ISSN: 2321-7782 (Online)
Volume 3, Issue 3, March 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Survey on Secure Mining of Association Rules in Horizontally Distributed Databases

Maheshkumar Ramrao Gangasagare[1]
Student of Master of Engineering
C.S.E. Department,
M.P.G.I., School of Engineering,
Nanded - India

Rafik Juber Thekiya[2]
Under The Guidance of
Assistant Professor, C.S.E. Department,
M.P.G.I., School of Engineering,
Nanded - India

Abstract: ***We propose a protocol for secure mining of association rules in horizontally distributed databases. The current leading protocol is that of Kantarcioglu and Clifton [1]. Our protocol, like theirs, is based on the Fast Distributed Mining (FDM) algorithm of Cheung et al. [2], which is an unsecured distributed version of the Apriori algorithm. The main ingredients in our protocol are two novel secure multi-party algorithms—one that computes the union of private subsets that each of the interacting players hold, and another that tests the inclusion of an element held by one player in a subset held by another. Our protocol offers enhanced privacy with respect to the protocol in [1]. In addition, it is simpler and is more efficient in terms of communication rounds, communication cost and computational cost.***

Keywords - Privacy preserving data mining, distributed computation, frequent item sets, association rules

I. INTRODUCTION

Data mining technology has emerged as a means of identifying patterns and trends from large quantities of data. Data mining and data warehousing go hand-in-hand: Most tools operate by gathering all data into a central site, then running an algorithm against that data. However, privacy concerns can prevent building a centralized warehouse data may be distributed among several custodians, none of which are allowed to transfer their data to another site. This paper addresses the problem of computing association rules within such a scenario. We assume homogeneous databases: All sites have the same schema, but each site has information on different entities.

We study here the problem of secure mining of association rules in horizontally partitioned databases. In that setting, there are several sites (or players) that hold homogeneous databases, i.e., databases that share the same schema but hold information on different entities. The goal is to find all association rules with support at least sand confidence at least c, for some given minimal support size sand confidence level c, that hold in the unified database, while minimizing the information disclosed about the private databases held by those players. The information that we would like to protect in this context is not only individual transactions in the different databases, but also more global information such as what association rules are supported locally in each of those databases.

The goal defines a problem of secure multi-party computation. In such problems, there are M players that hold private inputs, $x_1, \ldots, x_M$, and they wish to securely compute $y = f(x_1, \ldots, x_M)$ for some public function f. If there existed a trusted third party, the players could surrender to him their inputs and he would perform the function evaluation and send to them the resulting output. In the absence of such a trusted third party, it is needed to devise a protocol that the players can run on their own in order to arrive at the required output y. Such a protocol is considered perfectly secure if no player can learn from his view of the protocol more than what he would have learnt in the idealized setting where the computation is carried out by a trusted third party. Yao [3] was proposed a generic solution for this problem in the case of two players.

In our problem, the inputs are the partial databases, and the required output is the list of association rules that hold in the unified database with support and confidence no smaller than the given thresholds s and c, respectively. As the above mentioned generic solutions rely upon a description of the function f as a Boolean circuit, they can be applied only to small inputs and functions which are realizable by simple circuits. In more complex settings, such as ours, other methods are required for carrying out this computation. In such cases, some relaxations of the notion of perfect security might be inevitable when looking for practical protocols, provided that the excess information is deemed benign (see examples of such protocols in, [1]).

Computing association rules without disclosing individual transactions is straightforward. We can compute the global support and confidence of an association rule $AB \Rightarrow C$ knowing only the local supports of AB and ABC and the size of each database:

$$support_{AB \Rightarrow c} = \frac{\sum_{i=1}^{sites} support\_count\ ABC^{(i)}}{\sum_{i=1}^{sites} database\_size(i)}$$

$$support_{AB} = \frac{\sum_{i=1}^{sites} support\_count\ AB^{(i)}}{\sum_{i=1}^{sites} database\_size(i)}$$

$$confidence_{AB \Rightarrow c} = \frac{support_{AB \Rightarrow c}}{support_{AB}}$$

The protocol that we propose here computes a parameterized family of functions, which we call threshold functions, in which the two extreme cases correspond to the problems of computing the union and intersection of private subsets. Those are in fact general-purpose protocols that can be used in other contexts as well. Another problem of secure multi-party computation that we solve here as part of our discussion is the set inclusion problem; namely, the problem where Alice holds a private subset of some ground set, and Bob holds an element in the ground set, and they wish to determine whether Bob's element is within Alice's subset, without revealing to either of them information about the other party's input beyond the above described inclusion.

II. Background and Related Work

There are several fields where related work is occurring. We first describe other work in privacy-preserving data mining, then go into detail on specific background work on which this paper builds.

Previous work in privacy-preserving data mining has addressed two issues. In one, the aim is preserving customer privacy by distorting the data values. The idea is that the distorted data does not reveal private information and thus is "safe" to use for mining. The key result is that the distorted data, and information on the distribution of the random data used to distort the data, can be used to generate an approximation to the original data distribution, without revealing the original data values. The distribution is used to improve mining results over mining the distorted data directly, primarily through selection of split points to "bin" continuous data. Later refinement of this approach tightened the bounds on what private information is disclosed by showing that the ability to reconstruct the distribution can be used to tighten estimates of original values based on the distorted data.

More recently, the data distortion approach has been applied to Boolean association rules. Again, the idea is to modify data values such that reconstruction of the values for any individual transaction is difficult, but the rules learned on the distorted data are still valid. One interesting feature of this work is a flexible definition of privacy, e.g., the ability to correctly guess a value of "1" from the distorted data can be considered a greater threat to privacy than correctly learning a "0." The data distortion approach addresses a different problem from our work. The assumption with distortion is that the values must be kept private from whoever is doing the mining. We instead assume that some parties are allowed to see some of the data, just that no one is allowed to see all the data. In return, we are able to get exact, rather than approximate, results.

The other approach uses cryptographic tools to build decision trees. In this work, the goal is to securely build an ID3 decision tree where the training set is distributed between two parties. The basic idea is that finding the attribute that maximizes information gain is equivalent to finding the attribute that minimizes the conditional entropy. The conditional entropy for an attribute for two parties can be written as a sum of the expression of the form $(\upsilon_1 + \upsilon_2) \times \log(\upsilon_1 + \upsilon_2)$. The authors give a way to securely calculate the expression $(\upsilon_1 + \upsilon_2) \times \log(\upsilon_1 + \upsilon_2)$ and show how to use this function for building the ID3 securely. This approach treats privacy-preserving data mining as a special case of secure multiparty computation and not only aims for preserving individual privacy, but also tries to preserve leakage of any information other than the final result. We follow this approach, but address a different problem (association rules) and emphasize the efficiency of the resulting algorithms. A particular difference is that we recognize that some kinds of information can be exchanged without violating security policies; secure multiparty computation forbids leakage of any information other than the final result. The ability to share non sensitive data enables highly efficient solutions.

The problem of privately computing association rules in vertically partitioned distributed data has also been addressed. The vertically partitioned problem occurs when each transaction is split across multiple sites, with each site having a different set of attributes for the entire set of transactions. With horizontal partitioning, each site has a set of complete transactions. In relational terms, with horizontal partitioning, the relation to be mined is the union of the relations at the sites. In vertical partitioning, the relations at the individual sites must be joined to get the relation to be mined. The change in the way the data is distributed makes this a much different problem from the one we address here, resulting in a very different solution.

III. Private Association Rule Mining Overview

Our method follows the two-phase approach described above, but combining locally generated rules and support counts is done by passing encrypted values between sites. The two phases are discovering candidate itemsets (those that are frequent on one or more sites) and determining which of the candidate itemsets meet the global support/ confidence thresholds.

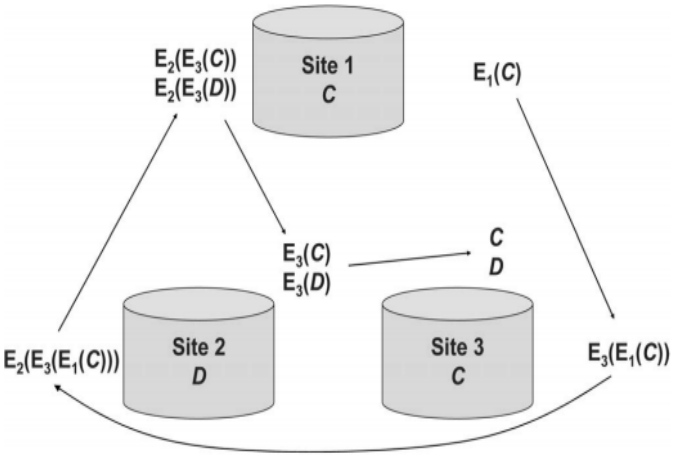

Fig. 1 Determining global candidate itemsets.

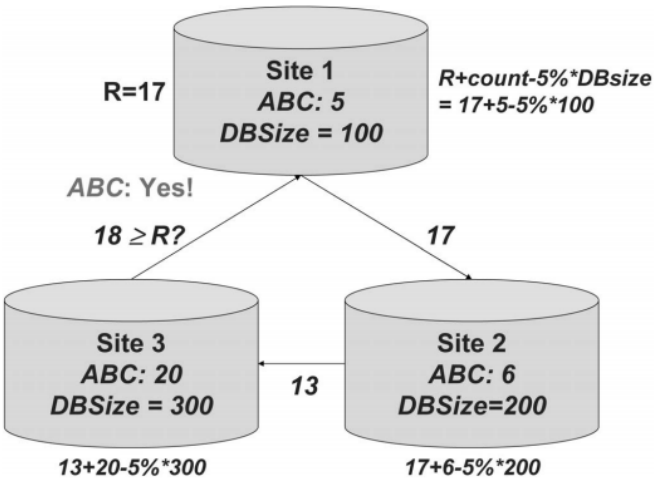

Fig. 2. Determining if itemset support exceeds 5 percent threshold.

The first phase (Fig. 1) uses commutative encryption. Each party encrypts its own frequent itemsets (e.g., Site 1 encrypts itemset C). The encrypted itemsets are then passed to other parties until all parties have encrypted all itemsets. These are passed to a common party to eliminate duplicates and to begin decryption. (In the figure, the full set of itemsets are shown to the left of Site 1, after Site 1 decrypts.) This set is then passed to each party and each party decrypts each itemset. The final result is the common itemsets (C and D in the figure).

In the second phase (Fig. 2), each of the locally supported itemsets is tested to see if it is supported globally. In the figure, the itemset ABC is known to be supported at one or more sites and each computes their local support. The first site chooses a random value R and adds to R the amount by which its support for ABC exceeds the minimum support threshold. This value is

passed to site 2, which adds the amount by which its support exceeds the threshold (note that this may be negative, as shown in the figure.) This is passed to site 3, which again adds its excess support. The resulting value is tested using a secure comparison to see if it exceeds the Random value. If so, itemset ABC is supported globally. This gives a brief, oversimplified idea of how the method works. Section 3 gives full details. Before going into the details, we give background and definitions of relevant data mining and security techniques.

IV. LITERATURE SURVEY

Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data

Data mining can extract important knowledge from large data collections—but sometimes these collections are split among various parties. Privacy concerns may prevent the parties from directly sharing the data and some types of information about the data. This paper addresses secure mining of association rules over horizontally partitioned data. The methods incorporate cryptographic techniques to minimize the information shared, while adding little overhead to the mining task.

Cryptographic tools can enable data mining that would otherwise be prevented due to security concerns. We have given procedures to mine distributed association rules on horizontally partitioned data. We have shown that distributed association rule mining can be done efficiently under reasonable security assumptions.

We believe the need for mining of data where access is restricted by privacy concerns will increase. Examples include knowledge discovery among intelligence services of different countries and collaboration among corporations without revealing trade secrets. Even within a single multinational company, privacy laws in different jurisdictions may prevent sharing individual data. Many more examples can be imagined. We would like to see secure algorithms for classification, clustering, etc. Another possibility is secure approximate data mining algorithms. Allowing error in the results may enable more efficient algorithms that maintain the desired level of security.

The secure multiparty computation definitions from the cryptography domain may be too restrictive for our purposes. A specific example of the need for more flexible definitions can be seen in Protocol 1. The “padding” set F is defined to be infinite so that the probability of collision among these items is 0. This is impractical and, intuitively, allowing collisions among the padded itemsets would seem more secure as the information leaked (itemsets supported in common by subsets of the sites) would become an upper bound rather than an exact value. However, unless we know in advance the probability of collision among real itemsets or, more specifically, we can set the size of F so the ratio of the collision probabilities in F and real itemsets is constant, the protocol is less secure under secure multiparty communication definitions. The problem is that knowing the probability of collision among items chosen from F enables us to predict (although generally with low accuracy) which fully encrypted itemsets are real and which are fake. This allows a probabilistic upper bound estimate on the number of itemsets supported at each site. It also allows a probabilistic estimate of the number of itemsets supported in common by subsets of the sites that is tighter than the number of collisions found in the RuleSet. Definitions that allow us to trade off such estimates and techniques to prove protocols relative to those definitions will allow us to prove the privacy of protocols that are practically superior to protocols meeting strict secure multiparty computation definitions.

More suitable security definitions that allow parties to choose their desired level of security are needed, allowing efficient solutions that maintain the desired security. Some suggested directions for research in this area are given in. One line of research is to predict the value of information for a particular organization, allowing trade off between disclosure cost, computation cost, and benefit from the result. We believe some ideas from game theory and economics may be relevant.

In summary, it is possible to mine globally valid results from distributed data without revealing information that compromises the privacy of the individual sources. Such privacy-preserving data mining can be done with a reasonable increase in cost over methods that do not maintain privacy. Continued research will expand the scope of privacy-preserving data mining,

enabling most or all data mining methods to be applied in situations where privacy concerns would appear to restrict such mining.

Privacy Preserving Association Rule Mining in Vertically Partitioned Data

Privacy considerations often constrain data mining projects. This paper addresses the problem of association rule mining where transactions are distributed across sources. Each site holds some attributes of each transaction, and the sites wish to collaborate to identify globally valid association rules. However, the sites must not reveal individual transaction data. We present a two-party algorithm for efficiently discovering frequent itemsets with minimum support levels, without either site revealing individual transaction values.

The major contributions of this paper are a privacy preserving association rule mining algorithm given a privacy preserving scalar product protocol, and an efficient protocol for computing scalar product while preserving privacy of the individual values. We show that it is possible to achieve good individual security with communication cost comparable to that required to build a centralized data warehouse. There are several directions for future research. Handling multiple parties is a non-trivial extension, especially if we consider collusion between parties as well. This work is limited to boolean association rule mining. Non-categorical attributes and quantitative association rule mining are significantly more complex problems. The same privacy issues face other types of data mining, such as Clustering, Classification, and Sequence Detection. Our grand goal is to develop methods enabling any data mining that can be done at a single site to be done across various sources, while respecting their privacy policies.

V. PRELIMINARY APPROACH

A] Definitions and Notations

Let D be a transaction database. As in [1], we view D as a binary matrix of N rows and L columns, where each row is a transaction over some set of items, A = $\{a_1, \ldots, a_L\}$, and each column represents one of the items in A. (In other words, the $\{i, j\}$ entry of D equals 1 if the i^{th} transaction includes the item a_j, and 0 otherwise.) The database D is partitioned horizontally between M players, denoted $P_1, \ldots, P_M$. Player P_m holds the partial database D_m that contains $N_m = |D_m|$ of the transactions in D, $1 \le m \le M$. The unified database is $D = D_1 \cup \ldots D_M$, and it includes $N := \sum_{m=1}^{M} N_m$ transactions.

An item set X is a subset of A. Its global support, supp(X), is the number of transactions in D that contain it. Its local support, $supp_m$ (X), is the number of transactions in D_m that contain it. Clearly, $supp(X) = \sum_{m=1}^{M} supp_m(X)$. Let s be a real number between 0 and 1 that stands for a required support threshold. An item set X is called s-frequent if $supp(X) \ge {}_sN$. It is called locally s-frequent at D_m if $supp_m(X) \ge {}_sN_m$.

For each $1 \le k \le L$, let F_s^k denote the set of all k-item sets (namely, item sets of size k) that are s-frequent, and $F_s^{k,m}$ be the set of all k-item sets that are locally s-frequent at D_m, $1 \le m \le M$. Our main computational goal is to find, for a given threshold support $0 < s \le 1$, the set of all s-frequent item sets, $F_s := \bigcup_{k=1}^{L} F_s^k$. We may then continue to find all (s, c)- association rules, i.e., all association rules of support at least ${}_sN$ and confidence at least c. (Recall that if X and Y are two disjoint subsets of A, the support of the corresponding association rule $X \Rightarrow Y$ is supp $(X \cup Y)$ and its confidence is $supp(X \cup Y)/supp(X)$).

B] The Fast Distributed Mining Algorithm

The protocol of [1], as well as ours, are based on the Fast Distributed Mining (FDM) algorithm of Cheung et al. [2],which is an unsecured distributed version of the Apriori algorithm. Its main idea is that any s-frequent item set must be also locally s-frequent in at least one of the sites. Hence, in order to find all globally s-frequent item sets, each player reveals his locally s-frequent item sets and then the players check each of them to see if they are s-frequent also globally. The FDM algorithm proceeds as follows:

1. **Initialization:** It is assumed that the players have already jointly calculated F_s^{k-1} . The goal is to proceed and calculate F_s^k.

2. **Candidate Sets Generation:** Each player P_m computes the set of all (k 1) item sets that are locally frequent in his site and also globally frequent; namely, P_m computes the set $F_s^{k-1,m} \cap F_s^{k-1}$. He then applies on that set the Apriori algorithm in order to generate the set $B_s^{k,m}$ of candidate k-item sets.

3. **Local Pruning:** For each $X \in B_s^{k,m}$, P_m computes $supp_m(X)$. He then retains only those item sets that

are locally s-frequent. We denote this collection of item sets by $C_s^{k,m}$.

4. **Unifying the candidate item sets:** Each player broadcasts his $C_s^{k,m}$ and then all players compute $C_s^k := \bigcup_{m=1}^{M} C_s^{k,m}$.

5. **Computing local supports**: All players compute the local supports of all item sets in C_s^k.

6. **Broadcast mining results:** Each player broadcasts the local supports that he computed. From that, everyone can compute the global support of every item set in C_s^k . Finally, F_s^k is the subset of C_s^k that consists of all globally s-frequent k-item sets.

In the first iteration, when k = 1, the set $C_s^{1,m}$ that the *m*th player computes (Steps 2-3) is just $F_s^{1,m}$, namely, the set of single items that are s-frequent in D_m. The complete FDM algorithm starts by finding all single items that are globally s-frequent. It then proceeds to find all 2-item sets that are globally s-frequent, and so forth, until it finds the longest globally s-frequent item sets. If the length of such item sets is K, then in the (k + 1)th iteration of the FDM it will find no (k +1) item sets that are globally s-frequent, in which case it terminates.

C] A Running Example

Let D be a database of N = 18 item sets over a set of L = 5 items, A = {1, 2, 3, 4, 5}. It is partitioned between M = 3 players and the corresponding partial databases are:

D_1 = {12, 12345, 124, 1245, 14, 145, 235, 24, 24},

D_2 = {1234, 134, 23, 234, 2345},

D_3 = {1234, 124, 134, 23}.

For example D_1 includes $N_1 = 9$ transactions, the third of which (in lexicographic order) consists of three items—1, 2 and 4.

Setting s = 1/3, an item set is s-frequent in D if it is supported by at least 6 = sN of its transactions. In this case,

F_s^1 = {1, 2, 3, 4},

F_s^2 = {12, 14, 23, 24, 34},

F_s^3 = {124},

$F_s^4 = F_s^5 = \emptyset$.

and $F_s = F_s^1 \cup F_s^2 \cup F_s^3$. For example, the item set 34 is indeed globally s-frequent since it is contained in 7 transactions of D. However, it is locally s-frequent only in D_2 and D_3.

In the first round of the FDM algorithm, the three players compute the sets $C_s^{1,m}$ of all 1-item sets that are locally frequent at their partial databases:

$C_s^{1,1}$= {1, 2, 4, 5}, $C_s^{1,2}$ = {1, 2, 3, 4}, $C_s^{1,3}$= {1, 2, 3, 4}.

Hence, C_s^1 = {1, 2, 3, 4, 5}. Consequently, all 1-item sets have to be checked for being globally frequent; that check reveals that the subset of globally s-frequent 1-item sets is F_s^1 = {1, 2, 3, 4}.

In the second round, the candidate item sets are:

$C_s^{2,1}$= {12, 14, 24}, $C_s^{2,2}$ = {13, 14, 23, 24, 34}, $C_s^{2,3}$= {12, 13, 14, 23, 24, 34}.

(Note that 15, 25, 45 are locally s-frequent at D_1 but they are not included in $C_s^{2,1}$ since 5 were already found to be globally infrequent.) Hence, C_s^2 = {12, 13, 14, 23, 24, 34}. Then, after verifying global frequency, we are left with F_s^2 = {12, 14, 23, 24, 34}.

In the third round, the candidate item sets are:

$C_s^{3,1}$= {124}, $C_s^{3,2}$ = {234}, $C_s^{3,3}$= {124}.

So, C_s^3 = {124, 234} and, then, F_s^3 = {124}. There are no more frequent item sets.

VI. Conclusion

Survey proposed a detailed analysis of the protocol for secure mining of association rules in horizontally distributed databases that improves significantly upon the current leading protocol [1] in terms of privacy and accuracy. One of the main ingredients in our proposed survey is a novel secure multi-party protocol for computing the union (or intersection) of private subsets that each of the interacting players holds. Another ingredient is survey on a protocol that tests the inclusion of an element held by one player in a subset held by another. Those surveys exploit the fact that the underlying problem is of interest only when the number of players is greater than two. One research problem that this study suggests to devise an efficient protocol for inequality validation that uses the existence of a semi-honest third party. Such a protocol might enable to further improve upon the communication and computational costs of the second and third stages of the protocol. Other research problems that this study suggests is the implementation of the techniques presented here to the problem of distributed association rule mining in the vertical setting, the problem of mining generalized association rules, and the problem of subgroup discovery in horizontally partitioned data.

References

1. M. Kantarcioglu and C. Clifton, “Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data,” IEEE Trans. Knowledge and Data Eng., vol. 16, no. 9, pp. 1026-1037, Sept. 2004.
2. D.W.L. Cheung, J. Han, V.T.Y. Ng, A.W.C. Fu, and Y. Fu, “A Fast Distributed Algorithm for Mining Association Rules,” Proc. Fourth Int’l Conf. Parallel and Distributed Information Systems (PDIS), pp. 31-42, 1996.
3. D. Beaver, S. Micali, and P. Rogaway, “The Round Complexity of Secure Protocols,” Proc. 22nd Ann. ACMSymp. Theory of Computing (STOC), pp. 503-513, 1990.
4. J. Vaidya and C. Clifton, “Privacy Preserving Association Rule Mining in Vertically Partitioned Data,” Proc. Eighth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD), pp.639- 644, 2002.
5. R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules in Large Databases,” Proc 20th Int’l Conf. Very Large Data Bases (VLDB), pp. 487-499, 1994.
6. R. Agrawal and R. Srikant, “Privacy-Preserving Data Mining,” Proc. ACM SIGMOD Conf., pp. 439-450, 2000.
7. Y. Lindell and B. Pinkas, “Privacy Preserving Data Mining,” Proc. Crypto, pp. 36-54, 2000.
8. X. Lin, C. Clifton, and M.Y. Zhu, “Privacy-Preserving Clustering with Distributed EM Mixture Modeling,” Knowledge and Information Systems, vol. 8, pp. 68-81, 2005.
9. H. Grosskreutz, B. Lemmen, and S. R€ uping, “Secure Distributed Subgroup Discovery in Horizontally Partitioned Data,” Trans. Data Privacy, vol. 4, no. 3, pp. 147-165, 2011.

AUTHOR(S) PROFILE

Maheshkumar Ramrao Gangasagare, received the B.E. degree in Computer Engineering from Savitribai Phule Pune University, Pune in 2012 and diploma in Diploma in Advanced Computing from C-DAC institute Pune in 2013. He is a M.E. candidate at Swami Ramanand Teerth Marathwada University, Nanded.

Rafik Juber Thekiya, received the B.E. degree in Computer Science and Engineering from Swami Ramanand Teerth Marathwada University, Nanded in 2006. Also, he received M.Tech. in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad in 2012. He served as a Lecturer for 3 years and Assistant Professor for 2 years.