International Journal of Modern Engineering Research (IJMER)
www.ijmer.com *Pp-47-53* *ISSN: 2249-6645*

Efficient Pattern Mining from Temporal Data through Correlation based Periodicity Detection

Jency Premalatha M.
Assistant Professor, Department of Computer Science and Engineering, Jerusalem College of Engineering, Pallikaranai, Chennai(Dist), Tamilnadu, India

Abstract: *Periodicity Mining captures the evolution of a data value over time. The existing Periodicity Mining Process is Text-Based which can be applied only to text data. The project proposed deals with the Periodic Patterns in Multimedia Data which includes text as well as audio and images. Multimedia data such as digital images and audio can be treated as temporal values, since a timestamp is implicitly attached to every instant of the signal. A Cross Correlation based approach is proposed for periodic mining of multimedia data which has its main application in pattern recognition. In multimedia data mining, when the same signal is compared to phase shifted copies of itself, the procedure is known as* autocorrelation *Basically Cross Correlation is a mathematical tool for finding repeating patterns in periodic signals by analyzing the degree of similarity between them. The periodic pattern retrieved from text data has its application in prediction, forecasting and detection of unusual activities. The patterns extracted from audio and image finds its application in content based retrieval, compression and segmentation.*

Keywords: *Auto-Correlation, Cross Correlation, Periodic Pattern Mining, Time series data.*

I. INTRODUCTION

Due to the increasing computerization in many applications ranging from finance to bioinformatics, vast amounts of data are routinely collected. To unearth useful knowledge from such databases there is need for a different framework. One such framework is provided by Periodicity Mining, a subfield of data mining that deals with the extraction of knowledge or patterns from time series data. A time series is a collection of data values gathered generally at uniform intervals of time to reflect certain behaviour of an attribute or entity. Examples of time series data are meteorological data containing several measurements, e.g., temperature and humidity; stock prices depicted in financial market; power consumption data reported in energy companies; and event logs monitored in computer networks. Periodicity analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of data. Periodic pattern mining or periodicity detection has a number of applications such as prediction, forecasting and detection of unusual activities.

A time series is mostly discretized before it is analyzed. The Data Discretization Technique [15] is used to reduce the number of values for a given continuous attribute by dividing the range of the attributes into intervals. This reduces and simplifies the original data. The time series T may be discretized by considering m distinct ranges such that all values in the same range are represented by one symbol taken from an alphabet set.

Basically, three types of periodic patterns [10] can be detected in a time series: 1) symbol periodicity, 2) sequence periodicity or partial periodic patterns, and 3) segment or full-cycle periodicity. A time series is said to have symbol periodicity if at least one symbol is repeated periodically. For example, in time series T = abd acb aba abc, symbol a is periodic with periodicity p = 3. Similarly, a pattern consisting of more than one symbol may be periodic in a time series; and this leads to partial periodic patterns. For instance, in time series T = bbaa abbd abca abbc abcd, the sequence ab is periodic with periodicity p = 4. Finally, if the whole time series can be mostly represented as a repetition of a pattern or segment, then this type of periodicity is called segment or full-cycle periodicity. For instance, the time series T = abcab abcab abcab has segment periodicity of 5 i.e., T consists of only three occurrences of the segment abcab.

In general, the different types of periodic patterns symbol, sequence and segments take different kinds of occurrences [10]: 1) Perfect 2) Imperfect or Asynchronous. A time series is said to have perfect periodicity, if every next occurrence of the pattern is exactly p positions away from the current occurrence starting from the first occurrence. Imperfect Periodicity of a time series misses the expected occurrence of the perfect periodicity. The patterns may be misaligned due to the intervention of random noise. The misalignment is accepted only upto certain threshold value.

The proposed paper addresses the problem of discovering submerged periodic patterns from Multimedia data such as audio, images and text. Multimedia Mining [15] is the process of arriving at the extraction of useful information from Multimedia data such as the colour, texture, shape, notes and other characteristics. The Periodic Patterns of the text data including numeric and alpha numeric data can be used to predict the fore coming behavior of entities or attributes and arrive at statistical information. It also finds its applicability in anomaly detection or analysis. The periodic pattern of the colour pixels in the image found can be used for applications such as image compression, content based retrieval and segmentation. For this mining, a cross correlation based approach is proposed.

The rest of the paper is organized as follows. Section II describes the previous work on periodicity mining with their limitations that lead to the need for a new approach for multimedia periodic mining. The Problem is defined in Section III. Section IV provides a brief introduction to the proposed approach with the algorithms and examples. In Section V the experimental results are shown. The results are followed by a brief conclusion in section VI.

II. LITERATURE SURVEY

Over the last decade, many interesting techniques of periodicity mining were proposed to detect various types of periodicities namely symbol, sequence and segment periodicity. Few of the techniques are described below with their significance and limitations.

2.1 Apriori based Alogorithm

Apriori [1] is a seminal algorithm that is used to find sequences of frequent itemsets. It is an interesting algorithm for mining imperfect periodicity for a given single period. The algorithm gains its name from the fact that it uses prior knowledge of the frequent itemset properties [15]. Apriori uses an iterative approach known as a level-wise search. First the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item and collecting those items that satisfy the minimum support count. This frequent 1-itemset is used to find frequent 2-itemsets. The benefits of using this algorithm are that it is efficient, scalable and easy to implement. The drawback is that it requires huge number of Candidate Key generation and repeated scans of the database.

2.2 Integrated Technique (Data Cube, Bit-Array and Apriori Mining Technique)

It is often useful to mine segment-wise or point-wise periodicity in which only some of the segments in the time sequence have cyclic behaviour. This integrated technique of data cube, bit-array and apriori mining technique [5] is used to mine all partial and complete periodic patterns in regard to a fixed length period.

Data cube provides an efficient structure and convenient way for interactive mining of multiple-event periodicity. To facilitate periodicity mining, two data cubes namely reference cube and working cube are constructed. A reference cube is built with the time related attribute as its measure with the time and the other non time related attributes as the dimensions. A working cube is constructed from the reference cube which includes the time related attribute of one object and one or more non time related attributes. The time related attribute is encoded using bit-array which includes an existent and a non-existent value. The Apriori technique is then applied on the working cube to mine periodic patterns. The major advantage of this integrated technique is that the data cube is constructed in a single scan of the database.

2.3 Time Warping Algorithm

The periodicity detection algorithms discussed before, do not take into account the presence of noise, which is inevitable in almost every real-world time series either due to unreliable data sources or unreliable transmission. The Time Warping periodicity detection algorithm [6] deals efficiently with all types of noise to produce more accurate results. Based on time warping, the algorithm warps (extends or shrinks) the time axis at various locations to optimally remove the noise. Time warping is computed dynamically and hence is commonly referred to as DTW (Dynamic Time Warping). The time and space complexity of computing the DTW distance is $O(n^2)$ which is considered as a major drawback of this algorithm. An online version of Warping is also available. The key issue to perform the WARP algorithm online is to maintain the DTW matrix over the continuous arrival of data.

2.4 Convolution based Approach

The convolution based approach [9] provides a scalable and computationally efficient algorithm, which discovers symbol and full cycle periodic patterns of obscure periods [14], without affecting the time complexity. Convolution is a mathematical tool for determining the similarity of two input sequences. Convolution can be computed by the Fast Fourier transform (FFT), which reduces the time complexity of the convolution to O(NlogN). The major advantage of this approach is that, not only does it determine the candidate periodic symbols, but it also determines their corresponding periods and locates their corresponding positions. This algorithm scans the time series only once to convert the input sequence into a binary vector according to the proposed mapping and performs the convolution on the binary vector, and analyzes the resulting values to determine the periodic patterns.

The algorithm is well resilient to replacement noise, which can tolerate up to 50 percentage in the data. When the other types of noise namely insertion and deletion gets involved separately or mixed with replacement noise, the algorithm performs poorly. This algorithm performs well for data sets with perfect periodicity, but it fails to perform well when the time series contains misaligned patterns.

2.5 Suffix Tree based Noise Resilient (STNR) Algorithm

STNR [10] is a comprehensive approach capable of analyzing the whole time series or a subsection of it to detect different types of periodic patterns which effectively handles different types of noise. The algorithm uses suffix tree as the underlying data structure, which can detect symbol, sequence (partial), and segment (full cycle) periodicity in time series simultaneously.

Suffix tree is a famous data structure [11], [12], [13] that has been proven to be very useful in string processing. A suffix tree for a string represents all its suffixes, for each suffix of the string there is a distinguished path from the root to a corresponding leaf node in the suffix tree. Each edge is labelled by the string that it represents. Each leaf node holds a number that represents the starting position of the suffix yield when traversing from the root to that leaf. Each intermediate node holds a number which is the length of the substring read, when traversing from the root to that intermediate node. Figure 1 shows a suffix tree for the string abcabbabb$, where $ denotes end marker for the string; it is a unique symbol that does not appear anywhere in the string. The suffix tree data structure used in this STNR allows the design of the algorithm such that its worst case complexity is $O(kn^2)$, where k is the maximum length of periodic pattern and n is the length of the analyzed portion (whole or subsection) of the time series. The algorithm is proven to be successful with replacement, insertion, deletion, or a mixture of these types of noise.

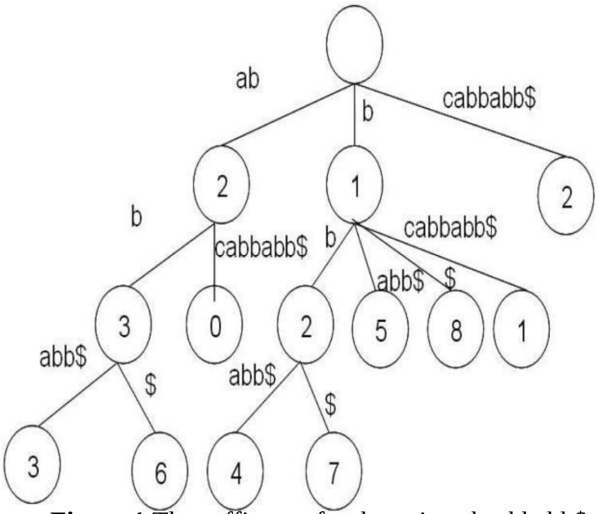

Figure 1 The suffix tree for the string abcabbabb$.

The algorithm requires only a single scan of the data to construct the suffix tree and the suffix tree is traversed only once to find all periodic patterns in the time series.

2.6 Concluding Remarks on Existing System

Different types of periodicities. The ability to handle the imperfectly occurring periodicities is limited to certain techniques at the cost of poor memory management and restricted types of periodicity detection. Only few techniques are resilient to noise, but those techniques possess greater response time. Among the various periodicity mining techniques, the only technique that can be applied to multimedia data represented in the form of digital signals is Convolution. But this technique also has its own limitations. As FFT is used, it is restricted to determine only symbol and segment periodicities. It does not perform well, when the time series contains imperfect patterns. Its resilience is also limited to replacement noise.

III. PERIODICITY DETECTION PROBLEM

Assume that a sequence of n time-stamped feature values is collected in a time series. For a given feature x, let xi be the value of the feature at time-stamp i. The time series of feature x is represented as T = x_1, x_2 ....., x_{n-1} where n is the length of the time series. Let T be discretized into symbols taken from an alphabet set ∑ with enough symbols, i.e., |∑| represents the total number of unique symbols used in the discretization process. In other words, in a systematic way T can be encoded as a string derived from ∑. For instance, the string abbcabcdcbab could represent one time series over the alphabet ∑ = {a, b, c, d}.

The problem is to develop an algorithm for multimedia data, capable of detecting in an encoded time series the Symbol, Sequence, and Segment periodicity that can occur both perfectly and imperfectly even in the presence of noise.

IV. PROPOSED SYSTEM

A cross correlation based approach is proposed for this periodicity detection in Multimedia data. Multimedia data such as digital images and audio signals can be treated as time series values, since a timestamp is implicitly attached to every instant of the signal.

4.1 Preface to Cross Correlation

A **Cross Correlation** based approach is proposed for periodic mining of multimedia data which has its main application in pattern recognition. In multimedia data mining, when the same signal is compared to phase shifted copies of itself, the procedure is known as *autocorrelation* Basically Cross Correlation is a mathematical tool for finding repeating patterns in periodic signals by analyzing the degree of similarity between them.

4.2 Autocorrelation

The existing Periodicity Mining techniques are Text-Based which are more appropriate for numerical and alpha numerical data for deriving periodic patterns. Of the various mining techniques, not all techniques detect all the Autocorrelation, a special case of cross correlation is the mathematical representation of the degree of similarity between a given time series and a lagged version of itself over successive time intervals. It is the same as calculating the correlation between two different time series, except that the same time series is used twice, once in its original form and once lagged one or more time periods. The term can also be referred to as **"Lagged Correlation"** or **"Serial Correlation"**. The correlation result reaches a maximum at the time when the two signals match best. If the two signals are identical, this maximum is reached at t = 0 (no delay). In signal processing, the autocorrelation is often used without the normalization, that is, without subtracting the mean and dividing by the variance. When the autocorrelation function is normalized by mean and variance, it is sometimes referred to as the autocorrelation coefficient. The discrete autocorrelation R at lag j for a discrete signal x_n is

$$R_{xx}(j) = \sum_n x_n \, \overline{x}_{n-j}.$$

It can be simply stated as the as the sum of the product of the attribute value with the attribute values to the right.

4.2 Discretization by Equal Width Interval Binning

Discretization techniques can be used to reduce the number of values for a given continuous attribute, by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values. During this pre processing process, the alphanumeric values are converted to their corresponding ASCII values. No changes are made to the numeric data. In case of images, the colour ones are converted to their grey scale equivalent and the pre processing is done based on the intensity values. For audio the discretization process is based on the frequency.

The discretization method used is Equal Width Interval Binning. The algorithm needs to first sort the attributes according to their values, and then find the minimum value, x_{min}, and the maximum value, x_{max} of that attribute. Interval width, w, is then computed by,

$$\mathbf{w = (X_{max} - X_{min})/k} \qquad (2)$$

where k is the user-defined parameter as the total number of intervals needed. The interval boundaries are specified as $x_{min} + w_i$, where i = 1,...,k-1.

Example: Consider the sequence of events X = {65 86 74 79 85} which has to be discretized for an interval range of 3. The interval width based on the above formula will yield a value of 7. By Equal Width Interval Binning method, the values from 65 to 72 will be assigned the label 'a', and the values from 73 to 79 will be assigned the label 'b' and the values from 80 to 86 will be assigned the label 'c'. The sequences of events are thus discretized as X = {a c b b c}

4.4 Algorithm for Detecting Symbol Periodic Pattern

The algorithm for finding the Symbol pattern takes the discretizwd time series as input and retrieves the patterns.

Overview: The discretized input sequence is converted into a matrix in which every row corresponds to a specific symbol. Auto Correlation is applied on every row of the matrix separately. The non zero elements of the correlated output indicates the number of occurrences of that symbol starting from that position. From this output, the index positions and periodic rates are derived.

Input: A discretized time series sequence T = x_1, x_2 ....., x_{n-1}, of length n.

Output: Symbol Periodic Patterns, Number of Occurrences,

Perfect and Imperfect Periodic Rates and the Index

Positions.

Algorithm:

1) For the input sequence T, build a m*n matrix M of binary values, in which every row corresponds to a particular symbol, where m is the Interval range specified by user and n is the size of the input sequence. The existence of a symbol is denoted by '1' and the non existence by '0'.
2) Apply auto correlation to every row of the matrix M separately to find the correlation for every symbol using the formula,

(3)
$$R_{xx}(j) = \sum_n x_n \, \overline{x}_{n-j}.$$

where R is the discrete autocorrelation, j is the lag for a discrete signal $x_{n.}$

3) Every non-zero element of the resulted sequence represents the total number of occurrences of that symbol from that position. The first non-zero element represents the total number of occurrence of that symbol.
4) The symbol that exceeds the minimum threshold percentage of occurrence is considered as a frequent symbol pattern.
5) The Index positions of the non-zero elements represent the starting position of the symbol pattern.
6) The Periodic Rates are derived from the index positions ($PR_i = P_i - P_{i-1}$).
   i) If the periodic rate of the symbol is the same in a minimum threshold percentage, it is considered as perfect periodic rate.

ii) If the periodic rate does not satisfy the above condition, it is considered as imperfect periodic rate.

Example:

Consider a discretized sequence {a b a b a} for an interval range of 2. The 2 * 5 Matrix M produced for the input sequence is given as follows,

$$M \Rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

In this matrix, the first row represents symbol 'a' and the second row represents symbol 'b'. The application of autocorrelation on each of the rows separately will produce the below result,

$$R \Rightarrow \begin{bmatrix} 3 & 0 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

In the correlated output R, every non-zero element represents the total number of occurrences of the symbol starting from that position. In that, the first element represents the total number of occurrences of the symbol. In this example, the output 3 in the first row represents the total number of occurrence of symbol 'a' and 2 in the next row represents the total number of occurrence of symbol 'b'. The index positions of the non-zero elements are derived from the matrix. From that index position, the perfect and imperfect periodic rates are computed. In this example symbols 'a' and 'b' has occurred with a perfect periodic rate of 1.

4.5 Algorithm for detecting Sequence and Segment Periodic Patterns

The algorithm for finding the sequence and segment periodic patterns also takes the discretized time series together with the matrix.

Overview: The non zero elements of the Binary Matrix is auto correlated with the adjacent element of every other row until a zero value or end of the series is reached. The resulted sequence is searched in the rest of the series. If the entire time series can be represented as a repetition of the same sequence, then it is declared as a segment pattern. From this output, the number of occurrences, the index positions and periodic rates are derived.

Input: A discretized time series sequence T = x_1, x_2 ....., x_{n-1}, of length n and the Binary Matrix M with the symbols that are not eligible for candidate patterns removed based on the execution of the above algorithm..

Output: Sequence and Segment Periodic Patterns, Number of Occurrences, Perfect and Imperfect Periodic Rates and the Index Positions.

Algorithm:

1) From the matrix M, remove the rows corresponding to the symbols that are not frequent.
2) Every non-zero element of the row is auto correlated with the adjacent element of every other row until a zero value or end of the series is reached. The formula used is as follows,

$$R_{xx}(j) = \sum_n x_n \overline{x}_{n-j}. \quad (4)$$

where R is the discrete autocorrelation, j is the lag for a discrete signal x_n.

3) The resulted sequence is searched in the rest of the series.
   i) If found, it is declared as a valid sequence if it exceeds the minimum threshold percentage.
   ii) If not found, the sequence is shrink and searched in the sequence until a two bit sequence is reached.
   iii) If the entire time series can be represented as a repetition of the same sequence, then it is declared as a segment pattern.
4) The Index position of the sequence represents the starting position of the sequence pattern.
5) The Periodic Rates are derived from the index positions ($PR_i = P_i - P_{i-1}$).
   i) If the periodic rate of the symbol is the same in a minimum threshold percentage, it is considered as perfect periodic rate.
   ii) If the periodic rate does not satisfy the above condition, it is considered as imperfect periodic rate.

Example:

Consider the Binary Matrix M in which the rows correspond to 'a', 'b' and 'c' respectively,

$$M \Rightarrow \begin{bmatrix} 1&0&0&0&1&0&0&0&0&0&1&0&0&0 \\ 0&1&0&1&0&0&1&0&1&1&0&1&0&1 \\ 0&0&1&0&0&0&0&1&0&0&0&0&1&0 \end{bmatrix}$$

The application of the above algorithm will produce a sequence of 'abcb'. The Index positions of the sequence are observed to be 1, 6 and 11. From this it is concluded that the sequence has occurred with a perfect periodic rate of 5. Based on the algorithm, initially the sequence 'abcb' is found, since only zeros are present in the fifth column. This sequence is searched in the rest of the time series. As this sequence is present in two more positions of the time series, it is considered as a valid sequence.

V. RESULT

This section consists of an extensive experimental study of the adopted approach for periodic pattern mining. The results are examined in terms of the various performance parameters namely the time performance, accuracy and their resilience to noise.

5.1 Time Performance

The time complexity of the Auto Correlation approach for pattern mining is in the order of $O(n^2)$.

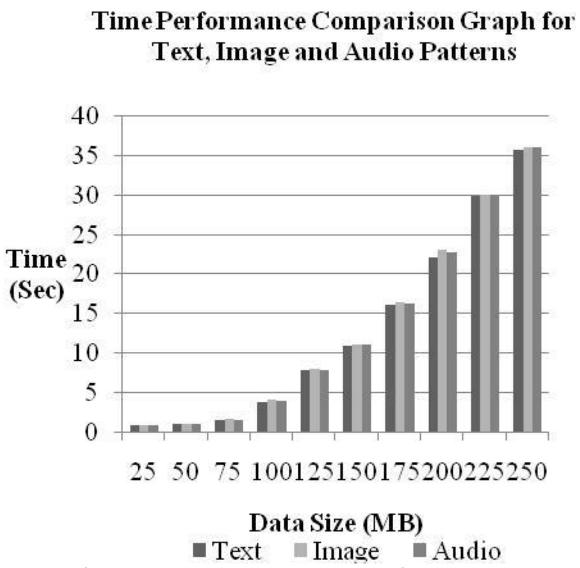

Figure 2 Time Performance Comparison Graph for Text, Image and Audio Patterns

The algorithm is examined to be efficient as it has the capability to detect all different kinds of periodicities (Symbol, Sequence & Segment) at their different occurrences (perfect & imperfect) within this time complexity. The time complexity is approximately the same for all different kinds of data namely the text, images and audio. The algorithm poses some scalability issues in terms of performance degradation, when applied to too huge volume of data, due to the comparison of every element with every other element. Figure 2 shows the time performance comparison for the different kinds of multimedia data

5.2 Accuracy

The accuracy measure is the ability of the algorithm to detect the different periodicities that are embedded in the time series. The parameters that determine accuracy are data distribution, alphabet size (number of unique symbols in the data), size of the data (number of symbols in the data), period size, and the type and amount of noise in the data. For an inerrant time series sequence (perfectly periodic with 0 percent noise), the correlation algorithm for both Symbol and Sequence or Segment can find all periodic patterns with 100 percent confidence regardless of the data distribution, alphabet size, period size, and data size. This is an instant benefit of using auto correlation which guarantees identifying all repeating patterns.

5.2 Noise Resilience

The completeness of the algorithm is not affected by the presence of noise in the time series. The Auto Correlation algorithm is resilient to Insertion, Deletion and Replacement noise. The algorithm is made resilient to noise by first determining the candidate patterns rather than the periodic rates of the patterns. However, if the noise ratio is more in accordance with the length of the time series, then the algorithm may report more number of imperfect or asynchronous periodic patterns instead of perfect periodic patterns.

VI. CONCLUSION

Analysing huge volume of time series data streams to unearth any hidden regularities is important in many applications ranging from finance to manufacturing processes to bioinformatics due to its interestingness and usefulness measure. To make it more significant and interesting, these periodicity mining technique approach has been extended to mine multimedia data which includes text, audio and images. In this paper, we have introduced a new periodicity mining technique based on correlation to recognize the hidden periodic patterns from multimedia data. The algorithm reports the Patterns together with the Perfect and Imperfect Periodic Rates, their Index Positions and the total number of occurrences of each pattern. The scalability issues of the algorithm can be dealt by performing Fast Fourier Transform for the auto correlation calculation, which can also bring down the time complexity to O(NlogN). The algorithm can also be extended to mine association rules from the patterns derived.

References

[1] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc. 11th Int'l Conf. Data Eng., Mar. 1995.

[2] J. Han, G. Dong, and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Databases," Proc. 15th Int'l Conf. Data Eng., Mar. 1999.

[3] C. Berberidis, W. Aref, M. Atallah, I. Vlahavas, and A.Elmagarmid, "Multiple and Partial Periodicity Mining in Time Series Databases," Proc. European Conf. Artificial Intelligence, July 2002.

[4] S. Ma and J. Hellerstein, "Mining Partially Periodic Event Patterns with Unknown Periods," Proc. 17th IEEE Int'l Conf. Data Eng., Apr. 2001.

[5] J. Han, W. Gong, and 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.

[6] M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid, "WARP: Time Warping for Periodicity Detection," Proc. Fifth IEEE Int'l Conf. Data Mining, Nov. 2005.

[7] Jiawei Han and Micheline Kamber, University of Illinois at Urbana-Champaign, "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, 2006.

[8] J. Yang, W. Wang, and P. Yu, "Mining Asynchronous Periodic Patterns in Time Series Data," Proc. Sixth Int'l Conf. Knowledge Discovery and Data Mining, Aug. 2000.

[9] K.-Y. Huang and C.-H. Chang, "SMCA: A General Model for Mining Asynchronous Periodic Patterns in Temporal Databases,"IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 774-785, June 2005.

[10] M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid, "Periodicity Detection in Time Series Databases," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 7, pp. 875-887, July 2005.

[11] F. Rasheed and R. Alhajj, "STNR: A Suffix Tree Based Noise Resilient Algorithm for Periodicity Detection in Time Series Databases," Applied Intelligence, vol. 32, no. 3, pp. 267-278, 2010.

[12] R. Grossi and G.F. Italiano, "Suffix Trees and Their Applications in String Algorithms," Proc. South Am. Workshop String Processing, pp. 57-76, Sept. 1993.

[13] M. Dubiner et al., "Faster Tree Pattern Matching," J. ACM, vol. 14, pp. 205-213, 1994.

[14] R. Kolpakov and G. Kucherov, "Finding Maximal Repetitions in a Word in Linear Time," Proc. Ann. Symp. Foundations of Computer Science, pp. 596-604, 1999.

[15] M. Elfeky, W. Aref, and A. Elmagarmid, "Using Convolution to Mine Obscure Periodic Patterns in One Pass," Proc. Ninth Int'l Conf. Extending Data Base Technology, Mar. 2004.