ISSN 2250-3153

Weather Report Calculation using Suffix Trees

K.Lavanya[1], R.Buvaneshwari[2], S.P.Ramya[3]

[1,2] PG Student, Department of Information Technology, Periyar Maniammai University
[3]Asst.Professor, Department of Information Technology, Periyar Maniammai University
Vallam, Thanjavur, India

[1]lavanya62.cse@gmail.com
[2]buvi.sumathi@gmail.com
[3]way2ramya@gmail.com

Abstract- A time series is a collection of data values are gathered.Periodic pattern mining or periodicity detection has a number of applications, such as prediction, forecasting, detection of unusual activities, etc. The problem is not trivial because the data to be analyzed are mostly noisy and different periodicity types (namely symbol, sequence, and segment) are to be investigated. The whole time series or in a subsection of it to effectively handle different types of noise (to a certain degree) and at the same time is able to detect different types of periodic pattern. This can detect symbol, sequence (partial), and segment (full cycle) periodicity in time series. The algorithm uses suffix tree as the underlying data structure; symbol means particular area in time series, sequence means particular time period and segment it is used half day temperature .In this paper we are using noise resilient and which can be used two nodes. So we are using sensor network, it works on heat temperature measuring .The temperature is varied from normal condition some incident occur in this place and we have to send an alert message to people.

Index Terms- time series, periodicity detection, suffix tree, symbol periodicity, segment periodicity, sequence periodicity noise resilient, sensor networks.

I. INTRODUCTION

A time series is a collection of data values gathered. Generally at uniform interval of time to reflect certain behavior of an entity. Real life has several examples of time series such as weather conditions of a particular location, spending patterns, stock growth, transactions in a superstore, network delays, power consumption, computer network fault analysis and security breach detection, earthquake prediction, gene expression data analysis [1],[10], etc. A time series is mostly characterized by being composed of repeating cycles. For instance, there is traffic jam twice day when the schools are open; number of transactions in a superstore is high at certain periods during the day, certain days during the week, and so on . In other words, periodicity detection is a process for finding temporal regularities within the time series, and the goal of analyzing a time series is to find whether and how frequent a periodic pattern (full or partial) is repeated within the series . Let T =e0; e1; e2; . . . ; en_1 be a time series having n events, where ei represents the event recorded at time instance i; time series T maybe discredited by considering m distinct ranges such that all values in the same range are represented denoted by ∑.In general, three types of periodic patterns 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 =abd acb aba abc, symbol a is periodic with periodicity p =3, starting at position zero (stP os = 0). Similarly, a pattern consisting of more than one symbol maybe periodic in a time series; and this leads to partial periodic patterns. 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 (p = 5) starting at the first position (stP os = 0), i.e., T consists of only three occurrences of the segment abcab. 1) Identifying the three different types of periodic patterns, 2) handling asynchronous periodicity by locating periodic patterns that may drift from their expected positions up to an allowable limit, and 3) investigating periodic patterns in the whole time series as well as in a subsection of the time series insertion, deletion, or any mixture of these types of noise [24].

It can also detect periodicity within a subsection of a time series and applies various redundant period pruning techniques to output a small number of useful periods by removing most of the redundant periods. The algorithm looks for all periods starting from all positions which have confidence greater than or equal to the user-provided periodicity threshold. Finally, the different aspects of the algorithm, its applicability, and effectiveness have been demonstrated using both real and synthetic data.

1. The development of suffix-tree-based comprehensive algorithm that can simultaneously detect the symbol, sequence, and segment periodicity;
2. Finding periodicity within subsection of the series;
3. Identifying and reporting only useful and non redundant periods by applying pruning techniques to eliminate redundant periods.
4. Detailed algorithm analysis for time performance and space consumption by considering three cases, namely the worst case, the average case, and the best case;
5. A number of optimization strategies are presented; they do improve the running time as demonstrated in the related experimental results, although they do not improve the time complexity;

II. RELATED WORK

The first category includes algorithms that require the user to specify the period, and then look only for patterns occurring with that period. The second classes, on the other hand, are algorithms which look for all possible periods in the time series. In this paper could be classified under the second category; it does more than the other algorithms by looking for all possible periods starting from all possible positions within a pre specified range, whether the whole time series or a subsection of the time series. User to provide the expected period value which is used to check the time series for corresponding periodic patterns. For example, in power consumption time series, a user may test for weekly, biweekly, or monthly periods. However, it is usually difficult to provide expected period values; and this approach prohibits finding unexpected periods, which might be more useful than the expected ones.

- In this paper, we concentrate on the first case; we need to develop an algorithm capable of detecting in an encoded time series (even in the presence of noise) symbol, sequence, and segment periodicity, which are formally defined next. We start by defining confidence because it is not always possible to achieve perfect periodicity and hence we need to specify the degree of confidence in the reported result.
- **Definition 1 (Perfect Periodicity):** Consider a time series T, a pattern X is said to satisfy perfect periodicity in T with period p if starting from the first occurrence of X until the end of T every next occurrence of X exists p positions away from the current occurrence of X. It is possible to have some of the expected occurrences of X missing and this leads to imperfect periodicity.
- **Definition 2 (Confidence):** The confidence of a periodic pattern X occurring in time series T is the ratio of its actual periodicity to its expected perfect periodicity. Formally, the confidence of pattern X with periodicity p starting at Position.
- Con f(p,stops,x) = Actualperiodicity(p,stPos,x) / Perfect Periodicity (p, stops,x)
- **Definition 3 (Symbol Periodicity):** A time series T is said to have symbol periodicity for a given symbol s with period p and starting position stP os if the periodicity of s in T is either perfect or imperfect with high confidence.
- **Definition 4 (Sequence Periodicity):** A time series T is said to have sequence periodicity (or partial periodicity) for a pattern(partial periodic pattern) X starting at position stP os, if $|X| \geq 1$and the periodicity of X in T is either perfect or imperfect with high confidence.
- **Definition 5 (Segment Periodicity):** A time series T of length n is said to have segment periodicity for a pattern X with period p and starting position stP os if p = |X| an the periodicity of X in T is either perfect or imperfect with high confidence.

A. *Periodicity Detection*

In the first phase, we build the suffix tree for the time series and In the second phase, we use the suffix tree to calculate the periodicity of various patterns in the time series. One important aspect of our algorithm is redundant period pruning.

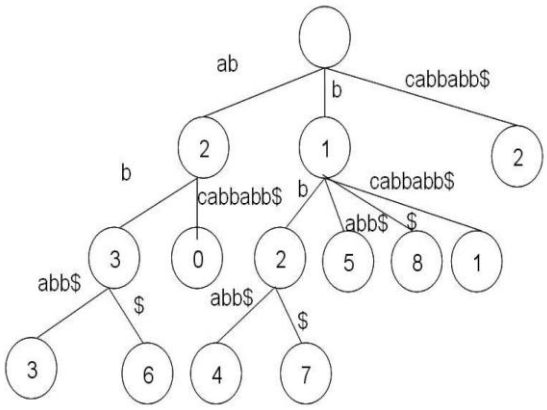


Fig. 1: The suffix tree for the string abcabbabb$.

The benefit of redundant period pruning, the algorithm does not waste time to investigate a period which has already been identified as redundant.

B. *First Phase—Suffix-Tree-Based Representation*

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. The path from the root to any leaf represents a suffix for the string. Since a string of length n can have exactly n suffixes, the suffix tree for a string also contains exactly n leaves..

C. *Periodicity Detection in Presence of Noise*

Three types of noise generally considered in time series data are replacement, insertion, and deletion noise. In replacement noise, some symbols in the discredited time series are replaced at random with other symbols. In case of insertion and deletion noise, some symbols are inserted or deleted, respectively, randomly at different positions (or time values). Noise can also be a mixture of these three types; for instance, RI type noise means the uniform mixture of replacement (R) and insertion (I) noise.

- The distance between the current occurrence and the current reference starting position;
- B represents the number of periodic values that must be passed from the current reference starting position to reach the current occurrence.
- Variable C represents the distance between the current occurrence and the expected occurrence.

Algorithm 2. Noise Resilient Periodicity Detection Algorithm
Input: a time series of size n and time tolerance value tt;
Output : positions of periodic patterns;

1.	for each occurrence vector occur_vec of size k for pattern, repeat
1.1	for j=0;j<n/2;j++
1.1.1	p=occur _vec[j+1]-occur_vec[j];
1.1.2	StPos=Occur_vec[j];endPos=occur_vec[k];
1.1.3	for i=j;i<k;i++;
1.1.3.1	A=occur_vec[i]-currStpos;
1.1.3.2	B=Round(A/p);
1.1.3.3	C=A-(p*B);
1.1.3.4	If((-tt≤ C≤+tt)AND (Round((preOccur-currSTPos)p)!=B))
1.1.3.4.1	currStPos=Occur_vec[i];
1.1.3.4.2	preoccur=occur _vec[i];

1.1.3.4.3 increment count(p);
1.1.3.4.4 sumPer+=(p+C);
1.1.4 end for
1.1.5 mean p=SumPer-(p/(count(p)-1));
1.1.6 conf(p)=count(p)/(perfect_periodicity(p,StPos, x)
1.1.7 if(conf(p)≥threshold)add p to the period list;
1.2 end for
2. end for
End Algorithm

Algorithm 2 contains three new variables, namely A, B, and C;

Periodicity Detection in a Subsection of Time Series

The periodicity detection algorithm calculates all patterns which are periodic starting from any position less than half the length of the time series $(stPos < \frac{n}{2})$ and continues till the end of the time series or till the last occurrence of the pattern pattern to be part of the same periodic section. Consequently, having the distance between two occurrences more than dmax may potentially mark the end of one periodic section and/or the start of new periodic section for the same pattern.

Redundant period pruning techniques

Periodicity detection algorithms generally do not prune or prohibit the calculation of redundant periods; the immediate drawback is reporting a huge number of periods, which meaningful periodic patterns within the large pool of reported periods.

Optimization Strategies Of The Algorithm

There are some optimization strategies that we selected for the efficient implementation of the algorithm. These strategies, though simple, have improved the algorithm efficiency significantly. We do not include these into the algorithm pseudo code so as to keep it simple and more understandable. Some of these strategies are briefly mentioned in the text below.

1. Recall that each edge connecting parent node v to child node u has its own occurrence vector that contains values from leaf nodes present in the sub tree rooted at u. Accordingly, edges are sorted based on the number of values they have to carry in their occurrence vectors; and edges that qualify to be processed in each step of the algorithm are visited in descending order based on the size of their occurrence vectors.
2. The occurrence vector for each intermediate edge as that would mostly result in huge number of redundant sub vectors. Rather, a single list of values is maintained and each intermediate edge keeps the starting and ending index positions of its occurrence list, and sorts only its concerned portion of the list. As the sorting might also require a huge number of shifting of elements, we maintain the globally unified occurrence vector as a linked list of integers so that the insertion and deletion of values do not disturb large part of the list.
3. Periodicity detection at the first level (for edges directly connected to the leaves) is generally avoided because experiments have shown that in most cases, the first level does not add any new period. This is based on the observation that in time series periodic patterns mostly consist of more than one symbol.
4. Intermediate edges (directly or indirectly) connected to significantly small number of leaves can also be ignored. For example, if the subtree rooted at the child node connected to an edge contains less than 1 percent leaves, there is less chance to find a significant periodic pattern there.
5. Very small periods (say less than five symbols) may also be ignored. Periods which are larger than 30 percent (or 50 percent) of the series length are also ignored so that the infrequent patterns do not pollute the output. Sequences which are smaller than or equal to half the length of the series.
6. Similarly, periods smaller than edge value (the length of the sequence so far) are ignored..
7. The collection of periods is maintained with two levels indexing; separate index is maintained on period values and starting positions. This facilitates fast and efficient search of periods because we check the existing collection of periods a number of times.

Algorithm Analysis

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

III. PROPOSED SYSTEM

A. Sensor Network

i. Procedure

Procedure is a description of how a sensor works, i.e., how a certain type of stimuli is transformed to a digital representation, perhaps a description of the scientific method behind the sensor. Consequently, sensors can be thought of as implementations of sensing methods where different methods can be used to derive information about the same type of observed property. Sensing methods can also be used to describe how observations where made: e.g., how a sensor was positioned and used. Simplifying, one can think of sensing as recipes for observing.

It is a collection of specialized transducers with a communications infrastructure planned to check and record conditions at various locations. Normally verify issues are direction of wind, pollutant levels, and illumination intensity. In addition to one or more sensors, every node in a sensor network is typically prepared with a radio transceiver, small microcontroller or other wireless transmission device, and battery as an energy source. The cost of sensor nodes is also variable, ranging from thousands of dollars to a small number of pennies, depending on the actual size of the network and the complexity required of individual sensor nodes as well as their applications. Size and cost constraints on sensor nodes result in equivalent constraints on resources such as energy, computational speed, memory and bandwidth.

This network consists of many detection stations called sensor nodes, each of which is small, lightweight and moveable. Each sensor node is set in a transceiver, transducers, and source of power. They generates electrical signals basis on sensed physical effects and phenomena. The microcomputers are manages and keep the sensor output. The transceiver may be hard wired or wireless. They accept commands from a central computer and transmitting data to that system. The power for every sensor node is derived from the electric utility or from a battery.

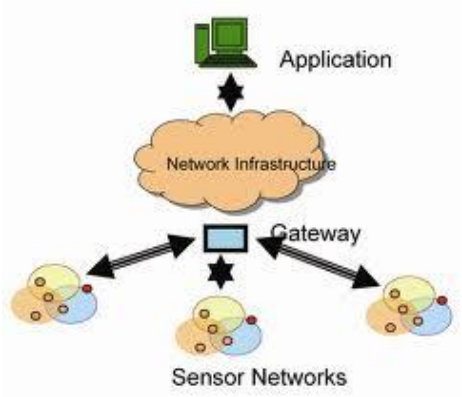
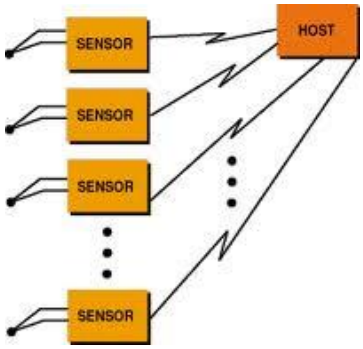


Fig. 1: Sensor Networks

This type of network is a worldwide exchange for sensor information. It permits researchers and scientists to split data with authorized partners. Authorized client can manages the sensor installations, visualize collected data and divide up it with trusted partners. This sensor networks are currently used in many resident application areas, such as environment and traffic control, healthcare applications, home automation. This network generally creates a wireless ad-hoc network, means that every sensor supports a multi-hop routing algorithm where nodes function as forwarders, transmitting data packets to a base station.



A landslide finding system, make use of this type of network to find out the small movements of soil that may happen during a landslide. And through the data collection, it is easy to know the occurrence of landslides long before it really happens

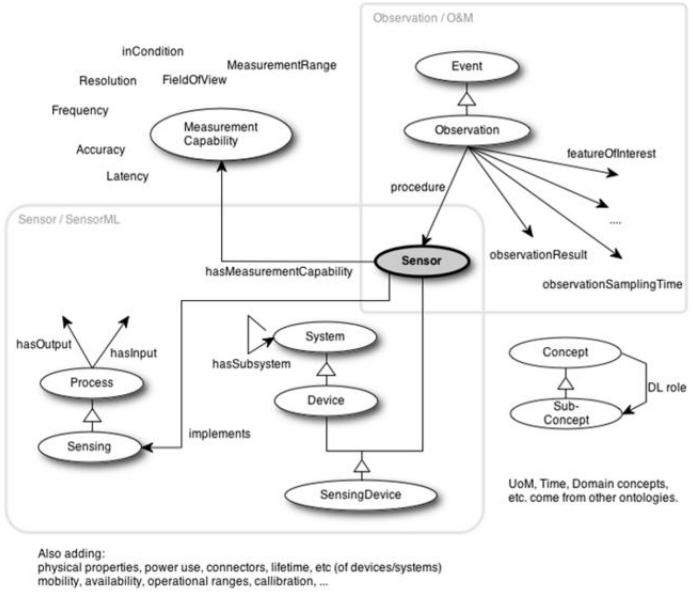


Fig. 2: Sensor network view

Sensors are physical objects that perform observations, i.e., they transform an incoming stimulus into another, often digital, representation. Sensors are not restricted to technical devices but also include humans as observers. A clear distinction needs to be drawn between sensors as objects and the process of sensing. We assume that objects are sensors while they perform sensing, i.e., while they are deployed. Furthermore, we also distinguish between the sensor and a procedure, i.e., a description, which defines how a sensor should be realized and deployed to measure a certain observable property. Similarly, to the capabilities of particular stimuli, sensors can only operate in certain conditions. These characteristics are modelled as observable properties of the sensors and include their survival range or accuracy of measurement under defined external conditions. Finally, sensors can be combined to sensor systems and networks. Many sensors need to keep track of time and location to produce meaningful results and, hence, are combined with further sensors to sensor systems such as weather stations.

ii. Observations

Observations act as the nexus between incoming stimuli, the sensor, and the output of the sensor, i.e., a symbol representing a region in a dimensional space. Therefore, we regard observations as social, not physical, objects. Observations can also fix other parameters such as time and location. These can be specified as parts of observation procedure. The same sensor can be positioned in different ways and, hence, collect data about different properties. In many cases, sensors perform additional processing steps or produce single results based on a series of incoming stimuli. Therefore, observations are rather contexts for the interpretation of the incoming stimuli than physical events.

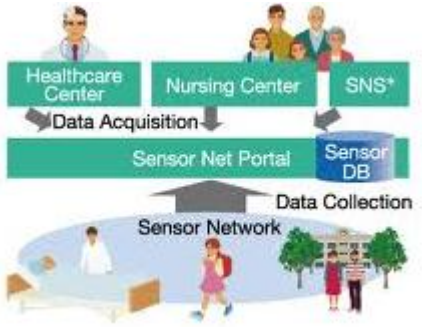
iii. Observed Properties

Properties are qualities that can be observed via stimuli by a certain type of sensors. They inhere in features of interest and do not exist independently. While this does not imply that they do not exist without observations, our domain is restricted to those observations for which sensors can be implemented based on certain procedures and stimuli. To minimize the amount of

ontological commitments related to the existence of entities in the physical world, observed properties are the only connection between stimuli, sensors, and observations on the one hand, and features of interests on the other hand.

iv. Application of Sensor Networks

- Traffic monitoring
- Air Traffic control
- Video surveillance
- Industrial automation
- Medical device monitoring
- Monitoring of weather conditions



B. DNA Sequence Mining

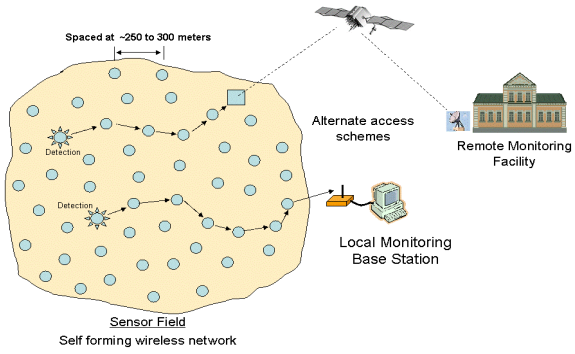


Fig. 3: DNA Sequence Mining

i. Accuracy

The first sets of tests are dedicated to demonstrate the Completeness of STNR in the sense that it should be able to find a period once it exists in the time series. We test how STNR satisfies this on both synthetic and real data.

ii. Synthetic Data

The synthetic data have been generated in the same way as done .The parameters controlled during data generation are data distribution (uniform or normal), 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.

iii. Real Data

For real data experiments, we have already used the Walmart data in testing a previous version of the algorithm and the results reported in demonstrated the efficiency of the algorithm for the general case where the period lasts till the end of the time series;

C. Wireless Sensor Technology

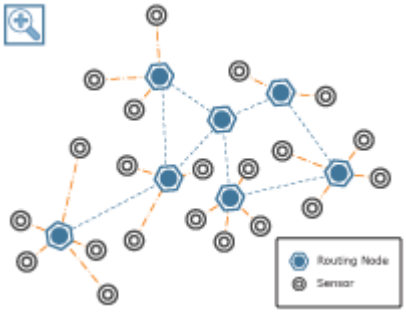


Fig. 4: Wireless Sensor Technology.

Wireless sensor networks are constituted from clusters of devices using sensor technologies deployed over a specific area, wirelessly communicating data to a central system. Sensor networks continually monitor physical properties, processes, chemical or magnetic properties, using viable and emerging communication infrastructures. With a range of software which enhances business intelligence through data extraction and mining, sensor networks will allow an entire class of systems to be designed and deployed with breakthrough results in diverse marketplaces.

Wireless sensor networks rely on emerging technologies such as communication technologies (RF communication, ad hoc networking routing), semiconductor technologies (MEMS CMOS microprocessor), embedded systems and micro sensor technologies.

Wireless sensor networks possess the potential to revolutionize business in a similar way to the emergence of the internet by providing a large number of users with various forms of information. In fact, sensor networking enjoys an enormous application potential in various fields, including:

- Environmental and healthcare: sensing ocean temperature, gathering information about a patient's condition
- Critical industrial areas: monitoring oil containers, verifying chemical gas substance concentration
- Warehouse and supply chain: monitoring currents states and history of goods with critical conservation conditions
- Military: surveillance and reconnaissance

A wireless sensor network consists of a large number of tiny sensor nodes, each of which is equipped with a radio transceiver, a small microprocessor and a number of sensors. These nodes are able to autonomously form a network through which sensor readings can be propagated. Since the sensor nodes have some intelligence, data can be processed as it flows through the network.

Given the hardware limitations and physical environment in which the nodes must operate, along with application-level requirements, the algorithms and protocols must be designed to provide a robust and energy-efficient communications mechanism. Design of physical-layer methods such as modulation, and source and channel coding also fall in this

category. Channel access methods must be devised, and routing issues and mobility management solved.

IV. TIME PERFORMANCE

This section reports the results of the experiments that measure the time performance of STNR compared to Par Per, CONV, and WARP. We test the time behavior of the compared algorithms by considering three perspectives: varying data size, period size, and noise ratio.

Varying Data size

The periodic patterns for a specific period. The synthetic data used in the testing have been generated by following uniform distribution with alphabet size of 10 an embedded period value of 32. This is because Par Per(and other similar algorithm(s)) have been designed to find only patterns with a specified periodicity while STNR is general and finds the periodicity for all the patterns which are periodic for any period value starting and ending anywhere in the time series. Par Per only finds partial periodic patterns while STNR can find all the three types of periodic patterns in the data, namely symbol, sequence, and segment periodicity.

Varying Period Size

This set of experiments is intended to show the behavior Of STNR by varying the embedded period size. For this experiment, we fixed the time series length and the number of alphabets in the series and vary the embedded period size.

Varying Noise Ratio

The next set of experiments measure the impact of noise ratio on the time performance of STNR. For this experiment, we fixed the time series length, period size, alphabet size, and data distribution and measured the impact of varying noise ratio on time performance of the algorithm. We tested two sets of data; one contains replacement noise and the other contains insertion noise.

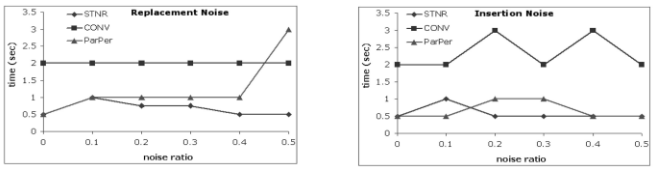


Fig. 5: Time performance of the algorithm by varying noise ratio.

Which carry less than 1 percent leaf nodes (or patterns which appear rarely), mostly noise is caught into these infrequent patterns and does not affect the time performance of STNR by reasonable margin.

Effect of Data Distribution

The data distribution seems to take less amount of time compared to the normal distribution. Since the shape of the suffix tree depends on the data distribution, it is very understandable that the different data distributions would take different amounts of time.

Optimization Strategies

we will present three experiments to show the effect of the employed optimization techniques; we mainly test for: 1) sorted and unsorted occurrence vectors of edges in terms of number of values they carry 2) calculating the periodicity and executing STNR by including/excluding the edges at the first level of the suffix tree ignoring or considering the edges whose occurrence vectors carry fewer values.

Tree Traversal Guided by Sorted Edges

The tree traversal is guided by the number of leaves a nodes (or edge) carry. The edge which leads to more leaves (i.e., the sub tree rooted at the immediate child following the edge has more leaves) is traversed first and the edge which leads to fewer number of leaves is traversed later.

Periodicity Detection at First Level

Periodicity detection can be avoided for occurrence vectors at the first level of the suffix tree for many data sets because usually the patterns at first level are subsets of the larger patterns. But this depends heavily on the data set and the periodic pattern size (or length).

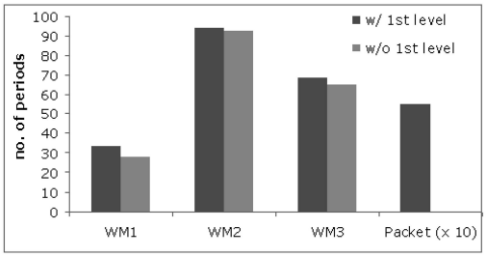


Fig - Periodicity detection at first level

Table 5 Periodic Pattern Found in P14593

Per	stpos	StPosMod	EndPos	confidence	pattern	repetitions
4	106	2	162	0.33	d	5
4	108	0	369	1	gn	66
4	115	3	369	1	agn	64
4	122	2	369	0.94	aagn	58

For the second experiment, we applied STNR on the protein sequence P14593 which is the code for circus porosities protein. It is the immune dominant surface antigen on the sporozoite. The sequence of this protein has interesting Both protein sequences studied are antigen proteins which are used for immune responses. The repeating sequence is used to allow the antibodies to detect the antigens. To further analyze the antigen protein sequences, all antigen protein sequences are analyzed to discover similar patterns among difference sequences.

V. CONCLUSION

The periodic detection can find symbol, sequence (partial periodic), and segment (full cycle) periodicity in the time series. It can also find the periodicity within a subsection of the time series. We performed several experiments to show the time behavior, accuracy, and noise resilience characteristics of the data. The reported results demonstrated the power of the employed pruning strategies. Currently, we are working on online periodicity detection where the suffix tree is constructed online, i.e., extended while the algorithm is in operation. This type of stream mining has a large number of applications

especially in sensor networks, ubiquitous computing, and DNA sequence mining. We are also implementing the disk-based solution of the suffix tree to extend capabilities beyond the virtual memory into the available disk space.

REFERENCES

[1] M. Ahdesma¨ki, H. La¨hdesma¨ki, R. Pearson, H. Huttunen, and O.Yli-Harja, "Robust Detection of Periodic Time Series Measured from Biological Systems," BMC Bioinformatics, vol. 6, no. 117, 2005.

[2] C. Berberidis, W. Aref, M. Atallah, I. Vlahavas, and A.Elmagarmid, "Multiple and Partial Periodicity Mining in TimeSeries Databases," Proc. European Conf. Artificial Intelligence, July2002.

[3] H. Brown et al., "Sequence Variation in S-Antigen Genes of Plasmodium falciparum," Molecular Biology and Medicine, vol. 4, no. 6, pp. 365-376, Dec. 1987.

[4] C.-F. Cheung, J.X. Yu, and H. Lu, "Constructing Suffix Tree for Gigabyte Sequences with Megabyte Memory," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 1, pp. 90-105, Jan. 2005.

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

[6] 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.

J. Fayolle and M.D. Ward, "Analysis of the Average Depth in a Suffix Tree under a Markov Model," Proc. Int'l Conf. Analysis of Algorithms, pp. 95-104, 2005.

[7] E.F. Glynn, J. Chen, and A.R. Mushegian, "Detecting Periodic Patterns in Unevenly Spaced Gene Expression Time Series Using Lomb-Scargle Periodograms," Bioinformatics, vol. 22, no. 3 pp. 310-316, Feb. 2006.

[8] J. Han, Y. Yin, and G. Dong, "Efficient Mining of Partial Periodic Patterns in Time Series Database," Proc. 15th IEEE Int'l Conf. DataEng., p. 106, 1999.

[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] 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.

[11] M.V. Katti, R. Sami-Subbu, P.K. Rajekar, and V.S. Gupta, "Amino Acid Repeat Patterns in Protein Sequences: Their Diversity andStructural-Function Implications," Protein Science, vol. 9, no. 6, pp. 1203-1209, 2000.

[12] N. Kumar, N. Lolla, E. Keogh, S. Lonardi, C.A. Ratanamahatana, and L. Wei, "Time-Series Bitmaps: A Practical Visualization Tool for Working with Large Time Series Databases," Proc. SIAM Int'

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

[14] F. Rasheed, M. Alshalalfa, and R. Alhajj, "Adapting Machine Learning Technique for Periodicity Detection in Nucleosomal Locations in Sequences," Proc. Eighth Int'l Conf. Intelligent Data Eng. and automated Learning (IDEAL), pp. 870-879, Dec. 2007.

[15] 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.

AUTHORS

First Author– K.Lavanya, PG Student, Department of Information Technology, Periyar Maniammai University, Vallam, Thanjavur, India
Email id - lavanya62.cse@gmail.com

Second Author- R.Buvaneshwari, PG Student, Department of Information Technology, Periyar Maniammai University, Vallam, Thanjavur, India. Email id - buvi.sumathi@gmail.com

Third Author - S.P.Ramya Asst.professor , Department of Information Technology, Periyar Maniammai University, Vallam, Thanjavur, India

Email id –way2ramya@yahoo.co.in