

A New TP-PTP Miner Algorithm for Mining Temporal Patterns

S.Soundariya,

Post-Graduate Scholar, Department of Computer Science And Engineering, MSEC,
Ramanathapuram, India

Abstract_ Sequential pattern mining is a topic in data mining concerned with finding statistically relevant patterns between data examples where the values are delivered in a sequence. Sequential mining applications using time interval based event data have attracted considerable efforts in discovering patterns from events. Here challenging issues are the relationship between two intervals is complex and how to effectively and efficiently mine interval based sequences. For solving complex relation, I develop two novel representations such as end point and end time representation. I propose two algorithms called Temporal Pattern Miner and Probabilistic Temporal Pattern Miner which discover three types of interval-based sequential patterns called temporal pattern, occurrence-probabilistic temporal pattern and duration-probabilistic temporal pattern. I use three types of pruning techniques which reduces the search spaces of the mining process. Experimental studies show that both algorithms are able to find three types of patterns efficiently.

Index Terms_ data mining, representation, temporal pattern, sequential pattern, interval based event.

I. INTRODUCTION S

Sequential pattern mining is an active research topic in data mining because of its widespread applicability. This type of

application always considers the order relation and the time issue in our daily lives. Sequential pattern mining mainly deals with extracting positive behaviors that can be used to predict an event based on the activity in the preceding sequence of events. However, finding sequential patterns is a difficult issue since mining may require generating or examining a large number of intermediate subsequence combinations. In various real-world scenarios, some events which intrinsically tend to persist for periods of time rather than being instantaneous occurrences, cannot be treated as “time points.” In such cases, the data are typically a sequence of events with both start and finish times. For example, Adopted sensor technology to monitor the electricity usage of all household appliances. Specifically, power meters are deployed to collect appliance usage log data. The times that each appliance is turned on and off can be easily identified. Obviously, such appliance usage log data are interval-based Data. Much of the existing research mainly focuses on discovering patterns from time point-based event data. These approaches are hampered by the fact that they can only efficiently handle instantaneous events not event intervals. The features of time intervals and time points vary substantially; the pairwise relationship between two time interval-based events is intrinsically complex [2]. This complex relationship is a critical problem in the endeavor to

design an efficient and effective time interval-based pattern (or **temporal pattern**) mining algorithm, since it may increase candidate generation and the workload for counting the support of candidate sequences. The extracted patterns encode the characteristics of the original temporal sequence and can be used for data summarization and pattern detection. Since activities cohere typically beyond modalities, a stand-alone pattern is seldom meaningful. The correlations among the patterns across modalities endow the pattern with meanings. Also temporal data mining techniques allow for the possibility of computer driven, automatic exploration of the data.

II. RELATEDWORK AND AIM OF RESEARCH

A number of recent studies have investigated the mining of interval-based events. To the best of our knowledge, most of these studies are based on Allen's temporal relations [2]. However, Allen's temporal relations are binary in nature, and may exhibit problems to describe the relationships among more than two intervals. An appropriate representation is crucial. In this section, I discuss various representations and mining methods. Furthermore, I review several studies on probabilistic sequential pattern mining.

Kam et al. [8] proposed a compact encoding method, named hierarchical representation, to efficiently express the temporal relationships among intervals. However, this method may suffer from two ambiguous problems. First, the same relationships among event intervals can be mapped to different temporal patterns. Second, a temporal pattern can represent different relationships among event intervals. Hoppner [7] proposed an unambiguous representation, relation matrix, which exhaustively lists all binary

relationships among event intervals in a pattern. Temporal representation [9] utilizes the relationship among end time points to express the temporal pattern unambiguously. Patel et al. applied additional counting information to achieve an unambiguous expression called the augmented hierarchical representation. Each Allen descriptor contains a counter that counts the number of relation occurrences. TSKR considers the noise tolerance and expresses the temporal concepts of coincidence for interval patterns. The pattern represented in TSKR is robust and easily understandable. SIPO uses the partial order among semi-intervals to create an abstraction that can represent many examples with similar properties; however, this representation was based on closed sequential patterns and closed itemsets; hence, the mining is time-consuming. Coincidence-Representation involves segmenting intervals into disjointed slices to avoid the processing of complex relationships. In this paper, I utilize the arrangement of endpoints to describe an interval sequence directly. The proposed representation includes time information on the occurrence of each endpoint, without expressing the complex relationships among endpoints.

Several Apriori-like algorithms have been proposed to discover temporal patterns in interval-based data. Villafane et al. proposed a mining method to discover time interval-based sequential patterns by transforming data sequences into containment graphs. Kam et al. [8] proposed an Apriori-like algorithm to discover temporal patterns based on hierarchy representation transforms an event sequence into id-lists then merges the id-lists iteratively to generate temporal patterns. Laxman et al. [10] extended the original framework of frequent episode discovery in event sequences by

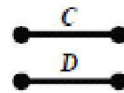
incorporating event duration constraints. IEMiner uses several optimization strategies to reduce the search space and remove non-promising candidate temporal sequences. Yoshida et al. efficiently mined sequential patterns by using temporal annotations that quantify the duration between successive symbols. In addition, several pattern-growth algorithms have been introduced to increase the efficiency of temporal pattern discovery. ARMADA was proposed to find frequent temporal patterns in a large database. This algorithm requires candidate generation to determine the relationship for growing patterns from local frequent intervals.

Obviously, the complex relationships among intervals create a huge search space and complicate the mining processes for temporal patterns. From the analysis above, we know that the complex relationship is a critical concern when designing **priori-like algorithms**, in each iteration, the complex relationships among intervals may lead to generating a huge number of candidate sequences and creating a tedious workload of support counting for candidate sequences. Moreover, for **pattern-growth algorithms**, the complex relationships within temporal pattern-mining may increase the complexity. Numerous additional low-level processes and operations are required, which is prohibitively expensive in terms of time and space when mining temporal patterns. The motivation of our study is that, if we can simplify the processing of complex relationships among event intervals within temporal patterns, we may design an efficient and effective pattern-growth algorithm.

Table 1. An example Database with Four Interval-sequences

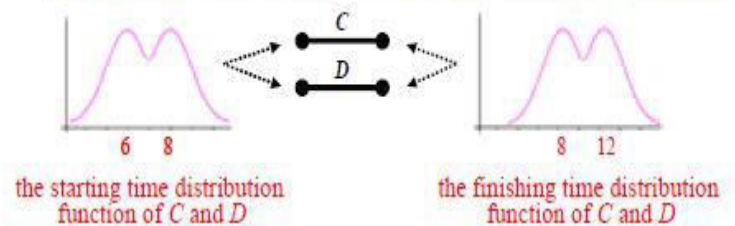
SID	event symbol	start time	finish time	interval duration	event sequence
1	A	1	4	3	
	B	3	6	3	
	C	6	12	4	
	D	8	12	4	
2	B	3	6	3	
	C	6	12	4	
	D	8	12	4	
3	C	6	8	2	
	D	6	8	2	
4	A	1	3	2	
	C	6	8	2	
	D	6	8	2	
	E	12	4	2	

* given the minimum support = 3



(a) A temporal Pattern

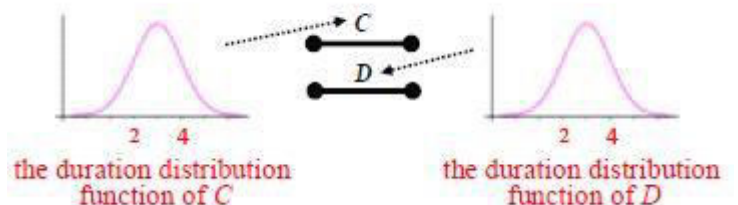
* with including occurring time distribution function



(b) Occurrence-

Probabilistic Temporal Pattern

* with the duration distribution function



(c) A duration-Probabilistic Temporal Pattern

III. METHODOLOGY AND DISCUSSION

A. Representation of Data Set

Representation of data set is the first process in our approach. Currently, time interval-based mining problem is much more arduous than the time point-based mining problem. Here two time intervals may overlap, then the relationship among event intervals is more complex than that of the event points. In this paper, two new representations are developed called endpoint representation and endtime representation, to effectively express temporal patterns. My observation says that, the complex relationships among event intervals are the major bottleneck for mining temporal patterns. Endpoint representation utilizes the endpoint arrangements to express the relations among intervals in sequence unambiguously. The time information is also critical for numerous applications. I develop another representation, called endtime representation, which not only expresses relations among intervals but also reveals the occurrence time.

B. Pruning Techniques

The Pruning Techniques is the second module in our process. These are the properties of endpoints; we propose three pruning strategies called scan-pruning, point pruning, and postfix-pruning for efficiently and effectively reducing the searching space. First, to calculate the support of all endpoints in the database, It is unnecessary that scanning each sequence from the beginning to the end. Instead, we only need to scan from the start of each sequence and stop at the first finishing endpoint which has a corresponding starting endpoint in prefix. Because of endpoints which always appear in pairs in a pattern, a frequent sequence will never become a pattern if it has no chance of obtaining all pairs of endpoints in its subsequent growth. This process is called scan pruning. Then the next process is, the starting and finishing endpoints definitely occur in pairs in a sequence. We only need to project the frequent finishing endpoints which have the corresponding starting endpoints in their prefixes. This process is called point pruning, for pruning non-qualified patterns before constructing the projected database when constructing a projected database, some endpoints in the postfix sequences need not be considered we use point pruning. With respect to a prefix sequence, finishing endpoint in a projected postfix sequence is essential if it has corresponding starting items. When constructing the projected database, only the essential endpoints in the postfix sequences are collected. All nonessential items are eliminated because they can be ignored during the discovery of temporal patterns. For eliminating the non essential items we use the last pruning method is called postfix pruning, for projecting a database.

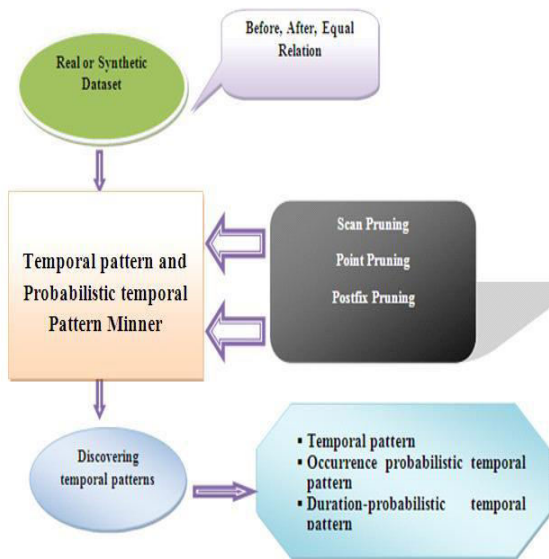


Fig. 1. System Architecture








C. Temporal Pattern and Probabilistic Temporal Pattern Miner Algorithm

The final module in our approach explained as, temporal database, and event intervals associated with the same sequence ID are grouped into an interval sequence. It is first transformed the temporal database into the endpoint representation, and then scans the database to calculate the count of each endpoint concurrently. Then it removes infrequent endpoints below the given support threshold. For each frequent starting endpoint, we build the projected database and it is processed recursively to discover sets of all temporal patterns.

Finally, outputs all temporal patterns. This output is transformed into endtime representation, and then finds frequent endpoints and removes infrequent ones then includes the concept of probabilistic function calculation. Frequent endpoint can be appended to the original prefix to generate a new frequent sequence.

I also use the time information in database to estimate the occurrence- and duration-probability functions. If all endpoints in a frequent endpoint sequence appear in pairs, i.e., every starting (finishing) endpoint has a corresponding finishing (starting) endpoint, we can output this frequent endpoint sequence, including its occurrence and duration probability function, as the occurrence- and duration-probabilistic temporal pattern, respectively.

Table2. Proposed Representations Of Allen's 13 Temporal Relations

temporal relation	inversed relation	pictorial example (s: starting time, f: finishing time)	endpoint representation	endtime representation
<i>A before B</i>	<i>B after A</i>		$A^+ A^- B^+ B^-$	$(A^+ A^- B^+ B^-)$ $(s_a f_a s_b f_b)$
<i>A overlaps B</i>	<i>B overlapped-by A</i>		$A^+ B^+ A^- B^-$	$(A^+ B^+ A^- B^-)$ $(s_a s_b f_a f_b)$
<i>A contains B</i>	<i>B during A</i>		$A^+ B^+ B^- A^-$	$(A^+ B^+ A^- B^-)$ $(s_a s_b f_a f_b)$
<i>A starts B</i>	<i>B started-by A</i>		$(A^+ B^+) B^- A^-$	$(A^+ B^+) B^- A^-$ $(s_a s_b f_b f_a)$
<i>A finished-by B</i>	<i>B finishes A</i>		$A^+ B^+ (A^- B^-)$	$(A^+ B^+ (A^- B^-))$ $(s_a s_b f_a f_b)$
<i>A meets B</i>	<i>B met-by A</i>		$A^+ (A^- B^+) B^-$	$(A^+ (A^- B^+) B^-)$ $(s_a f_a s_b f_b)$
<i>A equal B</i>	<i>B equal A</i>		$(A^+ B^+) (A^- B^-)$	$(A^+ B^+) (A^- B^-)$ $(s_a s_b f_a f_b)$

D. Algorithm Description

Temporal database, event intervals associated with the same sequence ID is grouped into an interval sequence. First transforms the temporal database into the endpoint representation, and then scans the database to calculate the count of each endpoint concurrently. Then removes infrequent endpoints below the given support threshold.

For each frequent starting endpoint, I build the projected database and recursively to discover sets of all temporal patterns. Finally, outputs all temporal patterns. This output is transformed into endtime representation, and then finds frequent endpoints and removes infrequent ones then includes the concept of probabilistic function calculation. Frequent endpoint can be appended to the original prefix to generate a new frequent sequence. I also use the time information in Database to estimate the occurrence- and duration-probability functions. If all

endpoints in a frequent endpoint sequence appear in pairs, i.e., every starting (finishing) endpoint has a corresponding finishing (starting) endpoint, we can output this frequent endpoint sequence, including its occurrence and duration probability function, as the occurrence and duration probabilistic temporal pattern, respectively. This output is transformed into endtime representation, and then finds frequent endpoints and removes infrequent ones then includes the concept of probabilistic function calculation. Frequent endpoint can be appended to the original prefix to generate a new frequent sequence. I also use the time information in Database. In contrast to TPSpan, P-TPSpan includes the concept of probabilistic function calculation. For a prefix, P-TPSpan first calls the procedures `count_support` and `point_pruning` to find all local frequent endpoints. Frequent endpoints s can be appended to the original prefix to generate a new frequent sequence. I also use the time information of s in $DB|$ to estimate the occurrence- and duration-probability functions f_s and g_s .

Then, build the projected database and call P-TPSpan recursively to discover sets of all temporal patterns. Finally, P-TPMiner outputs all occurrence- and duration probabilistic temporal patterns. In contrast to TPSpan, P-TPSpan includes the concept of probabilistic function calculation. For a prefix, P-TPSpan first calls the procedures `count_support` and `point_pruning` to find all local frequent endpoints. Frequent endpoints s can be appended to the original prefix to generate a new frequent sequence. If all endpoints in a frequent endpoint sequence appear in pairs, i.e., every starting (finishing) endpoint has a corresponding finishing (starting) endpoint, we can output this frequent endpoint sequence, including its

occurrence and duration probability function, as the occurrence- and duration-probabilistic temporal pattern, respectively. Finally, I can discover all patterns by constructing the projected database with the frequently extended prefixes and by recursively running P-TPSpan until the prefixes can no longer be extended. Note that, similar to TPSpan, P-TPSpan also utilizes three pruning strategies to effectively reduce the search space.

IV. EXPERIMENTAL RESULTS

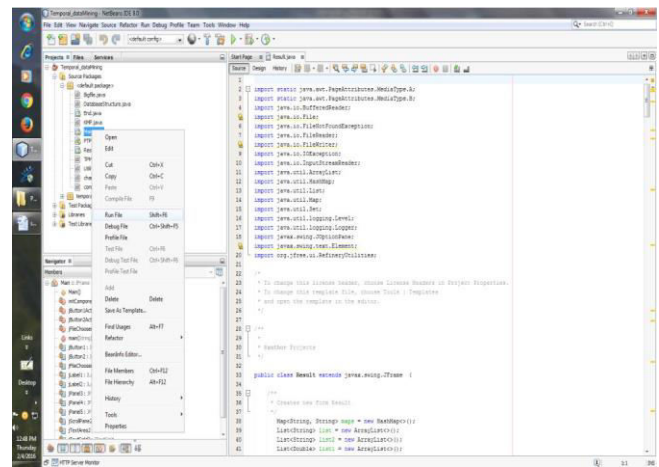


Fig. 2. Run The Project

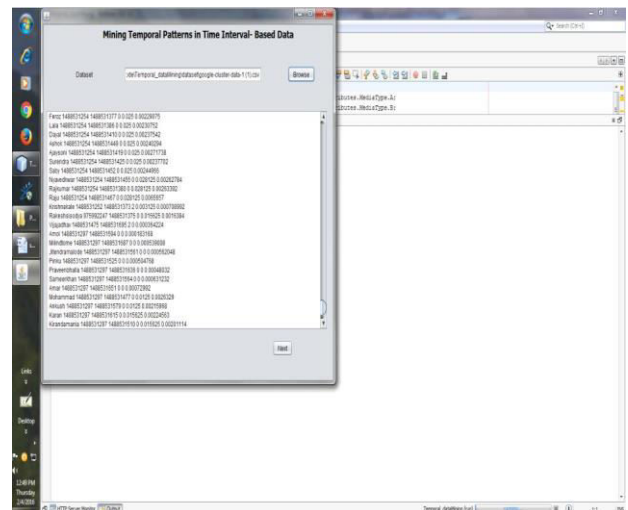


Fig. 3. Browse Datasets

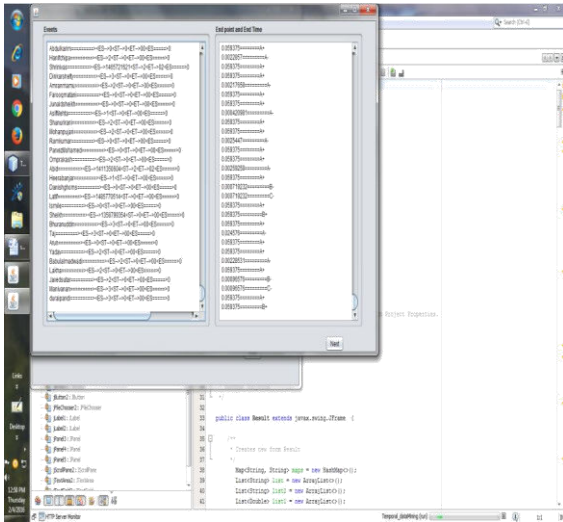


Fig. 4. End Point And End Time Representations

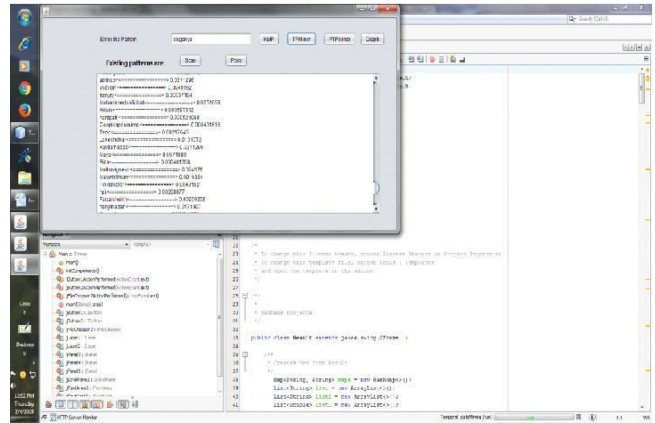


Fig. 7. TP Miner

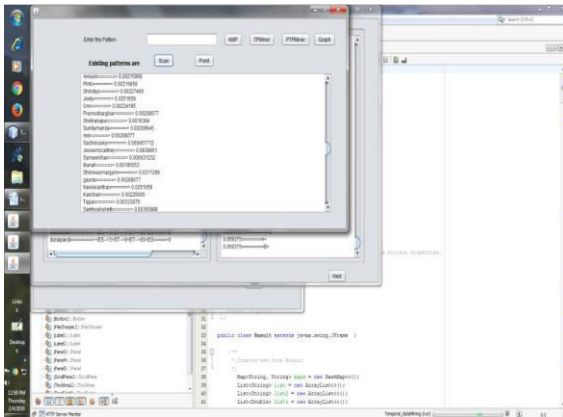


Fig. 5. Scan Pruning

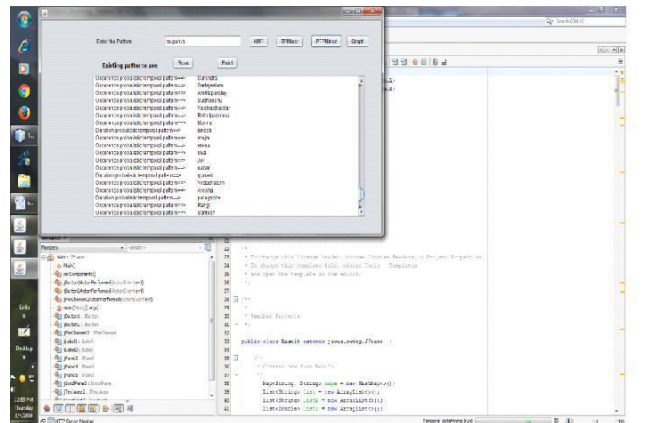


Fig. 8. PTP Miner

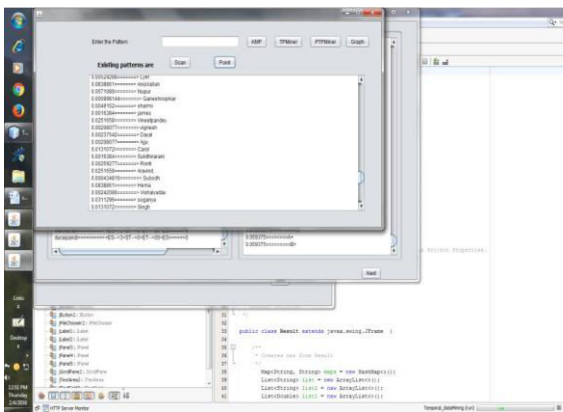


Fig. 6. Point Pruning

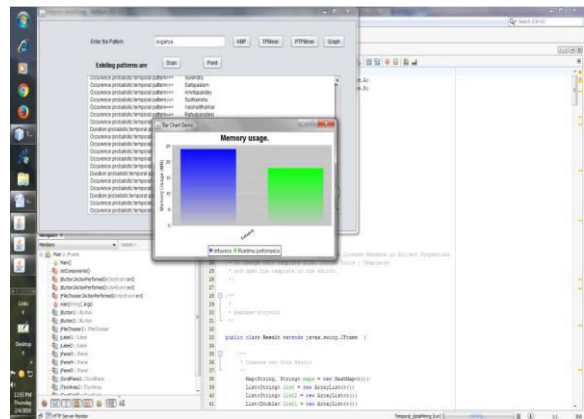


Fig. 9. Runtime Performance

V. CONCLUSION AND FUTURE WORKS

In this paper, for solving complex relations among the intervals we develop two new representations called endpoint representation and endtime representation.

we propose two algorithms TPMiner and P-TPMiner, are developed to efficiently discover three types of patterns: temporal pattern, occurrence - probabilistic temporal pattern, and duration probabilistic temporal pattern, based on the two representations and to describe the correlation among intervals and the probability of the occurring time and duration of each intervals. I also propose several pruning techniques to effectively reduce the search space. The experimental studies indicate that TPMiner and P-TPMiner are efficient and practical. In my future work, we implement the algorithm which is a combination of TPMiner and P-TPMiner called Temporal Pattern and Probabilistic Temporal Pattern Miner Algorithm with real time data sets.

ACKNOWLEDGMENT

It is my privilege to remember here the grace of GOD ALMIGHTY and all those people who have contributed directly or indirectly in the successful execution of my project. I express my sincere thanks and gratitude to our principal

Dr.J.Mohamed Jahabar., for his whole hearted support and help in completing this project. I am taking this opportunity to express my heartfelt gratitude to **Prof.**

R.Karthikeyan Head of the Department for his timely guidance and providing all necessary facilities at the right time for completing this project. Finally I express my deep appreciation to my parents and my friends for their moral support and encouragement throughout the project.

REFERENCES

[1] E. Winarko and J .F Roddick, "ARMADA-An algorithm for discovering richer relative temporal association rules from interval- based data," *Data and Knowledge Engineering*,

vol. 63, issue 1, pp. 76-90, 2007.

[2] A. Wong,D. Zhuang, G. Li, and E. Lee, "Discovery of Closed Patterns and Noninduced Patterns from Sequences," *IEEE Transactions on Knowledge and Data Engineering*, vol.24, no. 8, pp. 1408-1421, 2012.

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

[4] S. Wu and Y. Chen, " Discovering hybrid temporal patterns from sequences consisting of point- and interval-based events," *Data & Knowledge Engineering*, vol.68, issue 11, pp.1309–1330, 2009.

[5] J . Yang, W. Wang, and P . Yu, "InfoMiner: Mining Surprising Periodic Patterns," *Data Mining and Knowledge Discovery*, vol. 9, no. 2, pp. 189-216, 2004.

[6] J . Yang,W. Wang,P . S. Yu, and J . Han, "Mining Long Sequential Patterns in a Noisy Environment," *The 2002 ACM SIGMOD international conference on Management of data (SIGMOD'02)*, pp. 406-417, 2002.

[7] M. Yoshida,T . Iizuka, H. Shiohara and M. Ishiguro, " Mining Sequential Patterns Including Time Intervals," *Proceeding of SPIE - Data Mining and Knowledge Discovery: Theory, Tools, and Technology II* , vol. 4057, pp. 213-220, 2000.

[8] M. Zaki, "SPADE: An Efficient Algorithm for Mining Frequent Sequences," *Machine Learning*, vol. 42, no. 1-2, pp. 31-60, 2001. [35] A. Zakour, S. Maabout, M. Mosbah and M. Sistiaga, "Uncertainty Interval Temporal Sequences Extraction," *International Conference on Information Systems Technology and Management (ICISTM'*

12), pp. 259-270, 2012.

[9] Z . Zhao, D. Yan and W. Ng, “ Mining Probabilistically Frequent Sequential Patterns in Large Uncertain Databases,” *The 15th International Conference on Extending Database Technology (EDBT’ 12)*, pp. 74-85, 2012.

[10] R .Villafane, K. Hua and D. Tran, “Knowledge Discovery from Series of

Interval Events,” *Journal of Intelligent Information Systems*, vol.1



S.Soundariya received the B.E degree in Computer Science And Engineering from SACS M.A.V.M.M Engineering College, Tamilnadu, India, in 2014. Currently, S he is post graduate student in Computer Science And Engineering, Mohamed Sathak Engineering College(MSEC), Tamilnadu, India. Her current research area includes Data Mining and Big Data.