On Performance Evaluation of Mining Algorithm for Multiple-Level Association Rules based on Scale-up Characteristics

Suraj Srivastava, Harsh K. Verma and Deepti Gupta
Department of Computer Science and Engineering,
National Institute of Technology, Jalandhar, Punjab, India
Email: surajsriengg@gmail.com, vermah@nitj.ac.in and deepti_gupta49@yahoo.co.in

***Abstract*— Various methods for mining association rules at multiple conceptual levels focusing on different sets of data and applying different thresholds at different levels have been proposed in literature. These are ML_T2L1, ML_T1LA, ML_TML1, and ML_T2LA. It has been observed that these algorithms show higher processing time and processing cost as well as need large amount of memory space. This paper focuses on the comparative performance evaluation of the ML_TMLA algorithm that generates multiple transaction tables for all levels in one database scan with that of ML_T2L1 and ML_T1LA algorithms. The performance study has been conducted on different kinds of data distributions (three synthetic and one real dataset) and thresholds, which identify the conditions for algorithm selection. The Tool used for the experimental and comparative evaluation of the proposed algorithm with other algorithms is the AR Tool. It has been concluded that the ML_TMLA algorithm performs better than all the algorithms mentioned above.**

***Index Terms*— Data mining, Knowledge discovery in databases, Association rules, multiple-level association rules**

I. Introduction

The applications of computers, database technologies and automated data collection techniques require large amount of data to be stored into databases. It, thus, becomes necessary to analyze this data and turn it into useful knowledge. Data mining or Knowledge Discovery in Database (KDD) emerges as a solution to the data analysis problem. One of the data mining techniques that is used to discover interesting rules or relationships among attributes in databases is the Association rules. These rules help in discovering knowledge at multiple conceptual levels, which, in turn, provide a spectrum of understanding, from general to specific, for the underlying data. Mining association rules from large data sets has been a focused topic in recent research into knowledge discovery in databases [1, 2, 3, 4, 5, and 6].

It has been observed that the recent advances in data warehousing and OLAP technology it is a practice to arrange data at multiple levels of abstraction [7]. Therefore, the main focus of this study is exploration of efficient methods for multiple-level rule mining. There are various ways to explore efficient mining of multiple-level association rules. One possibility is the direct application of the existing single-level association rule mining methods to multiple-level association mining. One may apply the Apriori algorithm [2] to examine data items at multiple levels of abstraction under the same minimum support and minimum confidence thresholds. Second choice is the application of different minimum support thresholds and possibly different minimum confidence thresholds as well as mining associations at different levels of abstraction. This leads to mining interesting association rules at multiple concept levels, which may not only discover rules at different levels, but may also have high potential to find nontrivial, informative association rules because of its flexibility for focusing the attention to different sets of data and applying different thresholds at different levels.

When a single support threshold is used it allows many uninteresting rules to be generated together with the interesting ones if the threshold is rather low, but disallows many interesting rules to be generated at low levels if the threshold is rather high. Therefore, in their study, substantial efforts have been made on how to identify and remove the redundant rules across different levels.

The discovery of association rules constitutes a very important task in the process of data mining. The idea of discovering such rules is derived from market basket analysis where the goal is to mine patterns describing the customer's purchase behavior [12].

The problem of mining association rules can be stated as follows: I={i1 , i2 , ... , im } is a set of items, T={t1 , t2 , ... , tn } is a set of transactions, each of which contains items of the itemset I . Thus, each transaction ti is a set of items such that ti subset and equal I. An association rule is an implication of the form: X →Y, where X subset I, Ysubset I and X ∩Y= Ø. X (or Y) is a set of items, called itemset [8].

An example for a simple association rule would be {*bread*} → {*butter*}. This rule says that if bread was in a transaction, butter was in most cases in that transaction too. In other words, people who buy bread often buy butter as well. Such a rule is based on observations of the customer behavior and is a result from the data stored in transaction databases.

doi:10.4304/jait.2.4.234-238

Looking at an association rule of the form $X \rightarrow Y$, X would be called the antecedent, Y the consequent. It is obvious that the value of the antecedent implies the value of the consequent. The antecedent, also called the "*left hand side*" of a rule, can consist either of a single item or of a whole set of items. This applies for the consequent, also called the "right hand side", as well.

The most complex task of the association rule mining process is the generation of frequent itemsets. Many different combinations of items have to be explored which can be a very computation-intensive task, especially in large databases. As most of the business databases are very large, the need for efficient algorithms that can extract itemsets in a reasonable amount of time is high. Often, a compromise has to be made between discovering all itemsets and computation time. Generally, only those itemsets that fulfill a certain support requirement are taken into consideration. Support and confidence are the two most important quality measures for evaluating the interestingness of a rule as described.

To study the mining of association rules from a large set of transaction data, it has been assumed that the database contains (1) a transaction data set, T, which consists of a set of transactions (T_i, {A_p, . . . , A_q}), where T_i is a transaction identifier, A_i belongs to T (for i = p, . . . , q), and T is the set of all the data items in the item data set; and (2) the description of the item data set, D, which contains the description of each item in T in the form of (A_i, description i), where A_i belongs to T.

The necessity for mining multiple-level association rules or using taxonomy information at mining association rules has also been observed by other researchers such as in [9]. A major difference between this study and theirs is that they use the same support threshold across all the levels [9], whereas we have used different support thresholds for different levels of abstraction and different datasets (three synthetic & one real data set).

The paper is organized as follows: Section II discusses the methods for mining multiple-level association rules in depth. Section III evaluates the performance of the mining algorithms comparatively and reports the results obtained. Section IV concludes the paper.

II. Multiple-Level Association Rules Mining Methods

Methods for mining multiple-level association rules use a hierarchy-information encoded transaction table in iterative data mining.

A. *ML_T2L1 Algorithm*

Input: (1) T[l], a hierarchy-information-encoded and task-relevant set of a transaction database, in the format of (TID, Itemset), in which each item in the Itemset contains encoded conceptual hierarchy information, and (2) the minimum support threshold (minsup[*l*]) for each conceptual level *l*.

Output: Multiple-level large itemsets.

Method: A top-down, progressively deepening process which collects large itemsets at different conceptual levels as follows. Starting at level 1, derive for each level *l*, the large k-items sets, L[*l* ,k] for each k, and the set of large itemsets, LL[*l*] (for all k's), as follows:

```
(1) for (l:= 1; L[l, 1] ≠ Ø and l < max_level; l+ + ) do {
(2)     if l = 1 then {
(3)       L[l ,1] := get_large_l_itemsets(T[1], l );
(4)       T[2] := get_filtered_t_table(T[1], L[1,1]);
(5)     }
(6)     else L[l, 1] := get_large_1_itemsets(T[2],l);
(7)     for (k := 2; L[l, k – 1] ≠ Ø; k++) do {
(8)       Ck := get_candidate_set(L[l ,k – 1] );
(9)       foreach transaction t ЄT[2] do {
(10)        ct := gets_subbsets(Ck,t, );
(11)        foreach candidate cЄCt do c.support ++;
(12)      }
(13)      L[l ,k] := {c ε Ck  | c.support ≥ minsup[l] )
(14)  }
(15) LL [l ] := U_k L[l, k];
```

After finding the frequent itemsets, the set of association rules for each level l can be derived from the frequent itemsets LL [l]. based on the minimum confidence at this level, minconf[*l*]., as in [3]. Potential performance improvements of Algorithm ML_T2L1 have been considered by exploration of the sharing of data structures, and intermediate results and maximal generation of results at each database scan, etc. Generation for k > 1 has been performed on T [2], which may consist of much fewer items than T [1], the algorithm could be a potentially efficient one[8][9].

Performance improvements of Algorithm ML_T2Ll have been considered by exploration of the sharing of data structures and intermediate results and maximally generation of results at each database scan, etc. which leads to the following variations of the algorithm [10][11]:

ML_TlLA: using only one encoded transaction table (thus Tl) and generating L[*I*, l] for all the levels at one database scan (thus LA).

ML_TMLl: using multiple encoded transaction tables and generating L [*I*, l] for one corresponding concept level.

ML_T2LA: using two encoded transaction tables (T[1] and T[2]) and generating L[*I*, l] for all the levels at one database scan.

C. *Algorithm ML_TMLA*

INPUT: (1) T[1], a hierarchy information-encoded and task-relevant set of a transaction database, in the format of <TID, Itemset>, in which each item in the Itemset contains encoded conceptual hierarchy information, and (2) the minimum support threshold (minsup[l]) for each conceptual level l.

OUTPUT: Multiple-level large itemsets.

The procedure is described as follows:

(1) {L[1,1],……..L[max_*l*,1]} := get_all_large_1_ itemsets (T[*l*]);

(2) {T[*l*+1],L[*l*+1,1]} := get_filtered_T_table (T[*l*],L[*l*,1]);
(3) for (k := 2; L[*l*,k-1] ≠ Ø; k++) do begin
(4) for (*l* := 1; *l* < max_l; *l*++) do
(5) if L[*l*,k-1] ≠ Ø then begin
(6) C[*l*] := get_candidate_set (L[*l*, k-1]);
(7) foreach transaction t**Є**T[*l*+1] do begin
(8) D[*l*] := get_subsets (C[*l*], t); // Candidates contained in t
(9) foreach candidate c **Є** D[*l*] do c.support ++;
(10) end
(11) L[*l*,k] := { c**Є**C[*l*,*k*] | c.support ≥ minsup[*l*]}
(12) end
(13) end
(14) for (*l*:= 1; *l* < max_ *l* ; *l* ++) do LL[*l*] =U_k [*l*,k];

According to Algorithm ML_TMLA, the discovery of large support items at each level proceeds as follows.

Step 1: At the first scan of T[*l*], large 1-itemsets L [*l*, 1] for every level *l* can be generated in parallel, because the scan of an item *i* in each transaction *t* may increase the count of the item in every L[*l*,1] if its has not been incremented by t. After the scanning of T[*l*], each item in L [*l*,1] whose parent (if *l* > 1) is not a large item in the higher level large 1-itemsets or whose support is lower than minsup[*l*] will be removed from L[*l*,1].

Step 2: The first scan of T[*l*] generates the large 1-itemsets L [*l*, 1] which then serves as a filter to filter out from T[*l*] any small items or transactions containing only small items. A new table T [*l*+1] results from this filtering process and is used in the generation of large k-itemsets at level *l*. The filtered transaction table T[*l*+1] is derived by "get_filtered_t_table (T[*l*], L[*l*,1])", which uses L[*l*, 1] as a filter to filter out (a) any item which is not large at level 1, and (b) the transactions which contain no large items.

Step 3: T [*l* + 1] is generated at the processing of each level ***l*,** for *l* > 1. This is done by scanning T[*l*] to generate the large 1-itemsets L [*l*,1] which serves as a filter to remove from T[*l*] any small items or transactions containing only small items and results in T[*l*+ 1], which will be used for the generation of large k-itemsets (for k > 1) at level *l* and table T [*l* + 2] at the next lower level.

Step 4: After the generation of large 1-itemsets for each level *l*, the candidate set for large 2-itemsets for each level *l* can be generated by the apriori-gen algorithm [5]. The get_subsets function will be processed against the candidate sets at all the levels at the same time by scanning T[*l*] once, which calculates the support for each candidate itemset and generates large 2-itemsets L[*l*,2]. Similar processes can be processed for step-by-step generation of large k-item-sets L [*l*, k] for k > 2.

Step 5: For each transaction t in T[2].for each of t's K-item subset c, increment c's support count if c is in the candidate set C[*l*,k]. Then collect into L[*l*,k] each c(together with its support) if its support is no less than minsup[*l*].

Step 6: The large itemsets at level *l*, LL [*l*], is the union of L [*l*, k] for all the k's.

After finding the large itemsets, the set of association rules for each level *l* can be derived from the large itemsets LL [*l*] based on the minimum confidence at this level, *minconf* [*l*]. This is performed as follows [12]. For every large itemset r, if a is a nonempty subset of r, the rule "a → r - a" is inserted into rule_set [*l*] if support(r)/support(a) ≥ *minconf* [*l*], where *minconf* [*l*] is the minimum confidence at level *l*.

III. Results And Discussions

The comparative performance evaluation has been done on different kinds of data distributions (three synthetic and one real dataset) and thresholds, which identify the conditions for algorithm selection [13]. The AR Tool has been used to generate the results and perform the comparative investigations of ML_T1LA, ML_TML1 and MLTMLA algorithms.

The performance of the different multiple-level association rule mining algorithms has been experimentally evaluated, in the context of scale-up (number of transactions (thousands) and transaction size) on different datasets. Four different datasets, one real and three synthetic have been used in the performance comparison of the above mentioned algorithms. The synthetic datasets used have been generated using the AR tool. Table 1 summarizes the names and parameter settings for each dataset. For the synthetic datasets N (Number of items) was set to 1000 and |L| (Number of maximal potentially large itemsets) was set to 2000. We chose three values for |T|: 5, 10, and 20. We also chose three values for |I|: 2, 4, and 6. The number of transactions was set to 100,000. The real dataset used in our experiments was MUSHROOMS (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mushroom/agaricus-lepiota.data).

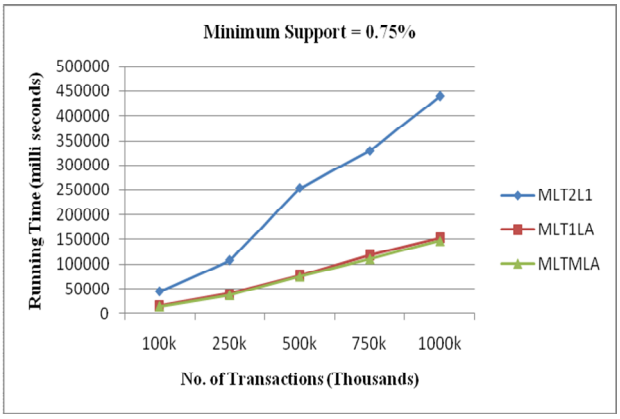
TABLE I.
Parameter Settings Of Datasets

Name	# of objects	Average Size	# of items
T100000 AT10 I1000 P2000 AP 4dB	100k	10	1000
T100000 AT20 I1000 P2000 AP 6dB	100k	20	1000
T100000 AT5 I1000 P2000 AP 2dB	100k	5	1000
MUSHROOMS	8416	23	128

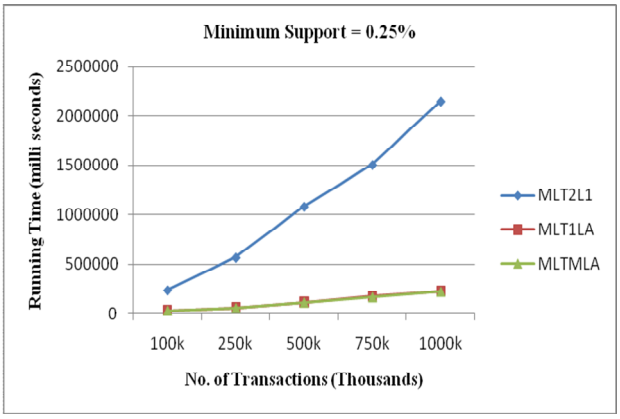
A. Experimental Results-Scale-up

Fig.1 depicts the performance comparison of ML_T2L1, ML_T1LA, and ML_TMLA algorithms with respect to scale-up characteristics i.e., number of transactions under fixed support levels 0.75 and 0.25% in case of dataset T100000 AT5 I1000 P2000 AP 2dB. It has been concluded that the initial increase in the running time is due to increase in the size of the global candidate set. However, the size of the global candidate set does not increase correspondingly as more and more local large itemsets are common. The execution time is relatively

linear from 750,000 transactions to 1000,000 transactions for the ML_TMLA algorithm when the minimum support threshold is fixed as 0.25%.



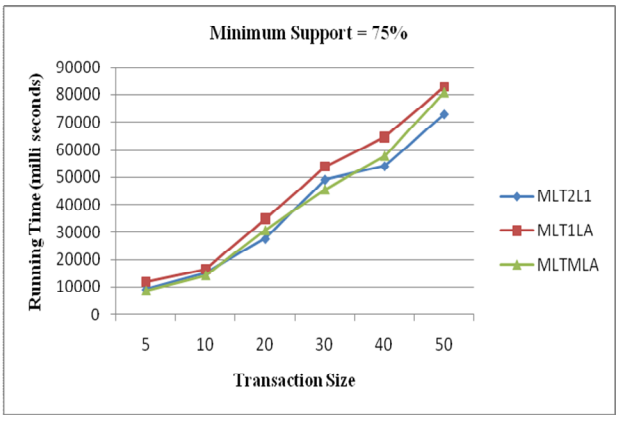
(a)



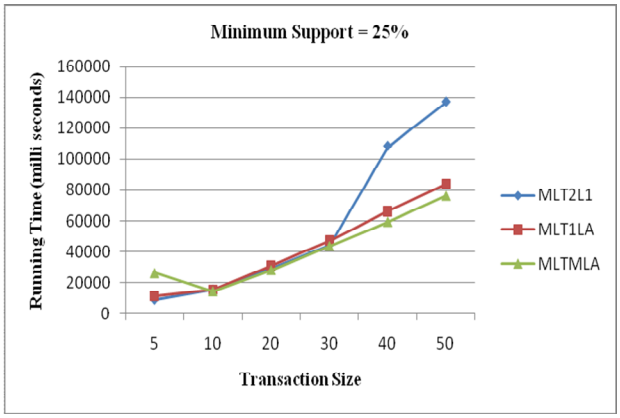
(b)

Fig.1 Scale-up (Number of Transactions) comparison for Minimum Support (a) 0.75% (b) 0.25%

Fig.2 indicates the performance comparison of ML_T2L1, ML_T1LA, and ML_TMLA algorithms with respect to scale-up characteristics i.e., transaction size under various support levels in case of dataset T100000 AT5 I1000 P2000 AP 2dB.



(a)



(b)

Fig.2 Scale-up (Transaction Size) comparison for Minimum Support (a) 75% (b) 25%

It has been noticed that the ML_TMLA algorithm exhibits marginally inferior scale-up as compared to ML_T2L1 algorithm when the minimum support is high (75%). This is because the ML_TMLA algorithm spends more and more time initializing the data structures without deriving much benefit in processing cost. However, for lower minimum support (i.e., high processing cost) of 25%, the scale-up of the ML_TMLA is superior to that of both the ML_T2L1 and the ML_T1LA algorithms because the processing cost increases slower than that of either ML_T2L1 or ML_T1LA algorithm.

C. Cross-Level Assoociation Rule

Fig.3 shows the performance comparison of ML_T1LA, MLT1LA-C, ML_TMLA, and MLTMLA-C based on varying the minimum support from 1% to 5% in case of dataset T100000 AT5 I1000 P2000 AP 2dB. Here ML_T1LA-C and ML_TMLA-C are the revised versions of ML_T1LA and ML_TMLA respectively that find cross-level rules.

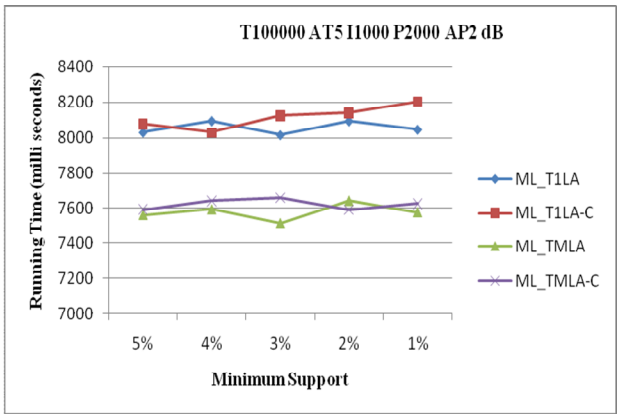


Fig.3 Execution Time comparison for Minimum Support varying from 1% to 5%

It has been observed that the ML_T1LA-C and ML_TMLA-C algorithms for mining cross-level association rules have higher execution time than the ML_T1LA and ML_TMLA algorithms respectively for minimum supports from 1% to 5%. This is found to be so as there are many more frequent itemsets at high levels and the support computations more complex.

IV. Conclusion

This paper focuses on the comparative performance evaluation of the ML_TMLA algorithm that generates multiple transaction tables for all levels in one database scan with that of ML_T2L1 and ML_T1LA algorithms. The algorithm has been evaluated on the basis of scale-up parameter, that is, number of transactions and transaction size. It has been observed that the ML_TMLA algorithm exhibits marginally inferior scale-up as compared to ML_T2L1 algorithm when the minimum support is high (75%). In case of lower minimum support of 25%, the scale-up of the ML_TMLA is superior to that of both the ML_T2L1 and the ML_T1LA algorithms. The execution time is relatively linear from 750,000 transactions to 1000,000 transactions for the ML_TMLA algorithm with the minimum support threshold fixed at 0.25%. Further, it has been noticed that the ML_T1LA-C and ML_TMLA-C algorithms for mining cross-level association rules have higher execution time than the ML_T1LA and ML_TMLA algorithms respectively for minimum supports from 1% to 5%. It has, thus, been concluded that the proposed algorithm executes fast and shows better scale-up characteristics.

References

[1] Agrawal, R., Imielinski, T., and Swami, A., "Mining association rules between sets of items in large databases", *in Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data*, pp. 207-216, Washington, D.C., May 1993.

[2] Agrawal, R. and Srikant, R., "Fast algorithms for mining association rules", *in Proc. 1994 Int. Conf. Very Large Data Bases*, pp. 487-499, Santiago, Chile, September 1994.

[3] Agrawal, R. and R Srikant, , "Mining sequential patterns", *in Proc. 1995 Int. Conf. Data Engineering*, pp. 3-14, Taipei, Taiwan, March 1995.

[4] Klemettinen, M., Mannila, H. and Ronkainen, P., Toivonen, H., and Verkamo, A. I., "Finding interesting rules from large sets of discovered association rules", *in Proc. 3rd Int '1 Conf. on Information and Knowledge Management*, pp. 401-408, Gaithersburg, Maryland, Nov. 1994.

[5] Park, J.S., Chen, M.S. and Yu, P.S., "An effective hash-based algorithm for mining association rules", *in Proc. 1995 ACM-SIGMOD Int. Conf. Management of Data*, pp. 175-186, San Jose, CA, May 1995.

[6] Piatetsky-Shapiro, G., "Discovery, analysis, and presentation of strong rules", *in Knowledge Discovery in Databases*, pp. 229-238, AAAIIMIT Press, 1991.

[7] Chaudhuri, S. ,and Dayal, U., "An Overview of Data Warehousing and OLAP Technology", *ACM SIGMOD Record*, vol. 26, pp. 65±74, 1997.

[8] Liu, Bing, "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data", Springer, 2007.

[9] Agrawal R. ,Srikant R., "Mining Generalized Association Rules", *Proc. 1995 Int'l Conf. Very Large Data Bases*, pp. 407,419, Zurich, Sept. 1995.

[10] Han Jiwawei,Fu Yongjian, "Mining Multiple-Level Association Rules in Large Databases", *IEEE*, 1999.

[11] Han, Jiawei and Yongjian, Fu, "Discovery of Multiple-Level Association Rules from Large Databases", *Proceedings of the 21st VLDB Conference*, Zurich, Swizerland, 1995.

[12] Han, Jiawei, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ISBN 1558609016, 2005.

[13] Wasilewska Anita, "Mining Association Rules in Large Databases", spring, 2007.

Suraj Srivastava received his BTECH in Computer Science and Engineering from Prasad Institute of Technology Jaunpur, Uttar Pradesh, India in 2007 and MTECH in Computer Science and Engineering from Dr B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India in the year 2009. His MTECH thesis was on "Performance Evaluation of Mining Algorithm for Multiple-Level Association Rules". He is currently working as Assistant Professor in the Department of Computer Science and Engineering, V.B.S. Purvachal University Jaunpur, Uttar Pradesh, India. His professional research activity lies in the field of Databases, Data mining and Information Security.

Harsh K. Verma received his PhD degree in Computer Science and Engineering from Punjab Technical University, Jalandhar and Master's degree from Birla Institute of Technology, Pilani. He is presently working as Associate Professor in the Department of Computer Science and Engineering at Dr B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India. He has published more than 20 research papers in various Journals and Conferences of International repute. His teaching and research activities include Scientific Computing, Information Security, Soft Computing and Software Engineering.

Deepti Gupta received her BE in Computer Science and Engineering from University of Jammu, Jammu and Kashmir, India in 2006 and MTECH in Computer Science and Engineering from Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India in the year 2009. Her MTECH thesis was on "Performance Evaluation of Routing Protocols for Wireless Sensor Networks with Different Radio Models". She is currently pursuing full-time PhD in the Department of Computer Science and Engineering, Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India. Her professional research activity lies in the field of wireless sensor networks.