Hybrid Temporal Pattern Mining with Time Grain on Stock Index

Sheng Xiang Fan
Department of Computer Science and Information Management
Providence University
Taichung, Taiwan
e-mail: g9871016@pu.edu.tw

Jieh-Shan Yeh
Department of Computer Science and Information Management
Providence University
Taichung, Taiwan
e-mail: jsyeh@pu.edu.tw

Yaw-Ling Lin
Department of Computer Science and Information Engineering
Providence University
Taichung, Taiwan
e-mail: yllin@pu.edu.tw

Abstract*—Both sequential pattern mining and temporal pattern mining have become highly relevant data mining topics in this decade. In 2009, Wu and Chen proposed a representation for hybrid events and an HTPM mining method. However, their approach neither addresses nor analyzes the length of event time. An event representation may stand for the same event with extremely different time lengths, which may induces the loss of accurate mining results. This paper addresses this difficulty and explores different models and solutions. Firstly, this paper introduces the concept of the time grain, and proposes new hybrid models as well as the pattern mining algorithms associated with the concept of event length limit. Events in hybrid sequences are divided or distinguished according to a given threshold, to enable a detailed exploration of the more frequent hybrid sequence of events. Secondly, this paper utilizes the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) as the testing data, to examine the proposed model and the feasibility and effectiveness of the algorithm. This research addresses effectively the problem of hybrid events with different lengths, and enhance the performance of dynamic hybrid temporal pattern mining.

Keywords: Hybrid temporal pattern mining, sequential pattern mining, temporal pattern mining, pattern representation.

I. INTRODUCTION

Sequential pattern mining of point-based event databases [1] and temporal pattern mining of interval-based event databases [2] have become highly relevant research topics in data mining. Both models have extremely widespread applications, for example: transaction sequence analysis, DNA sequence analysis, stock trading trend analysis, and analysis of book and video lending records. Pattern mining is also widely used in decision analysis and prediction of future events.

In 2009, Wu and Chen proposed a representation for hybrid events and an HTPM mining method [3]. However, their approach neither addresses nor analyzes the length of event time. An event representation may stand for the same event with extremely different time lengths, which may induces the loss of accurate mining results. This study observed this difficulty and explored different models and solutions. First, this paper proposes new hybrid models and pattern mining algorithms with the concept of event length limit. Events in hybrid sequences are divided or distinguished according to a given threshold, to enable a detailed exploration of the more frequent hybrid sequence of events. Second, this paper utilizes the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) to examine the proposed model and the feasibility and effectiveness of the algorithm.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 proposes new hybrid models and pattern mining algorithms. Section 4 presents our experimental results and evaluates the performance of the proposed algorithm. Finally, Section 5 presents conclusions of the study.

II. RELEATED WORKS

In this section, we review the following three mining models: sequential pattern mining (point-based), temporal pattern mining (interval-based), and hybrid temporal pattern mining (hybrid).

A. Sequentail Pattern Mining

Sequential pattern mining was first proposed by Agrawal and Srikant in 1995. Patterns of point-based events that frequently occur in transaction databases are discovered given a set of sequences and a threshold, the minimum support. A formal definition of sequential pattern mining is given as follows:

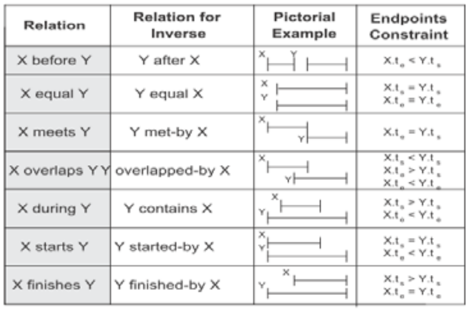
Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given a user-specified minimum support threshold, the purpose of sequential pattern mining is to determine all frequent subsequences.

Since Agrawal and Srikant [1] proposed AprioriAll, the most famous algorithm of sequential pattern mining, the majority of studies have also been primarily Apriori-style; for example, GSP [4] and SPADE [5]. Some algorithms, designed according to the Pattern-Growth method of the tree structure record type, have demonstrated high mining potency: for example, FreeSpan [6] and PrefixSpan [7].

B. Temporal Pattern Mining

In 2000, Kam and Fu proposed temporal pattern mining [8]. Temporal pattern mining is a variant of the sequential pattern mining problem, revealing interval-based event patterns rather than point-based event patterns from sequence databases. The expression method of this pattern is based on Allen’s two-interval events relationship. Thirteen possible relationships exist between two interval-based events, as shown in Figure 1. The relationships among more than two events are highly complicated in temporal pattern mining.

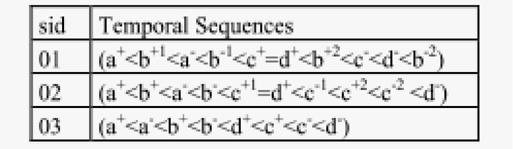
The first temporal pattern mining algorithm, the KF algorithm, was proposed by Kam and Fu. The KF algorithm, similar to the Apriori-gen method, generates long length patterns from short length patterns. Each loop includes two stages. The first stage is the candidate pattern generation stage, using the L_{k-1} pattern set to produce C_k (length k of candidate pattern set). The second stage determines which candidate patterns are frequent. The KF Algorithm stops while no additional frequency patterns are produced.



Relation	Relation for Inverse	Pictorial Example	Endpoints Constraint
X before Y	Y after X		$X.t_e < Y.t_s$
X equal Y	Y equal X		$X.t_s = Y.t_s$ $X.t_e = Y.t_e$
X meets Y	Y met-by X		$X.t_e = Y.t_s$
X overlaps Y	Y overlapped-by X		$X.t_s < Y.t_s$ $X.t_e > Y.t_s$ $X.t_e < Y.t_e$
X during Y	Y contains X		$X.t_s > Y.t_s$ $X.t_e < Y.t_e$
X starts Y	Y started-by X		$X.t_s = Y.t_s$ $X.t_e < Y.t_e$
X finishes Y	Y finished-by X		$X.t_s > Y.t_s$ $X.t_e = Y.t_e$

Figure 1. Allen's two-interval events relationship

Expression of the KF-method exhibits obstacles involving ambiguity. In 2007, Wu and Chen mitigated this ambiguity by defining a regular expression of two-interval event patterns. According to the regular expression, Wu and Chen developed a new interval pattern mining method called TprefixSpan [9]. As shown in Figure 2, Wu and Chen used "+" to express the event starting time, "-" to express event ending time, and "<" and "=" to indicate the relationship of the event times. Therefore, the expression ambiguity can be avoided.



sid	Temporal Sequences
01	$(a^+<b^{+1}<a^-<b^{-1}<c^+=d^+<b^{+2}<c^-<d^-<b^{-2})$
02	$(a^+<b^+<a^-<b^-<c^{+1}=d^+<c^{-1}<c^{+2}<c^{-2}<d^-)$
03	$(a^+<a^-<b^+<b^-<d^+<c^+<c^-<d^-)$

Figure 2. Example of KF-method expression

In 2010, Yeh and Ting [10] extended the traditional sequential pattern expression by combining "+" and "-" notations on event starting time and ending time, as shown in Figure 3. Based on the unique expression of event sequences, temporal pattern mining can be regarded as traditional sequential pattern mining. Therefore, Apriori-Like and SP&EPPF algorithms proposed by Yeh and Ting can be utilized to solve mining problems and to improve mining efficacy.

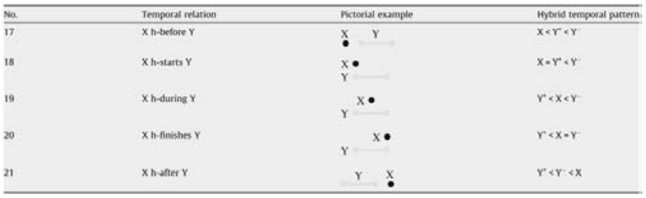
Patient	Temporal Sequence
01	$<a^+, b^+, a^-, b^-, (c^+, d^+), b^{2+}, c^-, d^-, b^{2-}>$
02	$<a^+, b^+, a^-, b^-, (c^+, d^+), c^-, c^{2+}, c^{2-}, d^->$
03	$<a^{+,}\ a^-, b^+, b^-, d^+, c^+, c^-, d^->$

Figure 3. Example of Yeh and Ting's expression

C. Hybrid Temporal Pattern Mining

In most situations, the sequences in the sequential pattern mining applications usually include both point-based events and interval-based events. In 2009, Wu and Chen proposed hybrid temporal pattern mining (HTPM), and also defined hybrid event sequence expression [3]. However, the hybrid temporal pattern mining problem cannot be resolved by any of the existing point- or interval-based methods, because the temporal pattern mining problem cannot be reduced to a sequential pattern mining problem, and the hybrid temporal pattern mining problem cannot be reduced to a temporal pattern mining problem.

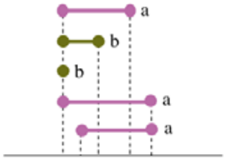
A hybrid temporal pattern consists of point-based events and interval-based events. Two-hybrid events are composed of three relationships of two point-based events, 13 relationships of two interval-based events, and five relationships of two hybrid events, as shown in Figure 4. The total number of hybrid temporal event relationships is 21.



No.	Temporal relation	Pictorial example	Hybrid temporal pattern
17	X h-before Y		$X < Y^+ < Y^-$
18	X h-starts Y		$X = Y^+ < Y^-$
19	X h-during Y		$Y^+ < X < Y^-$
20	X h-finishes Y		$Y^+ < X = Y^-$
21	X h-after Y		$Y^+ < Y^- < X$

Figure 4. Five relationships of two hybrid events

Similar to interval pattern expression, the expression of hybrid event patterns uses "+" and "-" to indicate event start time and event end time. A point event is labeled by the event ID. The relationship of two events is specified by "<" and "=". Figure 5 illustrates a hybrid event pattern.



$htp_1 = (a^{+0}=a^{+1}=b^+=b<a^{+2}<b^-<a^{-0}<a^{-1}=a^{-2})$

Figure 5. The expression of a hybrid event pattern

The HTMP method can also be implemented at the point-based event or with interval-based event mining.

III. DEFINTIONS AND PROPOSED MODEL

This section introduces the proposed hybrid temporal pattern mining model with time grain. First, definitions of notations used in the rest of the paper follow.

A. Notations and definitions

Let E be the $\{e_1, e_2, ..., e_n\}$ of all event types that may occur at the point-based and interval-based events

Definition 1: A ***point-based event*** (poE) is an event occurring at a certain time point. A poE is stored in the form (e, t_p) in a hybrid event sequence database, where $e \in E$ and t_p is the time at which e occurs. For example, let $E = \{a, b, c, d\}$; $(c, 6)$ is a point-based event.

Definition 2: An ***interval-based event*** (inE) is an event occurring over a time period. An inE is stored in the form $(e,[ts,te])$ in a hybrid event sequence database, where $e \in E$ and Ts and Te are the starting time and ending time of e, respectively. For example, $(a,[5,10])$ is an interval-based event.

Definition 3: A ***hybrid event sequence*** is composed of a series of point-based and/or interval-based events. A hybrid event sequence is represented as $S_i = \{E_1, E_2,.., E_{m_i}\}$, where E_j is either a poE or an inE. For example, $S_1 = \{(a,[5,10]),(c,6),(b,[6,12]),(c,8),(A,[8,12])\}$.

Definition 4: ***Pattern representation*** of a hybrid event sequence. This research adopted the pattern representation proposed by Yeh and Ting [YT10]. The poEs and inEs are ordered by the occurring time and the starting time. For example, the pattern representation of S_1 is $< a^+,(b^+,c),(a^{2+},c),a^-,(a^{2-},b^-) >$

In hybrid temporal pattern mining, both representations of the two patterns in Figure 6 are $< a^+,b^+,c^+,a^-,b^-,c^- >$. Obviously, the occurrence of Event c in the pattern on the right is much longer than the occurrence of Event c in the pattern on the left. This ambiguity of the pattern representation may induce some difficulties regarding the mining problem. For example, in weather forecasts, the event of raining during one day is quite different from the event of raining during five days.

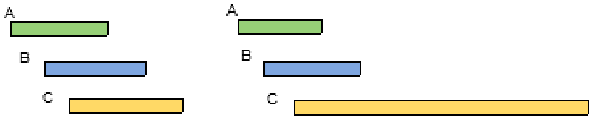


Figure 6. two patterns with the same representation

Therefore, this paper introduces the concept of the time grain to distinguish two events that have excessive length variance.

Definition 5: a ***time grain*** (TG) is a fixed length time interval, and is used to distinguish or to divide interval type events.

Definition 6: (***Event-distinguished Model***): Given a TG, the event-distinguished model differentiates events with the same event ID based on the given TG. If the difference between the shortest length of a given event and longest length of the given event is greater than the TG, we distinguish the given event by assigning the event ID different suffixes.

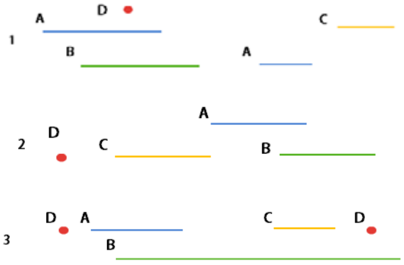


Figure 7. Example of three pattern sequences.

Consider the example provided in Figure 7 and Table 1, displaying three sequences {S_1, S_2, S_3} with event IDs A, B, C, and D. If TG = 5, the difference of the shortest length and the longest length of Event B is greater than 5. We distinguish Event B and assign it with different labels. The mean value of the shortest length and longest length of Event B is (3+10)/2=6. Event B with a length less than or equal to 6 is considered a long event, and is assigned a new label, B_1. Event B with a length greater than 6 is considered a short event, and is assigned a new label, B_2. Therefore, an event ID with distinct time lengths can be distinguished.

Event ID	Shortest of events	Longest of events	Difference
A	A(6,9) = 3	A(1,5) = 4	1
B	B(8,11) = 3	B(3,13) = 10	7
C	C(9,11) = 2	C(3,6) = 3	1

Table 1. The lengths of events in Figure 7

B. *Algorithm model*

We now introduce the proposed algorithm

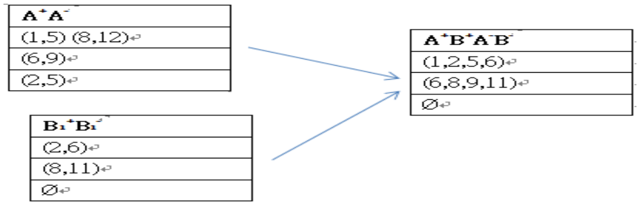
Consider the example provided in Figure 7, the proposed algorithm first computes the difference and the mean value of the shortest length and longest length of each event, as shown in Table 1. The algorithm then assigns different notations for the events with a difference greater than the given time grain (TG). As in Table 2, Event B in S_1 and S_2 is renamed B_1, and Event B in S_3 is renamed B_2.

S_1	S_2	S_3
A(1,5)	D(1)	D(1)
B_1(2,6)	C(3,6)	A(2,5)
D(4)	A(6,9)	B_2(3,13)
A(8,9)	B_1(8,11)	C(8,10)
C(10,12)		D(11)

Table 2. Sequences with new event labels.

Assume the minimum support (minS) is 50 %, that is, a frequent event must appear in at least two sequences. The algorithm now generates frequent 1-event patterns (L_1).

In the next step, the algorithm combines events in L_1 to generate the candidate 2-event patterns C_2. First, combine events of S_1, then combine S_2 and S_3. These combined events from S_1 and S_2 do not have to combine again, and not to previous events S_2 joined appear in above sequence. Figure 8 and Figure 9 demonstrate how to generate L_2 from L_1.



A^+A^-
(1,5) (8,12)
(6,9)
(2,5)

$B_1^+B_1^-$
(2,6)
(8,11)
∅

$A^+B^+A^-B^-$
(1,2,5,6)
(6,8,9,11)
∅

Figure 8. Generating $A^+B^+A^-B^-$ form A^+ A^- and B^+B^-

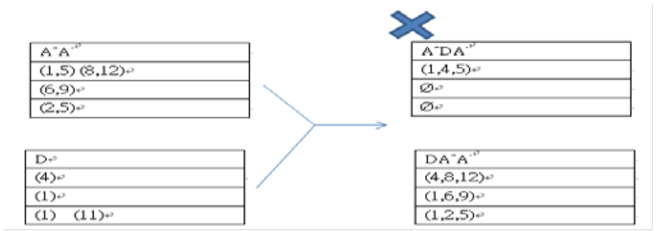


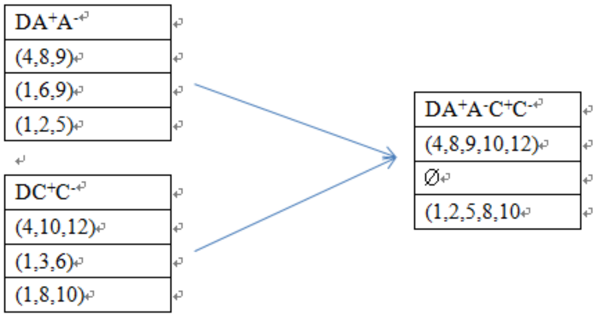
Figure 9. Generating DA^+A^- form A^+A^- and D

After generating all L_2, the algorithm uses L_2 to generate L_3. The algorithm repeats the steps until no additional patterns are generated. Table 3 is $L_{2.}$

$A^+B_1^+A^-B_1^-$	$A^+A^-C^+C^-$	$B_1^+DB_1^-$	DA^+A^-	DC^+C^-
(1,2,5,6)	(1,5,9,11)(8,9,10,12)	(2,4,6)	(4,8,9)	(4,10,12)
(6,8,9,11)	Ø	Ø	(1,6,9)	(1,3,6)
Ø	(2,5,8,10)	(3,11,13)	(1,2,5)	(1,8,10)

Table 3. 2-event patterns L_2

Definition 7 (combining event): two candidate events appearing on the same sequence are combined. The number of identical sequences must through minS, otherwise they cannot generate the next candidate item sets; for example, (A+B1+A-B1-){S_1,S_2, Ø} and (A+A-C+C-){S_1, Ø,S_3}. They have the same sequence S_1, but only one sequence cannot thought minS. Therefore, we can skip combining these events and delete them.



DA^+A^-
(4,8,9)
(1,6,9)
(1,2,5)

DC^+C^-
(4,10,12)
(1,3,6)
(1,8,10)

$DA^+A^-C^+C^-$
(4,8,9,10,12)
Ø
(1,2,5,8,10

Figure 9. Generating L_3 from L_2

IV. Data preparation

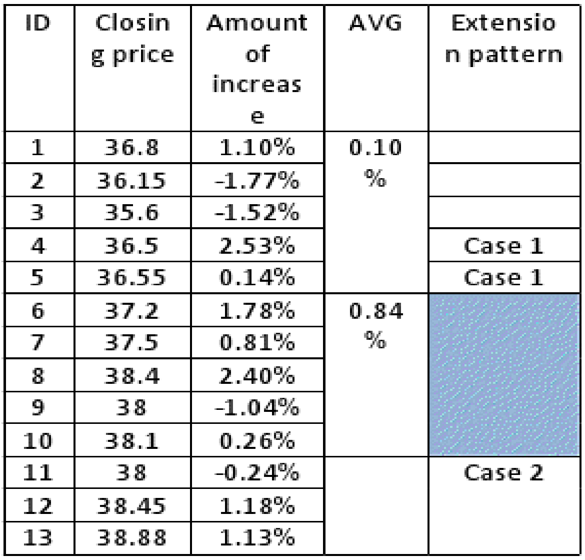
We now utilize the daily stock market closing price of the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) [1] to produce the hybrid sequence patterns. We propose a data preparation model and generate sequential patterns, according to the following steps.

1. According to the closing price, calculate the amount of increase (%). Assume that Di = the closing price of the i-th day, and that the amount of increase of $D_i = (D_i-(D_{i-1}))/D_{i-1}$.

2. Calculate the average of D_is in a given interval. Assume that we set the interval number = 5. We then calculate the average of D_1~D_5. The next interval is D_6~D_{10}.

3. Extension calculated interval pattern.

Next, we provide an example in Table 5. Assuming the original data is in Table 4, and according to Step 1, calculate the amount of increase (%). In Step 2, calculate the average of D_i and interval number = 5. Assume we set the minimum support at 0.6 %. The average of D_6~ID_{10} is through 0.6 %; therefore, it is an interval pattern. Step 3 is an extension of the pattern, and involves two cases. In the first case, if the pattern meets the same nature event, then extension. in the second case, assume that the small interval number = 3. If the pattern meets the small interval pattern through the minimum support, then extension. For example, D_6~D_{10} is a positive pattern. D_5 is consistent with Case 1; therefore, the extension pattern to D_5 converts to D_5~D_{10}. Next, D_{11} is not consistent with Case1; therefore, compare ID11 to case2. The average of D_{11}~D_{13} is 0.69 through the minimum support and is consistent with Case 2; therefore, extension pattern. The pattern is converted to D_5~D_{13}. Repeat the aforementioned steps until the cases can no longer be matched. Finally, the hybrid pattern sequence is generated as in Figure10.



ID	Closing price	Amount of increase	AVG	Extension pattern
1	36.8	1.10%	0.10 %	
2	36.15	-1.77%		
3	35.6	-1.52%		
4	36.5	2.53%		Case 1
5	36.55	0.14%		Case 1
6	37.2	1.78%	0.84 %	▨
7	37.5	0.81%		
8	38.4	2.40%		
9	38	-1.04%		
10	38.1	0.26%		
11	38	-0.24%		Case 2
12	38.45	1.18%		
13	38.88	1.13%		

Table 4. patterns from stock closing price

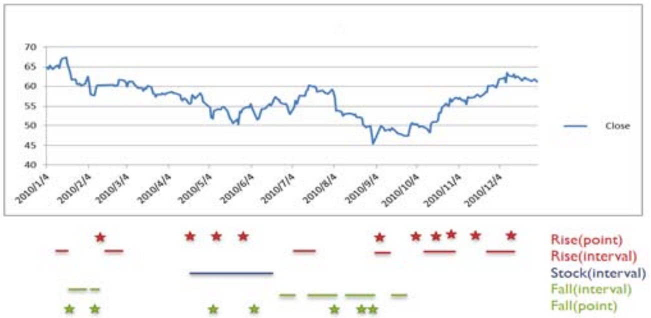


Figure 10. The hybrid pattern sequence of a stock closing price

V. Conclusions

In most actual situations, the sequences in sequential pattern mining applications usually include both point-based events and interval-based events. The hybrid temporal pattern mining problem cannot be resolved by any of the existing point- or interval-based methods. In 2009, Wu and Chen proposed hybrid temporal pattern mining (HTPM) and also defined hybrid event sequence expression. However, their expression cannot distinguish events with different occurrence times. This ambiguity of the pattern representation may induce some difficulties regarding the mining problem. Firstly, this paper introduces the concept of the time grain to distinguish two events that have an excessive length variance. Secondly, this paper proposes new hybrid models and pattern mining algorithms with the concept of event length limit. Events in hybrid sequences are divided or distinguished according to a given threshold, to enable a detailed exploration of the more frequent hybrid sequence of events. Finally, this paper utilizes the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) to examine the proposed model and the feasibility and effectiveness of the algorithm.

ACKNOWLEDGMENT

This work was supported by the National Science Council, Taiwan under grant NSC99-2632-E-126-001-MY3 and NSC99-2221-E-126-008.

REFERENCES

[1] R. Agrawal and R. Srikant, “Mining sequential patterns,” in Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, IEEE Computer Society Press, pp. 3-14, 1995.

[2] J. F. Allen, “Maintaining knowledge about temporal intervals,” Communications of the ACM, vol. 26, no. 11, pp. 832-843, 1983.

[3] S. Y. Wu and Y. L. Chen, "Discovering hybrid temporal patterns from sequences consisting of point- and interval-based events," Data & Knowledge Engineering, vol. 68, no. 11, pp.1309-1330, 2009.

[4] R. Srikant and R. Agrawal, “Mining sequential patterns: generalizations and performance improvements,” in Proceedings of the Fifth International Conference on Extending Database Technology (EDBT), Avignon, France, IBM Research Division, pp. 3-17, 1996.

[5] M. J. Zaki, “SPADE: an efficient algorithm for mining frequent sequences,” Machine Learning, vol. 42, no. 1 , pp. 31-60, 2001.

[6] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, M.C. Hsu, “FreeSpan: frequent pattern-projected sequential pattern mining,” in Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, Massachusetts, United States, ACM Press, pp. 355-359, 2000.

[7] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.C. Hsu, “PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth,” in Proceedings of the 17th International Conference on Data Engineering, Heidelberg, Germany, pp. 215–224, 2001.

[8] P. S. Kam and A. W. C. Fu, “Discovering Temporal Patterns for Interval-Based Events,” in Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery, Springer-Verlag, pp. 317-326, 2000.

[9] S. Y. Wu and Y. L. Chen, "Mining Nonambiguous Temporal Patterns for Interval-Based Events," IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 6, pp. 742-758, 2007.

[10] J. S. Yeh and C. H. Ting, "SP&EPPF: A Novel Algorithm for Sequential Interval-based Pattern Mining," in Proceedings of the 2010 International Conference on Data Mining (DMIN'10), pp. 343-349, 2010.

[11] Taiwan Stock Exchange Capitalization Weighted Stock Index, http://www.twse.com.tw/en/, May 2011.