



Periodic Pattern Mining – Algorithms and Applications

By G.N.V.G. Sirisha, M. Shashi & G.V. Padma Raju

S.R.K.R. Engineering College, India

Abstract- Owing to a large number of applications periodic pattern mining has been extensively studied for over a decade. Periodic pattern is a pattern that repeats itself with a specific period in a give sequence. Periodic patterns can be mined from datasets like biological sequences, continuous and discrete time series data, spatiotemporal data and social networks. Periodic patterns are classified based on different criteria. Periodic patterns are categorized as frequent periodic patterns and statistically significant patterns based on the frequency of occurrence. Frequent periodic patterns are in turn classified as perfect and imperfect periodic patterns, full and partial periodic patterns, synchronous and asynchronous periodic patterns, dense periodic patterns, approximate periodic patterns. This paper presents a survey of the state of art research on periodic pattern mining algorithms and their application areas. A discussion of merits and demerits of these algorithms was given. The paper also presents a brief overview of algorithms that can be applied for specific types of datasets like spatiotemporal data and social networks.

Keywords: *periodic pattern mining; time series; sequence mining; dna sequences; spatiotemporal data, approximate periodic patterns, partial periodic patterns; statistically significant patterns, recommender systems, animal migrations, gene mutations.*

GJCST-C Classification : D.2.11



Strictly as per the compliance and regulations of:



RESEARCH | DIVERSITY | ETHICS

Periodic Pattern Mining – Algorithms and Applications

G.N.V.G. Sirisha ^α, M. Shashi ^σ & G.V. Padma Raju ^ρ

Abstract- Owing to a large number of applications periodic pattern mining has been extensively studied for over a decade. Periodic pattern is a pattern that repeats itself with a specific period in a give sequence. Periodic patterns can be mined from datasets like biological sequences, continuous and discrete time series data, spatiotemporal data and social networks. Periodic patterns are classified based on different criteria. Periodic patterns are categorized as frequent periodic patterns and statistically significant patterns based on the frequency of occurrence. Frequent periodic patterns are in turn classified as perfect and imperfect periodic patterns, full and partial periodic patterns, synchronous and asynchronous periodic patterns, dense periodic patterns, approximate periodic patterns. This paper presents a survey of the state of art research on periodic pattern mining algorithms and their application areas. A discussion of merits and demerits of these algorithms was given. The paper also presents a brief overview of algorithms that can be applied for specific types of datasets like spatiotemporal data and social networks.

Keywords: *periodic pattern mining; time series; sequence mining; dna sequences; spatiotemporal data, approximate periodic patterns, partial periodic patterns; statistically significant patterns, recommender systems, animal migrations, gene mutations.*

1. INTRODUCTION

Sequence is an ordered list of elements from any domain. The order among the elements of a sequence may be implied by time order as in stock market data or by physical position as in DNA or protein sequences [1]. If the order is implied by time order the sequences are called event sequences. Sequences where the order is implied by physical position are called biological sequences. Frequently occurring subsequences are referred to as sequential patterns. Sequential patterns with high support extracted from sequence databases are called frequent sequential patterns while the regularly repeating patterns found in a lengthy sequence are called periodic patterns [51]. Periodic analysis is often performed over time-series data which consists of sequences of values or events typically measured at equal time intervals [5]. Periodic patterns are classified based on different criteria. Periodic patterns are categorized as frequent periodic

patterns and statistically significant patterns based on the frequency of occurrence. Frequent periodic patterns are in turn classified as perfect and imperfect periodic patterns, full and partial periodic patterns, synchronous and asynchronous periodic patterns, dense periodic patterns, approximate periodic patterns. Periodicity is of two types. The first type is fixed or known periodicity and the patterns with fixed periodicity are daily traffic patterns, daily sales patterns in super market. The second type is unpredictable periodicity and the patterns with this type of periodicity are found in datasets like power consumption, telecommunication traffic, maintenance of vehicles, web click streams, seasonal changes in climate, data sent by sensors, biological sequences. Finding periodicity should be part of mining process for these patterns. Applications of periodic patterns include prediction of drought and flood, stock market price prediction, earthquake prediction, repeat detection in DNA sequences, detection of recurrent illnesses and prediction of fraud and other outlier patterns. [10]. Generalized Sequential Patterns (GSP) [2] proposed by Srikant and Agrawal, Sequential Pattern Discovery using Equivalence Class (SPADE) [3] proposed by Zaki, Prefix and Suffix Projection (PrefixSpan) [4] proposed by Pei et al. are a few algorithms for finding frequent sequential patterns. This paper presents a survey of state of art research on periodic pattern mining.

II. CLASSIFICATION OF PERIODIC PATTERNS

a) Full and Partial Periodic Patterns

Periodic patterns can be classified as full periodic (complete periodic) or partial periodic. Full periodic pattern is a pattern where every position in the pattern exhibits the periodicity. Periodic patterns in which one or more elements do not exhibit the periodicity are called partial periodic patterns. if $\{a\}\{b\}\{c\}\{b\}\{c\}\{b\}\{c\}\{a\}\{c\}\{d\}$ is an input sequence $\{b\}\{c\}$ is a full periodic pattern with period 2. It is all also called as full periodic pattern because every position in the pattern exhibits the periodicity. The sequence $\{a\}\{b\}\{c\}\{a\}\{d\}\{c\}\{a\}\{c\}\{c\}$ contains a partial periodic pattern $\{a\}\{*\}\{c\}$ with period 3 where the second element is not exhibiting the periodic behavior. Partial periodicity is a looser kind of periodicity than full periodicity and its application is more general because of the mixture of periodic events and non-periodic events in real world data.

Author ^α: Assistant Professor, Dept. of CSE, SRKR Engineering College, Bhimavaram, A.P., India. e-mail: sirishagadraj@gmail.com

Author ^σ: Professor, Andhra University College of Engineering, Andhra University, A.P., India. e-mail: smogalla2000@yahoo.com

Author ^ρ: Professor & HOD, SRKR Engineering College, Bhimavaram, A.P., India. e-mail: gvpadmaraju@gmail.com

b) Perfect and Imperfect Periodic Patterns

Periodic patterns can also be classified as perfect and imperfect periodic patterns. A pattern X is said to satisfy perfect periodicity in sequence S with period p if starting from the first occurrence of X until the end of S every next occurrence of X exists p positions away from the current occurrence of X . $\{a\}\{b\}\{*\}$ is perfect periodic pattern with period 3 in the sequence $\{a\}\{b\}\{d\}\{a\}\{b\}\{v\}\{a\}\{b\}\{f\}\{a\}\{b\}\{c\}$. $\{a\}\{b\}\{*\}$ has occurred for 4 times starting from its first occurrence till the end of the sequence. It is possible to have some of the expected occurrences of X missing and this leads to imperfect periodicity. $\{a\}\{b\}\{*\}$ is an imperfect periodic pattern in the sequence $\{a\}\{b\}\{c\}\{d\}\{b\}\{f\}\{a\}\{b\}\{g\}\{a\}\{b\}\{v\}$ and hence the confidence on its periodicity is $3/4$. The occurrences of $\{a\}\{b\}\{*\}$ is missed in one of its expected positions. Perfect periodic patterns are first proposed by Ozden et al [6]. They developed an algorithm to find cyclic association rules that reoccur in every cycle for the entire time series. This algorithm finds cyclic association rules from time stamped transactional data. An association rule is called cyclic with cycle $c = (l, o)$ if the association rule holds in every l^{th} time unit starting with time unit t_o . Here t_i corresponds to the time interval $[i.t, (i+1).t]$ where t is the time unit referring to time granularity given as input by the user. For example if the time unit is an hour and if a person buys milk and newspaper daily in the interval 7A.M-8A.M., then newspaper- $\rightarrow$ milk or milk- $\rightarrow$ newspaper has the cycle $c = (24, 7)$ (e. starting at 7A.M every 24 hours we can observe the purchase of milk and newspaper together). Given a fixed length period as input Han et al. mined segment-wise periodic patterns by using data cube, bit-array and apriori mining techniques [17]. Segment-wise periodic patterns are full or partial periodic patterns whose number of occurrences in a time series database satisfy the minimum confidence threshold. Cyclic association rules repeat perfectly in the time series with 100% confidence.

Later on Han et al. developed an algorithm called max-subpattern hit set [13] which created a maxsubpattern tree to mine full imperfect periodic patterns and partial imperfect periodic patterns in two scans of the time series. The algorithm can find the periodic patterns occurring with a given period in two scans of the sequence. It is also extended to detect the patterns for a set of periods (period range is given as input). For a given period the time series is segmented in to different segments called period segments where the length of each period segment is equal to the given period. The time series is scanned once and all 1-patterns which are found to be frequent in the segmented time series are reported. A 1-pattern is a pattern where one position in the pattern is defined, for example $(a, *, *)$ or $(*, a, *)$ or $(*, *, a)$. These frequent 1-patterns are unioned to form max pattern. Now the

time series is scanned for the second time. During the second scan of the time series each period segment is intersected with the maxpattern. The result of the intersection called max sub-pattern is either inserted into the tree if it is not already present or its corresponding node count is incremented if it is already present in the tree. Finally the max subpattern tree is then traversed and all patterns whose count is greater than min_conf are output as frequent periodic patterns. Max sub-pattern hit set algorithm for multiple periods works in a similar way where max sub pattern trees for all periods are constructed parallelly with two scans of sequence. If the number of 1-patterns mined in the first scan of the sequence is very large, the resultant max-pattern constructed is also very complicated. This results in a very large max-subpattern tree and a lot of time is wasted traversing the tree for finding the frequent periodic patterns.

Aref et al. extended the max-subpattern hit set by introducing algorithms for incremental, on-line and merge mining of partial periodic patterns [7]. Incremental version of max subpattern hit set algorithm allows the users to mine partial periodic patterns in case of insertion and deletion updates of time series database. It requires at most one additional scan of old database. Online version of the algorithm allows users to modify the thresholds used in mining algorithm while the algorithm execution is in progress. Merge mining aims at merging the patterns that are mined from two or more databases. This merging process helps us in discovering the patterns from combined database without having to reapply the mining algorithm.

S-S.Chen et al. proposed an algorithm [26] to find periodic patterns of given periodicity using the encoded period segments database and Frequent Pattern tree. As in max-subpattern hit set algorithm, for a given period ' p ' the given long sequence is segmented into segments where the length of each segment is ' p '. For example if the given sequence $S_D = \langle a\{b, c\}\{b, e\}aeabcb \rangle$, and period $p = 3$, S_D is segmented into 3 segments. The first period segment $S_1 = \langle a\{b, c\}\{b, e\} \rangle$, the second period segment $S_2 = \langle aeb \rangle$ and the third period segment $S_3 = \langle acb \rangle$. In S_1 , a is at position 0, b and c are at position 1 and b and e are at position 2. So, S_1 can be encoded as $\langle (a0), (b1), (c1), (b2), (e2) \rangle$. Similarly $S_2 = \langle aeb \rangle$ is encoded as $\langle (a0), (e1), (b2) \rangle$ and $S_3 = \langle ace \rangle$ can be encoded as $\langle (a0), (c1), (e2) \rangle$. After encoding all the period segments in this way, S_1, S_2, S_3 can be treated as separate sequences of a sequence database and frequent pattern tree with multiple minimum supports is used to find all the frequent periodic patterns. The main advantage of this algorithm is in its encoding step. After encoding the period segments in the way illustrated above, any frequent pattern mining algorithm can be used for mining the frequent periodic patterns. Instead of using single minimum support to determine the

significance of a pattern, the user can specify different minimum supports for different events based on their real life occurrence frequency. This allows the algorithm to find the rare but important patterns from the data. Recently K.J. Yang et al. proposed projection based partial periodic pattern mining [9] that can efficiently mine all partial periodic patterns that occur with the given period. On giving a period value this algorithm divides the given sequence into segments whose length is equal to period. It then encodes these segments into event tuples. This encoding is similar to the encoding technique discussed above[26]. The usage of event tuples together with recursive projection based algorithm speeds up the mining of partial periodic patterns. Only one minimum support is used to find the significance of the pattern.

All the above algorithms can be applied on datasets with predictable periodicity. However as the algorithms divide a long sequence into equal size segments where the size is equal to given period, the above algorithms can only detect patterns occurring within a segment possibly missing the patterns that overlap across the segments.

c) *Perfect and Imperfect Periodic Patterns*

All the above periodic pattern mining algorithms require the user to give period as input. These algorithms are appropriate for all the applications where the data consists of natural periods like hour, day, week, month, quarter and year. Some data sets may have patterns that repeat with unexpected periods. In such cases we need algorithms that can extract potential periods and the patterns that appear with these periods.

Pioneering work in this direction is done by Ma et al. to find all partial periodic patterns with unknown periods [28]. Two algorithms were developed to find partially periodic patterns. The former also called period first finds potential periods first and then for each period discovers the associations between events occurring with that period. The potential periods are detected using chi-squared test. A level-wise algorithm is then used to mine partial periodic patterns that appear with these periods. The later algorithm also called association first finds associations between events first. For each temporal association pattern it then discovers its periodicity. These algorithms can find partial periodic patterns even in the presence of replacement, insertion and deletion noise. One of the drawbacks of this work is that some valid periods may not be identified as it considers the difference between adjacent time instants only as potential periods.

People belonging to various areas such as economics, digital signal processing and statistics have also proposed various methods for periodicity search in time series. These methods include the usage of autocorrelation function, FFT, DWT for finding the periodicity of the time series. They focused on finding the periodicity only and did not explore mining of

patterns that occur with the detected periodicity. Unlike these methods Berberidis et al. [19] have used autocorrelation to detect all candidate periods from a discretized time series and applied max subpattern hit set algorithm to find the patterns. This algorithm scans the time series once to create a binary vector of size n for every symbol in the alphabet of the time series. For each symbol in the alphabet it computes the circular autocorrelation vector that contains the frequency count of each period. Among the periods from 1 to $n/2$, periods whose occurrence frequency is less than minimum confidence threshold are filtered out and the rest are kept as candidate periods. For each candidate period all the symbols occurring significantly with that period are unioned to form max pattern and Han's max subpattern hit set algorithm is applied to mine all frequent partial periodic patterns that occur with that period. The drawback of this method is that for any period, circular autocorrelation produces unexpected frequency values when the length of time series is not an integral multiple of that period. It also produces wrong frequency values for periods when successive occurrences of a symbol are repeated frequently periodically. The overall complexity of the algorithm using FFT as a filter is $O(AN \log N)$ where A is the size of the alphabet and N is size of the time series.

Usage of different networking resources has lead to the generation of different time series data in telecommunications and network applications. These include total number and duration of calls, number of bytes or electronic mails sent out from one ISP to another, amount of web traffic at a site etc [18]. Sale of a specific item over time also represents time series data. Time series are approximately periodic (noisy signals). Application of FFT on such data involves filtering of data before processing to guarantee accuracy. Even with very good filtering techniques and with no phase shifts in data, FFT cannot find all periodicities that are present in the data. In order to identify periodicities in noisy time series, Indyk et al. have proposed an algorithm named representative trends [18] to detect the candidate periods called relaxed periods in $O(n \log^2 n)$ time. This algorithm identifies periodicities but not the patterns that repeat with those periodicities so we need to use a separate periodic pattern mining algorithm that uses the detected candidate periods as input to generate the periodic patterns. Convolution based periodicity detection algorithm [15], WARP [16] and STAGGER [22] are other algorithms that can detect potential periods and periodic patterns in one pass of the time series. P. Huang et al. used an algorithm based on frequent partial periodic pattern tree for wireless spectrum occupancy prediction [48]. This prediction helps unlicensed users to use licensed wireless spectrum bands that are under-utilized i.e. the unlicensed users can use the available slots that are left unused by licensed users. This in turn

helps in improving the channel utilization and reduce collision rate.

Dutta et al. have proposed algorithms to find perfect and imperfect calendar based periodic patterns [30][31][32]. These algorithms are used to mine yearly, monthly, daily, hourly periodicities from discrete as well as continuous time series datasets. They find calendar based periodicities of an interval based temporal pattern. An interval based temporal pattern is a pattern that occurs across a sequence of time-intervals in either a discrete or continuous domain. These algorithms have a time complexity $O(n \log n)$ for a continuous domain and only $O(n)$ for a discrete domain.

G. Lee et al. applied parallel computing to efficiently mine frequent partial periodic patterns at all valid periods [34]. They are able to speed up the process of mining partial periodic patterns by partitioning the periods into independent subsets based on independence property of prime numbers. Each independent subset of periods will be assigned to a processor for detection of periodic patterns. Generation of redundant patterns was avoided thus increasing the efficiency of mining process as well as effectiveness and interpretability of generated patterns. A periodic pattern is redundant if the knowledge provided by it is obtained from other periodic patterns. For example in the following patterns 'Stock A's price increases every 2 days', 'Stock A's price increases every 4 days', the second pattern is redundant. Partial periodic correlation is a set of offsets within a particular period such that the data at these offsets are correlated with certain user desired strength. These partial periodic correlations are identified in [35] using principal component analysis method.

d) Synchronous and Asynchronous Periodic Patterns

A pattern which occurs periodically without any misalignment is called as synchronous periodic pattern. In the sequence $\{a\}\{b\}\{c\}\{a\}\{d\}\{c\}\{a\}\{c\}\{c\}\{d\}\{e\}\{f\}$, $\{a\}\{*\}\{c\}$, $\{a\}\{*\}\{*\}$, $\{*\}\{*\}\{c\}$ are the synchronous partial periodic pattern. These patterns have repeated for three times consecutively in the sequence with a period 3. Asynchronous periodic patterns are patterns with some disturbance between the repetitions of the pattern. Disturbance is allowed not only in terms of missing occurrences but also as insertion of random noise events. $\{a\}\{*\}\{c\}$, $\{a\}\{*\}\{*\}$, $\{*\}\{*\}\{c\}$ are asynchronous periodic patterns in the sequence $\{a\}\{b\}\{c\}\{a\}\{c\}\{c\}\{c\}\{b\}\{a\}\{b\}\{c\}\{a\}\{d\}\{c\}$. The above patterns have appeared for four times in the sequence where there is a disturbance between second and third occurrences of the patterns. A number of algorithms like LSI [8], SMCA [10], OEOP [11] and E-MAP [12] were developed to find these asynchronous periodic patterns in a single long sequence. Longest Subsequence Identification (LSI) is the pioneering

algorithm to mine asynchronous periodic patterns. For each asynchronous periodic pattern the algorithm detects the longest subsequence containing it. LSI is an iterative level wise search algorithm. It works in three phases. In the first phase it detects all potential periods for all events. This requires single scan of the sequence. In the second phase all candidate *1-patterns* are validated. A *1-pattern* is a pattern where one position is the pattern is defined and rest of them are ***'s. For example $\{a\}\{*\}\{*\}$, $\{b\}\{*\}$ are *1-patterns* with periodicity 3 and 2 respectively. An *i-pattern* is a pattern where *i* positions in the pattern are defined for e.g. $\{a\}\{b\}\{*\}$ is a *2-pattern* with periodicity 3 where 2 positions out of 3 are defined. In the third phase an iterative level wise approach is used where in the *i*th iteration candidate *i-patterns* are formed from (*i-1*)-*patterns*. Validation of these *i-patterns* require single scan of the sequence. The second and third phases of LSI require multiple scans of the sequence. Simple Multiple Complex and Asynchronous periodic pattern miner (SMCA) is a four phase algorithm that detects all the subsequences containing asynchronous periodic patterns. It can work with sequence of event sets also. A sequence of event sets is a sequence where each time instant contains one or more events. The first algorithm SPMiner finds *1-patterns*. The second algorithm MPMiner finds *1-patterns* containing simultaneously occurring events. The third algorithm CPMiner finds complex patterns (*ipattern for i ≥ 2*). For all the *1-patterns*, *1-patterns* with simultaneously occurring events and complex patterns the fourth algorithm APMiner finds subsequences where their appearance is significant. A pattern is said to appear significantly if the number of occurrences of the pattern in a subsequence is greater than a threshold *min_rep*. One Event One Pattern (OEOP) algorithm uses a linked list structure to detect single event one patterns in a single scan of a sequence. OEOP can be used to replace the first phase of SMCA for data sets like data streams. It can mine all *1-patterns* of all the events at all periods in a single scan of the sequence. E-MAP was proposed as a further improvement over SMCA model to mine all patterns (*1-patterns*, *1-patterns* with simultaneously occurring events, complex patterns) in a single step and single scan of the sequence. Among these algorithms LSI can only handle sequence of events where as the other algorithms can handle sequence of event sets. All these algorithms can handle replacement, insertion and deletion noise that is present in the sequence. These algorithms can detect asynchronous periodic patterns that appear with periods in the range $\max 2 \leq p \leq L_{max}$. L_{max} is the maximum periodicity that the user wants to check in the sequence. These algorithms produce many redundant patterns and they can be extended to mine only closed and maximal periodic patterns. In order to capture the hierarchical nature of asynchronous periodic

patterns a meta-pattern model is proposed by Yang et al. [27].

e) *Patterns with Symbol, Sequence and Segment Periodicity*

Some authors have classified periodic patterns as symbol periodicity or singular patterns, sequence periodicity or partial periodic patterns and segment or full-cycle periodic patterns. A sequence is said to have symbol periodicity if at least one symbol is repeated periodically. For example in the sequence $\{x\}\{y\}\{y\}\{x\}\{u\}\{i\}\{x\}\{t\}\{r\}\{x\}\{k\}\{l\}$, symbol $\{x\}$ is periodic with periodicity 3. A pattern consisting of more than one symbol repeating with same periodicity in a sequence leads to sequence periodicity. $\{x\}\{y\}$ (*) is partial periodic pattern in $\{x\}\{y\}\{z\}\{y\}\{z\}\{z\}\{x\}\{y\}\{y\}\{x\}\{y\}\{x\}$. If the whole sequence can be mostly represented as a repetition of a pattern or segment then that type of periodicity is called segment or full-cycle periodicity. For example $\{x\}\{y\}\{z\}$ is a full-cycle periodic pattern in the sequence $\{x\}\{y\}\{z\}\{x\}\{y\}\{z\}\{x\}\{x\}\{x\}\{x\}\{y\}\{z\}$. The periodicity and confidence of the pattern $\{x\}\{y\}\{z\}$ are 3 and $\frac{3}{4}$ respectively.

Rasheed et al. proposed suffix tree based noise resilient algorithm (STNR) [14] to detect symbol, sequence and segment periodicities that occur in the entire sequence or a subsection of it. Suffix tree is a data structure used in string processing. Suffix trees can be used to solve exact string matching problem, the substring problem, longest common substring of two strings etc. A suffix tree represents all the suffixes of the string for which it is constructed. For each suffix of the string there is a distinguished path from the root to corresponding leaf node in the suffix tree. Given a long sequence of symbols, a suffix tree for the sequence helps us in finding the number of repetitions of all substrings (or subsequences) and the occurrence positions of these subsequences in the sequence. For detecting periodicity, periodicity detection algorithm is applied on the occurrence vector of each subsequence (substring) whose occurrence frequency satisfies the *min_conf* threshold. The major limitation of this algorithm is that it cannot detect patterns like $\{a\}\{*\}\{b\}$, $\{a\}\{b\}\{*\}\{c\}\{d\}$ where the elements represented by $\{*\}$ do not exhibit the periodicity. It also misses some valid periods because it only considers adjacent time instants in calculating the periods. For example if 0, 3, 5, 6, 10, 13, 15 are time instants at which an event occurs. STNR algorithm considers the difference between every pair of consecutive time instants as candidate period. In the above example 3,2,1,4,3,2 are considered as candidate periods. The period 5 which is present in the data will be unidentified. This algorithm being noise resilient can work with sequences having replacement, insertion and deletion noise. While some of the existing algorithms

[15][16][17][18] can only detect periods that span through the entire sequence, STNR can detect periodic patterns that span through the entire sequence as well as a subsection of it. This algorithm cannot be used on eventset sequences. The worst case time complexity of the algorithm is $O(k.n^2)$.

M.G. Elfeky et al. have developed a convolution based algorithm to detect symbol and segment periodic patterns [15]. The algorithm scans the time series once to convert it into binary vector according to proposed mapping. It applies modified convolution on the binary vector to find the underlying periodicities and corresponding symbol and segment periodic patterns in single pass of the time series. To reduce the time complexity, convolution is computed using FFT. The time complexity of the algorithm $O(n \log n)$. It works well in mining synchronous periodic patterns. It fails in the presence of insertion and deletion noise.

Noisy data exists in almost all real world databases due to different reasons like errors in data entry, data transmission errors. In streaming applications data elements may be dropped purposefully for processing reasons. We need noise resilient algorithms in such cases. Elfeky et al. developed an algorithm named WARP [16] that uses time warping to detect the underlying periodicities and periodic patterns. It uses time warping to extend or shrink time axis at various locations to optimally remove noise. WARP is tolerant to insertion and deletion noise. The time complexity of WARP is $O(n^2)$. Though it is tolerant to noise, it can only detect segment periodicities and cannot find symbol or sequence periodic patterns. An online version of WARP called Online WARP was also developed to work with datastreams.

For mining periodic patterns from data streams Elfeky et al. have proposed a one-pass, online and incremental algorithm named STAGGER [22]. STAGGER discovers potential periodicities using multiple expanding sliding windows. With the continuous flow of the streamed data the sliding windows expand in length in order to cover the whole stream. Short length windows help us in identifying the short periods early and in real time. Larger length windows help in identifying the longer periodicities. For each discovered periodicity, STAGGER incrementally maintains a max-subpattern tree to find the corresponding periodic patterns.

f) *Dense Periodic Patterns*

Most of the periodic pattern mining algorithms can mine the periodic patterns only if they appear in the entire time series. A pattern which occurs in small segments of time series will not be detected by such algorithms because of lack of sufficient support or confidence. STNR can detect periodic patterns that appear in the entire time series or a part of it, but the end user has to give the range i.e. start and end

positions of the segment of the time series from which he wishes to mine the periodic patterns. This is a difficult task. Sheng et al. has proposed an algorithm [25] to mine dense periodic patterns from time series database. For each unique symbol the algorithm finds dense fragments from the time series. A dense fragment is a segment of time series where the distance between every pair of consecutive occurrences of the symbol is less than the distance threshold $d_{\max}$. All the dense fragments of all the symbols whose lengths are greater than $\min_len$ are retained. Then a max-subpattern tree is used to generate the partial periodic patterns using only the dense fragments of the symbols of the alphabet. This algorithm also prunes some periods by finding the lower bound period for each symbol in its each dense fragment. All the periods that are less than lower bound period can be safely pruned.

III. APPROXIMATE PERIODIC PATTERNS

Noise and imprecision exists in most of the real world data e.g. gene mutations in DNA sequence, so there is a need to mine approximate patterns. The algorithms that we have discussed previously which tolerate replacement, insertion and deletion noise mine approximately repeating periodic patterns. Here approximation is allowed in occurrence position of the patterns. In this section we are going to discuss a few more algorithms that allow approximation in occurrence positions as well as structure of the patterns. Y.L. Zhu et al. has mined approximate periodic patterns from hydrological time series [23]. Multi-event asynchronous periodic patterns were discovered from long sequence of hydrological time series. A modified suffix tree representation and traversal together with dynamic candidate period intervals adjusting method was used to mine all patterns without omitting any periods. Another approach to mine approximate periodic patterns was proposed by Amir et al. [24]. Given an input sequence and a parameter $\varepsilon \in [0,1]$ this algorithm finds for each unique event the longest ε -relative error periodic pattern i.e. the longest subsequence $s_1, s_2, s_3, \dots, s_k$ of S for which there exists a number p such that the absolute difference between any two consecutive occurrences of event in the subsequence is at least p and at most $p(1+\varepsilon)$. p is the approximate period of the relative error periodic pattern. Zhang et al. have proposed an algorithm to find periodic patterns with gap requirement from DNA sequences [33]. The algorithm generates all periodic patterns of the form $a_1 g(N, M) a_2 g(N, M) \dots a_{l-1} g(N, M) a_l$ where a_i for $1 \leq i \leq l$ is any symbol from the sequence, $g(N, M)$ is the gap between symbols and N, M are two user supplied numbers that specify the minimum and maximum gap sizes between every pair of consecutive symbols in the pattern.

IV. SURPRISING PERIODIC PATTERNS

In some applications like genomic data, system traces, credit card transactions and seismic data, patterns that occur less frequently yet whose occurrence frequency is beyond prior expectation are considered interesting. But all the periodic pattern mining algorithms discussed so far prefer mining periodic patterns that occur frequently. J. Yang et al. had proposed an algorithm named InfoMiner [29] to mine rare but statistically significant periodic patterns. A pattern is considered significant if its occurrence frequency exceeds its expected frequency by a large margin. In order to identify rare and statistically significant patterns a new model called as information model is used instead of support model. Information gain measure is used to identify the degree of surprise of pattern in a data sequence. Information gain of a pattern $G(p)$ is given $G(p) = I(p) * (\text{Support}(p) - 1) \cdot I(p)$. $I(p)$ is the information carried with p or surprise of p . $I(p)$ is less for a frequent pattern than that of a rare event. In the context of mining surprising periodic patterns a pattern that never repeats is of no use. So $I(p)$ is multiplied by $\text{Support}(p) - 1$ to identify the true periodic patterns with statistical significance. To improve the efficiency of the mining process, pattern pruning based on bounded information gain is used.

The occurrence positions of a pattern are not taken into account by Infominer. For example in $S_1 = a_1, a_2, a_3, a_3, a_2, a_3, a_1, a_2, a_4, a_5, a_1, a_2$ and $S_2 = a_1, a_2, a_1, a_2, a_1, a_2, a_5, a_3, a_2, a_3, a_4, a_5$ pattern (a_1, a_2) has the same information gain. In the first sequence (a_1, a_2) is scattered where as in the second sequence it repeats perfectly in first half of . In some application like repeat discovery in bio-informatics, consecutive repetitions of a pattern are more significant than the scattered repetitions [20]. Some penalty is associated with the gap between pattern repeats. So we need a new measure that distinguishes between consecutive repeats and scattered repeats of a pattern. J. Yang et al. proposed an algorithm called InfoMiner+ [20] which makes this distinction. It uses generalized information gain (GIG) measure. It detects all partial periodic patterns that are significant in one or more subsequences of an event sequence. For each pattern the subsequence that maximizes the pattern's generalized information gain is reported. It can deal with replacement noise. This algorithm detects all statistically significant periodic patterns whose periodicity is less than or equal to the maximum period bound $L_{\max}$.

Periodic patterns not only help in predicting future events but also help in identifying anomalies in data. F. Rasheed et al. have developed a framework for periodic outlier pattern detection in time series sequences [44]. Frequent periodic patterns are the periodic patterns that satisfy the user specified $\min_sup$ or $\min_conf$. Though $\min_sup$ and $\min_conf$ measures

can help us in identifying frequent periodic patterns they cannot be used to find outlier periodic patterns. The outlier periodic patterns repeat less often, the repetitions may not be strictly periodic, and the period is generally larger than the regular frequent periodic patterns. Rasheed et al. proposed a new measure based on the relative frequency of the pattern and its coverage area to estimate the surprise of the pattern. Then they used STNR [14] to find the periodicity of these surprising or outlier periodic patterns. As this framework uses STNR in periodicity mining, it can work with noisy data where periodic repetitions are not strict.

If $f(x)$ represents the number of repetitions of the pattern ' X ', and $segLen(X)$ represents the segment length of the repetitions of ' X ', then X is candidate outlier pattern if $f(x) < \mu(f(x_i))$ AND $segLen(x) > \min SegLen$; $\forall i$ such that $|x_i| = |x|$ where $\mu(f(x_i))$ is mean of frequency of all patterns of length exactly the same as that of pattern X . The measure of surprise of a pattern X is defined as one minus the ratio of the frequency of X over the average frequency of all patterns with same length as X . $Surprise(X) = 1 - \frac{f(x)}{\mu(f(x_i))}$; $\forall i$ such that $|x_i| = |x|$.

A candidate outlier pattern ' X ' is an outlier periodic pattern iff $surprise(X) > surprise_{min}$ and $Conf(X, i_{st}, i_{end}, p) > Conf_{min}$ where $Conf(X, i_{st}, i_{end}, p)$ is confidence of pattern ' X ' repeating with period p within the segment starting and ending at i_{st} and i_{end} respectively; $Conf_{min}$ and $Surprise_{min}$ are respectively the minimum confidence and minimum surprise values provided by the user. STNR-Out mines small number of non redundant outlier periodic patterns.

V. MINING PERIODIC PATTERNS FROM SPATIOTEMPORAL DATABASES AND SOCIAL NETWORKS

Periodicity is a commonly seen phenomenon in moving objects as well as social networks. People usually go to workplace every day through more or less same route, birds and animals migration from one place to another also show yearly periodicity. Interactions between people and topic discussion in social media also show periodicity, for example delivering news about different budgets in news sites and posting of information corresponding to different events like annual conferences in blogs and websites are done periodically.

The periodic pattern mining algorithms designed for event/symbol sequences cannot be used to find spatiotemporal periodic patterns. This is because the repetitions of spatial locations $\langle x, y \rangle$ may not be identical. Even if objects follow same route regularly, they may not appear at exactly the same location regularly. For example a person may go from home to office in the same route every day between 9.00 A.M. to 10.00 A.M. But it is less likely that he will be at the same location on his route everyday at 9.30 A.M. [37].

Cao et al. have proposed an algorithm to find maximal periodic patterns from spatiotemporal data [37]. This algorithm mines periodic 1-patterns using spatial clustering. Then bottom-up and top-down mining techniques are used for generating longer patterns. They have also proposed algorithms which can mine periodic patterns that appear in only a time interval instead of whole time span. Another algorithm which can mine shifted or distorted instances of patterns is also proposed.

Algorithm for mining and indexing of spatiotemporal periodic patterns from historical spatiotemporal data was developed by N. Mamoulis et al [42]. Z. Li et al. define a periodic behavior as repeating activities at certain locations with regular time intervals. A two stage algorithm called Periodica [38] is used to detect periods and find the periodic behaviors. In the first stage reference spots which are dense regions that are frequently visited in the movement are identified. Then the periods associated with each reference spot are identified using Fourier transform and autocorrelation [45]. Since every period is associated with a reference spot, if we find the periods associated with each reference spot we can guarantee that all the periods in the movement are detected. A period may be associated with two or more reference spots. In the second stage all the reference spots associated with a period are considered together for mining the periodic behaviors. A periodic behavior is a statistical description of the periodic movement for a period. It is a probability distribution matrix which gives the probability that the object is at each reference spot at different sub intervals of a period. For example when we mine periodicities of a school student we may observe 24 hours period at three reference spots namely class room and play ground and house at different hours of the day. The periodic behavior for the student gives the probability with which the student is at different reference locations namely class room, play ground and house in each hour of the day. In [46] the authors have extended this work by explaining how periodic behaviors can be used for missing data interpolation and future movement prediction.

Z. Li et al, have developed probabilistic model to mine periodic patterns from noisy and incomplete observations [36]. Mobile, GPS and Sensor technologies help us to track human and animal movements. But the data collected by these devices have a large portion of missing and noisy data. The data is also unevenly sampled and the rate of sampling in some cases like data sent by sensors attached to animal is very low. Fourier transforms and autocorrelation generally require evenly sampled data collected at high sampling rate. So these methods fail in this case. So a new probabilistic model is proposed to mine periodic patterns in these types of datasets.

T. Jindal et al. have proposed a method to mine spatiotemporal periodic patterns from traffic data [41]. These patterns can be used to summarize the speed distribution of traffic along any road like inter-state highways, state highways or a local road.

L. Riotte-Lambert et al. have used Fourier and Wavelet analysis to extract periodic patterns from time series of presence/absence, arrival/departure from areas of interest [47]. They have also introduced reliable null models for assessing the statistical significance of the periods detected. This work is an extension of Z. Li et al.'s work [38]. They declare that some of the valid periods may be missed or false periods may be detected if we use a single threshold to find significant frequencies from a periodogram as in [38]. To avoid this, different thresholds were used for different frequencies when finding significant frequencies from a periodogram. Wavelet analysis makes it possible to determine both the periodicities and time intervals during which the different periods are found to appear in the data. Though it is theoretically possible to use only wavelet analysis for periodicity detection the authors suggest to first perform fourier analysis as it is very fast and convenient way to determine the frequencies that are potentially relevant and then apply wavelet analysis. This two stage approach helps in easy interpretation of wavelet power spectrum. Wavelet analysis helps in identifying periodic events that occur in whole time series or only a part of it.

M. Lahari et al. discussed periodic subgraph mining for dynamic social networks [43] and applied it on four real world dynamic networks like Enron e-mails, Plains Zebra. K. Ishida had presented an algorithm to find periodically discussed topics in blogs, news sites and spam [39].

M. Lv et al. have used user's long term activity regularity to find mobile user similarity for location based social network services [50]. A new branch of social network services called location based social network services (try to understand users activities and preferences) recommends places, activities, friends and other geo-related information to the users based on the knowledge of users activities and preferences. Finding user similarity based on their long-term activities is important to achieve this. In [49] the authors have proposed a method to find users similarity based on user's long term activity regularity using GPS trajectory data. So by mining periodic patterns we will be able to develop recommender systems.

S. Parthasarathy et al. have presented robust periodicity detection algorithms to mine periodicities from continuous time series datasets [50]. These algorithms combine short time fourier transform and autocorrelation analysis for finding the periodicity. Various algorithms were developed to find periodicity in stationary, non-stationary, noisy and incomplete datasets. These can also find multiple interleaved

periods and discover hidden relationships among attributes in multidimensional time series.

VI. PERIODIC PATTERNS VERSUS FREQUENT EPISODES AND FREQUENT CONTINUITIES

Frequent episodes and Frequent Continuities are two classes of patterns that are closely related to periodic patterns. K. Huang et al. has given the following similarities and dissimilarities between them [40]. An episode is defined to as total or partially ordered collection of events in a specific time window. An episode considers only order among events instead of the actual positions of events in a time window bound. Mannila et al. presented a framework for discovering frequent episodes from a single long sequence through a level-wise algorithm WINEPI [21].

Continuity as defined in [40] is a kind of causal relationship which describes a definite temporal factor with exact position between the records. It is similar to a periodic pattern, but without the constraint on the contiguous and disjoint matches. Frequent episodes are a loose kind of frequent continuities since they consider only the partial order between events, while periodic patterns are a strict kind of frequent continuities with constraints on the subsequent matches.

VII. CONCLUSIONS AND FUTURE WORK

Periodic pattern is a pattern that repeats itself with a specific period in a sequence. Periodic patterns can be mined from datasets like biological sequences, continuous and discrete time series data, spatiotemporal data and social networks. Different criteria are used to classify the periodic patterns. This paper presents an overview of different types of periodic patterns and their applications along with a discussion of the algorithms that are used to mine these patterns. We have also discussed about efficiency, user interaction needed and noise resilient nature of these algorithms.

Ideas for future research directions in this area are the following. Major limitation of all the frequent pattern mining algorithms is that they mine a large number of patterns when *min_sup* is set very low. Like any frequent pattern mining algorithm, frequent periodic pattern mining algorithms also generate a large number of periodic pattern when *min_sup* or *min_conf* is set low. In order to reduce the redundancy of the generated output and to improve the efficiency of mining process there is a need for the development of algorithms that mine closed and maximal periodic patterns. Approximate periodic pattern mining is a new sub area of periodic pattern mining and a few algorithms were developed recently to mine such patterns. It needs to be extended to eventset (multidimensional) sequences.

Many of the existing periodic patterns mining algorithms mine patterns from 1 dimension sequences (event sequences). These algorithms were not applicable on multi dimensional sequences. These have to be extended for multi dimensional sequences so that more hidden patterns can be found from such sequences also.

With the growing size of the datasets, development of incremental and distributed periodic pattern mining algorithms has become a necessity. Usage of periodic patterns in constructing classification/prediction and recommender systems should be further explored.

REFERENCES RÉFÉRENCES REFERENCIAS

1. G. Dong, J. Pei, *Sequence Data Mining, Advances in Database Systems*, Springer science, ISBN-13: 978-0-387-69936-3, 2007
2. R. Srikant, R. Agarwal, "Mining sequential patterns: Generalizations and Performance Improvements", In Proc. 5th Int. Conf. Extending Database Technology", pp. 3-17, Avignon, France, 1996.
3. M. J. Zaki "SPADE: An Efficient Algorithm for Mining Frequent Sequences", Journal of Machine Learning, Vol. 42, Issue 1-2, pp. 31-60, 2001.
4. J. Pei, J. Han, B.Mortazavi-Asl, J. Wang, H.Pinto, Q. Chen, U. Dayal and M.-C.Hsu, "Mining sequential patterns by pattern growth: The prefixspan approach", IEEE Trans. Knowledge and Data Engineering, pp. 1424-1440, 2004.
5. J. Han, M. Kamber, "Data Mining: Concepts and Techniques", 2nd ed., Morgan Kaufmann, 2006.
6. B. Ozden, S. Ramaswamy, A. Silberschatz, "Cyclic association rules", In Proc. 14th Intl. Conf. Data Engineering.
7. W. G. Aref, M. G. Elfeky, A.K. Elmagarmid, "Incremental, online and merge mining of partial periodic patterns in time series databases", IEEE Trans. Knowledge and Data Engineering, 16(3), pp. 332-342, 2004.
8. J. Yang, W. Wang, and P. S. Yu, "Mining Asynchronous Periodic Patterns in Time Series Data", *IEEE Trans. on Knowledge and Data Engineering*, Vol. 15, Issue 3, pp. 613-628, 2003
9. K. J. Yang, T. P. Hond, Y.M. Chen, G. C. Lan, "Projection-based partial periodic pattern mining for event sequences", *Expert Systems with Applications*, 40, pp. 4232-4240, Elsevier., 2013
10. K. Y. Huang, C.H. Chang, "SMCA: A General Model for Mining Asynchronous Periodic Patterns in Temporal Databases", *IEEE Trans. On Knowledge and Data Engineering*, Vol. 17, Issue 6, pp. 774-785, Jun, 2005.
11. J.S. Yeh, S.C. Lin, "A New Data Structure for Asynchronous Periodic Pattern Mining", *Proc. 3rd Int'l Conf. Ubiquitous Information Management and Communication*, pp. 426-431, 2009.
12. F. Maqbool, S. Bashir, and A.R. Baig, "E-MAP: Efficiently Mining Asynchronous Periodic Patterns", *Int'l Journal of Computer Science and Network Security*, Vol. 6, no. 8A, Aug, 2006.
13. J. Han, W. Gong, Y. Yin, "Efficient mining of partial periodic patterns in time series databases", *In Proc. 15th Intl. Conf. Data Engineering*, pp. 106-115.
14. F.Rasheed, M. Alshalalfa, R. Alhaji, "Efficient Periodicity Mining in Time Series Databases Using Suffix Trees", *IEEE Trans. On Knowledge and Data Engineering*, Vol.23, Issue 1, pp. 79-94, Jan, 2011.
15. M.G. Elfeky, W.G. Aref and A.K. Elmagarmid, "Periodicity Detection in Time Series Databases", *IEEE Trans. Knowledge and Data Engineering*, Vol. 17, no. 7, pp. 875-887, July 2005.
16. M.G. Elfeky, W.G. Aref, A.K. Elmagarmid, "WARP: Time Warping for Periodicity Detection," *Proc. 5th IEEE Int'l. Conf. Data mining*, Nov. 2005.
17. J. Han, W. Gong, Y. Yin, "Mining Segment-Wise Periodic Patterns in Time Related Databases," *Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining*, pp. 214-218, 1998.
18. P. Indyk, N. Koudas, S. Muthukrishnan, "Identifying representative trends in massive time series datasets using sketches," *Proc. Int'l Conf. Very Large Data Bases*, Sept. 2000.
19. C. Berberidis, W. Aref, M. Atallah, I. Vlahavas, A. Elmagarmid, "Multiple and Partial Periodicity Mining in Time Series Databases," *Proc. European Conf, Artificial Intelligence*, July 2002.
20. J. Yang, W. Wang, P. Yu, "InfoMiner+: Mining Partial Periodic Patterns with Gap Penalties," *Proc. Second IEEE Int'l Conf. Data Mining*, Dec. 2002.
21. H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovering frequent episodes in sequences", *Proc. First Int'l Conf. Knowledge Discovery and Data Mining (KDD'95)*, 1995, pp. 210-215.
22. 1. M.G. Elfeky, W.G. Aref, A.K. Elmagarmid, "STAGGER: Periodicity Mining of Data Streams using Expanding Sliding Windows", *Proc. Int'l Conf. Data Mining*, 2006. Pp. 188-199.
23. Y. L. Zhu, S.J. Li, N.N.Bao, D.S. Wan, "Mining approximate periodic pattern in hydrological time series", *Geophysical Research Abstracts*, Vol. 14, EGU2012-515-1, 2012, EGU General Assembly, 2012.
24. A. Amir, A. Apostolico, E. Eisenberg, G.M. Landau, A. Levy, N. Lewenstein, "Detecting approximate periodic patterns", *Proc. First Mediterranean Conf. Design and Analysis of Algorithms*, 2012. pp.1-12 , springer-verlag, Berlin.
25. C.Sheng, W. Hsu, M.L. Lee, "Mining dense periodic patterns in time series data", *Proc. 22nd Int'l Conf. Data Engineering*, April, 2006.
26. S.S. Chen, T.C.K. Huang, Z.M. Lin, "New and efficient knowledge discovery of partial periodic patterns with multiple minimum supports" , *Journal*

- of Systems and Software, Vol. 84, Issue 10, Oct, 2011, pp. 1638-1651.
27. J. Yang, W. Wang, P.S. Yu, "Discovering High-Order Periodic Patterns", *Int'l Journal of Knowledge and Information Systems*, Vol. 6, Issue 3, May 2004, pp.243-268
 28. S. Ma, J. Hellerstein, "Mining Partially Periodic Event Patterns with Unknown Periods", *Proc. 17th Int'l Conf. Data Engineering*, Heidelberg, Germany, 2001.
 29. J. Yang, W. Wang, P.S. Yu, "InfoMiner: Mining Surprising Periodic Patterns", *Proc. Seventh ACM Int'l Conference on Knowledge Discovery and Data Mining*, 2001, pp. 395-400.
 30. A.K. Mahanta, F.A. Mazarbhuiya, H.K. Baruah, "Finding calendar-based periodic patterns", *Journal of Pattern Recognition Letters*, Vol. 29, Issue 9, 2008. Pp. 1274-1284.
 31. M. Dutta, A.K. Mahanta, "Mining Calendar-Based Periodicities of Patterns in Temporal Data", In the *Proc. Third Int'l Conference on Pattern Recognition and Machine Intelligence*. LNCS 5909, Springer-Verlag. Pp. 243-248.
 32. M. Dutta, A.K. Mahanta, "Detection of Calendar-Based Periodicities of Interval-Based Temporal Patterns", *Int'l Journal of Data Mining and Knowledge Management Process*, Vol. 2, Issue 1, Jan, 2012.
 33. M. Zhang, B. Kao, D. W. Cheung, K. Y. Yip, "Mining Periodic Patterns with Gap Requirement from Sequences", *Journal of ACM Transactions on Knowledge Discovery from Data*, Vol. 1, Issue 2, Aug 2007.
 34. G. Lee, W. Yang, J.M. Lee, "A Parallel Algorithm for Mining Multiple Partial Periodic Patterns", *Journal of Information Sciences*, Vol. 176, 2006, pp. 3591-3609.
 35. Z. He, X.S. Wang, B.S. Lee, A.C.H. Ling, "Mining partial periodic correlations in time series", In *Journal of Knowledge and Information Systems*, Vol. 15, Issue 1, 2008, pp. 31-54.
 36. Z. Li, J. Wang, J. Han, "Mining Event Periodicity from Incomplete Observations", *Proc. 18th ACM SIGKDD Int'l. Conf. Knowledge Discovery and Data Mining*, 2012, pp. 444-452.
 37. H. Cao, N. Mamoulis, D.W. Cheung, "Discovery of periodic patterns in spatiotemporal sequences", *IEEE Trans. Knowl. Data. Eng.*, 19(4), 2007.
 38. Z. Li, B. Ding, J. Han, R. Kays, P. Nye, "Mining periodic behaviors for Moving Objects", In. *Proc. 16th ACM SIGKDD Int'l conference on Knowledge discovery and data mining*, 2010.
 39. K. Ishida, "Periodic Topic Mining from Massive Amounts of Data", In *Proc. Int'l Conf. on Technologies and Applications of Artificial Intelligence*, 2010.
 40. K.-Y. Huang, C.-H. Chang, "Efficient Mining of Frequent Episodes from Complex Sequences", *Journal of Information Systems*, Volume33, Issue 1, March 2008, pp. 96-114.
 41. T. Jindal, P. Giridhar, L.-A. Tang, J. Li, J. Han, "Spatiotemporal Periodical Pattern Mining in Traffic Data", In *Proc. UrbComp'13*, August, 2013.
 42. N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, D.W. Cheung, "Mining, indexing, and querying historical spatiotemporal data" In *Proc. 10th ACM SIGKDD Int'l conference on Knowledge discovery and data mining*, 2004, pp. 236-245.
 43. M. Lahiri, T.Y. Berger-Wolf, "Periodic Subgraph Mining in Dynamic Networks", *Journal of Knowledge and Information Systems*, vol.24, No. 3, Sep.,2010, pp. 467-497.
 44. F. Rasheed, R. Alhajj, "A Framework for Periodic Outlier Pattern Detection in Time Series Sequences", *IEEE transactions on Cybernetics*, vol.PP, No.99
 45. M. Vlachos, P. S. Yu, and V. Castelli, "On periodicity detection and structural periodic similarity", In *proc. of SIAM Int'l Conference on Data Mining*, 2005
 46. Z. Li, J.Han, B.Ding, and R. Kays, "Mining periodic behaviors of object movements for animal and biological sustainability studies", *Journal of Data Mining and Knowledge Discovery*, 2011.
 47. L. Riotte-Lambert, S. Benhamou, S. Chamaille-Jammes, "Periodicity analysis of movement recursions", *Journal of Theoretical Biology*, 2013, pp. 238-243
 48. P. Huang, C-J Liu, Li Xiao, J. Chen, "Wireless Spectrum Occupancy Prediction Based On Partial Periodic Pattern Mining", *IEEE 20th Int'l symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2012, pp. 51-58.
 49. M. Lv, L. Chen, G. Chen, "Mining user similarity based on routine activities", *Int'l journal of Information Sciences*, vol. 236, pp. 17-32, July 2013.
 50. S. Parthasarathy, S. Mehta, S. Srinivasan, "Robust Periodicity Detection Algorithms", In *Proc. 15th ACM Int'l conference on Information and Knowledge Management*, 2006.
 51. G.N.V.G. Sirisha, M. Shashi, G.V. Padma Raju, "Performance Analysis of Asynchronous Periodic Pattern Mining Algorithms", *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol II Advances in Intelligent Systems and Computing Volume 249*, 2014, pp 87-95.