# Efficient Pattern Mining for Wireless Sensor Networks Data

Manisha Rajpoot (Corresponding author)
Department of Computer Science and Engineering
Rungta College of Engineering and Technology, Bhilai, India
Email: manishamtech09@gmail.com

H R Sharma
Department of Computer Science and Engineering
Rungta College of Engineering and Technology, Bhilai, India
Email: hrsharma44@gmail.com

**Abstract**

Wireless Sensor Networks generate a large amount of data in the form of streams. Mining association rules on the sensor data provides useful information for different applications. In this paper, a total from partial (TFP) tree based approach is used to generate the set of all association rules from data. Our experimental results show that TFP techniques perform better result in case of sparse dataset and significantly comparable as SP-tree approach for the dense dataset.

**Keywords:** Association Rule Mining; Wireless Sensor Networks; Frequent Pattern.

## 1. Introduction

Wireless sensor networks have established their success in a variety of real world applications such as battlefields, smart buildings, toxic gas leaks, habitat monitoring [1] [2] [3]. WSN consist of a collection of lightweight (possibly, mobile) sensors with the capabilities of sensing, computing and transmitting. In general, while transmitting the detected events to sink (Possibly, fast) streams of raw sensor reading from every node generate a large amount of data. Wireless sensor datasets represent unique opportunities for not only applying but also advancing the science and engineering behind cutting-edge knowledge discovery techniques. The association rule is an important technique in knowledge discovery for the extracting pattern of data. The association rules for WSNs have received a great attention due to their importance in capturing the temporal relations among sensor nodes. The main objective of the wireless sensor association rules is to capture the temporal relations between sensor nodes based on common intervals of activities. An example of such a rule is ($s_1s_2 \Rightarrow s_3$; 85%, $\lambda$ ) which means that if we receive events from sensors $s_1$ and $s_2$, then there is a 85% chance of receiving an event from sensor $s_3$ within $\lambda$ units of time. The main step in the formation of association rules is to find the patterns of sensors that co-occur together and exceed a certain frequency (these patterns are called frequent association patterns) [1] [2].

Recently, extracting patterns from WNS data has received a great deal of attention by the data mining community. Different approaches for pattern analysis focusing on the data structure of algorithm have been successfully used for WNS data. The The FP-growth mining technique proposed in has been found to be one of the efficient algorithms in mining frequent patterns. The performance gain achieved by the FP-growth is mainly based on the highly compact nature of the frequent pattern tree, where it stores only the frequent items in frequency descending order. Boukerche and Samarah [1] proposed a representation structure, called a positional lexicographic tree (PLT), which is able to compress the sensor data residing in the database. However, construction of such data structures (e.g., FP-tree, PLT) requires two database scans, which is not suitable for generating association rules from the streams of sensor data, Moreover; mining PLT requires an extra mapping mechanism for the sensors to avector. Tanbeer et al. [2] proposed prefix-tree structure called sensor pattern tree (SP-tree) to capture the information and store them in a memory efficient highly compact manner, similar to a FP – tree[6][7].

Most of pattern discovery in WSN data is based on Apriori or FP-Growth framework. The Apriori like algorithms suffer a huge set of candidate sequence generation problem and multiple scans of databases. FP-tree utilizes a large branch of the tree of the sparse data set and it takes similar computation time as Apriori algorithm [4][5]. In this paper a total from partial (TFP) tree approach will be augmented for pattern discovery in WSNs data. It is perform better in both dense and sparse cases.

The remainder of the paper is organized as follows: In Section 2, related work in pattern mining for WSNs data will be reported. Section 3 illustrates the problem of mining association rules in WSNs. In Section 4, we describe TFP pattern mining methodology. The experiment and results analysis will be reported in Section 5 and finally study will be concluded in Section 6.

## 2. Related Work

Baukerche and Samarah [1] have proposed sensor association rules in which the event-detecting sensors are the main objects of the rules regardless of their values. To store the sensor's event-detecting status, this method uses a representation, called a positional lexicographic tree (PLT), constructed in lexicographic order of sensors. Each sensor maps to a unique integer, so that the lexicographic order is preserved. The first step in constructing the PLT is to scan the database once to obtain the set of frequent event–detecting sensors. The set of sensors that detect event at the same unit of time is processed together as an entry in the PLT. With the second database scan, the frequent sensors on each of such set are transformed into a position vector constructed by mapping the lexicographic distance between a sensor's identifier, and its parent's identifier, and then the vector is inserted into the PLT. This approach constructs as many PLTs as the number of distinct last sensors of all position vectors. The construction of PLTs terminates when all the position vectors from the database are inserted into respective PLT structures [7].

Similar to the FP-growth approach, PLT follows a pattern growth mining technique. The mining begins with the sensor having the maximum rank by generating the frequent patterns from its PLT in a recursive way. A conditional vector considering only the prefix part in the PLT for the sensor (under mining) is constructed. At this stage, the PLT of the conditional vector, if available, is also updated. For all of the sensors present in PLT structures, the mining process is the same. The computation required at each recursion to update the PLT involved in the prefix part of a pattern is not trivial. Therefore, the two database scans requirement and the additional PLT update operations during mining limit the efficient use of this approach in handling WSN data.

Tanbeer et al. [2] proposed a tree-based data structure called sensor pattern tree (SP-tree) to generate the set of all association rules from WSN data with one scan over the sensor database. The SP-tree is constructed in frequency-descending order, which facilitates an efficient mining using the frequent pattern (FP)-growth-based mining technique. Rashid et al. [8] developed a single pass tree structure that can capture important knowledge from the stream contents of sensor data in compact manner. This work focused on regularly frequent sensor pattern.

## 3. Association Rule Mining Problem in WSNs

Let S = {$s_1$, $s_2$, …, $s_n$) be a set of sensors, in a particular sensor network. It is assumed that the time space is divided into equally sized slots {t1, t2,.., tm} such that ti+1 – ti = $\lambda$, i $\in$[1, m-1] and $\lambda$ is the size of each time slot. A set P = {s1, s2, …, sk} $\subseteq$ S is called a pattern of sensors. A sensor database, SD, is defined to be a set of epochs in which each epoch is coupled E(Ets, X) such that X is a pattern of event detecting sensors that report events within the same time slot. Ets is the epoch's time slot. Let size (E) be the size of E, i.e., the number of sensors in X. We say an epoch E(Ets, X) supports a pattern X' if X'$\subseteq$X. The frequency of the pattern X' is SD is defined to be the number of epochs in SD that support it, i.e. Freq(X', SD) = |{E(Ets, X)|X'$\subseteq$X}|.

Pattern X' is said to be a frequent pattern if Freq (X', SD) $\geq$ min_sup, where min_sup is a user-given minimum support threshold (i.e., frequency) in percentage of SD size in number of epochs. Let FSD be the set of all frequent event, detecting sensor patterns is SD for a given min_sup.

Sensor association rules are implications of the form of X' $\Rightarrow$X'' where X' $\subset$ S, X'' $\subset$ S and X' $\cap$ X'' =$\varnothing$. The frequency of the rule X' $\Rightarrow$X'' is the frequency of the pattern (X' $\cup$ X''). The confidence of the rule is defined as Conf (X' $\Rightarrow$X'') = Freq ((X' $\cup$ X''), SD)/Freq (X', SD).

A rule is interesting if its frequency and confidence is greater than or equal to min_sup and the user given minimum confidence, min_conf, respectively. The problem of mining sensor rules, given an SD, a min_sum and a min_conf, is the problem of extracting all interesting association rules present on the SD [2].

## 4. Proposed Method

TFP structure an algorithm, which completes the summation of the final support counts, storing the results in a second set-enumeration tree (the T-tree, of Total support counts), ordered in the opposite way to the P-tree. The T-tree finally contains all frequent sets with their complete support counts [4][5][12]. This algorithm augmented and implemented for Wireless Sensor Networks. The methodology can be summarized as follow:

Step I. Preprocess WSNs data. It is accomplished by the execution of a spatial query. All the task relevant objects are collected into one database.

Step II. Arrange the data ascending order on the basis of time and identify stopping point with respect to input threshold.

Step III. In this step, the concept of partial support counting using the “P-tree” (Partial support tree) is used. The idea is to copy the input data (in one pass) into a data structure, which maintains all the relevant aspects of the input, and then mine this structure. A P-tree is a set enumeration tree structure in which to store partial counts for item sets. The top, single attribute, level comprises an array of references to structures of the form shown to the right, one for each column. Each of these top-level structures is then the root of a sub-tree of the overall P-tree.

The advantages offered by the P-tree table are [4]:

1. Reduced storage requirements (particularly where the data set contained duplicate rows).

2. Faster run times because the desired total support counts had already been partially calculated.

Step IV. In this step the P-tree is examined and creates T-tree [4]. The T-tree is generated in an Apriori manner. There are a number of features of the P-tree table that enhances the efficiency of this process:

1. The first pass of the P-tree will be to calculate supports for singletons and thus the entire P-tree must be traversed. However, on the second pass when calculating the support for "doubles" we can ignore the top level in the P tree, i.e. we can start processing from index 2. Further, at the end of the previous pass, we can delete the top level (cardinality = 1) part of the table. Consequently, as the T-tree grows in size the P-tree table shrinks.

2. To prevent double counting, on the first pass of the P-tree, we update only those elements in the top-level array of the T-tree that correspond to the column numbers in node codes (not parent codes). On the second pass, for each P-tree table record found, we consider only those branches in the T-tree that emanate from a top level element corresponding to a column number represented by the node code (not the parent code). Once the appropriate branch has been located we proceed down to level 2 and update those elements that correspond to the column numbers in the union of the parent and node codes. Repeat this process for all subsequent levels until there are no more levels in the T-tree to consider.

## 5. Experiment and Result

The experiments were conducted with the datasets synthetic data (T10I4D100K) generated from [10] and real datasets (kosarak, chess and connect-4) [11], earlier also used to compare the performance of pattern mining in wireless sensor network by other author [2]. In this experiments were performed in both dense and sparse data sets. Table 1 shows the characteristics of these datasets. T10I4D100K and kosarak are sparse and chess and connect-4 are dense data set [2].

Table 1. Characteristics of Dataset

| Dataset | Transaction | Items | Type |
|---|---|---|---|
| Chess | 3196 | 75 | Dense |
| Connect-4 | 67557 | 129 | Dense |
| Kosarak | 990002 | 41270 | Sparse |
| T10I4D100K | 100000 | 870 | Sparse |

We compared the performance of proposed total from partial tree approach with SP-tree and PLT. The algorithms were implemented in Java program language and run with Windows 2007 on a 3.20 GHz CPU and 4 GB memory. The experiments were conducted with constant confidence value 80% with variable minimum support values. The performance of algorithms with different data sets is shown through Fig. 1-4. The performance time included the tree creation, generation and association rule generation times.

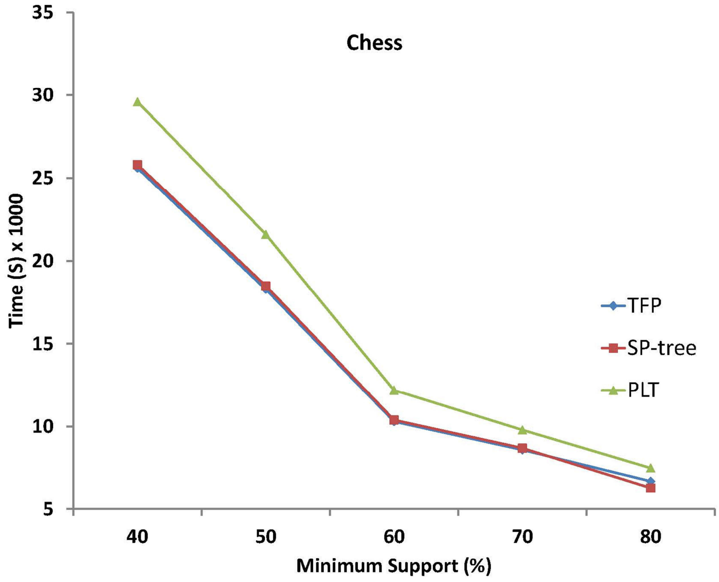

Figure 1.Comparison among TFP, SP-Tree, PLT with data set chess.

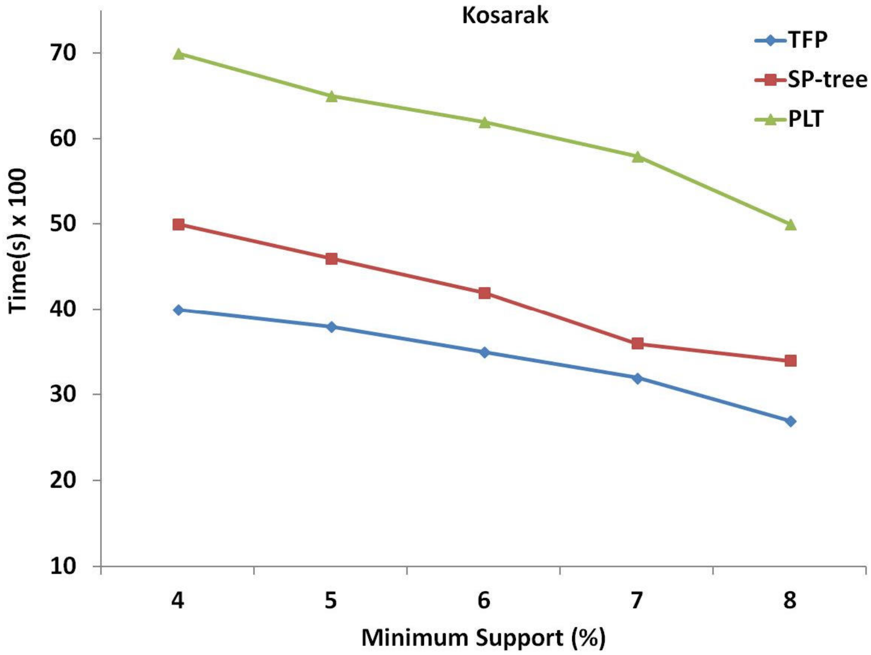

Figure 2. Comparison among TFP, SP-Tree, PLT with data set connect-4

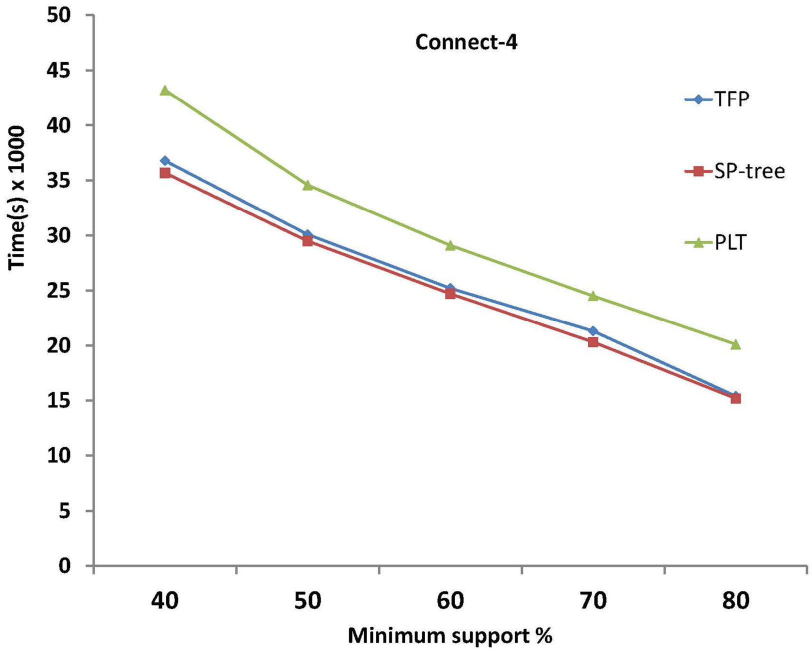

Figure 3. Comparison among TFP, SP-Tree, PLT with data set kosarak.

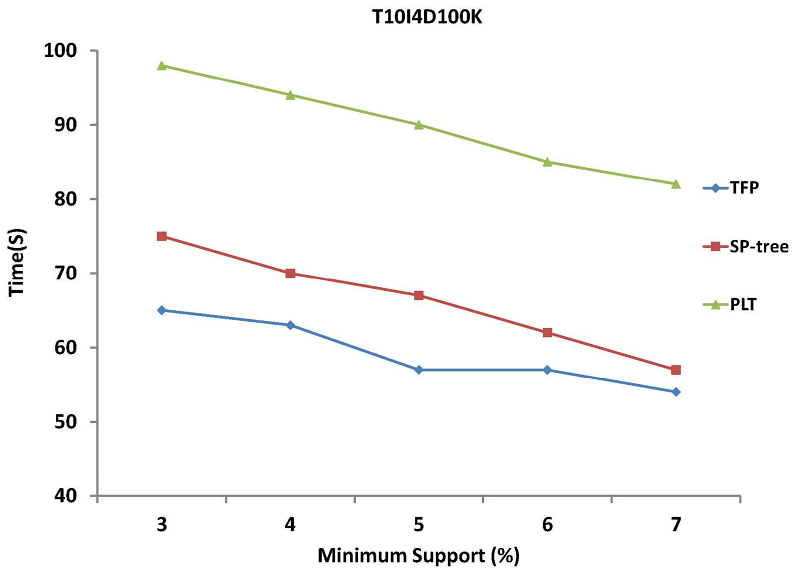

Figure 4. Comparison among TFP, SP-Tree, PLT with T10I4D100k.

It was observed that the performance of TFP and SP-Tree is significantly similar for dense datasets chess and connect. Both algorithms perform better than the PLT algorithm for the dense dataset. In case of sparse datasets, the TFP technique performs comparable better than SP-tree and PLT method.

## 6. Conclusion

In this paper, the feasibility of TFP technique for the pattern discovery in wireless sensor network was investigated. The performances of algorithm were tested for sparse and dense data. It was observed that TFP algorithm achieve good performance in both sparse and dense dataset.

## References

[1] A Boukerche, and S Samarah. “A Novel Algorithm for Mining Association Rules in Wireless Ad Hoc Sensor Networks”, IEEE Transactions on Parallel and Distributed Systems, vol. 19, no. 7, pp. 865-77, 2008.

[2] SK Tanbeer, CF Ahmed and BS Jeong (2009)"An Efficient Single-Pass Algorithm for Mining Association Rules from Wireless Sensor Networks, IETE Technical Review, Vol 26 (4), 2009, pp. 280-289.

[3] A Kongu et al. "Wireless Sesor Networks Fault Identification using Data Association", Journal of Computer Science 8(9), 2012, pp. 1501-1505.

[4] AK Akasapu, LK Sharma and G Ramakrishna, Efficient Trajectory Pattern Mining for both Sparse and Dense Dataset. International Journal of Computer Applications 9(5), 2010, pp.45–48.

[5] F Coenen, P Leng and S Ahmed, “Data Structure for Association Rule Mining: T-Trees and P-Trees”, IEEE Transactions on Knowledge And Data Engineering, 16(6) 2004, pp. 774-778.

[6] K. Roemer, “Distributed Mining of Spatio-Temporal Event Patterns in Sensor Networks,” Proc. of EAWMS ’06, 2006.

[7] M Rapoot and LK Sharma, “Comparative Study of Association Rule Mining for Sensor Data”, International Journal of Computer Applications 19(1), 2011, pp. 34-36.

[8] M Rashid, I Gondal, J Kamruzzaman, Regularly Frequent Patterns Mining from Sensor Data Stream, Neural Information Processing LNCS 8227, 2013, pp 417-424

[9] A Mahmood et al. "Data Mining Techniques for Wireless Sensor Networks: A Survey, Internation Journal of Distributed Sensor Networks, Volume 2013, Article ID 406316, 24 pages.

[10] IBM. QUEST Data Mining Project, http://www.almaden.ibm.com/ cs/quest.

[11] CL Blake, and CJ Merz. UCI repository of machine learning databases, University of California – Irvine, Irvine, CA, 1998.

[12] K Verma, OP Vyas, and R Vyas, "Temporal approach to association rule mining using t-tree and p-tree", Springer Machine Learning and Data Mining in Pattern Recognition, 2005, pp. 651-659.