Scope International Journal of Science, Humanities, Management and Technology. ISSN : 2455-068X
Vol.2 Issue 1 (2015) 8-17. Submitted 09/01/2016. Published 9/02/2016

Mining of Frequent Itemsets and Utility from Operational Data using Data Mining Technique

Vinoth Raja K, Dr. M. Thangamani
[1]Assistant Professor, St. Mother Theresa Engineering College, Vagaikulam, Tamilnadu, India
[2]Dr. M. Thangamani, Assistant Professor, Kongu Engineering College, Tamilnadu, India

ABSTRACT

Frequent Itemset Mining method generate the frequent patterns. The frequency of an itemset may not be a sufficient indicator of interestingness, because it only reflects the number of transactions in the database that contain the itemset. It does not reveal the utility of an itemset, which can be measured in terms of cost, profit, or other expressions of user preference. In this research, maximize profit, the itemset utilities should be decided by the quantity of items sold and the unit profit on these items. In proposed system, an algorithm named Utility Model-Growth for mining high utility itemsets from transaction databases are used. Utility Model-Tree maintains the information of high utility itemsets. The mining performance is enhanced significantly since both the search space and the number of candidates are effectively reduced.

1. INTRODUCTION

Data mining is the process of extracting hidden patterns from large amounts of data. It is sometimes referred to as Knowledge Discovery in Databases. The goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. It is commonly used in a wide range of profiling practices such as marketing, surveillance, fraud detection and scientific discovery. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.

Frequent itemset mining has become an important data mining task and a focused theme in data mining research. Frequent patterns are itemsets, subsequences or substructures that appear frequently in a data set i.e., with frequency more than the user-specified threshold. Finding frequent patterns plays an essential role in mining associations, correlations and many other interesting relationships among data.

1.1. ASSOCIATION RULE MINING

In data mining, association rules are useful for analyzing and predicting customer behaviour. They play an important part in shopping basket data analysis, product clustering, catalogue design and store layout. Association rule mining finding frequent patterns, associations, correlations or causal structures among sets of items or objects in transaction databases, relational databases and other information repositories. Mining association rules from massive amount of data in the database is interested for many industries which can help in many business decision making processes, such as cross-marketing, basket data analysis and promotion assortment.

Let I be the set of all items and T be the set of all transactions i.e $I=\{i_1,i_2,\ldots i_n\}$ and $T=\{t_1,t_2,\ldots t_m\}$. Each transaction ti contains the subset of items chosen from I. The transaction width is defined as the number of items present in a transaction. A collection of zero or more items is called as the itemset. The itemset has a property namely support count which is the

number of transactions contain a particular itemset. The support count for an itemset is calculated as follows

$$\sigma(X)=|\{t_i|X \text{ is subset of } t_i,\ t_i \epsilon T\}|$$

where X is the itemset , $\sigma(X)$ is the support count and |.| is the number of items in a set.
The strength of association rule can be measured in the terms of support and confidence. Support determines by how long the rule is applicable to a given data set and Confidence determines how frequently the item is appeared in the transactions

Association Rule Mining (ARM) process can be divided into two steps. The first step involves finding all frequent itemsets in databases. Once the frequent itemsets are found association rules are generated ARM is widely used in market-basket analysis. For example, frequent itemsets can be found out by analyzing market basket data and then association rules can be generated by predicting the purchase of other items by conditional probability. An association rule mined from market basket database states that if some items are purchased in transaction, then it is likely that some other items are purchased as well. Finding all such rules is valuable for guiding future sales promotions and store layout. One key step of association mining is frequent itemset mining, which is to mine all itemsets.

1.2 FREQUENT ITEMSET MINING

Frequent itemset plays the vital role in the data mining tasks such as find the interesting patterns from databases. Frequent itemset mining is the process of finding patterns like itemsets, subsequences and substructures that occur frequently in a database. The frequent itemset mining is motivated by problems such as market basket analysis. It also plays an essential role in the mining of many other patterns such as correlation and association rules. Mining of frequent itemsets is the set of items which are frequently purchased together in a transaction. Identifying frequent itemsets is one of the most important issues faced by the knowledge discovery and data mining. Mining frequent itemsets is the very crucial task to find the association rules between the various items. The frequent itemset mining is to identify all frequent itemsets. The generations of association rules are straight forward, once the frequent itemsets are identified. The frequent itemset is found out by using the methods such as Apriori Algorithm and vertical data format.

1.3 UTILITY MINING

The traditional ARM approaches consider the utility of the items by its presence in the transaction set. The frequency of itemset is not sufficient to reflect the actual utility of an itemset. For example, the sales manager may not be interested in frequent itemsets that do not generate significant profit. Recently, one of the most challenging data mining tasks is the mining of high utility itemsets efficiently. Identification of the itemsets with high utilities is called as Utility Mining. The basic meaning of utility is the interestedness/importance/profitability of items to the users. In utility based mining the term utility refers to the quantitative representation of user preference. The limitations of frequent or rare itemset mining motivated researchers to conceive a utility based mining approach, which allows a user to conveniently express his or her perspectives concerning the usefulness of itemsets as utility values and then find itemsets with high utility values higher than a threshold.

2. EXISTING SYSTEM

Existing approach uses Frequent pattern list and Transaction pattern list for the sparse and dense databases to find frequent itemsets. During mining process, the features of the real databases are diversified and do not offer much benefit. It consumes more memory space and time to find the complete frequent itemsets. The mining process should not be adaptive to the conditions of the database and does not use the appropriate data structure. It may loss infrequent but valuable itemsets and may present too many frequent but unprofitable itemsets to users. The important itemsets with high profits can't be found.

Problem Definition: The Existing Frequent Itemset Mining method does not considering purchased quantity and sales profit. The frequent itemset mining is to find an item that occurs in a transaction database above a user given frequency threshold, without considering the quantity or weight such as profit of the items. Each item in the supermarket has a different price and single customer will be interested in buying multiple copies of same item. Therefore finding only traditional frequent patterns in a database cannot fulfil the requirement of finding the most valuable itemsets that contribute the total profit in a retail business.

3. IMPLEMENTATION AND RESULT

The data structure named Utility Model Tree (UM-Tree) and the algorithm called Utility Model Growth (UM-Growth) are used for mining utility itemsets in the proposed work. To facilitate the mining performance and avoid scanning original database repeatedly, it uses Utility Pattern Tree to maintain the information of transactions and high utility itemsets.

The Elements in UM-Tree: In UM-Tree, each node N includes N.name, N.count, N.nu, N.parent, N.hlink and a set of child nodes. The details are introduced as follows. N.name is the item name of the node. N.count is the support count of the node. N.nu is called node utility which is an estimate utility value of the node. N.parent records the parent node of the node. N.hlink is a node link which points to a node whose item name is the same as N.name. Header table is employed to facilitate the traversal of UM-Tree. In the header table, each entry is composed of an item name, an estimate utility value, and a link. The link points to the last occurrence of the node which has the same item as the entry in the UM-Tree. By following the link in the header table and the nodes in UM-Tree, the nodes whose item names are the same can be traversed efficiently. The goal of utility mining is to discover all the high utility itemsets whose utility values are beyond a user specified threshold in a transaction database. It proposes UP-Growth and uses a special data structure called UP-Tree.

Structure of UP-Tree: The UM-Tree with two scans of the database. In the first scan Transaction Utility of each transaction is computed Transaction Weighted Utilization (TWU) of each single item is also calculated. Unpromising items are removed from the transaction and utilities are eliminated from the Transaction Utilities of the transaction databases. The remaining promising items in the transaction are sorted in the descending order of TWU. In the second scan Transactions are inserted into UM-Tree. The proposed system is divided into three modules containing removing Unpromising items, construction of UM Tree Finding Utility Items

Removing Unpromising Items: Consider the transaction database in Table.1 and the profit table in Table 2. Given a finite set of items $I = \{i_1, i_2, \ldots, i_m\}$. Each item i_p $(1 \leq p \leq m)$ has a unit profit $p(i_p)$. An itemset X is a set of k distinct items $\{i_1, i_2, \ldots, i_k\}$, where i_j I, $1 \leq j \leq k$, and k is the length of X. Fig.1 shows the screen shot of Removing Unpromising Items.

A transaction database D = {T_1, T_2, …, T_n} contains a set of transactions, and each transaction T_d ($1 \leq d \leq n$) has an unique identifier d, called TID. Each item i_p in the transaction T_d is associated with a quantity $q(i_p, T_d)$, that is, the purchased number of i_p in T_D.

Table 1 Sample Transaction Database

TID	TRANSACTION	TU
T1	(A,1) (C,1) (D,1)	8
T2	(A,2) (C,6) (E,2) (G,5)	27
T3	(A,1) (B,2) (C,1) (D,6) (E,1) (F,5)	30
T4	(B,3) (C,3) (D,3) (E,1)	20
T5	(B,2) (C,2) (E,1)(G,2)	11

Table 2 Profit Table

Item	A	B	C	D	E	F	G
Profit	5	2	1	2	3	1	1

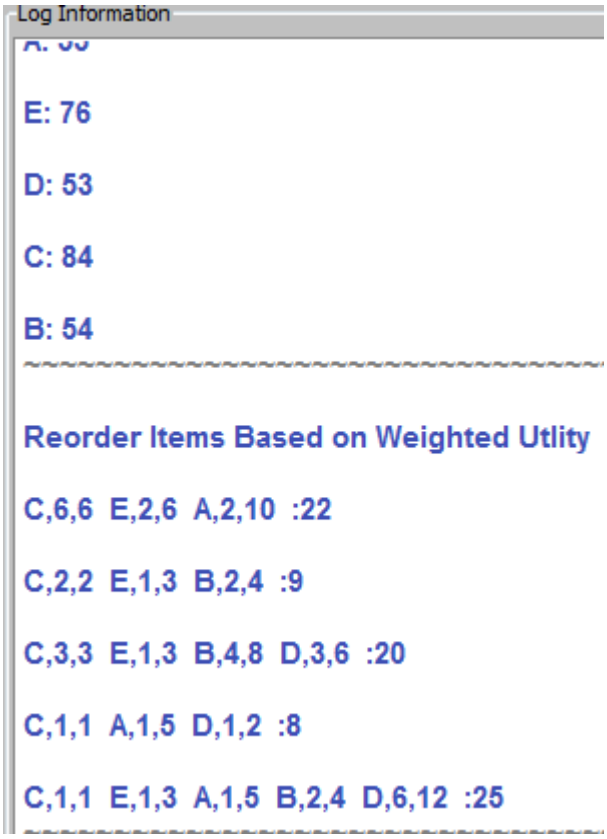

Fig.1 Removing Unpromising Items

Step 1: The utility of an item i_p in the transaction T_d is denoted as $u(i_p, T_d)$ and defined as $p(i_p) \times q(i_p, T_d)$. For example, in Table 1 and 2, $u(\{A\}, T_1) = 5 \times 1 = 5$.
Step 2: The utility of an itemset X in T_d is $u(X, T_d)$ and defined as $u(i_p, T_d)$. For example, $u(\{AC\}, T1) = u(\{A\}, T_1) + u(\{C\}, T_1) = 5 + 1 = 6$.
Step 3: The utility of an itemset X in D is denoted as u(X) and defined as u(X ,Td) For example, $u(\{AD\}) = u(\{AD\}, T1) + u(\{AD\}, T3) = 7 + 17 = 23$.
Step 4: The transaction utility of a transaction T_d is denoted as $TU(T_d)$ and defined as $u(T_d, T_d)$. For example, $TU(T_1) = u(\{ACD\}, T_1) = 8$.
Step 5: The transaction-weighted utilization of an itemset X is the sum of the transaction utilities of all the transactions containing X, which is denoted as TWU(X). For example, $TWU(\{AD\}) = TU(T1) + TU(T_3) = 8 + 30 = 38$. Suppose the minimum utility threshold min_util is 30. In the first scan of database, TUs of the transactions and the TWUs of the items are computed. They are shown in the last column of Table 3. As shown in Table 3, {F} and {G} are unpromising items since their TWUs are less than min_util. The promising items are reorganized in the header table in the descending order of TWU.

Table 3 Items and their TWUs

Item	A	B	C	D	E	F	G
TWU	65	61	96	58	88	30	38

An item i_p is called a promising item if $TWU(i_p)$ min_util. Otherwise, the item is called an unpromising item. Table 4 shows the reorganized transactions and their RTUs for the database. As shown in Table 4, unpromising items {F} and {G} are removed from the transactions T_2, T_3 and T_5, respectively. Besides, the utilities of {F} and {G} are eliminated from the TUs of T_2, T_3 and T_5, respectively. The remaining promising items {A}, {B}, {C}, {D} and {E} in the transaction are sorted in the descending order of TWU. The unpromising items and their utilities are eliminated from the transaction utilities during the construction of a UM-Tree. The principle is to discard the information of unpromising items from the database since an unpromising item plays no role in high utility itemsets and only the supersets of promising items are likely to be high utility.

Table 4 Reorganized Transactions

TID	Reorganized transaction	RTU
T_1'	(C,1) (A,1) (D,1)	8
T_2'	(C,6) (E,2) (A,2)	22
T_3'	(C,1) (E,1) (A,1) (B,2) (D,6)	25
T_3'	(C,3) (E,1) (B,3) (D,3)	20
T_5'	(C,2) (E,1) (B,2)	9

Construction of UM –Tree: The first reorganized transaction T_1' = {C, A, D} leads to create a branch in UM-Tree. The first node {C} is created under the root with {C}.count = 1 and {C}.nu = 8. The second node {A} is created under node {A} with {A}.count = 1 and {A}.nu =

8. The third node {C} is created as a child of node {A} with {C}.count =1 and {C}.nu = 8. When the next reorganized transaction T_2' = {C, E, A} is retrieved, the node utility of the node { C} is increased by 22 and {C}.count is increased by 1. Then, a new node {E} is created under {C} with {E }.count=1 and {E}.nu = 22. Similarly, a new node {A} is created under the node {E} with {A}.count=1 and {A}.nu = 22. The reorganized transactions T_3', T_3' and T_5' are inserted in the same way. Fig.2 shows the screen shot for remove minimum weighted utility item.

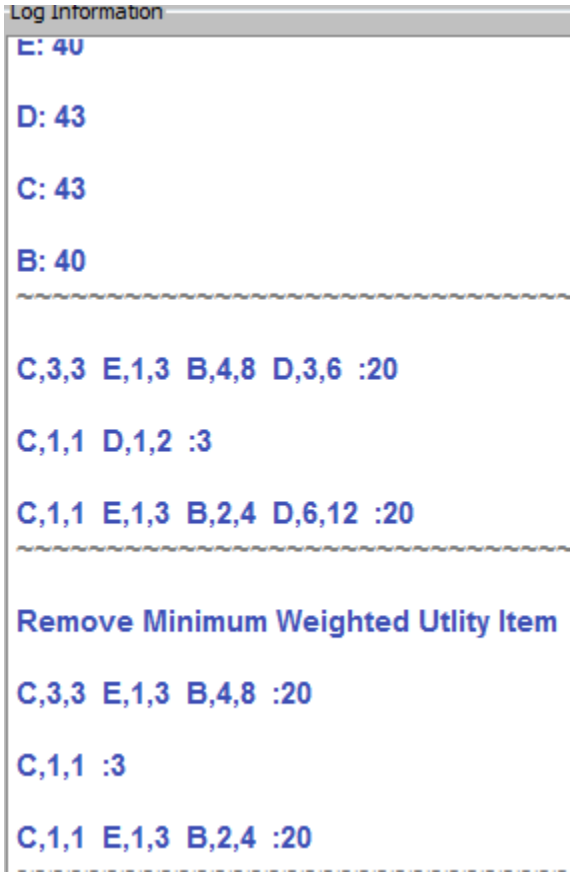

Fig.2 Finding the Path

Step 6: (Path utility of a path) The path utility of a path p_j = ($\{a_i\}\{a_{i+1}\}....\{a_n\}$) in $\{a_i\}$-CPB is equal to $\{a_i\}$.nu and is denoted as pu(p_j, $\{a_i\}$-CPB).For example, the path utility of the path {AC} in {D}-CPB is 8.

Step 7: (Path utility of an item in a path) For each item i_p in the path p_j , the path utility of an item i_p in a path p_j in $\{a_i\}$-CPB is equal to pu(p_j, $\{a_i\}$-CPB)and denoted as pu(i_p, p_j). For example, the path utility of {A} in the path {AC} is 8.

Step 8: (Path utility of an item in a database) The path utility of an item i_p in $\{a_i\}$-CPB. For example, the path utility of item {A} in {D}-CPB is equal to (pu ({A}, {AC}) + pu({A}, {BAEC})) = (8 + 25) = 33.

Suppose min_util is 30. The algorithm starts from the bottom of the header table and considers the item {D}. By tracing the nodes to root, three paths (D-A-C: 1, 8), (D-B-A-E- C: 1, 25) and (D-B-E-C: 1, 20) are found. For each path, the first number beside the path is is the path

utility, which is equal to {D}.nu. These paths are collected and shown in Table 5. By scanning, items and their path utilities are obtained, which is shown in Table 3.6. In Table 6, item {A} is an unpromising item since its path utility is less than min_util, i.e., 33<30. Then promising items {B}, {C} and {E} are arranged in the header table. Scan again to construct UM-Tree, when unpromising items are removed from the path and the remaining items are rearranged in the descending order according to their path utilities.

Finding Utility Itemsets: Fig.3 shows the utility itemsets. When a path is retrieved, each unpromising item is removed from the path and its minimum item utility in this path is eliminated from the path utilities. Consider the path shown in Table 7, the reorganized transactions are shown in the second column, and their path utilities which are reduced .When the first reorganized path {C} is inserted into Tree, the first node {C} is created under the root R' with {C}.count = 1 and {C}.nu = 3. When the second path { C, B, E } is inserted into the tree, {C}.count is increased by 1, and {C}.nu is increased by(20 – (miu({B}) × 1 + miu({E}) × 1)) = 20 – (3+3) = 13. After that, {C}.nu is equal to 16. The second node {B} is created under the node {C} with {B}.count = 1 and {B}.nu = (20 – miu({E}) × 1) = 20 – 3 = 17. The last node {E} is created under the node {B} with {E}.count = 1 and {E}.nu = 20. After inserting all paths in {D}-CPB, {D}-Tree is constructed completely.

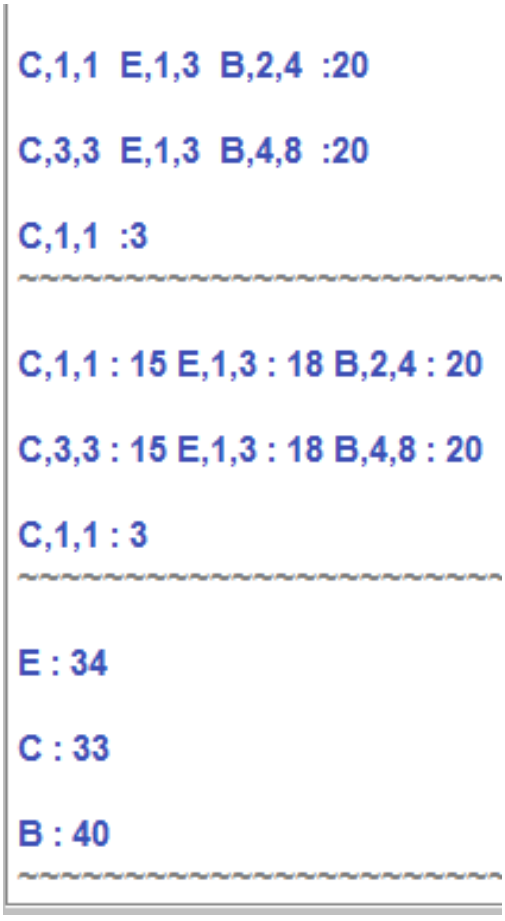

Fig 3 Utility Itemsets

Table 5 Finding Path

PATH	REORGANIZED PATH	PATH UTILITY
{A,C}	{C}	8
{B,A,E,C}	{CBE}	25
{B,E,C}	{CBE}	20

Table 6 Minimum Utility

ITEM	A	B	C	D
PATH UTILITY	33	35	53	35

Table 7 Path Utility

PATH	REORGANIZED PATH	PATH UTILITY
{A,C}	{C}	3
{B,A,E,C}	{CBE}	20
{B,E,C}	{CBE}	20

The Fig.4 shows that Utility mining performs better than Frequent Itemset Mining. It enhances overall profit. In this result shows the system of frequent patterns produce small portion of the overall profit. Utility mining refers to the quantitative representation of user preference.

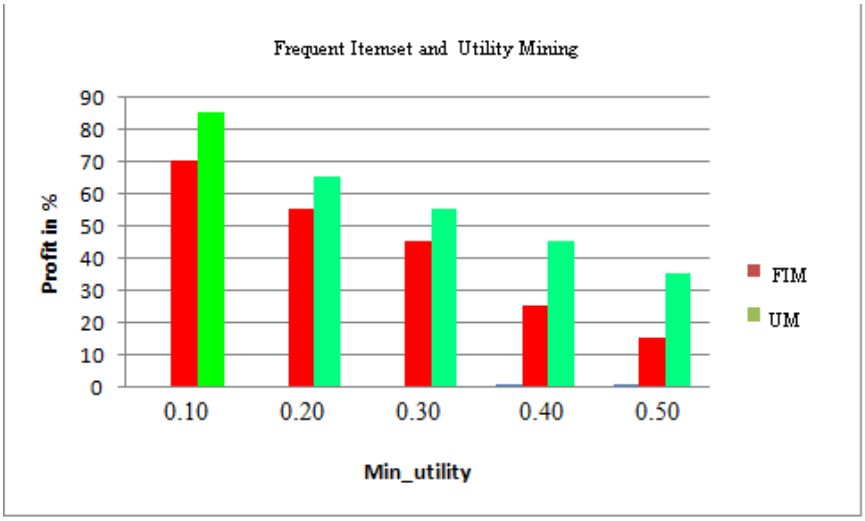

Fig.4 Minimum Utility Vs Profit

4. CONCLUSION AND FUTURE WORK

Frequent Itemset Mining method produces only the frequent patterns. In this research maximize profit, the itemset utilities should be decided by the quantity of items sold and the unit profit on these items. The result shows that it performs well especially when the database contains long transactions. In future, it can be extended using Utility Model Growth Plus Algorithms for reducing both run time and number of candidates.

References

1. Che-Tsung, Yang, Jen-peng Huang (2003), 'The Maintenance of Association Rules using Adjusting Frequent Pattern List', International Conference on Informatics, Cybernetics, and Systems.
2. Chan Yang, Che-Tsung (2007), 'Mining high utility itemsets', In Proceedings of Third IEEE International Conference on Data Mining, pp. 19-26, Nov., 2003.
3. Jiawei Han, Jian Pei and Yiwen Yin (2000), 'Mining frequent patterns without candidate generation', In Proceedings of 2000 ACM SIGMOD International Conference on management of data pp. 1–12.
4. Lie Tseng, Jeong Tanbeer and Lee (2009), ' Efficient tree structures for high utility pattern mining in incremental databases', In IEEE transactions on Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708-1721.
5. Rakesh Agrawal, Ramakrishnan Srikant (1994), 'Fast algorithms for mining association rules in large databases', In Proceedings of 20th International Conference on very large data bases pp. 478–499.
6. Vincent Tseng, Cheng-Wei Wu and Bai-En Shie (2010), 'UP-Growth: An Efficient Algorithm for High Utility Itemsets Mining', Proceedings on 16th ACM SIGKDD Conference of Knowledge Discovery and Data Mining (KDD'10), pp. 253-262.
7. Vincent Tseng, Cheng-Wei Wu and Bai-En Shie (2012), 'Efficient algorithms for mining high utility itemsets from transactional databases', IEEE transactions on knowledge and data engineering, Vol. 25, no. 8.

Authors Biography

Mr. K. Vinoth Raja received his M.E degree from the Anna University Chennai in 2006. He is currently working in St. Mother Theresa Engineering College, Thoothukudi, and Tamil Nadu. He is working for his Ph.D degree at Anna University, Chennai. His research interests in Bioinformatics with Big Data Mining and analysis. He is also the member of ISTE and CSI chapters of India.

Dr. M. Thangamani possesses nearly 20 years of experience in research, teaching, consulting and practical application development to solve real-world business problems using analytics. Her research expertise covers Medical data mining, machine learning, cloud computing, big data, fuzzy, soft computing, ontology development, web services and open source software. She has published nearly 70 articles in refereed and indexed journals, books and book chapters and presented over 67 papers in national and international conferences in above field. She has delivered more than 60 Guest Lectures in reputed engineering colleges on various topics. She has got best paper awards from various education related social activities in India and Abroad. She has organized many self-supporting and government sponsored national conference and Workshop in the field of data mining, big data and cloud computing. She continues to actively serve the academic and research communities. She is on the editorial board and reviewing committee of leading research journals, which includes her nomination as the Associate Editor to International Journal of Entrepreneurship and Small & Medium Enterprises at Nepal and on the program committee of top international data mining and soft computing conferences in various countries. She is also seasonal reviewer in IEEE Transaction on Fuzzy System, international journal of advances in Fuzzy System and Applied mathematics and information journals. She has organizing chair and keynote speaker in international conferences in India and countries like Malaysia, Thailand and China. She has Life Membership in ISTE, Member in CSI, International Association of Engineers and Computer Scientists in China, IAENG, IRES, Athens Institute for Education and Research and Life member in Analytical Society of India. She is currently working as Assistant Professor at Kongu Engineering College at Perundurai, Erode District.