International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

SPM: Study of Different Techniques of Sequential Pattern Mining

Sujit R Wakchaure, Dr. Rajeev G Vishwakarma

Submitted: 21/01/2023 **Revised:** 15/03/2023 **Accepted:** 09/04/2023

Abstract: An essential part of data mining is the process of finding unexpected and significant patterns hidden within databases. In the past several years, one trend that has emerged in the field of data mining is the development of algorithms for the purpose of locating patterns in sequential data. Sequential Pattern Mining (SPM) is one of the most well-known data mining activities that can be performed on sequences. Finding relevant subsequence's within a set of sequences is the goal of this process. The interestingness of a subsequence can be evaluated based on a number of factors, such as how frequently it occurs, how long it lasts, and how much profit it brings in. Because data is encoded as sequences in many domains, including genomics, e-learning, market basket analysis, information extraction, and webpage click-stream assessment, sequential pattern mining has numerous real - world applications. This is because sequences are used to organise the data in these fields. This paper provides a comprehensive review of recent research on sequential pattern mining and its various applications. The purpose of this article is to evaluate recent developments in sequential pattern mining as well as provide an overview to the field of sequential pattern mining. This article offers a structured study on SPM as well as an analysis of the approaches that are used by SPM. In addition to this, it discusses the concerns, research issues, and future developments that are associated with sequential pattern mining.

Keywords: *Sequential Pattern Mining, Apriori Technique, Frequent pattern growth technique*

1. Introduction

The process of collecting information from data that is recorded in databases in order to better comprehend the data and/or make decisions about the data is known as "data mining." Clustering, categorization, identifying outliers, and mining patterns are among the most important data mining activities [3]. Discovering patterns in databases are not just intriguing but also useful and unpredictable is the goal of pattern mining. The Apriori method was initially used for identifying frequent itemsets, which are sets of items (symbols) that appear frequently together within a dataset of customer transactions [14, 15]. Specifically, the technique was created to identify common item sets. For instance, the Apriori algorithm can be utilized to discover patterns in a traditional retail dataset, like "cranberry juice, veggies, and kiwi," which indicates that these items are frequently purchased together by consumers. Another application of the Apriori algorithm is in the field of artificial intelligence [16]. The capacity of pattern mining approaches to identify patterns that may be buried in big databases and that can be decipherable by humans, and therefore valuable for comprehending the data and making decisions, is the primary driver behind the growing interest in these techniques. One pattern that can be utilised to assess customer behaviour and make strategic decisions to improve sales, such as cross-promoting products and providing discounts, is "dairy product, choco cookies."

Although pattern mining has become increasingly popular as a result of its applications in a wide variety of fields, several pattern mining methods, such as those used for frequent itemset mining [4, 10]] and rule

Department of Computer Science & Engineering, Dr. A. P. J. Abdul Kalam University, Indore(M.P.) 452010
sujitw2777@gmail.com
Department of Computer Science & Engineering, Dr. A. P. J. Abdul Kalam University, Indore (M.P.) 452010
rajeev@mail.com

mining, are designed to analyse data in a manner that does not take into consideration the sequential arrangement of events. This is the case despite the fact that pattern mining seems to have become extremely popular as a result of its uses in a wide variety of fields. When such pattern mining approaches are applied to data that contains information about time or sequential ordering, this data will be suppressed because of the pattern mining procedures. This may lead to the inability to uncover essential patterns in the data, or it may lead in the discovery of patterns that might not be useful since it overlook the sequential connection between the occurrences or parts. Both of these outcomes are possible. In different spheres, the sequence in which things happen are comprised of essential. For the purpose of analysing texts, for instance, it is frequently useful to take into account the sequence in which words are presented in phrases [10]. When it comes to detecting network intrusions, the order in which events occur is also very crucial.

The activity of mining sequential patterns was suggested as a solution to this problem in order to address it [1, 4, 5, 10, 11, 12] It is a well-known method for evaluating sequential data. It involves finding interesting subsequences within a set of sequences, in which the interestingness of a subsequence can be quantified in terms of many parameters such as the number of times it occurs, how long it is, and how much money it makes. Because the data throughout many fields are naturally encrypted as sequences of symbols, sequential pattern mining has several real-world applications. Some of these fields include bioinformatics, e-learning, market basket assessment, text categorization, conserving energy in smart homes and webpage click-stream assessment [7, 9]. In addition, SPM can also be used to time series (for example, stock data), provided that discretization is carried out as a step in the pre-processing stage.

This paper is segmented into various parts. In part 2, related work of various methods of sequential pattern mining performed by experts is discussed in detail. In part 3, various techniques of sequential pattern mining, key characteristics and classification of it is discussed. In part 4 and 5, various research challenges, conclusion and future trend is described in detail.

2. Literature Survey

In 2019, Muhammad J. Alibasa et al. [1] propose a technique for identifying behavioural traits from online activities using a SPM algorithm, and utilizing these as attributes for predicting emotions. The research was published in the journal Computers in Human Behavior. The findings indicate that the technique can be utilised to analyse the connection between digital utilisation and mood, with the capability to predict the former with an accuracy of 80%, which is considerably higher than the benchmark accuracy of 71.1%. This approach offers an additional perspective from which to probe the application of digital technologies to research pertaining to wellbeing. In 2020, Ji-Soo Kang et al. [2] suggests the prefixSpan oriented pattern mining utilising time sliding weight from data streams. It uses a time sliding weight to generate a structure of anticipated database tree in order to find sequential patterns, and it does this by applying the weight. When calculating the label and assist of the window for the time sliding weight, a time window is implemented to the sequential data. This helps ensure that the results are accurate. A table is populated with the time weight that was computed for every pattern as part of the process of designing a projected database tree. At this point, the tree is modified by removing the node that has a time weight that is inferior to the value that serves as a reference. As a result of this, the tree needs to be reorganised each time the data is revised. Through the application of time weights, the re - ordering process is able to eradicate the sequence of less influence. As a result, it is feasible to build a projected database tree that is capable of identifying influential trends. The efficiency

of the proposed approach is analysed from three different vantage points. To begin, the traditional PrefixSpan algorithm and the newly proposed PrefixSpan algorithm depending on time sliding weights in terms of the pattern generation time as per size of data are contrasted and compared with one another. In the second step of the process, a methodology is put through cross-validation to determine whether or not it is suitable. Thirdly, the proposed method is contrasted with the GSP, SPADE, and prefixSpan methodologies using the F-measure. As a consequence of the evaluation, the suggested methodology enhances the accuracy 75% as much as PrefixSpan method and the sequential pattern techniques of GSP and SPADE. Regarding the comparison of F-measure depending on precision and recall, the suggested one enhances its effectiveness about 83%.

In 2020, Jerry Chun-Wei Lin et al. [3] illustrate a successful mining advancement to extract the sequential high-utility patterns of ambiguous databases. A PUL-Chain framework is designed and formed in this paper with many pruning methodologies to reduce the search space of requisite patterns for mining improved efficiency. In contrast to the traditional HUS Span, the experimental findings demonstrate evidently that if both in runtime and in terms of the number of candidates identified, the developed algorithms demonstrated the efficacy of the discovered patterns as well as its mining efficiency when compared to the elder HUS-Span model. In 2019, Sad Jabbour et al. [4] propose a novel formulation of the activity of gradual itemset mining as the issue of SPM. This new formulation is presented as an improvement over the previous one. Utilizing SPM algorithms, this original reduction makes it possible to retrieve gradual item sets. The viability of the conceptual methodology was demonstrated through the collection of test findings on multiple numerical datasets. Chunkai Zhang et al. [5] have come up with a proposal for an effective distributed algorithm for HUSPM in 2019. The proposed algorithm makes use of the benefits provided by various machines and multi-core processors, which enables it to rapidly compete with other algorithms designed for mining. The most significant contribution are outlined as follows. To begin, a method for partitioning the initial database that is relied on the SLST approach is suggested as a method. Second, a methodology for HUSPM that uses multiple threads has been suggested. In conclusion, an effective and distributed methodology that has been given the name PHUSP is suggested on the basis of the two points. The outcomes of the experiments demonstrate that the proposed method is significantly more efficient than the methodologies that are considered to be state of the art.

In 2017, Md.Mahamud Hasan et al. [6] Mining common itemsets is a skill that is possessed by many people and used in a variety of real world scenarios. The Apriori algorithm, in addition to the FP Growth method, is one of the algorithms that is utilised one of the most frequently in the extraction of frequent item sets. On the other hand, there is a challenge involved in defining the minimum support necessary (the threshold) in order to extract frequent item sets using the Apriori and FP Growth method. If the minimum support is reduced, an excessive number of frequent itemsets will be produced, which might also lead the Apriori and FP Growth method become more inefficient or even lead to a loss of memory. On the other hand, when the minimum support is exceeded, fewer itemsets are discovered than when it is set to a lower value. A solution to this issue is proposed in this work in the form of a method that makes use of the Binomial Distribution (BD) to consider suitable minimum support in an adaptive manner. It has been made easier to mine efficient frequent item sets, which has resulted in the proposed method performing significantly better than the benchmark that was previously used.

In 2020, Swati Nagori et al. [7] Data Mining is the process of extracting useful information from large databases using various techniques.

SPM, is one of the essential concept of data mining. The SPM is utilised in exploring the consecutive trends which occurs in large datasets. It likewise recognises the unrelenting subsequences as illustrations from a database. In today's era, an immense amount of information needs to be gathered and get stored in the databases. Different sectors are concentrating towards extracting sequential patterns from such databases. SPM is the most recent method that has been developed, and it consists of the researchers locating sequential patterns. This paper presents a rigorous investigation into various SPM approaches. In addition to this, it offers a structured research study on SPM as well as an investigation into their methods. In addition to this, it discusses the concerns, research issues, and future developments that are associated with SPM. In 2018, Ingle Mayur Rajendra et al. [8] Data mining is the process of uncovering data patterns and concepts in huge data sets through the use of procedures that are at the convergence of the database management system. This process is systematic and follows a specific order. In the field of data mining, the retrieval of High Utility Element sets (HUI) is a common challenge, and the purpose of this study is to address it. The most aggravating aspect of these HUIs, which stand for "sets of items with high utilisation and value," is their combination of frequently occurring items. Another issue that can be solved is the prevalent issue of pattern mining, which is part of data mining and consists of the process of looking for recurring patterns in transaction databases. This is an issue that can be addressed. In order to fix the issue of the set of HUI, some very specific data as well as the most advanced methods available are needed. To hold the HUI, many popular methods have been proposed for this issue, like "Apriori", FP growth, etc. However, the most prevalent TKO algorithms (retrieval of utility element sets) K in one step and TKU (retrieval of elements sets Top-K Utility) are being used at the moment. In this context, TKO refers to the Top K in one step, and TKU refers to the Top K in utility. In this paper, a novel framework is proposed to mine k upper HUI, where k is the required number of HUI to retrieve, which allows to tackle all of the problems that have been discussed previously. The extraction of element sets with a high utility is not a process that is done very frequently. Despite this, technology is becoming more and more integrated into our day-to-day lives, such as in online shopping and other areas. This is a component of the overall business analysis. The implementation of a hybrid effective method for Top K high utility itemsets is the primary focus of this paper's investigation into potential areas of future research. It does this by implementing a hybrid of TKU and TKO, which has improved performance characteristics and eliminates the shortcomings of both of the individual algorithms.

In 2018, Bhargav C. Kachhadiya et al. [9] there are thousands of databases that can be accessed online; therefore, in order to access data based on knowledge and forecast the nature of data, some strategies are needed. If all of the data are filtered mechanically, it could take anywhere from a few hours to several days. In order to access or recover the data that is mined through data mining, it is necessary to employ certain methods. Dig up those meaningful data, and then utilise one of the many accessible approaches, including sequential pattern mining, to get at the data you've discovered. There are a lot of options. The mining of sequential patterns is the most difficult operation in data mining since the manual processes involved might take a significant amount of time. The act of mining data provides a variety of patterns derived from the data sources. Mining for sequential patterns involves uncovering frequent sequential patterns that are in compliance with user requirements and provide information that is both accurate and meaningful. It finds usage in a wide variety of applications, including natural catastrophe analysis, marketing plan analysis, shopping analyzation, medical assessment, analysis of DNA sequences, and evaluation of online log data. As of right

now, numerous websites are receiving anywhere from hundreds to millions of customers each and every day, evaluation of who browsed which pages and what information they viewed can provide vital insight into the specifics of the present visitors. This will assist in analysing the data from the website in order to forecast specific behaviours. According WU Jia et al. [10] in the year 2019, there are a great deal of traditional algorithms available for SPM in this era. In this group of methodologies, the PrefixSpan algorithm is among the most frequently used. This algorithm makes utilization of the prefix projection technology, which allows it to effectively avoid candidate items and, to some extent, improve mining efficiency. However, it requires the construction of a large amount of projection databases. The PrefixSpan method and the construction of the projection database not only require a lot of memory, but also require a lot of additional scanning time. As a result, the PrefixSpan method is enhanced and the ISPA algorithm is suggested in this paper. The ISPA algorithm is able to significantly cut down on the number of projection databases that need to be built, which in turn helps to improve the effectiveness of SPM. First, by contrasting the mining outcomes produced by the two methodologies, it has been discovered that the ISPA method is the one capable of locating the most essential sequence pattern, thereby satisfying. Second, tests are carried out on three various aspects: the various supports, the multiple kinds of data sets, and the size of the data sets. It establishes beyond a reasonable doubt that the ISPA method is superior to the PrefixSpan algorithm.

In the year 2020, Shah Mohammed Nuruddin et al. [11] found that in order to manage enormous amounts of data in the modern era, a high-computing system is required for the handling of these real - time data. High Performance Computing (HPC) systems provide a significant amount of computing capacity, which is useful for the effective handling of large amounts of data. Data mining is a challenging field for data scientists because it involves dealing with a variety of data types to uncover previously unknown relationships and identify intricate patterns among them. As the amount of information grows, existing algorithms, and methods need a little special computing environment so they are able to operate in real time. Data mining is an important field overall, and one of its most important subfields is sequence pattern mining. SPM helps to analyse various essential data fields in order to locate sequence patterns. When it comes to finding sequential patterns, among the most effective techniques is called prefixspan. Moreover, because the sequential programme of Prefixspan is only executed on a single cpu at a time, the amount of time needed to compute a large sequence database is significantly increased. The High-Performance Computing system consists of multiple processors that are connected to one another via a high-speed linkage. These processors are able to simultaneously calculate the same task. The research framework aims to minimise the completion time of Prefixspan using a Diverse computing system. A method is suggested in which two main activities of Prefixspan can be incorporated in a parallel manner in GPU by utilizing the NVIDIA CUDA framework. These two major activities are to discover frequent item sets and build projected databases. This investigation demonstrates a new method for creating projected databases, just start index as well as ending index can be saved for a pattern in a single array. So, memory usage can decrease in the HPC system.

Wensheng Gan et al. [12] present an effective Projection-based Utility Mining (ProUM) technique in their work from 2019, which seeks to find high-utility sequential patterns within sequence data. The construction of the utility-array was developed so that it could save the essential information regarding sequence order and utility. Because it makes use of the projection technique in the generation of utility arrays, ProUM is able to considerably improve mining efficiency while

also lowering the amount of memory that is required. In addition to that, a new upper bound is suggested which is called as sequence extension utility. The effectiveness of ProUM is further enhanced by the application of a number of different pruning procedures. The outcomes of the experiments reveal that the suggested ProUM algorithm achieves much better than the algorithms that are considered to be state-of-the-art. In 2020, Yu-Hao Ke et al. [13] Finding motifs in promoter sequences is absolutely necessary for advancing the knowledge of the process of transcription control. When trying to anticipate promoter motifs, researchers commonly make use of known promoter traits seen in a range of species. Consequently, the findings are not going to be of much help to anyone. Promoter binding sites have very few characteristics that are shared amongst different species. In the current investigation, various sequence analysis methods were employed in order to locate potential promoter binding sites across a variety of species. Its goal is to make the existing algorithm better suited to the process of extracting sequential patterns that contain a predetermined amount of gaps. In addition to that, the way in which the proposed method is described within the context of a distributed setting. The method that is being proposed locates Transcription Start Sites (TSS) and then isolates potential promoter regions from DNA sequences in accordance with TSS. It did this by deriving the motifs in the various promoter regions while simultaneously taking into consideration the amount of gaps in the patterns in order to deal with nucleotides that are not relevant. It was demonstrated that the motifs created from promoter regions utilising the suggested methodology are able to withstand nucleotides that are not significant. The effectiveness of the suggested strategy was validated by a comparison with previously identified promoter motifs.

3. Mining Sequential Patterns

The sequential pattern mining is a very essential topic in data mining. Association rule mining which is an expansion of the concept, seems to have a broad range of practical applications, and is another extension of the concept. It is a solution to the problem of determining whether or not a particular database contains sequences that occur frequently [2]. It is a technique that considers enticing sequential patterns between large databases and furthermore finds out prevalent sub-sequences as patterns from a data set. Additionally, it searches for intriguing sequential patterns between large databases. In many different sectors, large amounts of data are being captured and stored, and those sectors are interested in determining the sequential patterns that can be found in their databases. Sequential pattern mining have several applications, some of which include the examination of customer buying habits, patterns of access to the web, disease therapies, the discovery of cell phone calling trends, and the analysis of DNA sequences [13]. The following is a discussion of a variety of sequential pattern methods:

3.1 Sequential Pattern Mining by Apriori Based Technique

The Apriori and the AprioriAll are the foundation for a group of algorithmic techniques that are heavily reliant on the apriori property and employ use of the Apriori-generate join methodology in order to produce candidate sequences [16]. These algorithms are known as apriori-dependent. According to the apriori property, "All nonempty subgroups of a frequent itemset should also be frequent." a prevalent itemset is a set that occurs frequently. Another way to characterise it is as antimonotonic.

3.1.1 Key characteristics of Apriori-based methodology are:

- **Breadth-first search:** Since they establish all of the k-sequences in the kth algorithm's iteration as they navigate the search space, apriori-based techniques are referred to as breath-first methodologies. This is because of the order in which they build the sequences.

- **Generate-and-test:** The very first algorithms that were ever developed for sequential pattern mining made use of this feature. The techniques that rely on this feature only demonstrate an inefficient pruning technique, produce an explosive number of potential sequences, and then evaluate every candidate sequence individually to determine whether or not it satisfies some user-specified constraints. As a result, these techniques utilize a large amount of memory in the beginning phases of the mining process.
- **Multiple scans of the database:** This function involves conducting a search through the primary database to determine whether or not a lengthy list of automatically generated candidate sequences occurs frequently. It is a very unfavourable quality shared by the vast majority of apriori-based methodologies and necessitates a significant amount of execution time in addition to an increased I/O cost.

3.1.2 Classification of Apriori based mining technique

3.1.2.1 Generalized Sequential Pattern Mining (GSP): The GSP algorithm will run through the data several times in succession. This method is not one that operates in the main memory. In the event that the candidates cannot be stored in memory, the methodology will generate only however many candidates as can be saved in memory, and then it will scan the data to figure out which candidates have the most support. The frequent sequences that outcome from all these candidates are written to disc, while candidates that do not meet the minimum support requirements are deleted. This method is repeated as many times as necessary until each and every candidate has been regarded. As can be seen in Figure 1, the first step of the GSP method searches over all frequent sequences and then sorts them according to the support they have, disregarding the sequences whose support is lower than the min sup threshold. The method then searches the database to gather the minimum support for every candidate pattern and produces candidate length (k+1) sequence data from length-k prevalent sequences utilising Apriori [18]. This process is repeated for every level, which corresponds to sequences of length-k. This procedure is repeated numerous times until there is either no prevalent sequence or even no candidate left.

Frequent Sequence
Itemsets
< (2) (3) (4) >
< (2) (3) (6) >
< (2) (6) (4) >
< (3) (4) (5) >
< (3) (6) (4) >
< (4) (5) (6) >
< (6) (4) (5) >

Candidate Generation
Length-1 Itemsets
< (2) (3) (4) (5)>
< (2) (3) (6) (4)>
< (2) (6) (4 5) >
< (3) (4) (5) (6) >
< (3 6) (4 5) >

Candidate Pruning
Itemsets
< (2) (3 6) (4)>

Fig1: Generation of candidate and pruning in GSP

3.1.2.2 Sequential Pattern Mining With Regular Expression Constraints (SPIRIT): Utilizing regular expressions as an adaptable tool for constraint configuration is the original and innovative idea behind the SPIRIT method [4]. It applies a standard user-specified regular expression limitation to the patterns that are mined, which makes it possible to impose limits that are both incredibly versatile and extremely powerful. In order to incorporate the constraining into the mining procedure, the method utilizes a version of the constraint that is adequately relaxed, which means it is less rigid. This is so that it can push the restricting

within the mining process. There are multiple variations of the method, which vary in the extent to which restrictions are adhered to in order to narrow the pattern search space while it is being computed. These variations are available. Regular expressions (REs) are used as a constraint configuration tool because of two major factors that motivated this choice. To begin, REs offer a straightforward and natural syntax that allows for the condensed configuration of families of sequential pattern families. Furthermore, REs have enough expressive power to specify a wide range of intriguing and non-trivial pattern restrictions. This is because REs are recursive expressions [19].

3.1.2.3 Sequential Pattern Discovery using Equivalence Classes (SPADE): To ensure that every candidate sequence group can indeed be entirely stored in the primary memory, the SPADE methodology is used to partition the candidate sequence data into categories based on the items [18, 19]. Furthermore to that, the ID-List method is implemented by this method in order to lessen the workload associated with computing support counts. An ID-list of a series maintains a list of pairs, each of which indicates a role in the database in which the series can be found. The very first value in a pair represents a customer series, and the second value relates to a transaction in that sequence that includes the last itemset of the series. Together, these two values make up the pair. The ID-list of sequence <(p, g) (q)> in the example database shown in Table:1 is <(1,2), (1,6), (4,3), (4,4),> where the pair (1,2) indicates that this sequence begins in the first customer series and concludes in the second transaction. It is important to keep in mind that the same customer series can contain multiple instances of the same sequence, and as a result, more than single pair will be noted.

Table 1: Example of SPADE

CID	Customer Sequence
1	(p,e,g) (q) (h) (f) (r) (q,f)
2	(q) (d,f) (e)
3	(q,f,g)
4	(f) (p,g) (q,f,h) (q,f)

The support count of a candidate k-sequence can be determined using this method by combining the ID-lists of the any two frequent (k-1) patterns that have the same (k-2)-prefix. Taking into consideration of the database used in Table 1, The SPADE algorithm combines the ID-lists of patterns <(p, g)(h)> and <(p, g) (f)>, which are respectively <(1,3), (4,3)> and <(1,4), (1,6), (4,3), (4,4)>, in order to calculate the support count of the series <(p, g) (h) (f)>. This allows the method to determine how much support there is for the sequence. As a consequence of this, the ID-list for the pattern <(p, g) (h) (f)> is <(1, 4), (1, 6), (4, 4),> which indicates that this series is present in both the first and the fourth customer patterns and consequently has a support count of 2. The SPADE methodology incurs a significant amount of overhead whenever it must recurrently integrate the ID-lists of candidate sequences that contain frequent sequences.

3.1.2.4 Sequential Pattern Mining (SPAM): The lattice idea has been incorporated into the SPAM algorithm in order to cut down on the cost of merging; however, every ID-list is represented as a lateral bitmap. The depiction of the database that is used by the SPAM is a linear bitmap data structure, which is displayed in tables 2 and 3, and is comparable to the ID-list that is used by the SPADE. This data structure can be entirely kept in the main memory. The size of main memories has reached gigabytes and is continuing to grow, which means that many datasets ranging from

medium to large sizes will soon get to be completely memory native.

Table 2: Sorting of Dataset by CID and TID

CID	TID	Itemsets
1	1	{p,q,d}
1	3	{q,r,d}
1	6	{q,r,d}
2	2	{q}
2	4	{p,q,r}

Table: 3 Sequence for each customer

CID	Sequence
1	({p,q,d} {q,r,d} {q,r,d})
2	({q } {p,q,r})

A database D is a collection of tuples (CID, TID, X), at which CID is a customer-id, TID is a transaction-id dependent on the transaction time, and X is an itemset so that X is a Subset of I. Every tuples in D is indicates transaction. CID is an abbreviation for customer identifier; TID is an abbreviation for transaction identifier; and X is an A succession of item sets that is arranged by increasing TID can be regarded as a representation of every transactions that share the same cid.

Table 2 displays the data set for the transaction, which is comprised of tuples of the information (customer id, transaction id, itemset). First, it is grouped by the customer's id, and then it is arranged by the transaction's id. The sequenced depiction of the database is displayed in Table 3. The sequence that begins with customer 2, which has a size of 2, and a length that is 4, respectively are taken into account.

3.2 Sequential Pattern Mining by Pattern Growth Based Techniques

Almost immediately following the apriori-based methods that were developed in the middle of the 1990s, the pattern growth-method [20, 21] was developed in the early 2000s as a remedy for the issue of generate-and-test. The most important thing to keep in mind is to skip the stage of candidate generation entirely and to concentrate the exploration on a more limited section of the preliminary database. The pattern-growth process is significantly impacted by the functionality of search space partitioning. Nearly every single pattern growth method starts with the construction of a depiction of the database that is going to be mined. Next, the method suggests a technique for partitioning the search space. Finally, the algorithm produces as few candidate patterns as feasible by expanding on the already mined frequent patterns and implementing the apriori property even as search space is navigated recursively in an effort to find frequent patterns. The first projected databases were used by the early methodologies, such as FreeSpan and PrefixSpan, with the former being the most impactful [2].

3.2.1 Major characteristics of pattern growth-based algorithms

- **Partitioning of the search space:** It is possible to divide the search space that is produced for a large number of candidate patterns in order to manage memory more effectively. There are several distinct approaches to dividing up the search space. As once search space has been partitioned, subsequent partitions of a smaller size can be extracted in parallel. These more complex methods for search space partitioning are sometimes referred to as split-and-project methods. These techniques involve projected databases and implicit search [20].
- **Tree projection:** Tree projection is a technique that is typically utilised in

conjunction with pattern-growth methodologies. In this case, algorithms create a physical tree data format that is a depiction of the search space. This framework is then traversed either breadth-first or depth-first in order to find sequences that occur frequently, or pruning is determined by the apriori property [21].

- **Depth-first traversal:** The depth-first search of the search space makes a big difference in effectiveness, and it also assist in the initial pruning of candidate sequences as well as in the mining of restricted sequences. When compared to breadth-first or post-order, two traversal methods that were utilised by some of the earliest methodologies, depth-first makes significantly less use of memory, provides a rather more directed search area, and generates fewer candidate sequences as a result. This is the primary reason for this effectiveness [20, 21].
- **Candidate sequence pruning:** During the early stages of the mining procedure, pattern-growth methods make an effort to make use of a data structure that enables them to eliminate candidate sequences. Because of this, an earlier display of a more constrained search space is produced, and the procedure itself is kept more aimed and limited [21].

3.2.2 Classification of Prefix Growth based mining Technique:

3.2.2.1 FREESPAN: - While preserving Apriori's fundamental heuristic, the development of FreeSpan [2] was motivated by the desire to make Apriori's candidate creation and testing significantly more cost-effective. FreeSpan, in particular, makes use of frequent items in order to iteratively produce the sequence database into predicted databases while simultaneously growing subsequence splits in every projected dataset. Every projection creates sections in the database, which limits subsequent testing to modules that are gradually more controllable and successively narrower. The drawback is that there will be a significant amount of duplicate sequences as a result of the fact that the same sequence may show up in far more than one predicted database. Nevertheless, the dimensions of every projected database typically shrink quite a bit with recursion.

The itemset denoted by the notation d1U.... Udl is referred to as a's projected itemset when applied to the sequence α = (d1.....dl). The following feature serves as the foundation for FreeSpan: Any series whose projected itemset is a collection of M is not permitted to be a sequential behaviour if the itemset M in issue has a low frequency. Mining sequential patterns is performed by FreeSpan by first partitioning the search area and then projecting the pattern sub databases in a recursive manner according to the projected item sets.

Let the list among all frequent items in sequential data set D be denoted by the notation fq_ list = (m1.....mn). The entire collection of sequential patterns in S can then be broken down into n distinct subsets. These subsets are as follows: (a) the set of frequent patterns that only comprise item m1; (b) those that comprise item x2 but not any items in m3;... ; mn; and so on. In a broader sense, the ith subset (1< I <= n) is a set of sequences that includes item xi but excludes any item that appears in the pattern {mi+1;... ; mn}. After that, the database projection can be carried out in the following manner: When p's projected database is derived from DB, the collection of frequent items that make up database is already known at that point in time.

It is simply necessary to project into p's projected dataset those items that are contained in M. By doing so, meaningless data is efficiently removed, and the dimensions of the projected dataset are reduced to its smallest possible size. One is able to mine the projected datasets and produce the full set of sequences in the provided partition without having to repeat themselves if they perform these steps in a recursive manner. The following illustration will provide additional information on the specific details:

Table: 4 Example of Freespan

SequenceId	Sequence
1	<p (pqr) (pr) d (rf) >
2	< (pd) r (qr) (pe) >
3	< (ef) (pq) (df) rq >
4	< eg (pf) rqr >

The sequence database indicated by D is listed in Table 4, and the minimum support required is 2. The following items are contained in the database: {p, q, r, d, e, f, g }. FreeSpan begins by scanning S, after which it gathers the evidence that supports each item and locates the most common items. The most frequent items are presented first, followed by the least frequent ones, in descending order of support (in the form "item: support"), which means that fq_ list = p: 4, q: 4, r: 4, d: 3, e: 3, and f: 3. It organises into six sequences of length one: <p>:4; <q>:4; <r>:4; <d>:3; <e>:3; <f>:3: The entire collection of sequential patterns in D can reportedly be broken down into six distinct subsets, as indicated by the fq_list. These subsets are as follows:

1. The ones comprising only item p.
2. The ones comprising item q but nothing after q in fq_list.
3. The ones comprising item r but nothing after r in fq_list, and so on, and, ultimately.
4. The ones comprising item fq.

By building six projected databases, one can extract the sequential patterns that are associated with the six different subsets that have been partitioned (obtained by conducting a single more search through the primary database).

3.2.2.2 WAP-MINE: - It is a methodology for mining pattern growth as well as tree structures using its WAP-tree structure. In this case, the sequence database is only analysed a total of two times in order to construct the WAP tree using frequent sequences and the support they provide. In addition, a "header table" is kept in order to highlight to the first incidence of every item included in a frequent itemset, and this information is then monitored in a threaded manner in order to extract the tree for frequent sequences that construct on the suffix. It has been reported that the WAP-mine [9] method has high flexibility than GSP and that it performs significantly better than GSP. WAP-mine has a memory consumption issue because it iteratively reconstructs innumerable intermediate WAP-trees while mining, and in specific as the amount of mined frequent patterns rises. Despite the fact that it only scans the database multiple times and is capable of avoiding the issue of producing explosive candidates as in apriori-based methodologies, WAP-mine only scans the database twice. This allows it to prevent the issue of producing explosive candidates. The PLWAP method [10], which constructs on the prefix by making use of position coded nodes, was able to address this issue successfully.

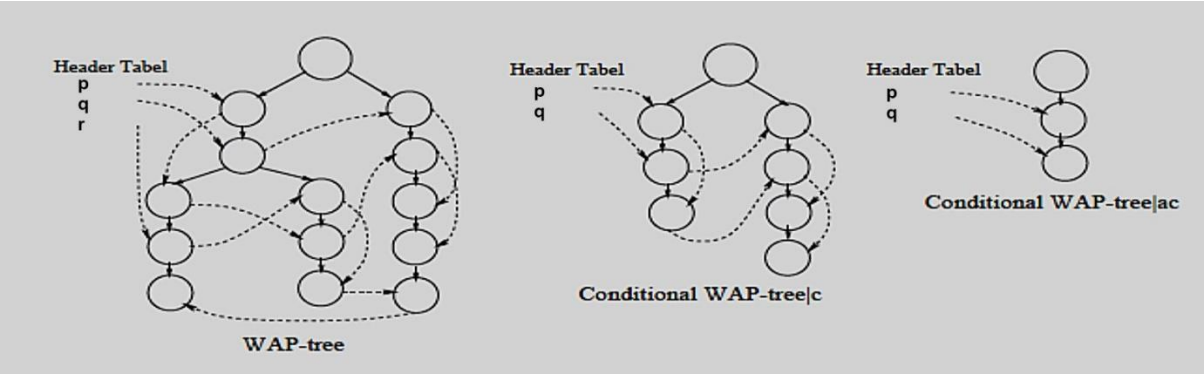

Fig 2: Classification of Prefix Growth based mining Technique

3.2.2.3 Prefix-projected Sequential pattern mining (PREFIXSPAN): - From the findings of the examination of the FreeSpan method, it is possible to see that there is still a possibility of having to incur significant costs when managing projected databases. It is possible to cut down on the size of the projected dataset as well as the time and money required to check the entire possible situation of a viable candidate sequence. Fixing the order of the items within every element is the first step one can take to prevent the time-consuming and potentially fruitless process of checking every possible pairs of a viable candidate sequence. One can reasonably assume that the items within a component of a sequence have always been listed in alphabetical order given that the elements can be mentioned in any order within that component without compromising the generalisation of the statement [2].

The existing research uses the projection scheme that is included in the PrefixSpan method to design the customer patterns into overlapping clusters that are referred to as projected databases [2]. This is done in such a way that all of the customer patterns that are included in every group contain the same prefix, which relates to a prevalent sequence. The instance database shown in Table 5 is used to demonstrate how the PrefixSpan method worked. Presuming that the min support count is two, the algorithm begins by searching the database for the most common 1-sequences, which are denoted by the notations <(p)>, <(q)>, <(e)>, <(f)>, <(g)>, and <(h)>. The projected database for every frequent 1-sequence is then generated by this algorithm once that step is complete. For example, the planned database of <(p)> can be seen in Table 5. The PrefixSpan algorithm begins the exploration of frequent 1-sequences in order to generate frequent 2-sequences with prefix <(p)> for this projected database. The PrefixSpan algorithm does this by generating the projected dataset in a recursive manner for every frequent k-sequence, with the goal of locating frequent (k+1)-sequences. To recursively create a huge proportion of projected databases using the PrefixSpan method clearly incurs a significant amount of expense.

Table: 5 Customer Database

CID	Customer Sequence
1	(p,e,g) (q) (h) (f)(r)(q,f)
2	(q)(d,f)(e)
3	(q,f,g)
4	(f)(p,g)(q,f,h)(q,f)

Table: 6 Projected databases of <p>

CID	Customer Sequence
1	(_, e, g)(q)(h)(f)(r)(q, f)
4	(_, g)(q, f, h)(q, f)

3.3 Extension of Mining Sequential Pattern

Mining sequential patterns has been the subject of extensive research over the past few years, and there is a wide variety of methodologies designed specifically for mining sequential patterns. In addition to this numerous variations of the initial formulation have been suggested, and these extensions may be connected to other kinds of time related trends or to the addition of time limitations. These extensions were inspired by the potential uses for the sequential patterns, and they have been proposed. In the following section, we will go over some extensions of such methodologies for specific purposes like multi - dimensional, shuttered, time interval, and constraint-based sequential pattern

mining. These are all examples of special purposes.

3.3.1 Multidimensional Mining of Sequential Pattern:

When mining sequential patterns with a single dimension, just one attribute, in addition to time stamps, is taken into consideration during the pattern discovery phase. While mining sequential patterns with different dimensions, numerous attributes can be taken into consideration. Mining sequential patterns in different dimensions, as opposed to mining sequential patterns in a single dimension, can result in the discovery of patterns that are both more insightful and much more useful [11]. For instance, a conventional sequential pattern can be acquired from the supermarket dataset by observing that soon after purchasing product a, the majority of customers then purchase product b within a predetermined amount of time. Nevertheless, through the use of multiple-dimensional sequential pattern mining, it is possible to discover various groups of people who have distinctive purchasing habits.

3.3.2 Identifying Time-interval Sequential Pattern:

Sequential patterns are able to tell what items are often bought together and on which order, but they are unable to offer information about the amount of time that elapses between items so that better decisions can be made. The solution to this issue is to generalise the mining issue into exploring time-interval sequences, that not only tells the sequence of items as well as the time intervals among successive items [11].

3.3.3 Closed Sequential Pattern Mining:

The sequential pattern mining techniques that have been developed up to this point have good performance when compared to databases that are comprised of short frequent sequences. Unfortunately, the effectiveness of such methodologies frequently degrades significantly when mining long frequent sequences or when using very weak support thresholds. This can happen either when using the thresholds. This should not come as a surprise: Assuming there is only one long frequent series in the database < (p1) (p2)... (p100) >, the algorithm will generate 21001 frequent subsequences if the minimum support is 1. However, all of the subsequence, with the exception of the longest one, will be superfluous because they will have the same support as < (p1) (p2)... (p100) >. Therefore, an alternative approach that is just as effective was suggested: rather than mining the entire set of frequent subsequence, one should only mine the frequent closed subsequence. This refers to the frequent subsequence that does not contain any super-sequences that share the same support. This mining technique will produce a significantly lower number of explored sequences in comparison to the conventional methods, but it will keep the same expressivity. This is because the complete set of frequent substrings, along with their supports, can be quickly derived from the acquired data [13].

3.3.4 Identifying Constraint Based Sequential Pattern:

Although there has been a significant improvement in the effectiveness of mining the entire collection of sequential patterns, sequential pattern mining nevertheless faces significant challenges in several areas, both in terms of its efficacy and its efficiency. On the one side, there is a possibility that a dataset will contain a great number of sequences. It's common for a user to have interest only in a tiny subsection of these patterns. It's possible that conveying the full set of sequences will make the extraction result hard to comprehend and even more difficult to put to use. In order to solve this issue, researchers have devised a method that systematically investigates the issue of pushing multiple constraints further into sequential pattern mining by utilising pattern growth methods [15].

4. Research Challenges

Discovering sequential patterns in relation to the original definition can currently be accomplished through the application of a number of different strategies. These kinds of

patterns are extremely well suited for an extremely wide variety of different applications. In spite of this, there are still a number of obstacles to investigate in this particular area of data mining. Among them are the following [7]:

• In order to make the process extremely systematic and scalable, a limited number of database scans are only performed.

• To be capable of encompassing a variety of different user-specific constraints.

• It is necessary for the methodology to be able to handle a large search space.

• During the course of the extraction and processing, the method must refrain from performing a second round of inspections on the database.

• To investigate constraints such as monetary and frequency constraints, as well as to analyse the results of these investigations with regard to execution speed, memory consumption, and scalability.

• To examine target aligned sequential pattern mining as well as its usage in some real-world datasets.

• To initiate the idea of object-aligning throughout this sequential pattern mining, which will allow for more flexibility in terms of extracting focused portions of the database? This will be accomplished via the use of sequential pattern mining.

• When there are larger number of databases, there is a possibility that obe will have to use distributed sequential pattern mining in order to obtain scalability.

The following are some additional possibilities for research that are mentioned in [7]:

• Come up with new algorithms that are both quicker and more memory-friendly.

• Develop brand new interactive methodologies that are less complicated and more intuitive to use.

• New algorithms have been suggested that are able to support a greater number of constraints and provide a higher level of visualisation power.

• Conceive of and create completely new pattern mining tasks that involve completely new obstacles,

• recommending improvements to existing problems that assist big data mining patterns, use parallel frameworks, and other such things;

• providing new apps for established methodologies.

5. Conclusion and Future Trend

Despite the fact that the concept of sequence data mining is relatively new, significant progress has been made in it over the past few years. The efficiency of the methodologies can be improved in a few different ways: with the help of the most recent structures, the most recent methods, or by the database systems in the memory storage. A variety of strategies have been proposed to address these issues, all of which are concerned with sequential pattern mining. This article divides the process of sequential pattern mining into two primary categories, namely, apriori and pattern-based algorithms, on the basis of the norms that have been proposed. There are a few prerequisites that need to be satisfied before we can consider an algorithm for mining sequential patterns as legitimate. To begin, a method that produces a search space that is as condensed as possible ought to be selected as the one to use. Both the pruning of early candidate pattern and the splitting of search space are features that are responsible for making this possible. Second, it is absolutely necessary to restrict the scope of this search area within the larger search area. There is also a methodology that can have a more specific searching process, similar to how depth-first search works. Thirdly, in order to find the genuine techniques of sequential pattern-mining, it is important to investigate methods that are not limited to tree projection. The following are some examples of potential application domains in which sequential pattern mining may play a significant role in the near future: sequential pattern mining with incomplete data, high utility item set mining, evaluating a

patient's prescribed medication or pathology test, the activity pattern of tourists and some campus placement trend of students. It's possible that another hot place where sequential pattern mining plays an important role is the development of a recommendation system and user behaviour analysis.

References

[1] Muhammad J. Alibasa, Rafael A. Calvo and Kalina Yacef. "Sequential Pattern Mining Suggests Wellbeing Supportive Behaviors", 2019, IEEE.

[2] Ji-Soo Kang, Ji-Won Baek and Kyungyong Chung. "PrefixSpan Based Pattern Mining Using Time Sliding Weight From Streaming Data", 2020, IEEE.

[3] Jerry Chun-Wei Lin, Gautam Srivastava, Yuanfa Li, Tzung-Pei Hong and Shyue-Liang Wang. "Mining High-Utility Sequential Patterns in Uncertain Databases", 2020, International Conference on Big Data (Big Data), IEEE.

[4] Saıd Jabbour, Jerry Lonlac and Lakhdar Saıs. "Mining Gradual Itemsets Using Sequential Pattern Mining", 2019, IEEE.

[5] Chunkai Zhang and Yiwen Zu. "An efficient parallel High Utility Sequential Pattern Mining algorithm", 2019, 21st International Conference on High Performance Computing and Communications; 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems, IEEE.

[6] Md.Mahamud Hasan and Sadia Zaman Mishu. "An Adaptive Method for Mining Frequent Itemsets Based on Apriori And FP Growth Algorithm", 2017, IEEE.

[7] Swati Nagori and Dr. Hemant Kumar Soni. "Issues and Research Challenges in Sequential Pattern Mining", 2020, International Conference on Advances and Developments in Electrical and Electronics Engineering (ICADEE), IEEE.

[8] Ingle Mayur Rajendra, Shri Chaitanya Vyas, Sanika Sameer Moghe, Deepali Deshmukh, Sachin Sakhare and Prof Sudhanshu Gonge. "Implementing a Hybrid of Efficient Algorithms for Mining Top-K High Utility Itemsets", 2018, IEEE.

[9] Bhargav C. Kachhadiya and Prof. Bhavesh Patel. "A Survey on Sequential Pattern Mining Algorithm for Web Log Pattern Data", 2018, 2nd International Conference on Trends in Electronics and Informatics (ICOEI), IEEE.

[10] WU Jia, LV Bing and CUI Wei. "An Improved Sequential Pattern mining Algorithm based on Large Dataset", 2019, International Conference on Power Data Science (ICPDS), IEEE.

[11] Shah Mohammed Nuruddin, Md. Didarul Islam, Md. Shafiqul Alam, Jesan Ahammed Ovi and Md. Ashraful Islam. "An Efficient Approach for Sequential Pattern Mining on GPU Using CUDA Platform", 2020, IEEE.

[12] Wensheng Gan, Jerry Chun-Wei Lin, Jiexiong Zhang, Han-Chieh Chao, Hamido Fujita and Philip S. Yu. "ProUM: High Utility Sequential Pattern Mining", 2019, International Conference on Systems, Man and Cybernetics (SMC), IEEE.

[13] Yu-Hao Ke, Jen-Wei Huang, Wei-Chen Lin and Bijay Prasad Jaysawal. "Finding Possible Promoter Binding Sites in DNA Sequences by Sequential Patterns Mining with Specific Numbers of Gaps", 2020, IEEE.

[14] Eirini Stamoulakatou, Andrea Gulino and Pietro Pinoli. "DLA: a Distributed, Location-based and Apriori-based Algorithm for Biological Sequence Pattern Mining", 2018, IEEE.

[15] Sudhakar Singh, Rakhi Garg and P. K. Mishra. "Observations on Factors Affecting Performance of MapReduce based Apriori on Hadoop Cluster", 2016, International Conference on Computing, Communication and Automation (ICCCA), IEEE.

[16] Mercy Nyasha Mlambo, Naison Gasela and Michael Bukohwo Esiefarienrhe. "Implementation and Analysis of Enhanced Apriori Using MapReduce", 2018, IEEE.

[17] S.Haseena, S.Manoruthra, P.Hemalatha and V.Akshaya. "Mining Frequent Item sets on Large Scale Temporal Data", 2018, 2nd International conference on Electronics,

Communication and Aerospace Technology (ICECA), IEEE.
[18] Mohammad Javad Shayegan Fard and Parsa Asgari Namin. “Review of Apriori based Frequent Itemset Mining Solutions on Big Data”, 2020, 6th International Conference on Web Research (ICWR), IEEE.
[19] Mohammad Javad Shayegan and Parsa Asgari Namin. “An Approach to Improve Apriori Algorithm for Extraction of Frequent Itemsets”, 2021, 7th International Conference on Web Research (ICWR), IEEE.
[20] Pan Zhaopeng, Liu Peiyu and Yi Jing. “An Improved FP-tree Algorithm for Mining Maximal Frequent Patterns”, 2018, 10th International Conference on Measuring Technology and Mechatronics Automation, IEEE.
[21] Lingxizhu, Yufeiguo and Jingyiwang. “Application of FPGrowth Algorithm of Sequential Pattern Mining on Container Maintenance Components Association”, 2020, 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), IEEE.