining and open addressing (e.g. linear probing, quadratic probing, double hashing) etc. But for both of these cases, maintenance of linked lists and pointers for a huge volume of collision domains is a tedious process. Moreover, cache performance exponentially decreases and extra space is consumed by the link pointers. Therefore, it is found that hashing is not a good choice to make for datasets where more common sequences are likely to occur. TRIE on the other hand is capable of handling longer as well shorter sequences with great ease. It also allows alphanumeric filtering and prefix matching ability which are the ideal requirements for a sequence mining paradigm generating voluminous common sequences. Due to its implementation feasibility and robustness of supporting algorithms like the AprioriAll, it is found to be a better option for mining sequential rules from datasets that produce repeated sequential occurrences.

In most sequence mining implementations, a large number of rules are generated. Not all of these rules are equally meaningful or important for the user. At the same time it is desirable that the rules generated out of the sequence mining process fulfils the user's needs. There are propped methods using average inter itemset distance as a parameter for reduction of such rules. Based on the performance evaluation matrices, the search space of sequence mining can be filtered and thus the large number of rules can be narrowed down to some targeted meaningful rules. Moreover, analyzing the real dataset in different situations like – "variable data size against fixed minimum support" or "fixed data size against variable minimum supports" is a necessity for most of the sequence mining applications. From these point of views, there is a need for multiple manipulation of datasets from secondary memory to the primary memory for rule discovery. But, these loading-unloading of data into the main memory leads to consumption of additional processing time and input/output overheads. For addressing the difficulties involved in such cases, a detailed investigation through extensive implementations are performed against the two datasets.

The AprioriAll algorithm with Hashing as the underlying data layout is tested with multiple database passes. It needs to construct the Hash tree again and again whenever the user defined parameters are changed resulting in increased latency of execution time in each mining phase. In another scheme, the AprioriAll algorithm is tested using Trie structure with single database pass. This implementation follows the Construct once, mine rules multiple times strategy allowing users to load the complete data into the main memory and analyze it using variable support thresholds. It limits the database scan to only once. It is observed that, the initial tree construction time for the TRIE is little bit higher than that of the Hashing for both the datasets because of the fact that Trie has to manage individual node pointers for each of the sub-sequences. But the Trie structure when used against variable thresholds on subsequent phases show considerable improvements in execution time in comparison to hashing.

Sometimes primary memory is a limiting factor where the processing time can be compromised. Alternately, in some cases, memory is in abundance but processing time must be in minimal. To maintain these tradeoffs, the sequence mining paradigms to be chosen effectively. Though no mining paradigm is perfect enough to fit into different sequence mining domains, but there is definitely a contributory significance of underlying data layouts and data retrieval techniques used with the mining paradigm as has been investigated thoroughly in the above sections.

VI. Conclusion and Future Works

In this paper, the AprioriAll sequence mining algorithm is implemented with an improvement by considering TRIE and Hash Tree as their underlying data structures and reduction in database scanning techniques. The TRIE data layout with a single database retrieval strategy is used with the algorithm for mining rules from the Educational dataset as well as the Nifty 50 Index Dataset. The objective of this work was to investigate the effect of data structures and multiple database passes on the efficacy of an algorithm during the worst case scenarios in the mining process. It is also taken into account how the selection of a proper data layout plays a vital role in finding sequential rules from a specific dataset. This study also provides an information about the time and space complexity tradeoffs for main memory consumption during sequence mining.

Extensive Experimentations against the two datasets show that the AprioriAll algorithms using the TRIE instead of the Hash as the underlying data structure have considerable

improvements in execution time in worst case situations. It is capable of finding targeted sequential rules from dense datasets against variable user-defined support thresholds instantaneously without taking the extra burden of multiple data load from the secondary memory. It uses the "*construct once and mine rules multiple times*" policy by loading the complete set of sub-sequences in a single prefix tree for manipulation in the main memory. This study of the algorithmic implementation fits better when there is a plenty of available primary memory and a higher possibility of occurring similar sequences in the dataset.

In practical situations, data is accumulated in exponential rates in comparison to the growth in hardware technologies such as storage devices or the primary memory capacity. Due to this imbalance, accommodating data in main memory during computation of rules from bigger datasets is an ever demanding problem. Design and Development of time-space efficient data structures and retrieval techniques for computing significant rules will always be a leading area of research. Investigations for the advancement and development of algorithmic paradigms for the performance efficiency in time-space tradeoffs will be carried out in the future endeavors.

ACKNOWLEDGMENT

The authors are grateful to the anonymous researchers and reviewers for their invaluable suggestions & comments for improving the quality of the article.

CONFLICT OF INTERESTS

The Authors declare that there is no Conflict of Interests for this research work.

REFERENCES

[1] C. H. Mooney and J. F. Roddick, "Sequential pattern mining - Approaches and algorithms," ACM Computing Surveys, vol. 45, no. 2. Feb. 2013. doi: 10.1145/2431211.2431218.

[2] A. Kour, "Sequential Rule Mining, Methods and Techniques: A Review," 2017. [Online]. Available: http://www.ripublication.com

[3] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Proc. 20th Int. Conf. Very Large Data Bases, pp. 487–499, 1994.

[4] R. Agrawal, R. Srikant, "Mining sequential patterns," Proc. 11th Int. Conf. Data Engineering, pp. 3-14, IEEE, March 1995.

[5] Y. Zhang and L. Paquette, "Sequential Pattern Mining in Educational Data: The Application Context, Potential, Strengths, and Limitations," pp. 219–254, 2023, doi: 10.1007/978-981-99-0026-8_6.

[6] D. Liu, "Construction of Higher Education Management and Student Achievement Evaluation Mechanism Based on Apriori Algorithm," Mob. Inf. Syst., vol. 2022, 2022, doi: 10.1155/2022/5375825.

[7] D. Y. Arora and D. Y. Arora, "Market Basket Analysis using Apriori Algorithm," Int. J. Innov. Res. Comput. Sci. Technol., vol. 2, no. 1, pp. 62–66, 2022, doi: 10.55524/ijircst.2022.10.3.12.

[8] W. A. Kamakura, "Sequential market basket analysis," no. May, 2014, doi: 10.1007/s11002-012-9181-6.

[9] V. Drakopoulou, "A Review of Fundamental and Technical Stock Analysis Techniques Journal of Stock & Forex Trading," vol. 5, no. 1, pp. 1–8, 2015, doi: 10.4172/2168-9458.1000163.

[10] D. D. Leeds, C. Chen, Y. Zhao, F. Metla, J. Guest, and G. M. Weiss, "Generalized Sequential Pattern Mining of Undergraduate Courses," pp. 2–6.

[11] Akshaya Kumar Mandal, Pankaj Kumar Deva Sarma, and Satchidananda Dehuri, "A Study of Bio-inspired Computing in Bioinformatics: A State-of-the-art Literature Survey," The Open Bioinformatics Journal 16.1, 2023.

[12] N. Mazumdar, P. K. Deva Sarma, "Sequential Pattern Mining Algorithms and their Applications," to be published.

[13] S. Tarun, R. S. Batth, and S. Kaur, "A Review on Fragmentation, Allocation and Replication in Distributed Database Systems," 2019 Int. Conf. Comput. Intell. Knowl. Econ., no. December, pp. 538–544, 2019, doi: 10.1109/ICCIKE47802.2019.9004233.

[14] F. Masseglia, F. Cathala, and P. Poncelet, "The PSP approach for mining sequential patterns," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 1510, no. November, pp. 176–184, 1998, doi: 10.1007/bfb0094818.

[15] B. Mallick, D. Garg, and P. S. Grover, "Incremental mining of sequential patterns: Progress and challenges," Intell. Data Anal., vol. 17, no. 3, pp. 507–530, 2013, doi: 10.3233/IDA-130591.

[16] J. Han and J. Pei, "Mining frequent patterns by pattern-growth," ACM SIGKDD Explor. Newsl., vol. 2, no. 2, pp. 14–20, 2000, doi: 10.1145/380995.381002.

[17] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and P. S. Yu, "A Survey of Parallel Sequential Pattern Mining," May 2018, doi: 10.1145/3314107.

[18] S. Itkar, U. Kulkarni, "Distributed sequential pattern mining: A survey and future scope," International Journal of Computer Applications, 1;94(18), Jan 2014

[19] J. Pei, J. Han, and W. Wang, "Mining sequential patterns with constraints in large databases," no. Section 6, p. 18, 2002, doi: 10.1145/584796.584799.

[20] K. B. Prajna, K. Abhishek, A. A. Shanbhag, B. V Shashank, and P. Ekshith, "Shopping Cart Analysis Using Apriori Algorithm and Association Rules," 2023 Int. Conf. Appl. Intell. Sustain. Comput., no. 2, pp. 1–5, 2023, doi: 10.1109/ICAISC58445.2023.10200434.

[21] Z. Ning, Z. Ouyang, and X. Deng, "An Improved Apriori Algorithm Based on Transaction Sequence Counting," 2023 IEEE 10th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)/2023 IEEE 9th Int. Conf. Edge Comput. Scalable Cloud, pp. 192–196, 2023, doi: 10.1109/CSCloud-EdgeCom58631.2023.00041.

[22] Sunita Parashar, "Performance Analysis of Apriori Algorithm with Progressive Approach for Mining Data," Int. Jour. of com. App. 31.1:13-18, 2011

[23] T. P. M and M. R. D, "Journal of Global Research in Computer Science Available Online at http://www.jgrcs.info A SURVEY OF HASHING TECHNIQUES AND ITS APPLICABILITY FOR," vol. 8, no. 9, pp. 1–5, 2017.

[24] C. Yan, X. Bai, J. Zhou, and Y. Liu, "Hierarchical hashing for image retrieval," Commun. Comput. Inf. Sci., vol. 772, pp. 111–125, 2017, doi: 10.1007/978-981-10-7302-1_10.

[25] M. Wilson, M. S. Nair, P. P. Nair, and A. M, "A perfect hashing to enhance the performance of Apriori algorithm," 2023 Second Int. Conf. Electr. Electron. Inf. Commun. Technol., pp. 1–6, 2023, doi: 10.1109/iceeict56924.2023.10157902.

[26] Cheval, Vincent, et al, "Hash Gone Bad: Automated discovery of protocol attacks that exploit hash function weaknesses," 32nd USENIX Security Symposium, 2023.

[27] J. F. Qu, B. Hang, Z. Wu, Z. Wu, Q. Gu, and B. Tang, "Efficient mining of frequent itemsets using only one dynamic prefix tree," IEEE Access, vol. 8, no. 2, pp. 183722–183735, 2020, doi: 10.1109/ACCESS.2020.3029302.

[28] C. Yue, G. Jiankui, W. Yaqin, X. Yun, and Z. Yangyong, "Incremental mining of sequential patterns using prefix tree," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4426 LNAI, no. 60573093, pp. 433–440, 2007, doi: 10.1007/978-3-540-71701-0_43.

[29] J. F. Qu, P. Fournier-Viger, M. Liu, B. Hang, and C. Hu, "Mining High Utility Itemsets Using Prefix Trees and Utility Vectors," IEEE Trans. Knowl. Data Eng., vol. PP, no. 8, pp. 1–14, 2023, doi: 10.1109/TKDE.2023.3256126.

[30] A. Bogarín, R. Cerezo, and C. Romero, "A survey on educational process mining," Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 8, no. 1, 2018, doi: 10.1002/widm.1230.
[31] N. Mazumdar, P.K. Deva Sarma, "Correlation analysis of stock index data features using sequential rule mining algorithms," In Proc. Int. Conf. on Data, Electronics and Computing, ICDEC 2022, in press.
[32] H. Liu, S. Huang, P. Wang, and Z. Li, "A review of data mining methods in financial markets," Data Sci. Financ. Econ., vol. 1, no. 4, pp. 362–392, 2021, doi: 10.3934/dsfe.2021020.
[33] Q. Al-Radaideh, A. Assaf, and E. Alnagi, "Predicting stock prices using data mining techniques," Int. Arab Conf. Inf. Technol., no. December 2013, p. 8, 2013, [Online]. Available: http://www.acit2k.org/ACIT/2013Proceedings/163.pdf
[34] S. Roy, P. Sharma, K. Nath, D. K. Bhattacharyya, and J. K. Kalita, "Pre-processing: A data preparation step," Encycl. Bioinforma. Comput. Biol. ABC Bioinforma., vol. 1–3, no. April 2020, pp. 463–471, 2018, doi: 10.1016/B978-0-12-809633-8.20457-3.
[35] S. Tapia-Fernández, D. García-García, and P. García-Hernandez, "Key Concepts, Weakness and Benchmark on Hash Table Data Structures," Algorithms, vol. 15, no. 3, 2022, doi: 10.3390/a15030100.
[36] B. W. E. Spangler, M. Gal-or, and J. H. May, "Introduction to Data Mining Association Rules," vol. 46, no. 12, pp. 67–72, 2003.

APPENDIX

***Appendix A*:** Flowchart of AprioriAll algorithm using Hashing and Multiple DB Scans for finding maximal sequences and sequential rules.

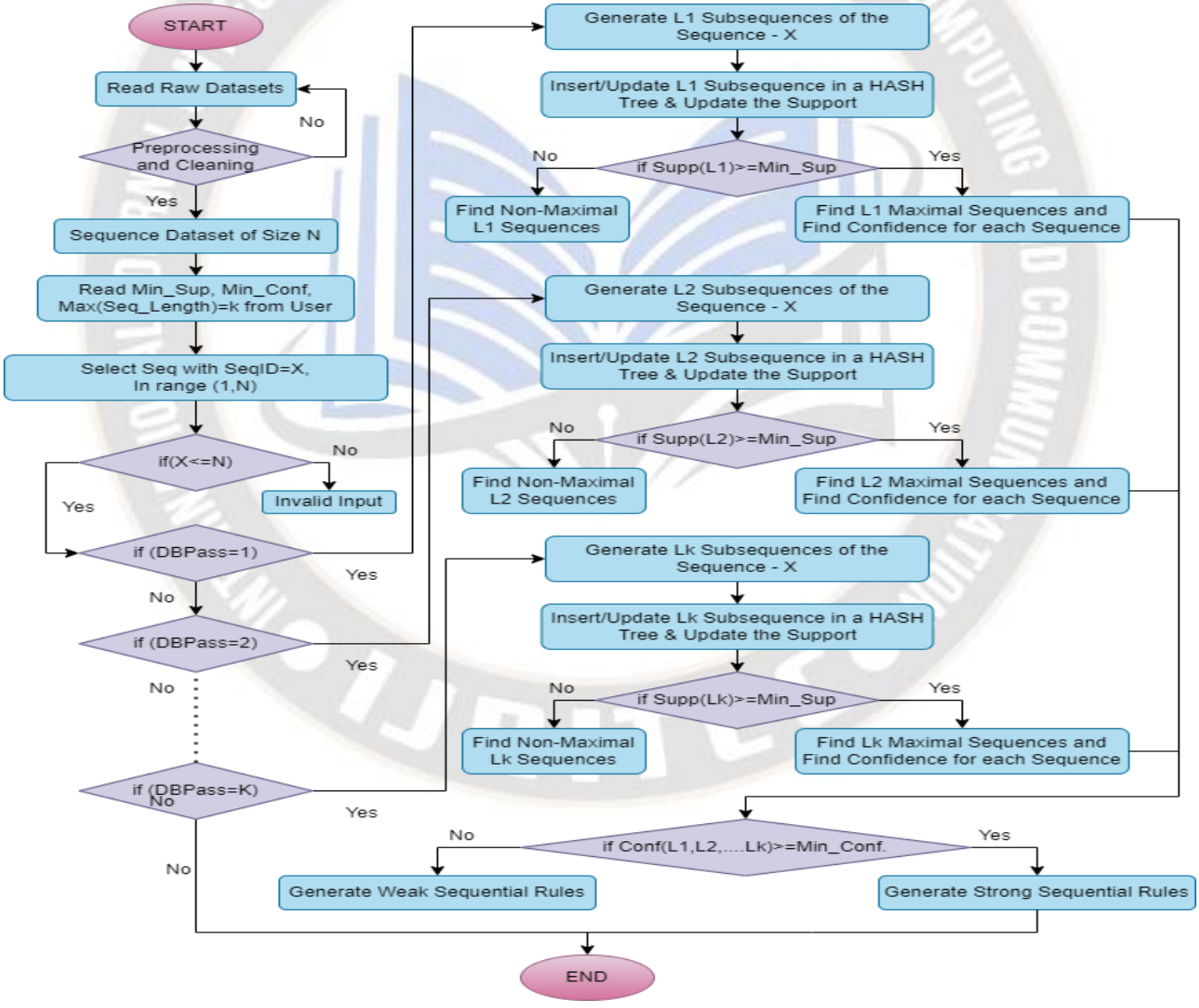

Figure 9. Process Flow Diagram of the AprioriAll Algorithm

Appendix B: Flowchart of Modified AprioriAll algorithm using single DB Pass and TRIE data structure for finding maximal sequences and sequential rules.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

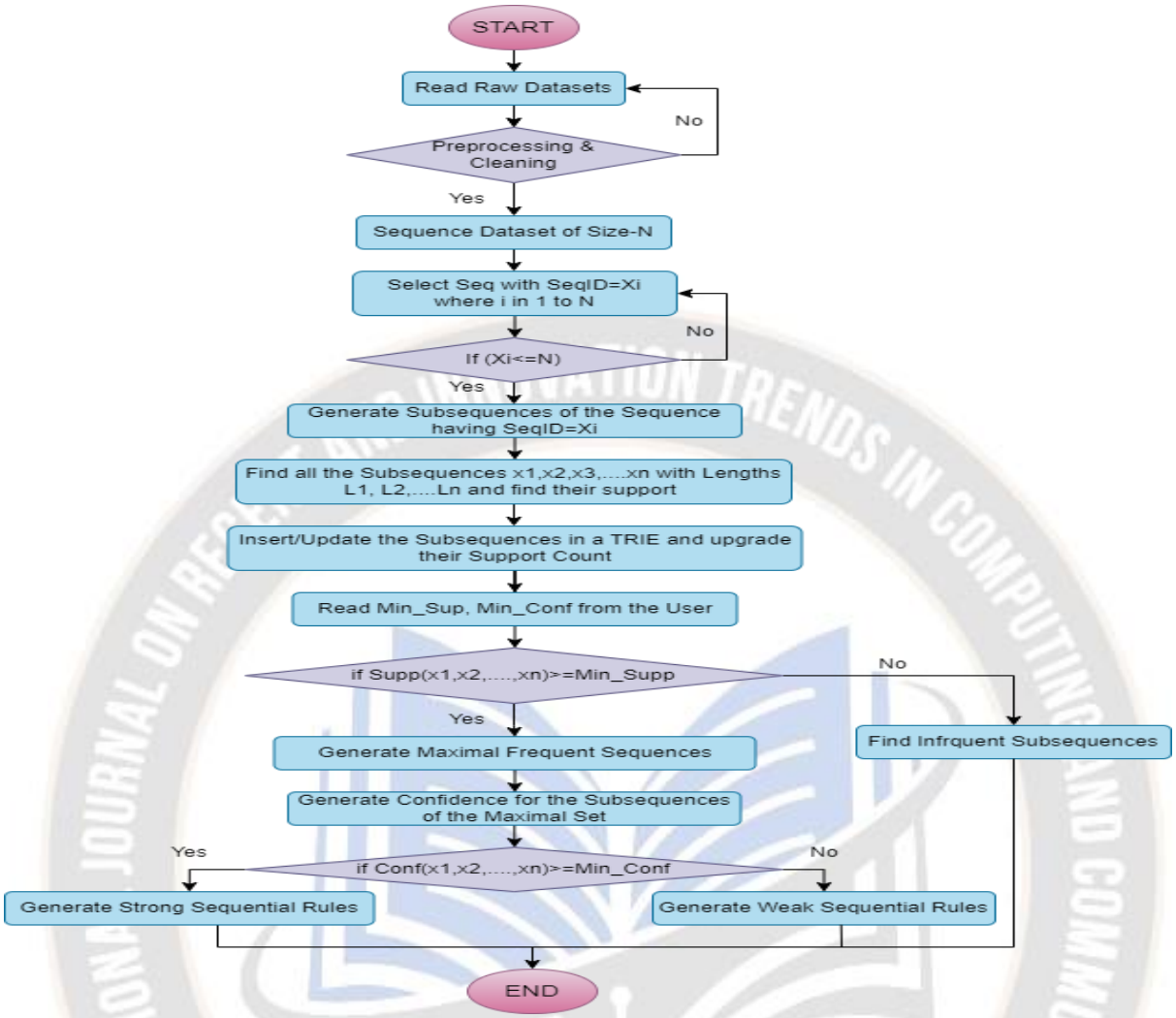

Figure 10. Process Flow Diagram of the *SinglePassAprioriAll* Algorithm