

## Enhanced User-driven Ranking System with Splay Tree

R. Jayashree\*, A.Christy

Sathyabama University, Chennai, India

\*Corresponding author, e-mail: jayashreeram77@gmail.com

### Abstract

*E-learning is one of the information and communication technology products used for teaching and learning process [35]. An efficient and effective way to construct trust relationship among peer users in e-learning environment is ranking. User-driven ranking systems are based only on the feedback or rating provided by the users. In [46-48] the authors provide a variety of trust and reputation methods. Certified Belief in Strength (CBS) [45] is a novel trust measurement method based on reputation and strength. In [38] author presents a recommendation system based on the relevant feedback review to predict the user's interests, that are ranked based on the recommendations history they provide previously. Users with higher rating obtain high reputation compared to less scored users. In question answering websites like StackOverflow, new or low scored users are ignored by the community. This discourages them and their involvement with the community reduces further down, as power law states, alleged low users are pushed to the bottom of the ranking list. Avoid this condition by encouraging less reputed users and prevent them from moving further down in ranking level. Thus, low reputed users are provided with few more chances to participate actively in the e-learning environments. A splay tree is a Binary Search Tree with self-balancing skill. The splay tree brings the recently accessed item to the top of the tree, thus active users are always on the top of the tree. A splay tree is used to represent user's ranks, and to semi-splay low ranked users again in the tree thus preventing them from further drowning in the ranking list. The focus of this research work is to find and enhance low reputed users in reputation system by providing few more chances to take part actively in the e-learning environment using the splay tree. Normalized discounted cumulative gain (NDCG) acts 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

**Copyright © 2018 Universitas Ahmad Dahlan. All rights reserved.**

### 1. Introduction

In e-learning environments, co-learners trust relationships play a vital role to establish collaborative activities. An overview of [35] demonstrates the utilization of existing information technology in the teaching and learning process. Here, the author focuses to develop quality e-learning software with ability to operate in many electronics. For collaboration activities in e-learning environments, the trust relationships among co-learners are imperative. An individual's privacy diminishes by expectations of trust [27]. Reputation is an effectual source for measuring trust, and it is obtained through user rating or ranking. A context-aware personalized recommendations system in [12] analyzes the user's context while they access the multimedia information. This method shows an experimental use of latent preferences for ranking items. Reputation is a contextual evaluation of a person's actions [27]. Privacy and protection rights are the key challenges that are needed to tackle when capturing and using contextual data. The two reputation systems are content-driven and user-driven. Content-driven reputation systems are based on the feedback provided by analysis of all interactions whereas user-driven reputation systems are based only on the feedback or ratings provided by the users. The splay tree brings the newly accessed items closer to the top of the tree. The locality of reference states that 80% of the accesses are to 20% of the items. A splay tree search operation moves searched user to the top of the tree. If the search is successful, then that user is splayed and becomes the new highest rated user. Else the last user accessed before reaching the NULL is splayed and becomes the new highest rated user. 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 is accessed once, then the same user X is likely to be accessed again. After locating user X, perform "splaying" operations to bring up X to the top of the tree. Do this in a way that

leaves the tree more or less balanced as a whole. Move active (recently accessed) user towards the root and inactive users slowly far-off from the root. Let user X is not a maximum-Ranked-User, that is, X has, at least, one ancestor, then just rotate X once [24]. Zig and Zag are single rotation operation in a splay tree. "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 searches does not occur because each search operation causes a rebalance. StackOverflow is a well known Question-Answering website that allows registered users to post their questions and to post answers to others' questions [23]. In general, users with good answers are ranked high. In all Question-Answering websites users with the highest reputation scores are marked as highly reputed users. Overall rating depends on the ratings of users with low reputation. The higher the vote weight of a user with great reputation compared to a vote of a low reputation user, the less the overall reputation will change due to the low reputation votes. In this paper, users with the fewer score are shuffled for a limited number of times so that they do not get ignored in the top reputation list. In other words, moderately reputed or less active users are given few more chances to participate 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 the quality of the results in a ranked list [17]. Contributors' reputations are predicted. Calculate users' overall rating and ranking and verify ranked contributors and their reputation scores using NDCG algorithm [17]. 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 the 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 discusses splay tree and NDCG in the 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.

## 2. Motivation and Background

In general, the study of the evaluation of ICT-based learning in SMA [26] and the effectiveness of learning with the help of internet in SMK [36] resulted in the fact that community awareness of the use of e-learning improved the qualifications and the attitude of co-learners. Reusability and maintainability qualities of FUOLC's e-learning [35] helps in the development of a software to anticipate the co-learners' need. Fuzzy based context method predicts the relevant instances among the relevant reviews and the combination of association rules and ontology mining carries out textual analysis [37]. Semantic analyzer compares the relationship between review and their context based on the fuzzy rules [37]. Over the data taxonomies, review comparison on distributed data can be efficiently carried out with association rules like Apriori [39]. In [37] the authors discusses on different types of association rule mining and the category of databases on which way the association rules are assigned.

Personalized summarization [10] proposes to customize the user's review history based on the customer's previous browsing history or by user's credentials. Click through approach [37] extracts the features of the users. In [37] the author finds the reviews related to the desired products for the current user and classify their opinions (positive or negative). Based on the user behavior [40] propose a user based contextual recommendation system in which author extracts the user behavior based on their preference cart by constructing clustered trees for items. In [41] the author analyzes the e-commerce websites and provides the recommendation to users based on the comparative results of the products. The result shows that context-aware systems mostly provide better recommendation than the available benchmark system. In [37] propose a service-based framework using hierarchical ontology which distributes the concepts of services hierarchically. In the review of literature [37] analysis the extracting features of context and the opinion of the given context. In this study, ontology helps in predicting the implicit knowledge of users. Context-based recommendation system [37]

mines the context from the user's reviews and discovers the level of associations between the reviews and links them. Evaluating Amazon dataset in [37] with the benchmark performance measures, precision, and recall, shows that the association rules are quite appropriate for review recommendation systems to provide the efficient classification. An ensemble model [43] reaches a more accurate decision by combining the decisions of a group of individual classifiers. As reported in the literature, using multiple individual classifiers (or base classifiers) in an ensemble model [49, 50], multiple base classifiers produce reliable and accurate predictions. The CBS method [45] proves to improve the accuracy rates of a number of base classifiers, FMM network [44]. Application of the ensemble model (MACS-CBS) in [45] is further evaluated using a group of EFMM classifiers [43]. Deploying a group of classifiers for decision making in an ensemble model allows to produce a more reliable and accurate predictions [49]-[50].

According to the 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]-[32]. User status is evaluated to predict the quality of future user contributions [30]. The predictive ability of the content reputation system is used to measure 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 importance of detecting anomalous users in his work. The potential expert users are identified 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 accurate reputation ranking lists [28]. SChord [1] is based on splay tree and implements Chord finger table with improved resource locating efficiency. Nodes hierarchy is related 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 contains a splay tree. Insert node  $n$  with  $s$  successors  $(n.id+2i)$ ,  $0 \leq i \leq s-1$  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 Merkle hash tree (MHT) is constructed 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. An element is verified using 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.

In [4], Randomized splay tree version is presented with chain splay technique for compressing data. An adaptive data compression algorithm called as the splay-prefix algorithm on the prefix code, where the code tree is restructured using semi-splaying. In semi-splaying technique leaf corresponding to the transmitted symbol is splayed 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] and self-adjusting tree [13, 21] ideas are implemented to design a cache management for Content-Centric Networking (CCN) [5]. The download time is related to the class popularity that is, the download time is very short for the content with high probability to access. The frequency of visits and the recent visit are considered 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 is used to the store length of the prefixes. The level of access determines binary search on prefix lengths [15]. Statistical Splay Tree Policy Filters (SSF-BSPL) [6] 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 [17]. 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 the website which later ordered to find ranking among users [17]. The registered users gain more mean points than anonymous users and registered users are more trusted [17].

In Splay Tree Packet Classification Technique (ST-PC) [7] integer values with their matching rules are stored in splay trees. Whereas in Self-Adjusting Binary Search on Prefix Length (SA-BSPL) [22], the prefix lengths and their corresponding hash tables with matching rules are stored which gives better-amortized analysis. System performance is affected significantly 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 filtering processing time is reduced. Key Insertion and Splay Tree encryption (KIST) [8] algorithms use the splay tree for encryption. Key injection algorithm in the cipher text is to compress and move inner nodes which are higher than the 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 moved closer together. Sleator and Trajan [21] proposed splay tree as optimized binary search tree which reduces average access time by moving more popular nodes closer to the root. In [16] introduce Minimum Linear Arrangement (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. In this scheme, unwantedly traffics are rejected in the early stages and thus decrease overall packets matching time. Statistical Splay Tree Policy Filters (SSF-BSPL) [10] system uses a mathematical model to decide statistical policy fields order for the next packet segment. Learning objects in the repository are ranked based on the citation numbers similar to Google page rank [11]. In Slivkins et al.'s [25] work, relevant documents are selected so as to obey the expected relevance rate  $\mu(x)$ , distributed according to a power-law, for each document  $x$ .

### 3. 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 run in  $\theta(\log m)$  worst-case time. Let  $\ell$  be the reputation-based sorted list of users where the 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 the searching time of user  $x$  in  $\ell$ . 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  $\ell$ . 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.

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

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

$$nDCG_k = DCG_k / iDCG_k$$

#### 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 the difference between current and previous user\_position is greater than a 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. Let  $T$  be a splayed tree, then  $T_i(u)$  is the subtree of node  $u$  in step  $i$ . Rank of  $u$  in level  $i$  of splaying,

$$R_i(u) = \log_2 (\text{SizeOf}(T_i(u))) \quad (1)$$

where  $\text{SizeOf}(T_i(u))$  is the sum of weights of all elements in the subtree rooted in  $u$ . In other words,

$$\begin{aligned} 2R_i(u) &= \text{SizeOf}(T_i(u)) \\ \text{if } u &= \text{leaf then } R_1(u) = 0; \\ \text{if } u &= \text{root then } R_n(u) = \log_2 n \end{aligned} \quad (2)$$

Where  $n$  is the total number of nodes in  $T_i$ . Time complexity depends on the height of the  $T$  and amortized time complexity depends on the rank of the  $T$ . The potential of a  $T$ ,  $P_t(T)$ , is the sum of ranks of all its nodes.

$$P_t(T) = \sum_{i=0}^n R_i \quad (3)$$

Let  $d$  be the depth of the node before and after splaying,

$$d = |\text{splay}_i - \text{splay}_{(i-1)}| \quad (4)$$

$\text{splay}_i$  and  $\text{splay}_{(i-1)}$  are user move before and after the splay operation respectively. Searching a key takes  $d$  time. Splaying node  $x$  consist of  $d/2$  splaying sub steps. Change in the user move after the splay operation is  $[\text{splay}_i - \text{splay}_{(i-1)}]$ . This is computed only for  $u$ , parent  $u$  and grandparent  $u$ .

Amortized Complexity,

$$A_{c_i} = w_i + \text{splay}_i - \text{splay}_{(i-1)} \quad (5)$$

$$A_{c_i} = w_i + d \quad (\text{From Equation 4}) \quad (6)$$

Here  $w_i$  is work done for the splay operation, which is computed as number of levels the target node rises or falls during a splay operation. Total user move of a  $T$  in  $i^{\text{th}}$  step of splaying,

$$\text{splay}_i = \sum_{u \in T_i} R_i(u) = \sum_{u \in T_i} \log_2 (\text{SizeOf}(T_i(u))) \quad [\text{from (1)}] \quad (7)$$

If step  $i$  initiates semi-splay operation then Amortized Complexity of splay tree is,

$$A_{c_i} < 1 + (R_i(u) - R_{i-1}(u))$$

In  $T$  with  $n$  nodes, Amortized cost  $C(n)$  of a search with splaying does not exceed  $(1 + 3 \log_2 n)$  upward moves from the specific node.  $C(n)$  is the number of rotation and  $C(n) = 1$  if no rotation occurs.

Amortized Cost of a semi-splay is at most  $1 + 3d$ , if it takes only one rotation.

Amortized time to splay a tree with root  $t$  at a node  $x$  is at most,

$$3(R(t) - R(x)) + 1 = O(\log(\text{SizeOf}(T_n(u)) / \text{SizeOf}(T_x(u)))) \quad 1 \leq x \leq d/2$$

Therefore, total access time is  $O((m+n) \log n + m) = C(n) + P_t(T)$  [From (2) and (3)]

That is the sum of amortized cost and potential difference, where  $n$  is the number of elements in the  $T$  and  $m$  is the number of accesses.

If  $a, b > 0$  and  $c > a + b$  then  $\log a + \log b \leq 2 \log c - 2$ . Total Amortized complexity  $\leq \sum (R_i(x) - R_{i-1}(x)) + 1$ ,  $1 \leq i \leq d/2$ . For semi-splay,

$$3(R(t) - R(x)) + 1 \leq 3(R(t)) + 1 = 3 \log n + 1$$

Thus splaying operation  $m_1$  and semi-splaying operation  $m_2$  takes  $O((m_1+m_2) \log n)$  time complexity which is better than  $O(mn)$  or  $O(n)$  worst case complexity of traditional ranking algorithm.

**4. 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 the rotation that pushes 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.

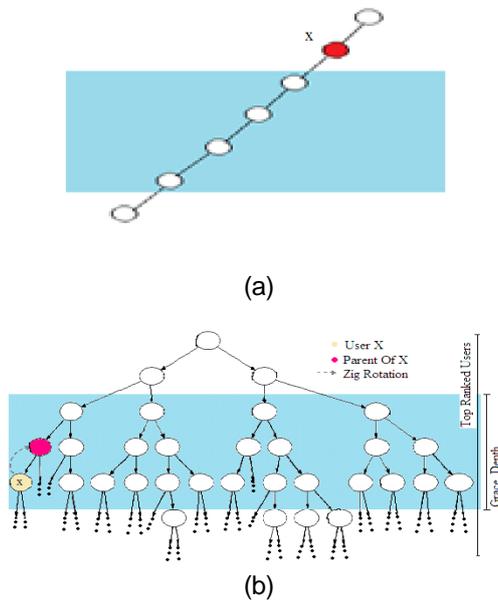


Figure 1. (a) Position of User  $x$  in  $\ell$  (b) Zig Rotation to Increase the Ranking of User  $x$  in  $t$

The tolerate-factor depends on the rank of the user in  $\ell$  (pos1) and  $t$  (pos2) as shown in Figure 1.

- a. If pos2 is in between the grace-upper and grace-lower values, then calculate 'mean' value of grace-upper and grace-lower.
- b. If pos2 is lesser than mean value and pos1 is lesser than grace-lower value, then the tolerate-factor is the max-tolerate-factor.
- c. If pos2 is lesser than mean value and pos1 is greater than grace-lower value, then tolerate-factor is mid-tolerate-factor.
- d. Otherwise, tolerate-factor is min-tolerate-factor.

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 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 a list of weak users and 'count' counts the number of times the element appears in the weak user list. Store user  $x$  in the weak user list.

```

Breaking_Power_law(user x)
if (t.Normalized_Discounted_Cumulative_Gain(ld, 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)
tolerate-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 tolerate-factor=max-tolerate-factor
    if pos2 <= mean and pos1 > grace-lower then tolerate-factor=mid-tolerate-factor
    if pos2 > mean then tolerate-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  $t_1$  within grace-depth in a top-down approach and in  $t_2$  through in-order tree traversal method. The grace-upper represents the maximum level in the splay tree,  $t_1$ , to consider for searching 'x'. The largest number of nodes in a binary tree of depth grace-upper is  $2^{(\text{grace-upper})} - 1$  where  $\text{grace-upper} \geq 1$ . The splay tree with  $2^{(\text{grace-upper})} - 1$  nodes take at least  $O(\log(2^{(\text{grace-upper})} - 1))$  comparisons to find a particular node and the total amortized time for a sequence of  $m$  operations is  $O(m \log(2^{(\text{grace-upper})} - 1))$ .

## 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 reputations are users' voting. Up-voted posts increase 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 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 indicates the quality of ranking.

When  $ndcg < 0.5$ , as shown in the Figure 2, then pick weak users based on their position in the splay tree. Find the difference between the 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. Figure 2 displays NDCG for all collected data before and after semi-splay. Below table shows the user performance improves remarkably after applying semi-splay (zig/zag) operations through the Breaking\_Power\_Law algorithm. User F level in the 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 a 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.

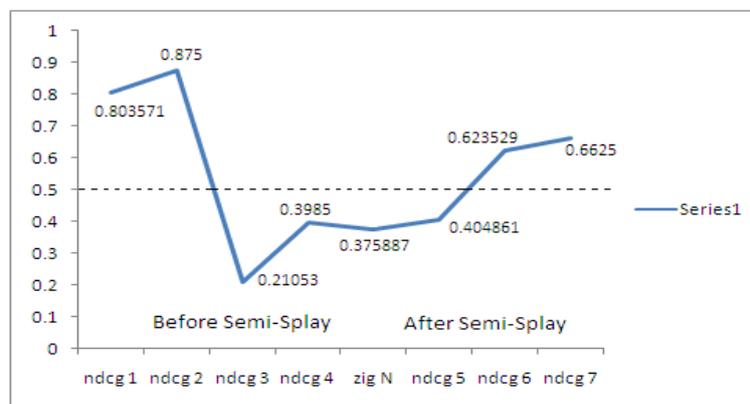


Figure 2. NDCG of Users Before and After Semi-Splay

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 result 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 4 (a) shows StackOverflow users' weekly rank report for the months January and February 2016. This clearly proves that if users get low reputation or ranking they fall steeply 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 as shown in figure.3. 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.

Table 1. Live Data Collected from StackOverflow Website with User Level in Splay Tree Before and After Semi-Splay

User ID	Collected data before semi-splay					Collected data after semi-splay
	Data 1	Data 2	Data 3	Data 4	Zig/Zag	Data 5
A	4	1	3	6	6	7
B	5	4	6	8	8	10
C	6	2	2	1	1	3
D	12	9	6	4	4	5
E	9	8	7	8	8	4
F	2	4	5	9	7	10
G	5	5	6	9	9	3
H	8	7	4	6	6	6
I	11	7	7	5	5	6
J	10	8	1	4	4	2
K	5	4	3	2	2	4
L	12	8	3	3	3	4
M	3	4	4	9	7	5
N	4	3	5	10	8	9
O	5	6	8	10	10	5
P	8	9	6	10	8	6
Q	6	6	5	7	7	5
R	7	9	9	11	11	6
S	4	5	3	7	5	4
T	11	9	4	3	3	1
U	10	8	5	2	2	2
V	2	2	4	7	7	8
W	5	3	5	8	8	9
X	4	5	2	5	5	6
Y	7	8	5	8	8	7

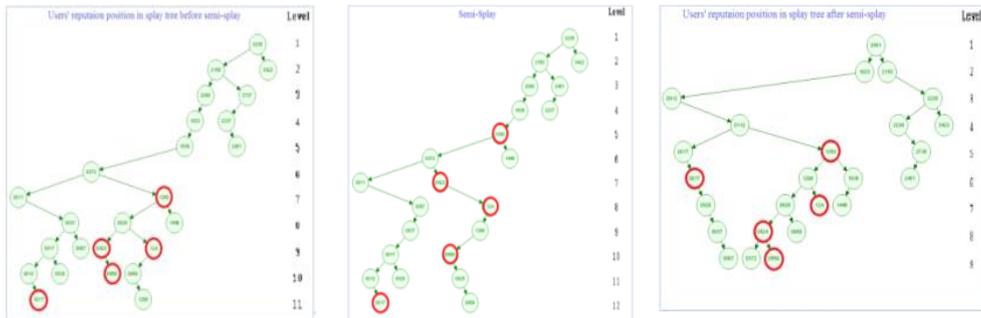


Figure 3. Splay Tree of Stackoverflow Users' Reputation Values

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 4 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 traditional ranking system in question-answering websites.

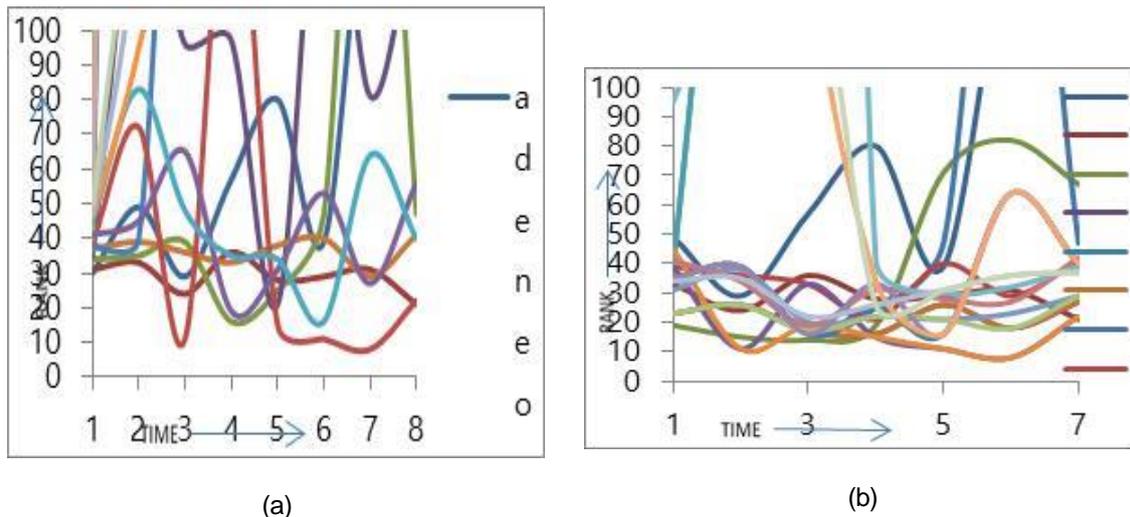


Figure 4. Weekly rank of StackOverflow Users from JAN to Feb '16 (a) Before Applying Breaking\_Power\_law Algorithm (b) After Applying Breaking\_Power\_law Algorithm

## 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 ranking system 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.

## Acknowledgment

The authors would like to thank StackOverflow for providing the data used in this paper.

## References

- [1] Wei Zhou, Zilong Tan, Shaowen Yao, Shipu Wang. A Splay Tree-Based Approach for Efficient Resource Location in P2P Networks. *The Scientific World Journal*. Article ID 830682. <http://dx.doi.org/10.1155/2014/830682>. Research Article. 2014; 11 pages.
- [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, 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, 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>. 2011: 1-6
- [7] Safaa Zeidan, 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, 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*. DOI 10.1109/IPDPS.2013.40. 2013: 395-406.
- [10] Zouheir Trabelsi, 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, Jörg Ott. *On content-centric router design and implications*. ACM CoNext Workshop ReARCH'10 Proceedings of the Re-Architecting the Internet Workshop. ISBN: 978-1-4503-0469-6 DOI 10.1145/1921233.1921240. 2010; 5:20-24
- [13] Jayashree R, Christy A. *Privacy and Reputation in Context Aware e-Learning, International Journal of Computer Applications*. 3rd National Conference on Future Computing. 0975–8887. 2014: 27-31.
- [14] Daniel Dominic Sleator, Robert Endre Tarjan. *Self-Adjusting Binary Search Trees*. *Journal of the Association for Computing Machinery*. A T&T Bell Laboratories. Murray Hill. NJ 0 1985 ACM 0004-541 1/85/0700-0652. 1985; 32(3): 652-686.
- [15] M. Waldvogel, G. Varghese, 1. Turner, B. Plattner. *Scalable High Speed IP Routing Lookups*. In Proceedings of the ACM SIGCOMM (SIGCOMM '97). ISBN: 0-89791-905-X DOI: 10.1145/263105.263136. 1997: 25-36.
- [16] J. Diaz, J. Petit and M. Serna. *A survey of graph layout problems*. *Journal ACM Computing Surveys (CSUR) Surveys Homepage archive*. ACM New York, NY, USA DOI: 10.1145/568522.568523. 2002; 34(3): 313-356.
- [17] Jayashree R, Christy A. *Improving the Enhanced Recommended System Using Bayesian Approximation Method and Normalized Discounted Cumulative Gain*. *Procedia Computer science*, Elsevier. 2015; 50: 216-222.
- [18] J Leitao, J Marques, J Pereira, L Rodrigues. *X-bot: A protocol for resilient optimization of unstructured overlay net-works*. *IEEE Transactions on Parallel and Distributed Systems*. 2012; 23: 2175-2188.
- [19] G Mitchison, R Durbin. *Optimal numberings of an N X N array*. *SIAM J. Algebraic Discrete Methods*. 1986; 7(4): 571–582.
- [20] R. Ravi, A Agrawal, PN Klein. *Ordering problems approximated: Single-processor scheduling and interval graph completion*. In Proc. ICALP. 1991.
- [21] T Srinivasan, M Nivedita, V Mahadevan. *Efficient Packet Classification Using Splay Tree Models*. *IJCSNS International Journal of Computer Science and Network Security*. 2006; 6(5): 28-35.
- [22] Viraj Anchan, Sarang Deshpande, Deep Doshi and Akshat Kedia. *Ranking Algorithm*. *International Journal of Advanced Research in Computer and Communication Engineering*. DOI 10.17148/IJARCCE.2015.4456 248. 2015; 4(4).
- [23] Muralidhar Medidi And Narsingh Deo. *Parallel dictionaries on AVL trees*. Parallel Processing Symposium. Proceedings. Eighth International. IEEE ISBN:0-8186-5602-6. DOI: 10.1109/IPPS.1994.288203. 1994: 878–882.
- [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*. 2013; 14: 399-436.
- [25] Robert Kleinberg, Aleksandrs Slivkins and Eli Upfal. *Multi-armed bandits in metric spaces*. In 40th ACM Symposium on Theory of Computing (STOC). 2008: 681–690.
- [26] Ferdinand S Leuwol. *Evaluasi Manajemen Pembelajaran Berbasis ICT di SMA se kota Ambon*. Tesis: PPs UNY Yogyakarta. 2008.
- [27] Luca de Alfaro, Thomas Adler, Ashutosh Kulshreshtha, Ian Pye. *Reputation system for open collaboration*. *Communications of the ACM*. doi: 10.1145/1978542.1978560. 2011; 54(8): 81–87.
- [28] XL 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. <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. ASONAM13. Niagara. Ontario. ACM 978-1-4503-2240-9 /13/08. 2013.

- [30] Adler B, de Alfaro L. *A content-driven reputation system for the Wikipedia*. Proceedings of the 16th International World Wide Web Conference, (WWW 2007), ACM Press. 2007.
- [31] Adler B, de Alfaro L, Pye I, Raman V. *Measuring author contributions to the Wikipedia*. Proceedings of WikiSym 08, International Symposium on Wikis, ACM Press. 2008.
- [32] Druck G, Miklau G, 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. *Springer-Verlag Berlin Heidelberg*. 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. 2013
- [34] Lada A Adamic. zipf, power-laws and pareto-a ranking tutorial. Information Dynamics Lab, HP Labs. Palo Alto, CA 94304. 2002.
- [35] Irma Salamah, M Aris Ganiardi. Development of E-learning Software Based Multiplatform Components. *Bulletin of Electrical Engineering and Informatics (BEEI)*. ISSN: 2302-9285, DOI: 10.11591/eei.v6i3.647. 2017; 6(3): 228-234
- [36] Rasyid Hardi Wirasasmita. Keefektifan Pembelajaran Berbantuan Internet di SMk Muhammadiyah Yogyakarta. Tesis: Program PPs UNY Yogyakarta. 2009.
- [37] Razia Sulthana A, Subburaj Ramasamy. Context Based Classification of Reviews Using Association Rule Mining, Fuzzy Logics and Ontology. *Bulletin of Electrical Engineering and Informatics (BEEI)*. ISSN: 2302-9285. DOI: 10.11591/eei.v6i3.682. 2017; 6(3): 250-255.
- [38] Ramesh LS, Ganapathy S, Bhuvaneshwari R, Kulothungan K, Pandiyaraju V, Kannan A. Prediction of User Interests for Providing Relevant Information Using Relevance Feedback and Re-ranking. *International Journal of Intelligent Information Technologies (IJIT)*. 2015; 11(4): 55-71.
- [39] Kotsiantis S, Kanellopoulos D. Association Rules Mining: A Recent Overview. *GESTS International Transactions on Computer Science and Engineering*. 2006; 32(1):71-82.
- [40] Song L, Tekin C, van der Schaar M. Online Learning in Large-scale Contextual Recommender Systems. *IEEE Transactions on Services Computing*. 2016; 9(3): 433-45.
- [41] Shi F, Ghedira C, Marini JL. Context Adaptation for Smart Recommender Systems. *IT Professional*. 2015; 17(6): 18-26.
- [42] Gang LV, Sheng-bing C. An Improved Entity Similarity Measurement Method. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2014; 12(4): 1017-22.
- [43] Mohammed Falah Mohammed, Taha H. Rassem. An Ensemble of Enhanced Fuzzy Min Max Neural Networks for Data Classification. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. ISSN: 1693-6930, Decree No: 58/DIKTI/Kep/2013 DOI: 10.12928/TELKOMNIKA.v15i2.6149. 2017; 15(2): 942-948.
- [44] Simpson PK. Fuzzy min-max neural networks-I: Classification. *IEEE Transaction on Neural Networks*. 1992; 3(5): 776-786.
- [45] Mohammed MF, Lim CP, Quteishat A. A Novel Trust Measurement Method Based on Certified Belief in Strength for a Multi-Agent Classifier System. *Neural Computing and Applications*. 2012; 24 (2): 421-429.
- [46] Mui L, Mohtashemi M, Halberstadt A. *A computational model of trust and reputation*. Hawaii International Conference on System Sciences-HICSS. 2002; 2431-2439.
- [47] Boukerche A, Xu L. *An Agent-Based Trust and Reputation Management Scheme for Wireless Sensor Networks*. IEEE Global Telecommunications Conference-GLOBECOM. 2005; 3: 5.
- [48] Wang P, Zhang Z. *A Computation Trust Model with Trust Network in Multi-Agent Systems*. International Conference on Active Media Technology (AMT). 2005; 389-392.
- [49] Polikar R. Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine*. 2006; 6: 21-45.
- [50] Quteishat A, Lim CP, Saleh JM. A neural network-based multi-agent classifier system with a Bayesian formalism for trust measurement. *Soft Computing*. 2011; 15(2): 221-231.