Scientific Journal of Impact Factor (SJIF): 5.71
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 02, February -2018

Enhancing user ranks in reputation system with splay tree by breaking power law

R. Jayashree[1], A. Christy[2]

[1]Assistant Professor, SRM University
[2]Assistant Professor, Sathyanbama University

Abstract - *The reputation system is an efficient and effective way to build trust relationships among co-learners in collaborative learning. In reputation system, users with higher point gets high rating compared to less scored users. As per power law, drive alleged low users to the bottom of the ranking list. Break this law by encouraging low reputed users in their early stage and preventing them from moving further down in ranking level. A splay tree is a Binary Search Tree with self-balancing skill which brings the recently accessed item to the top of the tree. A splay tree represents user's ranks. Low ranked users are semi-splayed in the tree thus preventing them from further drowning in the ranking list by enhancing their ranks in the reputation system. In this paper, we find and enhance low scored users' rank in reputation system by providing few more chances to take part actively in the e-learning environment using splay tree and normalized discounted cumulative gain (**NDCG**) which act as a decision part for identifying drowning users.*

***Keywords* -** *Power law; Splay tree; reputation system; ranking algorithm; Normalized Discounted Cumulative Gain; reputation in website; ranking in e-learning environment; semi-splay; rank improvement method.*

I. INTRODUCTION

Co-learners trust relationships play an imperative role to set up collaborative activities in e-learning environments. An individual's privacy diminishes by expectations of trust [27]. A contextual evaluation of a person's actions is reputation [27]. Obtain reputation through rating or ranking which is an effectual source for measuring trust. The power law in reputation system is a relationship in which a relative change in reputation gives rise to a proportional relative change in ranking, independent of the initial rank of the user. Thus by power law, high ranked users are very few, and many of users listed at the middle level and very huge at the low-level of ranking. The splay tree brings the newly accessed items closer to the top of the tree, thus the recently searched items are accessible in O(1) time if accessed again. The locality of reference states that 80% of the accesses are to 20% of the items. A splay tree search operation does the same standard Binary Search Tree (BST) searching, additionally, it moves recently accessed user to the crown of the tree. The splay tree allows searching and insertion operations to balance the tree so that future operations may run faster. Based on the heuristic, if user X accessed once, then same user X is likely to get accessed again. The active (recently accessed) user will move towards the root, and inactive users will slowly move far-off from the root. Let user X is not an Uppermost_User, that is, X has, at least, one ancestor, then "Zig" and "Zag" are just a single rotation, as in an AVL tree right and left rotation respectively [24]. "Zig-Zig"/ "Zag-Zag" consists of two single rotations of the same type, and "Zig-Zag"/ "Zag-Zig" consists of two rotations of the opposite type, similar to an LR imbalance correction [24]. This guarantees that even if the depths of some nodes get huge, a long sequence of O(N) searches does not occur because each search operation causes a rebalance. Stackoverflow is a well-known website that allows registered users to post their questions and to post answers to others' questions [23]. In general, users with good answers are high ranked. In all Question-Answering websites, like Stackoverflow, mark users with the top reputation scores as highly reputed users. Overall rating depends on the ratings of users with low reputation. In this paper, identified the less active users with good reputation scores and boosted up in the ranking list so that they do not get neglected in ranking list. In other words, moderately reputed or less active users are given few more chances to take part actively and thus, postponing them from getting eliminated from top scored list.

In our earlier work, we presented an online rating calculation method for reputation management which approximates expectation of contributors' performances to derive simple update rules for online ranking using Bayesian approximation method and Normalized Discounted Cumulative Gain as a metric for measuring ranking correctness [13]. DCG measures quality of the results in a ranked list [27]. Contributors' reputations are predicted. Calculate users' overall rating and ranking and verify ranked contributors and their reputation scores using NDCG algorithm [27].

In this paper, we discuss user reputation-based ranking system with the splay tree and NDCG as a significant concept and develop the re-ranking with semi-splay algorithm to prevent weak users from downfall thus, improve their ranks. The paper is organized as follows: Section 2 describes motivation and background, Section 3 discuss about splay tree and NDCG in reputation system. NDCG is to identify less active users. Section 4 describes how to break power law by increasing user level in the splay tree using semi-splay and present re-ranking-with-semi-splay algorithm. Section 5 describes and discusses experimental results and finally, section 6 concludes and describes future work.

II. MOTIVATION AND BACKGROUND

According to power law, small occurrences are extremely common, whereas large instances are extremely rare [34]. The quality and quantity of user's contributions compute their reputations. The good quality contribution preserves the introduced changes in subsequent revisions [30, 31, 32]. Evaluate user status to predict future user contributions' quality [30]. The predictive ability of the content reputation system measures its performance [27]. The design space characteristics influence the structure of a reputation system. D Movshovitz described the different experts versus non-experts activity patterns and highlighted the detecting anomalous users' importance in his work. Identified the potential expert users based on their business in the first few months of activity on the site. An Initial activity of a user when joining the site is indicative of his/her long-term contribution [29]. Enhancing or reducing the influences of the large-degree users could produce precise reputation ranking lists [28]. Based on splay tree, SChord[1] implemented Chord finger table with improved resource locating efficiency and related nodes hierarchy to the access frequency. Routing and caching are the two operations in Schord ring[1] where the routing process is to look up the closest preceding node and caching is searching and inserting (key, node) to its splay tree. Each node has a splay tree. Insert node n with s successors (n.id+2i), 0<=i<=s-1 into splay tree [1]. Stefan et al.[2] explored fully decentralized and self-adjusting network that minimizes the routing cost between arbitrary communication pairs. They proved by the empirical entropies of the sources and destinations that the overall cost is upper bounded. A new content authentication scheme proposed by Liangbin et al., in which constructed Merkle hash tree (MHT) based on an OBST [3]. The basic idea in MHT is to produce a short cryptographic description of a large data set. Parent node stores the concatenated children node values and verified an element's node's siblings in the path from the associated node to the root. An element's authentication cost depends on the computation time which is a linear to node's depth in MHT.

Dimitris et al. [4] presented the Randomized splay tree version with chain splay technique for compressing data. An adaptive data compression algorithm called as the splay-prefix algorithm on the prefix code, where restructured the code tree using semi-splaying. In semi-splaying technique, splay the leaf corresponding to the transmitted symbol so that it moves halfway to the root, thus moving other symbols automatically to the bottom of the tree. Comparing randomized with non-randomized versions based on rotations and time proves that randomized algorithm is much smaller than the deterministic text of the algorithm. Randomized version[4] achieves up to 3% reduction in the rotations and is preferable for the application of relatively small sequences of accesses on a large amount of data. The splay tree is very suitable for caching the recently accessed content to provide quick access again. The splay tree has good performance [14] since it is self-optimizing. For quick access, move frequently accessed nodes closer to the root. Packets sorted as binary search tree and then balanced tree [12]. Self-adjusting tree [13, 21] ideas are implemented to design a cache management for Content-Centric Networking (CCN) [5] and relate the download time to the class popularity that is, the download time is very short for the content with high probability to access. Consider the frequency of visits and the recent visit to evaluate the content popularity [5]. Overall packets matching time reduces with Splay tree by rejecting unwanted traffic in early stages and by accepting repeated packets with fewer memory accesses [6]. Splay tree changes dynamically according to the flow of traffic and to the store length of the prefixes. The level of access determines binary search on prefix lengths [15]. Statistical Splay Tree Policy Filters [6] (SSF-BSPL) optimize the early rejection of unwanted flows, and the acceptance of repeated wanted traffic through splaying properties. Filtering processing time for the unwanted packets reduces by arranging policy fields in descending order starting from the area with the highest rejection statistics. Normalized Discounted Cumulative Gain is a metric for measuring ranking correctness [27]. In software documentation retrieval environment application the place of recommended items in the list is important for the recommendation and normalized discounted cumulative gain (NDCG) is a frequently used metric for measuring ranking correctness, considering item ranking position [33]. Reputation is the sum of scores given by peer contributors of website which later ordered to find ranking among users [27]. The registered users gain more mean points than anonymous users and registered users are more trusted [27].

In Splay Tree Packet Classification Technique [7] (ST-PC) stores integer values with their matching rules in splay trees. Whereas in Self-Adjusting Binary Search on Prefix Length [22] (SA-BSPL) stores the prefix lengths and their corresponding hash tables with matching rules which gives better-amortized analysis. System performance is significantly affected by default-deny rule [7] which increases filtering processing time. Early packet rejection techniques reject the maximum number of packets as soon as possible; thereby reduce filtering processing time. Key Insertion and Splay Tree encryption [8] (KIST) algorithms use the splay tree for encryption. Key injection algorithm compresses the cipher text that moves inner nodes which are higher than specified layer. In cloud environment key insertion and splay tree-based outsourcing key management [8] provides an approach that is highly secure and flexible. SplayNet [9] is a distributed generalization of the splay tree where frequently communicating nodes are closer. Srinivasan et al.[21]

proposed splay tree as optimized binary search tree which reduces average access time by moving more popular nodes closer to the root. Harper [17] introduced Minimum Linear Arrangement [16] (MLA) problem to design error-correcting codes with minimum average absolute errors. The domains such as job scheduling [20] and nervous activity in the cortex [19] use MLA concept. Leitao et al. [18] study self-optimizing overlay networks with dynamic topology. Chen Avin et al.[9] designed a double splay algorithm to perform splaying in subtrees, Zouheir Trabelsi and Safaa Zeidan [10] proposed a mechanism based on multilevel filtering modules using the splay tree, to optimize filtering fields order according to traffic statistics. This scheme rejects unwanted traffics in the early stages and thus decreases overall packets matching time. Statistical Splay Tree Policy Filters [10] (SSF-BSPL) system uses a mathematical model to decide statistical policy fields order for the next packet segment. Rank the learning objects in the repository based on the citation numbers like Google page rank[11]. Slivkins et al.'s [25] work selects the relevant documents to obey the expected relevance rate μ(x), distributed according to a power-law, for each document x.

III. SPLAY TREE AND NDCG IN REPUTATION SYSTEM

Splay trees are the self-adjusting tree with amortized time bounds. In this tree move frequently accessed users towards highly reputed users, Uppermost_User. In this strategy, the highly active users stay close to the Uppermost_User and thus quickly found. Construct a splay tree t according to the users' active participation in the website; thus, the lastly accessed user is at the position 'Uppermost_User'. Let UserN(Y) be the some users ranked below the user Y then,

$$rank(Y)=log(userN(Y))$$

Let rank`(Y) be the user Y's rank before splaying. The time taken for searching a user is proportional to the depth of the user Y in t before splaying, that is, the number of links L from Uppermost-User to user Y. For "m" number of users, searching operations runs in θ(log m) worst-case time. Let ℓ be the reputation-based sorted list of users where highly reputed user is in the top of the list and least reputed is in the bottom. Searching time of user x in t is directly proportional to searching time of user x in ℓ. If the user x is both active and highly reputed, then position user x about at the same level or depth, in both the t and ℓ. But sometimes, users are more active at the beginning with the highest reputation and then they become inactive. In other cases, though users are active they score destitute status. Calculate users' reputation with their positive or negative votes. Inactive users may be pushed to more inactive/dead state and poorly reputed users to poorest/eliminated state. Avoid this situation by improving the level of users in the splay tree, thus proving chances for weak users to become active again and to gain more reputation.

3.1. Normalized Discounted Cumulative Gain (NDCG) in Reputation System

Considering users' ranking place, calculate normalized discounted cumulative gain which is a metric for measuring ranking correctness by comparing Discounted Cumulative Gain (DCG) to the ideal ranking. DCG measures the correctness of a ranked list based on users' reputation discounted by their place in the splay tree. Higher values of NDCG indicate better ranked lists and therefore better correctness [33]. DCG measures the results' quality in a ranked list provided by the Stackoverflow community. Computed metrics represent true differences in performance between users. The cumulative gain is the sum of the scores for each user's position in the splay tree. The Discounted part of NDCG sum up the score divided by the rank. In practice, the score is often divided by the log of the rank, which seems to better match the user reputation. Cumulative Gain (CG) is the predecessor of DCG and does not include a result position in the usefulness of a result set consideration. Changes in the ordering of search results unaffected the value computed with the CG function. DCG is in place of CG for a more exact measure.

3.1.1. Discounted Cumulative Gain

The premise of DCG is to penalize the highly reputed users appearing lower in a search result list as the graded reputation value reduces which is logarithmically proportional to the result position.

$$DCG_k=\sum_{i=1}^{k} 2rep_i-1/log_2(i+1)$$

The Normalized part in NDCG allows comparing DCG values between different users. The best ranking is known as the "ideal DCG," or iDCG. *iDCGk* is the most possible (ideal) *DCG* for a given set of reputation,

$$nDCG_k=DCG_k / iDCG_k$$

ALGORITHM: Normalized_Discounted_Cumulative_Gain (user_id, user_position)
/ Measure the gain of a user (contributor) based on his/her place, in the splay tree. OptimalDCG is the ideal (greatest*

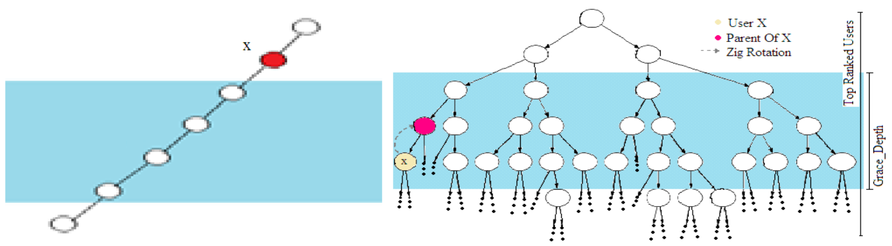

```
possible) dcg. The ndcg varies from 0.0 to 1.0, with 1.0 representing the ideal ranking of the entities. Let k be the
maximum number of users ranked */
{dcg = CalculateDcg(parameters.K,user_ position);
                optimal=CalculateOptimalDCG(parameters.K, user_position);
                ndcg = dcg / optimal;
                        if (optimal <= 0) {
                   ndcg = (dcg == optimal) ? 1.0 : 0.0;}
  }
    function CalculateOptimalDCG(int k, user_reputation){
          return CalculateDcg(k, user_ position.Values.OrderByPosition (i => i));    }
    function CalculateDcg(int k,  user_ position) {
          i = 1;
          dcg = 0.0d;
          foreach (user_ position in  range k) {
     dcg += (Math.Pow(2, user_ position) - 1.0) / Math.Log(1 + i, 2);
             i++;        }
          return dcg;    }
    }
```

Compute NDCG for users in the splay tree. Select users whose rank drops from their earlier position when ndcg of that ranking list is less than 0.5. That is, if ndcg<0.5 then select users with difference between current and previous user_position is greater than fixed value, say n. Semi-splay the selected users. Select low ranked users using NDCG method as stated above, considering n as 3. Rotate tree so that selected users' are taken up two steps forward in ranking. Breaking_Power_Law algorithm improves these user positions in t by rotating the tree assuming that user's parent as the root.

IV. BREAKING POWER LAW

Power law implies that very few users are ranked high, and a large number of users are listed at the middle level and very huge at the low level. Power distribution shows that very low scored elements are massive in numbers, in other words, small occurrences are extremely common, whereas significant instances are extremely rare. A power law is a pragmatic relation between the frequency of an event '*f*' and size of the event 'S' with fixed power 'p' and constant 'c'. That is, $f = cS^{-p}$.

Break the power law by boosting up low ranked users through rotation that push those users two steps forward in ranking. Breaking_Power_Law algorithm improves the weak user position in t by rotating the tree assuming that user's parent as the root. The grace-upper and grace-lower are the upper and lower levels in t with constant values. The grace-depth is the range between the grace-upper and grace-lower. The tolerate-factor represents the number of times to do a Zig/Zag operation to improve a weak user's rank.



(a) (b)

Figure.1. (a) Position of user x in ℓ (b) Zig rotation to increase the ranking of user x in t

The tolerate-factor depends on the rank of the user in ℓ (pos1) and t (pos2) as shown in Figure.1. Let us consider max-tolerate-factor=3, mid-tolerate-factor=2 and min-tolerate-factor=1. Increase the ranking of a user by doing an additional splay operation, as shown in the Figure.1 (b), considering user x's parent as a maximum ranked user with subtree. The tolerate factor is the number of times to rotate the x. Calculate the tolerate factor for the x using grace-upper and grace-lower values, where grace-upper is the upper limit and grace-lower is the lower limit of the ranks. Let a[] be list of weak users and 'count' counts the number of times the element appears in weak user list. Store user x in the weak user list.

ALGORITHM: Breaking_Power_law(user x)

if (t.Normalized_Discounted_Cumulative_Gain(Id, Pos) <0.5) then
for each user x in t do
if ((x.currentPosition –x.previousPosition) >3) then

```
pos1=Searching(ℓ,x)
pos2=Searching(t,x)
tolarate-factor=0, max-tolerate-factor=3, mid-tolerate-factor=2, min-tolerate-factor=1
if pos2 >= grace-lower  and pos2 <= grace-upper then
      mean=(grace-upper + grace-lower)/2
        if pos2<=mean and pos1 <= grace-lower then tolarate-factor=max-tolerate-factor
        if pos2<=mean and pos1 > grace-lower then tolarate-factor=mid-tolerate-factor
        if pos2>mean then tolarate-factor=min-tolerate-factor
a[i]=x; i++;
for  j=0 to i do
            if a[j]=x then count= count+1
          if count>tolerate-factor then
                 print "The element cannot be rotated anymore"
/*Move parent below child and one of child's children below parent using 'Zig/Zag' operation for the splay tree t. Let 'parent-of-x' be the parent of user x*/
         else { if parent-of-x.left = x then
                        parent-of-x.Left=x.right
                        x.Right=parent
                 else  parent-of-x.Right=x.left
                        x.Left=parent}
```

The splay time at a node 'n' is proportional to the time to reach an item in node 'n'. Though the size of the tree grows as the number of active users' increases, the depth of the tree is based on the grace-depth value. In other words, Zig/Zag operation occurs only when the searching user's position is within the specified grace-depth. Search user 'x' in t1within grace-depth in a top-down approach and in t2 through in-order tree traversal method. The grace-upper represents the maximum level in the splay tree, t1, to consider for searching 'x'. The largest number of nodes in a binary tree of depth grace-upper is $2(grace\text{-}upper)-1$ where $grace\text{-}upper \geq 1$. The splay tree with $2(grace\text{-}upper)-1$ nodes take at least $O(\log (2(grace\text{-}upper)-1))$ comparisons to find a particular node and the total amortized time for a sequence of m operations is $O(m \log (2(grace\text{-}upper)-1))$.

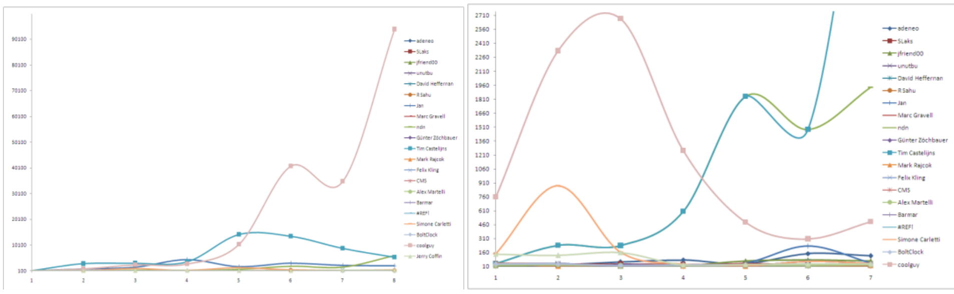
5. Experimental Results and Discussions

Reputation is a user's identity that reflects user's familiarity with the website, the amount of users' subject knowledge and the level of respect peers have on the user. Sometimes reputation also determines a user's privileges within the system. Gaining more reputation and trust can make a user access give new functionality. As user gain reputation, they gain abilities and responsibilities. The primary factors that determine reputation is users' voting. Up-voted posts increases users' status, the reverse is true for posts which are down-voted. User's up-votes are more heavily weighted than down-votes. Reputation lost from the reputation cap is not awarded on the following days in the Question-Answering websites such as StackOverflow. Reputation cap is to prevent users from gaining privileges and trust too quickly. StackOverflow badges are similar to ranking, awarded to users for achieving an individual score in a specific tag. Tag score is the combined total number of up-votes and down-votes accumulated on answers under that particular tag. Tumbleweed badge in StackOverflow is to bring attention to a neglected user and this encourages people to stay on the website. Our Breaking_Power_Law algorithm is much similar to the tumbleweed badge, but neglected users' ranking increases unknowingly to other users/voters. Data collected from the StackOverflow website with user id, reputations, latest answered/questioned time and level. Live data collected from the StackOverflow website for 25 users as shown in the table 1, with first 4 digits of user reputation scores and their level (position) in the splay tree. Compute NDCG for collected data with users' position in the splay tree. Calculated NDCG indicate the quality of ranking.

When ndcg<0.5, then pick weak users based on their position in splay tree. Find the difference between current and previous position of users when ndcg<0.5. If this difference is greater than n, in our experiment we considered n=3, then semi-splay these users using Zig/Zag rotation. Below table shows the user performance improves remarkably after applying semi-splay (zig/zag) operations through Breaking_Power_Law algorithm. User F level in tree has not increased as user M and S. This proves that early application of Breaking_Power_Law algorithm improves users' performance and participation interest in a better way. Calculated ndcg is 0.405797<0.5 for collected data '4'. Construct s splay tree based on reputation list 4 and users' response/activity time, that is, hours ago. In the ranking list 4, ndcg is less than 0.5, thus find the differences among present and previous users position in the splay tree and spot users with [(current_position)-(previous_position)] >3. Users satisfying the above conditions in our data are F, M, N, P and S. Semi-splaying these users' shows that their ranking quality is much improved.

In our work, the Breaking_Power_Law algorithm is implemented and compared with the existing Question-Answering ranking algorithm. Finding weak users and boost up their levels to break power law is the main goal of our algorithm. The implementation result shows that the poor users are identified using NDCG and saved from further drowning to the lower level of the splay tree. Boosting up of participants' degree in the splay tree yields them the chances to get active. Thus, these users may become active again. In the traditional ranking methods, the weak participants are not in the limelight, and results in high ranked members alone to grab voter's attention, and thus, only those groups of participants alone stay in the top-level of the ranking lists, that is, in the ranking cap. Our algorithm overcomes this and breaks power law. The studied statistical data of StackOverflow from the tail of top 50 weekly ranked users' list conclude that nearly 75% of users' weekly ranks drop down to the lower levels. Users ranked 35, and above are in the critical place of dropping steep into the ranking list. Figure.2 shows StackOverflow users' weekly rank report for the months January and February 2017. This clearly proves that if users get low reputation or ranking they fall steep into the ranking list.

Consider grace-lower as 35 and grace-upper as 45. The users in grace-depth are zig-zig or zag-zag rotated as they are the left or right child of their parents respectively. So that they can be moved two levels high from their current position, just above their parent node, thus preventing them from dropping steep into the list for a limited number of times. If user 'x' is within the grace-depth then the user is zig/zag rotated. Thus, her parent 'y' becomes her child in the splay tree. Though the rank changes, the parent's reputation, and trust values remain unchanged, thus, the chances of losing up-votes and voters trust are very less. In other words, y's probability to get pushed down in the ranking list is very less. Now, user y's expected reputation value in the next move becomes x's expected reputation value.



(a) (b)

Figure .2 Weekly rank of StackOverflow users from Jan to Feb '16 (a) Before applying Breaking_Power_law algorithm (b) After applying Breaking_Power_law algorithm

The experimental results show that our algorithm has increased weak users' performance by increasing their chances to remain in the ranking cap by breaking the power law. Figure.3 depicts the changes in user ranks before and after breaking the power law through semi-splay. In our work, 85% of users retain in higher ranks which are almost 60% more than the existing question-answering website.

6. Conclusion and Future Work

The collaboration activities in e-learning environment require trust relationships among co-learners that obtained through reputation and ranking. The splay tree is a self-adjusting binary search tree where highly active/ranked users are near the root node. In our paper, the splay tree represents user reputation. Arrange users in the splay tree based on their participation frequency. Highly active user occupies the root node. Calculate NDCG and identify weak users, who are all in the critical level of getting ignored or losing the trust of voters. To break the power law, do zig/zag rotation which raises their grade levels to save them from getting very low ranks. In our work, we examined existing StackOverflow data with our algorithm and proved that 60% of users get saved from drowning. In our future activities, group users based on the subject of expertise and accordingly ranked using multiple splay trees. One user may have more than one subject of interest. Thus, represent a user as a node in more than one splay tree. These users connect the splay trees forming a splaynet. Ranking in splaynet is our future goal.

References

[1] Wei Zhou, Zilong Tan, Shaowen Yao and Shipu Wang. "A Splay Tree-Based Approach for Efficient Resource Location in P2P Networks. The Scientific World Journal," Article ID 830682. 11 pages. http://dx.doi.org/10.1155/2014/830682. Research Article. 2014
[2] Stefan Schmid, Chen Avin, Christian Scheideler, Michael Borokhovich, Bernhard Haeupler and Zvi Lotker. "SplayNet: Towards Locally Self-Adjusting Networks," IEEE/ACM TRANSACTIONS ON NETWORKING. IEEE. 1063-6692. 2015.
[3] Liangbin Li, Jinlin Wang, Erli Niu and Xue Liu. "Improving Content Authentication Efficiency of Online

Multimedia Service. Second International Conference on Networks Security. Wireless Communications and Trusted Computing," 978-0-7695-4011-5/10 IEEE. DOI 10.1109/NSWCTC.2010.171. 2010.
[4] Dimitris Antoniou, Ioannis Kampouris, Evangelos Theodoridis and Athanasios Tsakalidis. "Splaying for Compression: An Experimental Study," Panhellenic Conference on Informatics 978-0-7695-4389-5/11. IEEE.DOI 10.1109/PCI.2011.19. 2011
[5] Haopei Wang, Zhen Chen, Feng Xie and Fuye Han. "A Data Structure for Content Cache Management in Content-Centric Networking," Third International Conference on Networking and Distributed Computing. 2165-5006/12 IEEE. DOI 10.1109/ICNDC.2012.11. 2012.
[6] Zouheir Trabelsi and Safaa Zeidan. "Splay Trees based Early Packet Rejection Mechanism against DoS Traffic Targeting Firewall Default Security rule," Iguacu Falls. Brazil. ISBN: 978-1-4577-1017-9. DOI http://doi.ieeecomputersociety.org/10.1109/WIFS.2011.6123123. pp:1-6. 2011.
[7] Safaa Zeidan and Zouheir Trabelsi. "A Survey on Firewall's Early Packet Rejection Techniques. International Conference on Innovations in Information Technology," 978-1-4577-0314-0/11 IEEE. 2011.
[8] A Mercy Gnana Rani and A Marimuthu. "Key Insertion and Splay Tree Encryption Algorithm for Secure Data Outsourcing in Cloud.," World Congress on Computing and Communication Technologies. 978-1-4799-2877-4/14 IEEE. DOI 10.1109/WCCCT.2014.14. 2014.
[9] Chen Avin, Bernhard Haeupler, Zvi Lotker, Christian Scheideler and Stefan Schmid Ben Gurion. "Locally Self-Adjusting Tree Networks.," IEEE Computer Society. pp: 395-406. DOI 10.1109/IPDPS.2013.40. 2013.
[10] Zouheir Trabelsi and Safaa Zeidan. "Multilevel Early Packet Filtering Technique based on Traffic Statistics and Splay Trees for Firewall Performance Improvement," IEEE ICC - Communication and Information Systems Security Symposium. 978-1-4577-2053-6/12 IEEE. 2012.
[11] Neil Y. Yen, Franz F. Hou, Louis R. Chao and Timothy K. Shih. "Weighting & Ranking the E-Learning Resources," Ninth IEEE International Conference on Advanced Learning Technologies. 978-0-7695-3711-5/09 IEEE. DOI 10.1109/ICALT.2009.36. 2009.
[12] Somaya Arianfar. Pekka Nikander. and Jörg Ott. "On content-centric router design and implications," ACM CoNext Workshop ReARCH'10 Proceedings of the Re-Architecting the Internet Workshop. ACM New York. Article No. 5. pp: 20. 23. 24. ISBN: 978-1-4503-0469-6 DOI 10.1145/1921233.1921240 . 2010.
[13] R. Jayashree and A. Christy (2014) 'Privacy and Reputation in Context Aware e-Learning', International Journal of Computer Applications (0975 – 8887) 3rd National Conference on Future Computing February 2014. PP.27-31.
[14] Daniel Dominic Sleator and Robert Endre Tarjan. "Self-Adjusting Binary Search Trees. Journal of the Association for Computing Machinery," Vol. 32. No. 3. pp. 652-686. A T&T Bell Laboratories. Murray Hill. NJ 0 1985 ACM 0004-541 1/85/0700-0652. 1985.
[15] M. Waldvogel. G. Varghese. 1. Turner. B. Plattner. "Scalable High Speed IP Routing Lookups," In Proceedings of the ACM SIGCOMM (SIGCOMM '97). pp. 25-36. ISBN: 0-89791-905-X DOI: 10.1145/263105.263136. 1997.
[16] J. Dıaz, J. Petit and M. Serna. "A survey of graph layout problems," Journal ACM Computing Surveys (CSUR) Surveys Homepage archive. Volume 34, Issue 3, pp: 313-356. ACM New York. NY. USA DOI: 10.1145/568522.568523. 2002.
[17] L. H. Harper. Optimal assignment of numbers to vertices. J.SIAM . (12):131–135. 1964.
[18] J. Leitao, J. Marques, J. Pereira and L. Rodrigues. "X-bot: A protocol for resilient optimization of unstructured overlay net-works," IEEE Transactions on Parallel and Distributed Systems. Vol 23. pp: 2175-2188. 2012.
[19] G. Mitchison and R. Durbin. "Optimal numberings of an N X N array," SIAM J. Algebraic Discrete Methods. Volume 7. No.4. pp:571–582. 1986.
[20] R. Ravi, A. Agrawal and P. N. Klein. "Ordering problems approximated: Single-processor scheduling and interval graph completion," In Proc. ICALP. 1991.
[21] T. Srinivasan, M. Nivedita and V. Mahadevan. "Efficient Packet Classification Using Splay Tree Models," IJCSNS International Journal of Computer Science and Network Security. Volume 6. No.5. pp. 28-35. 2006.
[22] Viraj Anchan, Sarang Deshpande, Deep Doshi and Akshat Kedia. "Ranking Algorithm. International Journal of Advanced Research in Computer and Communication Engineering," Vol. 4. Issue 4. Copyright to IJARCCE DOI 10.17148/IJARCCE.2015.4456 248. 2015.
[23] Muralidhar Medidi And Narsingh Deo. "Parallel dictionaries on AVL trees. Parallel Processing Symposium. Proceedings," Eighth International. IEEE ISBN:0-8186-5602-6. pp: 878 – 882. DOI: 10.1109/IPPS.1994.288203. 1994.
[24] Aleksandrs Slivkins, Filip Radlinski and Sreenivas Gollapudi. "Ranked Bandits in Metric Spaces: Learning Diverse Rankings over Large Document Collections," Journal of Machine Learning Research. 14 pp: 399-436. 2013.
[25] Robert Kleinberg, Aleksandrs Slivkins and Eli Upfal. "Multi-armed bandits in metric spaces," In 40th ACM Symp. on Theory of Computing (STOC) . Pages 681–690. 2008.
[26] Mohd Anwar and Jim Greer. "Facilitating Trust in Privacy-Preserving E-Learning Environments," IEEE Transaction on Learning technologies; VOL. 5. NO. 1. 1939-1382/12 IEEE. 2012.
[27] R. Jayashree and A. Christy (2015) 'Improving the Enhanced Recommended System Using BayesianApproximation Method and Normalized Discounted CumulativeGain', 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15), Procedia Computer Science 50 (2015) 216 – 2221877-0509 © 2015.

Published by Elsevier
[28] X L Liu. "Ranking online quality and reputation via the user activity," Research Center of Complex Systems Science. University of Shanghai for Science and Technology. Shanghai 200093. PR China. 2015. http://dx.doi.org/10.1016/j.physa.2015.05.043. 2015.
[29] D. Movshovitz. "Analysis of the reputation system and user contribution on a question answering website: stack overflow," IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. ASONAM'13. Niagara. Ontario. CAN Copyright 2013 ACM 978-1-4503-2240-9 /13/08. 2013.
[30] Adler B, de Alfaro L. "A content-driven reputation system for the Wikipedia," Proc. Of the 16th Intl. World Wide Web Conf. (WWW 2007); ACM Press. 2007.
[31] Adler B, de Alfaro L. Pye I, Raman V. "Measuring author contributions to the Wikipedia," Proc. of WikiSym 08: International Symposium on Wikis; ACM Press. 2008.
[32] Druck G, Miklau G and McCallum A. "Learning to predict the quality of contributions to Wikipedia," Proceedings of AAAI: 23rd Conference on Artificial Intelligence. 2008.
[33] Martin P. Robillard, Walid Maalej, Robert J. Walker, Thomas Zimmermann (2013) 'Recommendation Systems in Software Engineering', ISBN 978-3-642-45134-8 ISBN 978-3-642-45135-5 (eBook). DOI 10.1007/978-3-642-45135-5. Springer-Verlag Berlin Heidelberg.
[34] Lada A. Adamic(2002) 'zipf, power-laws and pareto-a ranking tutorial',. Information Dynamics Lab, HP Labs. Palo Alto, CA 94304. http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html