A New Method for Improving the Performance of Linkage Pattern Mining

Saerom Lee, Takahiro Miura, and Yoshifumi Okada

***Abstract*—Linkage pattern mining is a data mining technique that finds frequent patterns appearing repeatedly across multiple sequential data. This technique does not assume similarity or correlation between the frequent patterns in a linkage pattern; hence it is expected as a promising approach for discovering causal association among events in multiple sensor data, such as physiological signals in different regions and crustal movements at different points. However, existing methods have focused only on detecting linkage patterns without noises/fluctuations in sequential data. The objective of this study is to develop a new noise-robust linkage pattern mining method. The proposed method excludes pseudo patterns derived from noises by closed itemset mining from interval graphs regarding frequent patterns so that only noiseless and maximal linkage patterns can be extracted. In this paper, the proposed method is applied to artificial sequential datasets in which linkage patterns are embedded. As a result, it is shown that the proposed method can adequately detect not only embedded linkage patterns without noise but also previously undetectable embedded linkage patterns with noise.**

***Index Terms*— linkage pattern, interval graph, closed itemset, sequential pattern mining**

I. Introduction

Sequential pattern mining is a promising and effective data mining method for finding frequent patterns in large-scale sequential data. Since Agrawal et al. [1] constructed the foundations of sequential pattern mining in 1995, a variety of new effective algorithms have been developed [2, 3] and have also been applied in a wide range of fields, such as Web log analysis [4], market basket analysis [5], behavior analysis [6], and DNA sequence analysis [7]. Research into sequential pattern mining can be broadly classified into two types: the targeting of single-sequence data and that of multiple sequence data. The former aims to find repeating and frequently occurring patterns in sequential data (frequent patterns or episodes) [8-12]. The latter focuses on detecting the same or similar subsequences among sequential data [13-15].

Manuscript received December 20, 2013. This work was supported in part by Grant-in-Aid for Young Scientists (B) (247002) of JSPS. S. Lee is with Division of Production and Information Systems Engineering, Muroran Institute of Technology, 27-1, Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan (e-mail: saerom@cbrl.csse.muroran-it.ac.jp). T. Miura is with IT Platform Division Group, Information & Telecommunication Systems Company, Hitachi, Ltd., 292, Yoshida-cho, Totsuka-ku, Yokohama, Kanagawa 244-0817, Japan (e-mail: takahiro.miura.mw@hitachi.com). Y. Okada is with College of Information and Systems, Muroran Institute of Technology, 27-1, Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan (corresponding author to provide phone: +81-143-5408; fax: +81-143-5408; e-mail: okada@csse.muroran-it.ac.jp)

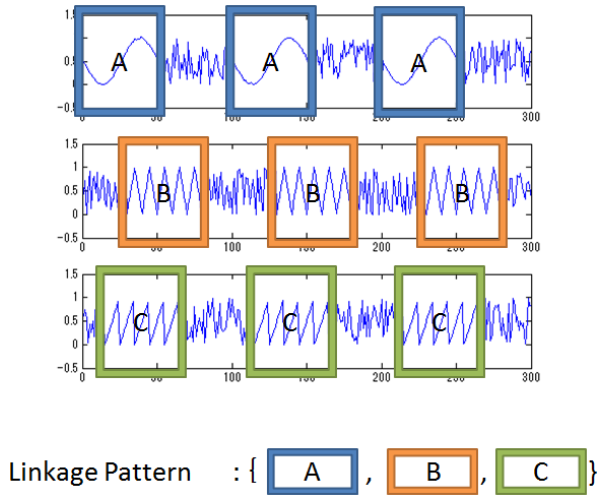

Fig. 1. A linkage pattern repeating across three sequential data

Recently, Miura and Okada [16] proposed a method for mining linkage pattern that is a set of patterns repeating across multiple sequence data. In Miura's method, linkage patterns were extracted by using an interval graph representation of frequent patterns in the sequential data. The feature of linkage pattern mining lies in the point that it does not assume similarity or correlation among different sequential data patterns. Figure 1 shows an example of a linkage pattern {A, B, C} that appears across three sets of sequential data. As we can see from this diagram, even if patterns that frequently occur in respective sequential data do not show similarity to each other, the set of those patterns is extracted as a linkage pattern if it continually appears within the same time frame. In [16], it was demonstrated that Miura's method showed good performance on sequential data without noises/fluctuations, but meanwhile it was also suggested that noise or fluctuations within the sequential data can significantly affect the accuracy of extracting linkage patterns.

The aim of this study is to improve Miura's method and to develop a noise-robust linkage pattern mining method. In the proposed method, closed itemset mining is employed to exclude randomly generated noise patterns and to obtain only frequent and maximal patterns among different interval graphs. In this paper, we show the comparative results for the performances between the proposed method and Miura's methods (hereinafter referred to as "the previous method") using artificial sequential data.

This paper is structured as follows. Section 2 defines closed itemsets. Section 3 discusses problems with the previous method and the procedure of the proposed method. Section 4 explains the experimental performance evaluation methods using artificial sequential datasets. Section 5 states the results

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

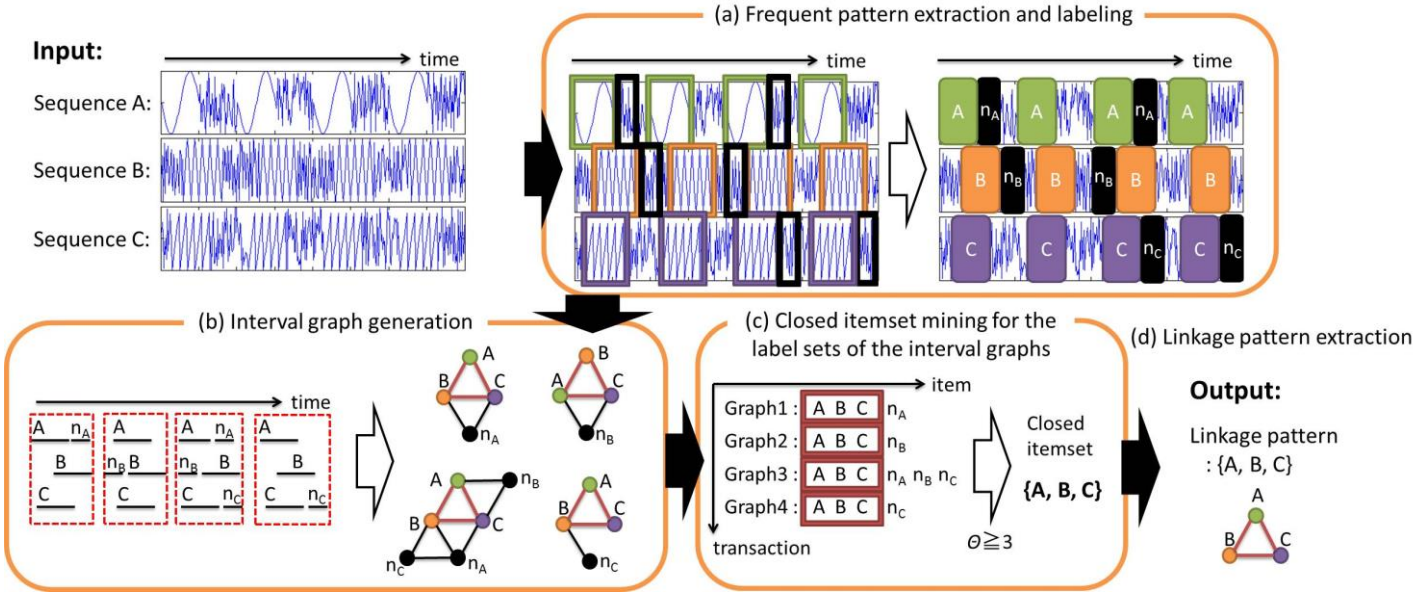

Fig. 2. Procedure of the presented method

of these experiments and presents some observations. Section 6 provides an overall summary of the paper.

II. Definition of Closed Itemset

Let $I = \{1, 2, \ldots, n\}$ be the set of *items*. A *transaction database* on I is a set $T = \{t_1, t_2, \ldots, t_m\}$ such that each t_i is included in I. Each t_i is called a *transaction*. $\|T\|$ is the size of the transaction database. A set $P \subseteq I$ is called an *itemset*. A transaction including P is called an *occurrence* of P. The set of occurrences of P is expressed as $T(P)$. The size of a set of occurrences for P is referred to as the *frequency* of P.

An itemset P is called a *closed itemset* if no other itemset Q satisfies $T(P) = T(Q)$, $P \subseteq Q$. For a given constant called a *minimum support* (hereafter *minsup*), P is frequent if $|T(P)| \geq minsup$. A frequent and closed itemset is called a *frequent closed itemset*.

In this paper, an exhaustive search on closed itemsets that occur in more than a *minsup* in a transaction database is referred to as a *closed itemset mining*.

III. Method

Figure 2 shows the procedure of the proposed method. In this figure, the figure 2a, 2b, and 2d are steps implemented in the previous method: extracting and labeling frequent patterns from each sequence (Figure 2a), generating interval graphs based on overlapping labels on the time axis (Figure 2b), and outputting the linkage pattern (Figure 2d). In the proposed method, a new step (Figure 2c) is introduced; the closed itemset mining from the generated interval graphs. This resolves the problem that linkage patterns are contaminated by noise data as presented in the previous methods. These steps are explained in detail below.

A. Frequent pattern extraction and labeling

First, normalization and discretization are executed on each sequential data as pre-processing. In the normalization, sequential data are converted to a scale from 0 to 1. In the discretization, the range of normalized data (0–1) is divided at the D stages, and a discrete value from 0 to D-1 is allocated to the data.

Next, repeatedly occurring frequent patterns are extracted from each sequential data using Mannila's algorithm [8]. This algorithm uses the maximum window width w and minimum number of occurrences θ of the frequent pattern as input parameters, where w and θ are natural numbers.

Labeling is the process of applying the same label to the same frequent pattern. This process is run after excluding frequent patterns with a length less than or equal to $w/2$. When multiple frequent patterns occur within the same sequential data and the same time frames, labeling is performed for the maximum length frequent pattern.

B. Interval graph generation

Hereafter, a labeled frequent pattern is referred to as a *label*. In this step, interval graphs are generated from the interval representation of each label. An interval graph is obtained by associating each label with a node and an overlap of any two labels on the time axis between sequential data with an edge [17-19]. In other words, an interval graph is a set of frequent patterns that occur in a linked manner in the same time frame between different sequential data.

The previous method outputs the interval graph with the highest frequency as a linkage pattern. However, frequent patterns that are constructed as a result of noise (pseudo patterns) cause the following problems. If different pseudo pattern labels are attached to all of the same interval graphs, the maximum frequency will be 1. In this case, these interval graphs are considered as completely different ones in spite of an identical linkage pattern. This is the critical problem that reduces the accuracy of linkage pattern mining.

C. Extraction of linkage patterns based on closed itemset

Since pseudo patterns tend to randomly occur on the time axis, the probability that the same pseudo pattern is included in multiple same interval graphs will be extremely low. Therefore, it is expected that pseudo patterns can be excluded by extracting label sets that commonly occur in multiple

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

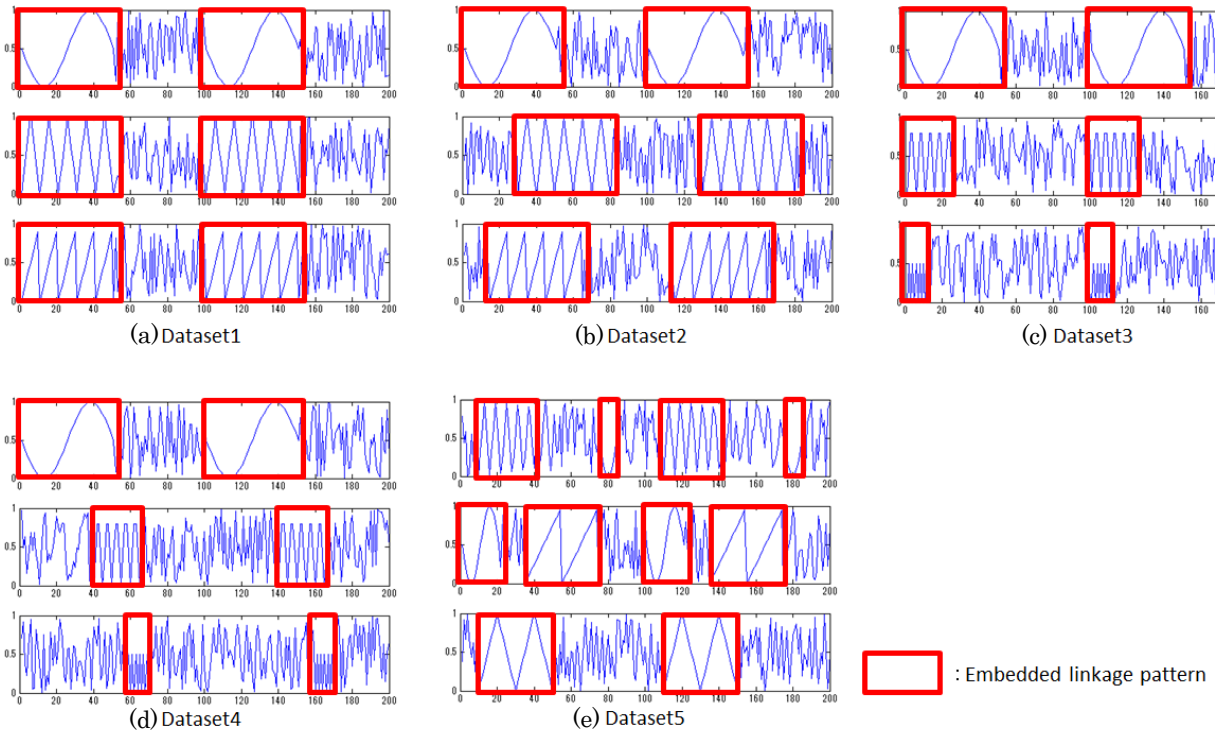

Fig. 3. Artificial datasets

interval graphs. In the proposed method, pseudo patterns are excluded by mining closed itemset on the obtained interval graphs.

Figure 2c presents the manner of excluding pseudo patterns from interval graphs. Each interval graph is seen as a transaction, and each node in the interval graph is regarded as an item. By applying the closed itemset mining to this transaction database, we can extract the maximal node sets (closed itemsets) that are shared in *minsup* or more interval graphs. Finally, the closed itemset with the highest frequency is output as the linkage pattern. By the above step, it is possible to extract linkage patterns with greater accuracy, since randomly constructed pseudo patterns can be excluded. Figure 2c illustrates an example of how the pseudo patterns n_A, n_B, and n_C are excluded; only the authentic linkage patterns {A, B, C} are appropriately extracted.

In this study, we use linear closed itemset miner (LCM) [14] that is a fast and exhaustive closed itemset mining algorithm.

IV. Experiments

The performance of the proposed method was evaluated using artificially created sequential datasets (artificial datasets).

A. Artificial datasets

Each artificial datasets was composed of three sequential data. The sequential data were generated by inserting 10 linkage patterns (embedded linkage patterns) into random sequential data created using uniform random numbers. For this experiment, we created five non-noise artificial datasets (Dataset1–Dataset5) that include no noise within embedded linkage patterns. Figure 3 shows a section of each artificial dataset. The formats of linkage patterns embedded in each dataset are as follows. Dataset1 is an artificial dataset in which equal length frequent patterns were embedded with the same start time across the three sequential data (Figure 3a). Dataset2 is an artificial dataset in which equal length frequent patterns were embedded with different start times across the three sequential data (Figure 3b). Dataset3 is an artificial dataset where different length frequent patterns for each of the three sequential data were embedded at the same time (Figure 3c). Dataset4 is an artificial dataset in which frequent patterns with different lengths for each of the three sequential data were embedded at different times (Figure 3d). Dataset5 was an artificial dataset in which one or two types of frequent patterns were embedded with different lengths and different start times for each of the three sequential data (Figure 3e).

In addition, five artificial data sets (Dataset1_noise–Dataset5_noise) that include with noise in embedded linkage patterns were created by adding fluctuations to each time point in the linkage patterns. The fluctuations were generated using normal random numbers (standard deviation = 0.01).

B. Parameter settings

For frequent pattern extraction, the minimum number of occurrences (θ) was fixed at five, and the maximum window widths (w) were set to natural numbers within the range of $2 < w < 10$. The range of w was determined on the basis of the following. When $w \leq 2$, since all the data at each time point will be labeled, data points with the number of occurrences greater than θ are all labeled. Further, when $w \geq 2\theta$, since frequent patterns with length greater than $w/2$ are extracted from all windows, each sequential data will be continuously and closely labeled. In any cases of the above two, as the continuous overlapping of intervals occurs between the sequential data, in extreme cases, only one interval graph will be generated from all of the sequential data. The minimum support (*minsup*) in closed itemset mining was set to five.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

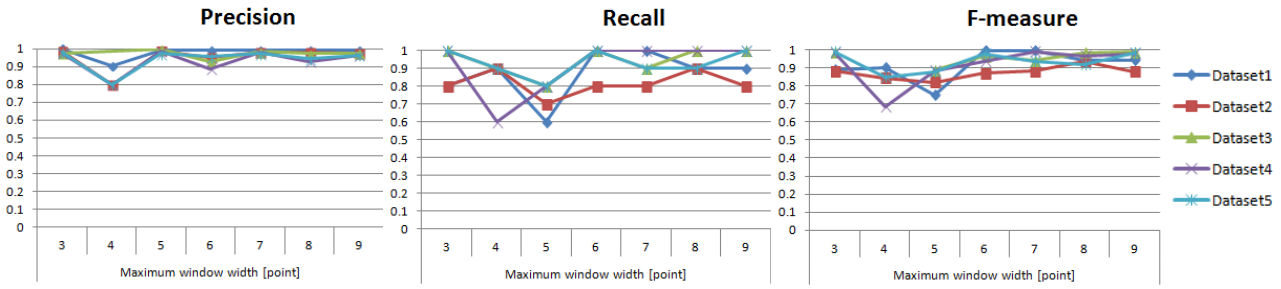

Fig. 4. Extraction accuracies by the previous method

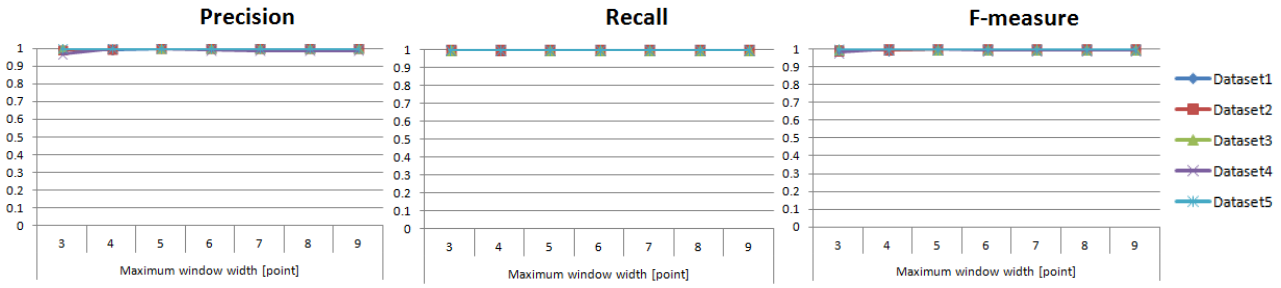

Fig. 5. Extraction accuracies by the proposed method

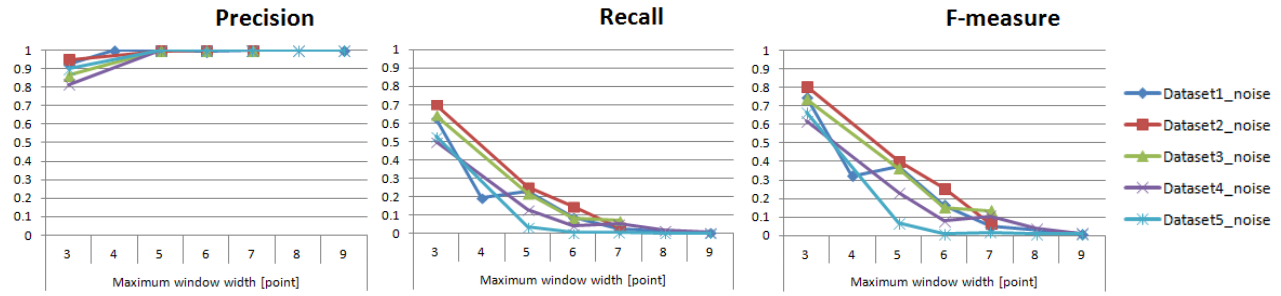

Fig. 6. Extraction accuracies for the datasets with noise

C. *Extraction accuracy of linkage patterns*

The extraction accuracies of embedded linkage patterns of the previous methods and the proposed method were compared using the 10 artificial datasets created above. *Precision*, *recall*, and *F-measure* were used as evaluation indexes. These indexes were calculated according to the following formulas.

$$Precision = CDP / DDP$$
$$Recall = CDP / EDP$$
$$F\text{-}measure = 2 * Precision * Recall / (Precision + Recall)$$

Here CDP is the number of data points in the correctly detected areas of the embedded linkage patterns, DDP is the number of data points in the areas of the embedded linkage patterns detected by the method, and EDP is the number of data points in the embedded linkage patterns.

V. Results and Discussion

A. *Extraction accuracy for non-noise datasets*

Figure 4 and 5 are graphs of precision, recall and F-measure when the proposed method and the previous method were applied to the five non-noise datasets. In this graphs, the scores in different maximum window width (w) are presented.

As a result, the previous method shows unstable scores for the different w. This is caused by the pseudo patterns randomly formed by noise being added to embedded linkage patterns. In contrast, the proposed method demonstrates 100% extraction accuracy for all w values. This means that the noises included in the interval graphs were suitably excluded by closed itemset mining.

B. *Extraction accuracy for noise datasets*

In the previous method, the accuracy of extracting linkage patterns shows 0% for all the datasets, because only one interval graphs is generated. This is due to the fact that the pseudo patterns exist throughout the sequence data. Figure 6 is graphs of precision, recall and F-measure for the five datasets with noise (Dataset1_noise－Dataset5_noise). These graphs show the scores for different w.

Precision shows 80% or more for all w. In particular, when w is 5 or more, embedded linkage patterns are perfectly extracted from all datasets. This is because pseudo patterns are suitably excluded at the step of closed itemset mining.

Recall tends to decrease as the w increases. In particular, when w is 5, the score dramatically decrease in all the datasets. This is because the number of frequent patterns extracted

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

from each sequence dramatically decreases in $w \geqq 5$; therefore, the obtained interval graphs are also dramatically reduced.

F-measure decrease significantly by the influence of the drastic decline of recall values. As explained in section 4.2, we can see that w should be specified to a smaller value in the range of $2 < w < 2\theta$ to get higher extraction accuracy.

VI. Conclusion

We have proposed a new noise-robust linkage pattern mining method based on closed itemset mining. In the proposed method, closed itemset mining was employed to exclude randomly generated noise patterns and to obtain only frequent and maximal patterns among different interval graphs. In this paper, we used artificial data to compare the performance of the proposed method and the previous method. The results showed that the extraction accuracy of linkage patterns was significantly improved by the proposed method. In particular, the proposed method was able to appropriately detect linkage patterns with noise which were not detected at all in the previous method.

In the future, we will address increasing the speed of the frequent pattern mining algorithm. Moreover, we will apply the method to large-scale real sequential data that includes noise and fluctuations, such as vital data and crustal movement data. In addition, the practical applicability of the method will be evaluated in terms of the extraction accuracy and the computational time.

Acknowledgment

This work was supported in part by Grant-in-Aid for Young Scientists (B) (247002) of JSPS.

References

[1] Agrawal, R. and Srikant, R. 1995. Mining Sequential Patterns. Proc. of The 11th Int'l Conf. on Data Engineering. 3-14.

[2] Tak-chungm F. 2011. A review on time series data mining. Engineering Applications of Artificial Intelligence. Volume 24, Issue 1. 164-181.

[3] Zhao, Q. and Bhowmick, S. S. 2003. Sequential Pattern Mining: A Survey. Technical Report. CAIS. Nanyang Technological University. Singapore. No. 2003118.

[4] C.I. EZEIFE, YI LU. 2005. Mining Web Log Sequential Patterns with Position. Coded Pre-Order LinkedWAP-Tree. Data Mining and Knowledge Discovery, 10, 5–38, 2005c Springer Science, Business Media. Inc. Manufactured in The Netherlands.

[5] Xintao Wu, Ying Wu, Yongge Wang, Yingjiu L, Privacy-Aware Market Basket Data Set Generation: A Feasible Approach for Inverse Frequent Set Mining, 2005, Proceedings of the Fifth SLAM International Conference on Data Mining, pp 103-114

[6] Andreas D. Lattner, Andrea Miene, Ubbo Visser, and Otthein Herzog, Sequential Pattern Mining for Situation and Behavior Prediction in Simulated Robotic Soccer, 2006, RoboCup 2005: Robot Soccer World Cup IX Lecture Notes in Computer Science Volume 4020, pp 118-129

[7] Karaca M., Bilgen M., Onus A. N., Ince A. G. and Elmasulu S. Y. 2005 Exact Tandem Repeats Analyzer (E-TRA): A new program for DNA sequence mining. J. Genet. 84, 49–54

[8] Ohtani, H. Kida, T. Uno, T and Arimura, H. Efficient Serial Episode Mining with Minimal Occurrences. 2009. The Third International Conference on Ubiquitous Information Management and Communication.

[9] Wen-Chi, P and Zhung-Xun, Liao. Mining sequential patterns across multiple sequence databases. 2009. Data & Knowledge Engineering Volume 68, Issue10. 1014-1033.

[10] Gong, C. Xindong, W. and Xingquan, Z. Mining Sequential Patterns across Time Sequences. 2008. New Generation Computing, 26. 75-96.

[11] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. and Hsu, M.- C. 2001. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. Proc. of the 17th Int'l Conf. on Data Engineering. 215-224.

[12] Mannila, H., Toivonen, H. embedded and Verkamo, A.I. 1997. Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery 1. 259-289.

[13] Sakurai, Y. Faloutsos, C and Yamamuro, M. Stream monitoring under the time warping distance. 2007. In Proc. of ICDE. 1046-1055.

[14] Sakurai, Y., Papadimitriou, S. and Faloutsos, C. 2005. BRAID: Stream Mining through Group Lag Correlations. In Proc. of ACM SIGMOD Conference. 599-610.

[15] Zhu, Y. and Shasha, D. 2002. StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. In Proc. Of VLDB. 358-369.

[16] Takahiro Miura and Yoshifumi Okada, "Detection of Linkage Patterns Repeating across Multiple Sequential Data", International journal of computer applications, Vol.63, No.3, pp.14-17, 2013.

[17] Miyoshi, N. Shigezumi, T. Uehara, R andWatanabe, O. 2009. Scale free interval graphs. Theoretical Computer Science Volume 410, Issue 45. 4588-4600.

[18] Korte, N. and Mohring, R.H. 1979. An incremental linear-time algorithm for recognizing interval graphs. SIAM Journal on Computing, vol. 18. 68-81.

[19] Lueker, G.S. and Booth, K.S. 1979. A linear time algorithm for deciding interval graph isomorphism. Journal of the ACM, vol. 26. 183-195.

[20] Takeaki Uno and Hiroki Arimura, Data Intensive Computing : No.2 Frequent Itemset Mining Algorithms(<Lecture Series>Intelligent Computing and Related Issues (2)), The Japanese Society for Artificial Intelligence 22(3), pp.425-436, 2007.