Journal of Theoretical and Applied Information Technology
15th August 2017. Vol.95. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN INTELLIGENT SERVICE RECOMMENDATION MODEL FOR SERVICE USAGE PATTERN DISCOVERY IN SECURE CLOUD COMPUTING ENVIRONMENT

[1] K. MANGAYARKKARASI, [2]DR.M.CHIDAMBARAM

[1]Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore, India

[2]Assistant Professor, Rajah Serfoji College, Bharathidasan University, Thanjavur, Tamilnadu, India

E-mail: [1]kanthimangai@gmail.com, [2]chidsuba@gmail.com

ABSTRACT

With the increase in the usage of web services in all business activities, the enormous amount of data is created and stored on different cloud servers. It is a challenging task to recommend the appropriate web services according to the demands of the user. The web recommendation techniques mainly concentrate on the mining of the association patterns among the web services from the historical compositions. But, the negative patterns denote the incorrect combination of web services. An accurate service recommendation model is presented by combining the positive patterns with the negative patterns in the large web service network. This paper proposed an intelligent service recommendation model for service usage pattern discovery in secure cloud computing environment. A RuleScore algorithm is proposed for predicting the future service collaboration based on the mined rules. The experiments on the real-time and synthetic datasets show that the proposed model ensures effective recommendation of web services in a large-scale network.

Keywords: *Association Rule Mining, Cloud Computing, RankScore, Semantic Tag, Web Service Recommendation*

1. INTRODUCTION

The cloud involves the usage of computational sources such as hardware and software resources that are delivered as a service over the Internet to the users or clients across worldwide. The cloud computing defines the usage of shared set of virtual and physical computational resources such as servers, data storage resources and network resources. It offers high flexibility, scalability and reliable hosting services to the users. Fig.1 depicts the cloud model comprising service models and deployment models. The service models of the cloud computing architecture are

Software as a Service (SaaS): It enables access to the applications to the users or clients through the Internet.

Platform as a Service (PaaS): It allows the user to create own applications that run on the infrastructure of the Cloud Service Provider (CSP).

Infrastructure as a Service (IaaS): It enables usage of resource such as virtual server, network connection and load balancers.

The deployment models of the cloud computing architecture are described below

Private Cloud: It is a secure environment that can be accessed by a single business organization. It can be maintained and managed by that organization or other third-party agency. The private clouds provide the computational service within a virtualized environment using a group of computing resources. Microsoft Azure is one of the private clouds that enable the clients to create the private cloud infrastructure using Windows server and dynamic data center tool kit. The main advantages of the private cloud are higher security and data control.

Public Cloud: In a public cloud, the CSP hosts and shares the services and infrastructure to the clients through Internet. The Amazon Elastic Compute Cloud is one of the public clouds. The public cloud services are available through the Internet, by enabling remote access to the clients and business enterprises from multiple locations. The public clouds are highly vulnerable to the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

security attacks when compared to the private clouds, due to the high levels of data accessibility.

Community Cloud: It is a multi-client infrastructure that is shared among a group of various business organizations having a common set of the computing requirements such as security policies, audit requirements and facilities. The main goal of the community cloud is to provide the benefits of both public and private clouds.

Hybrid Cloud: It integrates both the private cloud and public cloud to provide individual cloud services within the same business organization. Hence, the profit of the business organization is increased by employing the public cloud services for all non-sensitive operations and private cloud services during the requirements.

The challenges in the cloud computing are

- Lack of resources for user demands
- Lack of energy-efficient resource allocation schemes
- Resource sharing and adoption
- Data transfer bottlenecks
- Security threats

1.1 Data Mining Techniques

Data mining is done over a large volume of data to obtain information for making better decisions. The root of the data mining lies in the statistics, mathematics, information theory, artificial intelligence, machine learning techniques, etc. It searches the huge databases and data warehouses to find the hidden data or patterns. The data mining technique is the best technique for predicting the future patterns and trends to make wiser decisions for the business organizations.

1.2 Motivation and Objective

With the rapid increase in the number of available web services, manual selection of web services from a huge one is inefficient. Various automatic composition techniques (1-7) are proposed to improve the efficiency in the selection of web service. Among them, frequent pattern-based composition techniques (1), (5), (6), (8-10), receive a great deal of interest. With the rapid growth in the number of service compositions, most compositions follow the common business models and usage patterns (6). If the dataset of previous service compositions is given, the frequently occurred service composition patterns can be found out and recommendation of service compositions can be made based on the patterns.

The frequent patterns provide positive rules that summarize the relationships between correlations, collaborations and match between services in created compositions (2). There are negative rules in the compositions representing the functionality non-correlation, conflicts or competitions between services (11). If two services satisfy a negative rule, they do not cooperate in a same composition. The service recommendation is improved by considering the negative rules. Due to critical challenges, implementation of negative rules is an insignificant task. It is difficult to find the negative rules and combine with the positive rules. As there is no explicit information provided on the non-correlation of services with each other, it is impossible to find negative rules from the previously created service compositions. The direct method is to obtain the never-collaborated services as uncorrelated services. But, this suffers from the following problems: The number of never collaborate service pairs are much greater than ever-collaborated service pairs, as the network is partial and sparse (12). This brings data imbalance in model training (13). As the never collaborated services may originate from the incompletion of dataset, they cannot be simply considered as negative relations. There is a probability of cooperation between these never collaborated services in future (14). Introduction of negative rules will not improve the prediction accuracy (15, 16). In extreme situation, the recommendations given by the negative rules and positive rules are identical. If it is true, there is no improvement in prediction accuracy (17).

The data owners want to upload their data to the cloud environment. This causes an increase in the security concerns such as data confidentiality, privacy, authentication and access control. The continuous rise in the usage of the cloud computing system provides features such as mobility, high resource availability and affordable services. The major issues in the cloud computing environment are the resource availability and security. An efficient model should be developed to satisfy these issues. To address this issue, this paper proposes an intelligent service recommendation model for the discovery of service usage pattern in a secured cloud computing environment. As the negative patterns aid in the improvement of the recommendation performance, the positive and negative rules should be combined for prediction. The main objective of the proposed model is to recommend the best services to the users based on their demands.

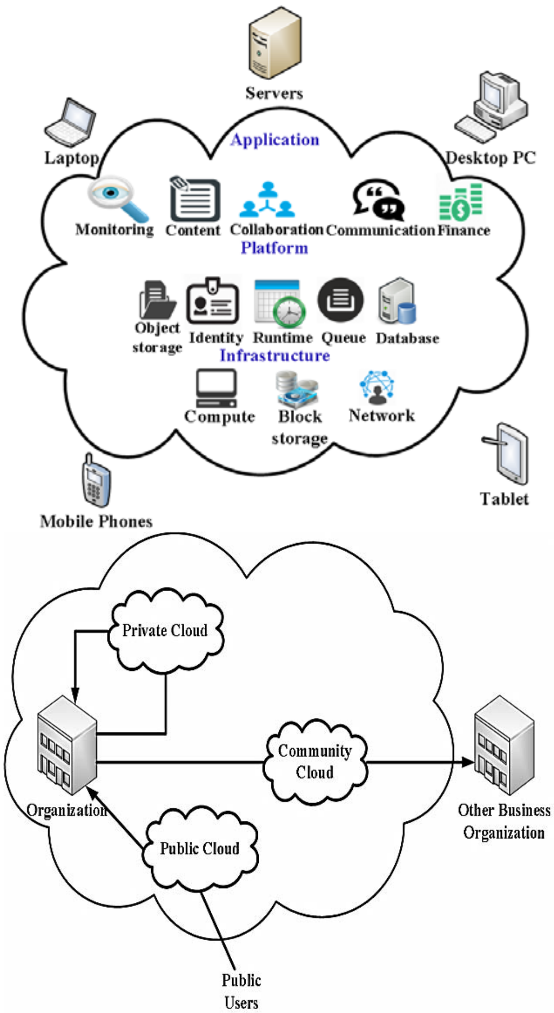

Fig.1 Cloud computing environment and deployment models in the cloud environment

1.3 Contributions of proposed work

The main contributions of the proposed service recommendation model are

- A novel association rule mining algorithm is proposed for finding the positive and negative rules in a constant representation of semantic tags from the previously generated service compositions that address the data imbalance issues.
- A rule-based service recommendation model called as RuleTree and a rule evaluation model referred as RuleScore are proposed to predict the future service collaboration based on the mined rules and prove the value of negative rules in the generation of service compositions.

Various experiments are conducted on both the real-time and synthetic datasets to prove that the proposed model outperforms the baseline algorithms.

The paper is schematized as follows: An overview of the research works relevant to the service recommendation techniques is described in Section II. Section III presents the cloud data mining framework, including rank-based and weighted association rule mining and cloud data mining architecture. Section IV explains the association rule mining concept. Section V presents the tag-based service recommendation and Section VI explains the RuleScore algorithm. Section VII gives the experimental setup including the description about the real-time and synthetic datasets. Section VIII illustrates the performance analysis of the proposed model. The comparative analysis of the proposed model with the existing service recommendation method is described in Section IX. Section X describes the conclusion of the proposed intelligent service recommendation model.

2. EXISTING SERVICE RECOMMENDATION TECHNIQUES

A keyword-aware service recommendation method is proposed for efficient recommendation of appropriate services to the users (18). A novel framework that exploits a user-centric strategy is presented to enable personalized Quality of Service (QoS) evaluation of the cloud services (19). A novel service recommendation approach is proposed to improve the service selection efficiency for multi-tenant Software-as-a-Service (SaaS) (20). Jung et al. (21) introduced a platform for recommending various cloud configurations based on the user preferences. A real-time multi-criteria QoS-aware decision making technique is presented for selecting the Infrastructure-as-a-Service (IaaS) (22). Guo et al. (23) reviewed various cloud service recommendation methods focusing on the collaborative filtering and discussed the research challenges and opportunities for service recommendation. The Preference-based cLoud Service Recommender providing optimization as brokerage service is discussed (24). The recommender uses a holistic multi-criteria decision making approach. The design and implementation of the cloud-based service recommendation system used for discovering and suggesting the best mobile services to the users is described (25). A mechanism for selecting the optimal public cloud service at the IaaS and PaaS is recommended (26). Aznoli and Navimipour (27)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

conducted a systematic review and comparison of recommender systems based on the scalability, accuracy, availability and trust attributes. A cloud service recommendation approach is proposed based on the trust measurement for potential cloud users by using ternary interval numbers (28). A correlated QoS ranking algorithm is proposed to predict the personalized ranking for the selection of service by an active cloud user (29). A survey of recent cloud service recommendation approaches based on the security is presented (30). However, the existing recommendation approaches do not distinguish the positive and negative preferences of the users from their reviews for more accurate predictions. To overcome this issue, this paper presents an accurate service recommendation model by combining the positive patterns with the negative patterns in the large web service network.

3. CLOUD DATA MINING FRAMEWORK

The data mining techniques are required to predict the future patterns of the next generation cloud computing environment. The association rule mining technique is combined with the cloud computing environment. The proposed model allows the users or the business organizations to centralize the server management, data storage, hardware and software resources. It allows the signification reduction in the time required to identify the best resources for the demands of the users and provides secure services for the users and business organizations. The proposed model allows the user to retrieve meaningful information from the cloud data store. It provides information about the interests and behavior of the clients, resource accessibility by the clients and security information.

3.1 Rank-based and Weighted Association Rule Mining

This comprises two measures

- Weighted Condensed Support confidence (WCS)
- Weighted Condensed Confidence (WCC)

$$WCS(Z) = \begin{cases} \frac{\sum_{k=1}^{m} W_k(Z)}{m'(Z)}, if\ |Z| > 1 \\ \frac{\sum_{k=1}^{m} W_k(Z)}{m}, if\ |Z| = 1 \end{cases} \quad (1)$$

Where $m'(Z)$ is described as follows

$$m'(Z) = \max_{(\forall g_i \notin Z, Q=|Z|)} \{\sum_{k=1}^{m} BIT_{k1}, \sum_{k=1}^{m} BIT_{k2}, \ldots, \sum_{k=1}^{m} BIT_{Q}\} \quad (2)$$

Where Q represents the total number of datasets in a data storage $S|Z| > 1$.

$$WCC\ (A \to C) = \frac{WCS(A \cup C)}{WCS(A)} = \frac{WCS(Z)}{WCS(A)} \quad (3)$$

Apriori High Utility Discovery Data Set (HUDS)

Input: Data matrix ***D(rows=datasets, columns=samples)***
Output: Set of rules ***Rules***, support ***RuleSupp***, confidence ***RuleConf***

Step 1: RankApriori-HUDS()
Step 2: {
Step 3: Normalize the Data Matrix D
Step 4: Calculate rank of datasets
Step 5: Assign weights wt(:) to all datasets to their ranks rank(:)
Step 6: Select initial seed value for using k-means clustering
Step 7: Initialize k=1
Step 8: Find Frequent Data Sets (FDS)
Step 9: $FDS_k = \{i | i \in A1 \wedge WCS(i) \geq \min\_wsupp\}$
Step 10: Repeat
Step 11: k=k+1
Step 12: Generate Candidate DataSets CDS_k from FDS_{k-1}
Step 13: For each CDS, $c€CDS_k$
Step 14: {
Step 15: Calculate WCS(:) for each CDS c
Step 16: If $(WCS(:) \geq \min\_wsupp)$ then
Step 17: {
Step 18: $FDS_k \leftarrow [FDS_k : c]$
Step 19: Generate rules, rule(:) from the FDS c
Step 20: Determine WCC(:) for every rule(:)
Step 21: For every rule, $r€rule(:)$
Step 22: {
Step 23: If $(WCC(r) \geq \min\_wconf)$ then
Step 24: {
Step 25: Store the value of r in resultant rule-list rules with its WCS and WCC
Step 26: $Rules \leftarrow r, RuleSupp \leftarrow WCS(r)\ and\ RuleConf \leftarrow WCC(r)$
Step 27: }
Step 28: }
Step 29: }
Step 30: }
Step 31: until $KDS_k = \phi$
Step 32: }

3.2 Cloud Data Mining Architecture

Fig.2 illustrates the cloud data mining model architecture integrated with the proposed model. The proposed model provides great management capabilities in terms of ranking datasets and providing security. As the proposed model identifies the best frequently used datasets, the users will get the best datasets for their queries. A secure communication is enabled between the users and datasets.

4. ASSOCIATION RULE MINING

Association rule mining is an emerging research topic in the data mining field, for the discovery of interesting relationship between the database items. The most prevalent frequent pattern mining algorithms are Apriori, Eclat and FP-growth. The frequent patterns of the items are discovered from the huge transactional database by using the co-occurrence metrics such as support, confidence and lift. Also, association rule mining is the recent approach in the service computing.

Shafiq et al. (31) proposed a semantic Frequent Pattern (FP) Growth-based algorithm to find the possible service correlations from the service execution logs to improve the service ranking and recommendation performance. The mined rules are used for prefiltering the web services to reduce the search space during the ranking and recommendation process. But, this approach demands an additional task to generate semantic logs from the original log data before the rule mining.

Khanh Dam (9) employed association rule mining for extracting the web service patterns based on the assumption that the web services that are changed frequently in the past will be changed in future. Using the knowledge of co-changed patterns, the possible impact on the service ecosystem is predicted if certain web services are updated. The integration of the collaborative filtering (CF) with the association rule mining improves the service recommendation prediction accuracy. But, CF becomes ineffective when the invocation histories of the users are sparse.

Rong et al. (32) proposed a novel algorithm for ranking the services, by combining the CF and association rule mining. The users are divided into various groups based on the similarity of the service invoking log of the users. Then, the personal association rules are generated for each user from the service compositions. These methods considered the positive rules only. The useful negative patterns in the service compositions are missed by ignoring the negative rules.

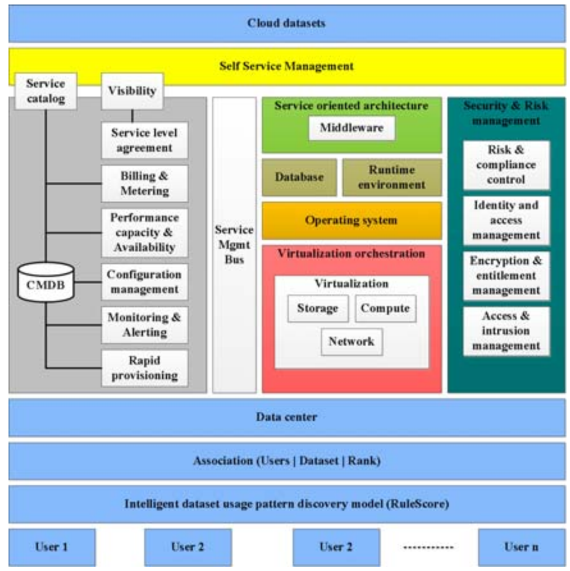

Fig.2 Cloud data mining architecture integrated with intelligent dataset usage pattern discovery model

The aim of the rule mining is to find the candidate rules that are used for the service recommendation. A rule can be considered as a candidate, if at least one service connection in the training dataset obeys the rule. The set of all candidate rules $C = C_{p \cup C_n}$ is represented based on the definition of candidate rule:

$$R_C = \{ \forall r | \exists c \in C, \ such \ that \ c \Vdash r \} \quad (4)$$

As the number of candidate rules is huge, it consumes more time to select the proper rules from this huge candidate set for service recommendation. Naturally, two observations about the candidate rules can facilitate the selection of proper rule. If a rule is obeyed by a small amount of connections, it is not able to create a summary about the common composition patterns. A rule obeyed by a large number of connections or having a large support value is selected for the service recommendation. If a rule is obeyed by the equivalent number of positive and negative connections, the different classes of the training samples cannot be distinguished effectively. Hence, there is a possibility of selecting a rule with high significance value from the service recommendation. There should be a gap between the number of positive and negative connections that obey the rule. The support value of the rule ‘r’ that measures the number of connections that obey the rules is defined as

$$supp(r) = \left| \{ \forall c \in C_{p \cup C_n} | c \Vdash r \} \right| \quad (5)$$

The significance of the rule ‘r’ measures the gap between the positive and negative connections. It is given as

Journal of Theoretical and Applied Information Technology
15th August 2017. Vol.95. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$sig(r) = \frac{|\{\forall c \in C_p | c \Vdash r\}| - |\{\forall c \in C_n | c \Vdash r\}|}{|\{\forall c \in C_p | c \Vdash r\}| + |\{\forall c \in C_n | c \Vdash r\}|} \quad (6)$$

If the number of positive connections that obey the rule is greater than the negative connections or $sig(r) > 0$, then the rule is called as a positive rule. Otherwise, it is called as negative rule. If the minimal support and significance threshold for the rules are given, the candidate generation algorithm is described as follows

Candidate Rule Generation Algorithm

Input: Training dataset of service connections C_p and C_n, support threshold supp_min and significance threshold for candidate generation sig_min
Generate_Candidate_Rules
$(C_p, C_n, supp_{min}, sig\_min)$
Step 1: $R_0 \leftarrow \emptyset$
Step 2: $R \leftarrow \emptyset$
Step 3: **for** every connection $c = \langle S_1, S_2, f(S_1, S_2)\rangle$ in $C_{p \cup C_n}$ **do**
Step 4: $R_0 \leftarrow R_{0 \cup \{\forall r | c \Vdash r\}}$
Step 5: **for** every element $r \in R_0$ **do**
Step 6: **if** $supp(r) \geq supp\_min$ **and** $|sig(r)| \geq sig\_min$ **do**
Step 7: $R \leftarrow R \cup \{r\}$
return R

5. TAG-BASED SERVICE RECOMMENDATION

The collaboration network constructed by the services and their compositions is represented as $N = \langle S, M, T\rangle$. The term 'S' represents the set of all services $S = \{s_1, s_2, \dots \dots, s_k\}$, 'M' denotes the set of all compositions $M = \{m_1, m_2, \dots \dots, m_p\}$ and 'T' denotes the set of all semantic tags $T = \{t_1, t_2, \dots \dots, t_q\}$. The service instances are the nodes in the network. If two services are used collaboratively in a same composition, then the corresponding nodes are connected by a positive edge termed as positive service connection. If two services are never used collaboratively in any composition, they are connected by a negative service connection. Fig.3 shows the illustration of the collaborative network.

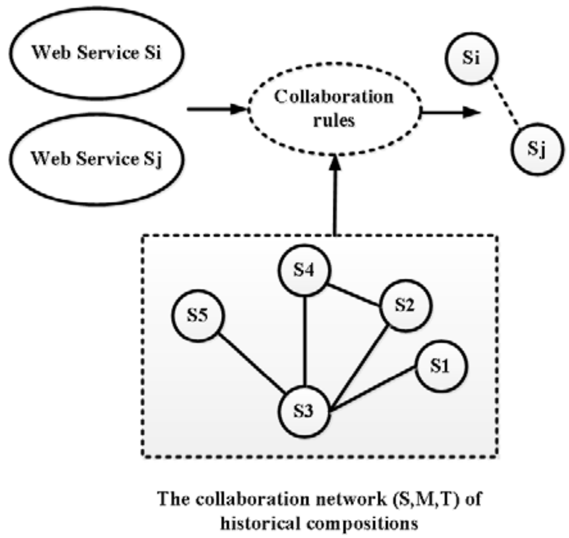

Fig.3 Semantic tag-based service recommendation

The service recommendation task contains different aspects for facilitating the development of composition. This work focuses on the discovery of potential collaboration rules among the set of web services, historical compositions and semantic tags to initiate the development of future composition. In the framework of service network, the prediction of the future compositions and future sign of the service connections are identical. The future sign indicates the probability of the negative connections to become positive. The tag-based service recommendation can be formulated as
If two services S_i and S_j with annotated tags are given, there is a need to find the collaboration between the services to create the compositions in the near future. The web services are represented as

$$S_i = \left\{t_{S_i^{(1)}}, \dots \dots, t_{S_i^{(r_i)}}\right\} \quad (7)$$

$$S_j = \left\{t_{S_j^{(1)}}, \dots \dots, t_{S_j^{(r_j)}}\right\} \quad (8)$$

The future sign $f(S_1, S_2)$ of the service connection $\langle S_1, S_2 f(S_1, S_2)\rangle$ is predicted based on the rules in the service collaboration network. It implies that our service recommendation approach is a combination of association rule mining and link prediction.

5.1 Selection of service connection

The strategy for selecting the service connection for rule mining is presented in this section. Sampling of the positive and negative connections suffers from the imbalance problem. To address the imbalance problem, two metrics are used so that the size of the negative connections becomes equal to the positive connections.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5.5.1 Positive connection:

There exists a positive connection between the services that are collaboratively used in at least one composition. The set of positive connection is defined as

$$C_P = \left\{\langle S_i, S_j, 1\rangle \,\middle|\, \forall_{S_i,S_j} \in S, S_i \neq S_j, f(S_i, S_j) = 1\right\} \quad (9)$$

Positive connection selection algorithm

Generate_positive_connections $\langle S, M, T\rangle$
Step 1: $C_p \leftarrow \emptyset$
Step 2: for every connection $m = \left\{S_m^{(1)}, S_m^{(2)}, \dots, S_m^{(1)}\right\}$ in M **do**
Step 3: for every two services S_i, S_j in m **do**
Step 4: $c \leftarrow \langle S_i, S_j, 1\rangle$
Step 5: $C_p \leftarrow C_p \cup \{c\}$
return C_p

5.5.2 Generation of Negative connection:

Despite of the existence of a huge number of negative connections, most of them are unreliable negative connections. Most of the negative connections emerge from the unpopular services or lack of composition log data. It does not contain information about the negative patterns. Two metrics for measuring the negative connection, such as popularity and cold time are used to filter the invaluable data. Two observations can distinguish the reliable negative connections from the worthless connections. If two services are very popular in the creation of composition and never cooperate with each other even in one composition, they form a reliable negative connection. If two services harmonize in the network for a long time and never cooperate with each other even in one composition, they form a credible negative connection.

Popularity-product

The popularity-product of the negative connection $c = \langle S_i, S_j, -1\rangle$ is defined as

$$product\,(c) = pop(S_i) \times pop(S_j) \quad (10)$$

Where $pop(S_i) = |\{\forall m \in M | S_i \in M\}|$ represents the popularity of the service S_i.

Cold-Time

The cold-time of a negative connection $c = \langle S_i, S_j, -1\rangle$ is defined as

$$time(c) = now - max\left(date(S_i), date(S_j)\right) \quad (11)$$

Where $date(S_i)$ indicates the publication date of the service S_i. The popularity product is a measurement of non-cooperation based on the structural scale. The cold-time is a measurement of non-cooperation based on the time scale. Fig.4 depicts the distribution of negative connections. The X-axis represents the cold-time and Y-axis indicates the popularity product. The figure shows the sample probability of each negative connection, where k=1.

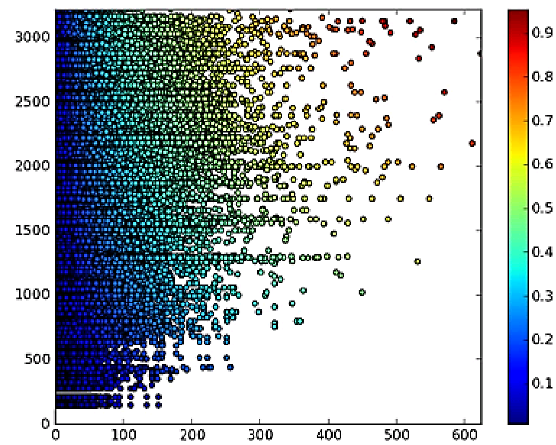

Fig.4 Distribution of negative connections

The negative connections that are located in the top right corner of the graph represent the credible negative connections. Further negative connections move to the top right direction. Higher probability of the negative connections implies the selection of these connections for negative rule mining.

The popularity-product and cold-time are combined and an undersampling approach is proposed to generate a set of credible negative connection. The undersampling approach is a widely used technique for solving the imbalanced issues. This approach generates equal datasets of different classes, by selecting a portion of data from the outnumbered class. There is a possibility of selecting the negative connections with maximum popularity-product and longer cold-time for rule mining. Hence, the sample probability of a negative connection $c = \langle S_i, S_j, -1\rangle$ is defined as follows

$$p(c) = \frac{1}{N} * [product(c) * time(c)]^k \quad (12)$$

Where 'N' is the normalization constant to ensure that $\sum_C p(c) = 1$ and 'k' is a control parameter. Larger value of 'k' applies more weightage on the association with the maximum popularity-product and longer cold-time.

Negative connection generation algorithm

Generate_Negative_Connections $\langle S, M, T\rangle, l$
Step 1: $C_0 \leftarrow \emptyset$
Step 2: for every two services S_i, S_j in S **do**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 3: if $\beta_m \in M, \{S_i, S_j\} \subseteq$m **do**
Step 4: $c \leftarrow \langle S_i, S_j, -1\rangle$
Step 5: $c_0 \leftarrow C_o \cup \{c\}$
Step 6: Sample 'l' instances with replication with the distribution p(c) from C_o as C_n
Return C_n

Thus, the undersampling approach generates a set of meaningful negative service connections. A balanced dataset $C = C_{p \cup C_n}$ is obtained along with the positive connections.

6. RULESCORE

A RuleTree algorithm provides composition recommendations without estimating the importance of the rules. The contribution of each rule to the composability of two services is not known. To evaluate the rules, a RuleScore algorithm is proposed to assign score to each collaboration rule. The RuleScore uses the Adaboost algorithm (33) to create a sequence of RuleTree. Each RuleTree is assigned with a coefficient that indicates the contribution to the composability. The depth of the RuleTree is set to 1. Each tree contains only one rule as its non-leaf node and two leaf nodes are labeled as 1 and -1. During each iteration of main loop, a RuleTree is constructed over the present training dataset of service connections with the present weights. Then, the weight of all training samples is updated based on the correct classification of the training samples by the RuleTree.

RuleScore Algorithm

Input: Connection set 'C', candidate rule set 'R' and number of iterations 'q'
RuleScore (C,R,q)
Step 1: Initialize every connection 'i' of 'C' with equal weight $D_1(i) = 1/|C|$
Step 2: **for** every rule r in R **do**
Step 3: $Score(r) \leftarrow 0$
Step 4: **for** $j = 1, \dots, q$ **do**
Step 5: training a RuleTree $h_j(r) \leftarrow RuleTree(C, R, 1) s.t r \in R$ over C with respect to the weight distribution D_j
Step 6: $\epsilon_j \leftarrow the\ misclassification\ rate\ of\ h_j(r)\ over\ C$
Step 7: $\alpha_j \leftarrow 0.5\ \ln\left((1-\epsilon_j)/\epsilon_j\right)$
Step 8: $Score(r) \leftarrow Score(r) + \alpha_j$
Step 9: **for** connection 'i' that is correctly classified by h_j **do**
Step 10: $D_{j+1}(i) \leftarrow D_j(i)e^{-\alpha_j}$
Step 11: **for** connection 'i' that is misclassified by h_j **do**
Step 12: $D_{j+1}(i) \leftarrow D_j(i)e^{\alpha_j}$
Step 13: $S_{j+1} \leftarrow \sum_i D_{j+1}(i)$
Step 14: **for** every connection 'i' of C **do**
Step 15: $D_{j+1}(i) \leftarrow D_{j+1}(i)/S_{j+1}$
return R

The weight of all training samples is set to $1/|C|$ and scores of all candidate rules are initialized as 0. A rule is selected to create a RuleTree $h_j(r)$ over current training samples. The score of the rule is updated in accordance with the misclassification rate ϵ_j of $h_j(r)$. The weight of all samples are updated according to ϵ_j and based on the correct classification of the training samples. A score, either positive or negative is assigned to a portion of rules in the candidate rule set. The score 0 is assigned to other undesired rules. Fig.5 depicts the illustration of the RuleScore algorithm. The RuleScore algorithm generates a set of scored collaboration rules. For a scored rule, if $Score(r) > 0$, it is a positive rule. Otherwise it is a negative rule. The RuleScore can make accurate recommendations on the creation of composition based on the scored rules. Let us consider web services S_1 and S_2, annotated tags and scored collaboration rules 'R' are given, then the rules that are obeyed by the connection $c = \langle S_1, S_2, f(S_1, S_1)\rangle$ is represented as follows

$$R(S_1, S_2) = \{\forall r \in R | \langle S_1, S_2, f(S_1, S_1)\rangle \Vdash r\} \quad (13)$$

The composability score of service S_1 and S_2 is calculated by summing the scores of all satisfied rules $R(S_1, S_2)$

$$F(S_1, S_2) = \sum\nolimits_{\forall r \in R(S_1,S_2)} Score(r) \quad (14)$$

$\langle S_1, S_2, f(S_1, S_2)\rangle$ satisfies more positive rules and less negative rules. Higher the value $F(S_1, S_2)$, the probability of service composability is high. Hence, if $F(S_1, S_2) > 0$, the services S_1 and S_2 are composable. If $F(S_1, S_2) < 0$, the services are non-composable.

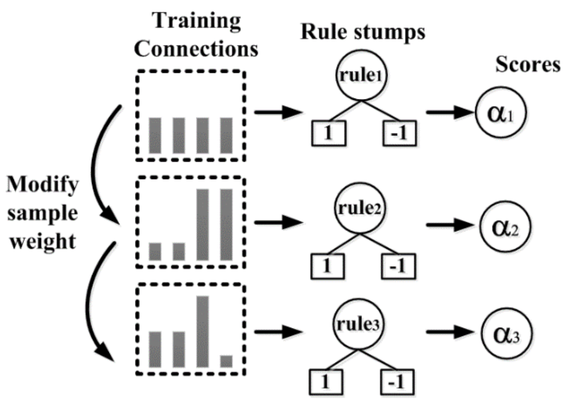

Fig.5 Illustration of RuleScore Algorithm

7. EXPERIMENTAL SETUP

7.1 Dataset Introduction

7.1.1 Real-time Evaluation dataset

A real-time dataset of web services and compositions is collected from programmableweb (4). A filtered collection of 1377 services and 2679 compositions is obtained, after removing the services that never cooperate with each other and compositions that contain a single service. This collection is divided into training and testing dataset. The training dataset is generated from the compositions that are created before December 31, 2012. If the services cooperate with the compositions before December 31, 2012, a training sample of positive connection is generated. Otherwise, the negative connection training samples are generated. The testing dataset is generated from the compositions that are created before June 30, 2014. A testing sample of positive connections is generated if there is cooperation between two services with the compositions before June 30, 2014. Otherwise, a testing sample of negative connections is generated. Table I summarizes the details of the real-time dataset.

7.1.2 Large Scale Synthetic Dataset

The synthetic dataset is created from the real-time evaluation dataset by duplicating and renaming the existing services, which combine ten times to a large scale dataset. It is used to evaluate the efficiency of the recommendation methods. Table II summarizes the details of the synthetic dataset.

7.2 RuleScore-based Recommendation

This proposed model performs learning of an estimator $F(S_1, S_2)$ of a sequence of scored rules over the balanced training dataset. For two web services S_1 and S_2, if the estimate $F(S_1, S_2) > 0$, the services are composable. Otherwise, the services are not composable. By default, the minimum support for the generation of candidate rule is set as 0.002 and minimum significance is set to 0.1. The number of iterations is set as 2000. The default sample power for the negative connections is set as 1.

Table 1: Details of Real-time Dataset.

Items	Number
Web services	1377
Tags used to annotated services	2577
Compositions created before December 31, 2012	1983
Compositions created after June 30, 2014	696
Training positive connections	12875
Training negative connections	11087
Testing positive connections	1000
Testing negative connections	1000

Table 2: Details of Large Synthetic Dataset

Items	Number
Web services	13770
Tags used to annotated services	25770
Compositions created before December 31, 2012	19830
Compositions created after June 30, 2014	6960
Training positive connections	128750
Training negative connections	110870
Testing positive connections	10000
Testing negative connections	10000

8. PERFORMANCE ANALYSIS

The performance of the proposed RuleScore algorithm is evalauted by comparing it with the Rank-based Weighted Association Rule Mining technique (S-RWARM) for secure cloud computing environment (34). Fig.6 shows the graph illustrating the variation in the execution time with respect to the threshold value. The execution time decreases with respect to the increase in the threshold value. This indicates that the proposed RuleScore algorithm requires minimum execution time than the S-RWARM technique. The proposed RuleScore algorithm identifies the best dataset among the available datasets on the cloud environment. Hence, it requires minimum time to predict the best data corresponding to the user requirement.

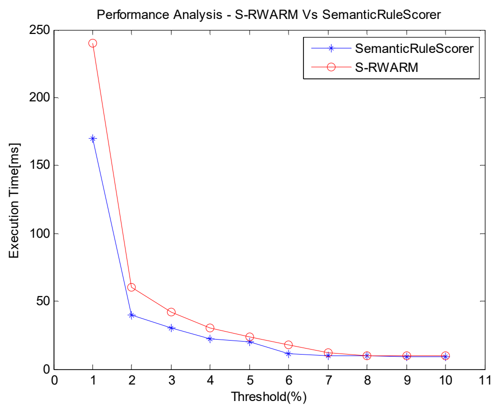

Fig.6 Execution Time vs threshold

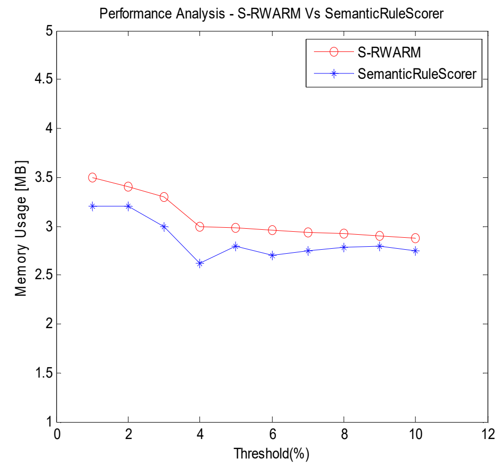

Fig.7 Memory Usage Analysis

Fig.7 depicts the memory usage analysis of the proposed RuleScore model and S-RWARM technique. The memory usage decreases with the increase in the threshold value. The proposed model consumes lesser memory space than the S-RWARM. Hence, the storage space of the server is utilized efficiently for the execution of other tasks. Fig.8 presents the selection efficiency of the dataset related to the demand of the users. The S-RWARM technique ranks the dataset based on the demands of the user. The proposed model ensures accurate recommendation to the user based on the scored collaboration rules. Fig.9 shows the utilization rate of the dataset by the users. The S-RWARM technique shows the usage rate of the users in a dispersed manner. The users with maximum, intermediate and minimum dataset usage rate are plotted randomly. Our proposed semantic RuleScore depicts the usage rate in a hierarchical way in order of maximum, intermediate and minimum dataset usage rate. Due to the accurate recommendation of the services, the dataset can be utilized efficiently by the user. Hence, the semantic RuleScore is found to be efficient than the S-RWARM technique.

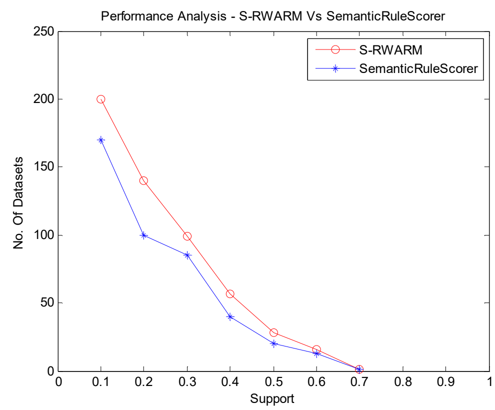

Fig.8 Dataset selection efficiency

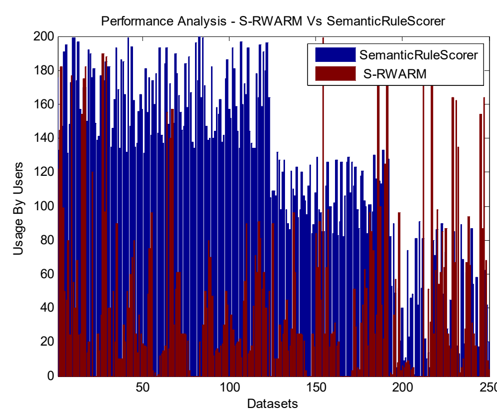

Fig.9 Dataset utilization rate

9. COMPARATIVE ANALYSIS

Our proposed model is compared with the keyword-aware service recommendation method KASR-ASC and KASR-ESC (18), user-based algorithm using Pearson Correlation Coefficient

(UPCC) (35) and Item-based algorithm using PCC (IPCC) (36).

Fig.10 shows the MAE and NMAE analysis of UPCC, IPCC, KASR-ASC, KASR-ESC and proposed semantic Rulescore algorithm. The Mean Absolute Error (MAE) measures the recommendation quality and Normalized MAE (NMAE) measures the recommendation accuracy. More accurate recommendation can be obtained if the MAE and NMAE are low. The MAE of UPCC, IPCC, KASR-ASC and KASR-ESC is 67.92%, 54.65%, 47.53%, 41.93% and proposed semantic Rulescore algorithm is 39%. The NMAE of UPCC, IPCC, KASR-ASC and KASR-ESC is 17.69%, 13.73%, 11.37%, 10.59% and proposed semantic Rulescore algorithm is 9%. As the MAE and NMAE of the proposed semantic Rulescore algorithm are lower than the existing methods, the recommendation accuracy is high. The Mean Average Precision (MAP) is used to evaluate the quality of Top-K service recommendation list. Fig.11 depicts the MAP analysis of UPCC, IPCC, KASR-ASC, KASR-ESC and proposed semantic Rulescore algorithm. The MAP of the KASR-ASC and KASR-ESC is higher than the IPCC and UPCC. The MAP of the proposed semantic Rulescore algorithm is higher than the IPCC, UPCC, KASR-ASC and KASR-ESC. From the comparative analysis, it is concluded that the proposed semantic Rulescore algorithm is efficient than the IPCC, UPCC, KASR-ASC and KASR-ESC.

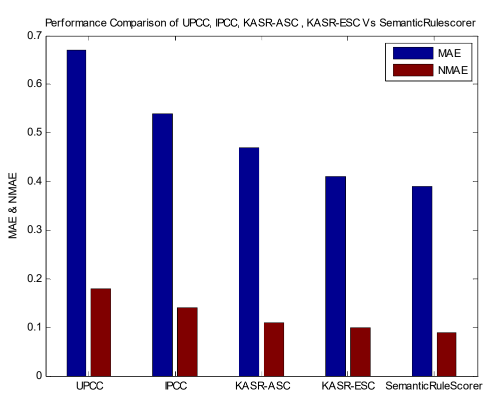

Fig.10 MAE and NMAE analysis of UPCC, IPCC, KASR-ASC, KASR-ESC and proposed semantic Rulescore algorithm

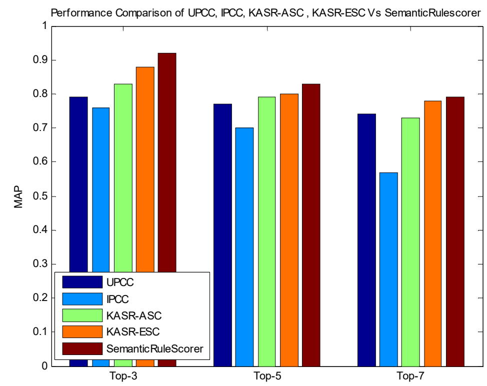

Fig.11 MAP analysis of UPCC, IPCC, KASR-ASC, KASR-ESC and proposed semantic Rulescore algorithm

The service recommendation methods are effective in solving the information overload for the consumers. A tag-based intelligent service recommendation model is presented in the paper by combining the positive and negative collaboration rules. This provides a comprehensive and significant illustration of recent trend for the creation of service compositions. A balanced dataset of positive and negative service connections is collected based on the popularity and time metrics. A reduced set of candidate rules is mined from the service connections by using the support and significance metrics. The recommendation of service composition is made based on the candidate rules. With the rule reduction strategy, our proposed model is highly effective for large dataset. The proposed semantic rulescore gained better performance than the Apriori algorithm.

10. CONCLUSION

Service recommendation methods are proven to be efficient for solving the problem of information overhead for the customers. This paper proposed an intelligent service recommendation model by combining the positive and negative collaboration rules. Our proposed model includes a strategy for sampling the positive and negative service connection based on the metrics such as popularity product and cold-time. The RuleScore algorithm generates a set of scored collaboration rules for recommending the compositions based on the mined rules. From the experimental results, it is proven that the proposed service recommendation model achieves better performance than the existing technique. The proposed Semantic Rulescore algorithm requires minimum execution time and memory usage than the S-RWARM technique. The Semantic Rulescore algorithm yields maximum dataset utilization rate and minimum dataset selection efficiency than S-RWARM. The proposed Semantic Rulescore algorithm is compared with IPCC, UPCC, KASR-ASC and KASR-ESC. The comparative analysis shows that the proposed semantic Rulescore algorithm yields minimum MAE and NMAE and higher MAP than the existing methods. The proposed model can be applicable to a huge dataset by adopting the rule reduction strategy.

As the proposed approach depends on the semantic tags on the web services, the applicability of the proposed methodology is limited when there is no clear definition of the tags. It is a real-time challenge to deal with the similarities and variations between different tags. In future, our proposed work is enhanced to handle the issues about the semantic tags.

REFERENCES:

[1] 1. Han Y, Chen S, Feng Z. Mining integration patterns of programmable ecosystem with social tags. Journal of grid computing. 2014;12(2):265-83.

[2] 2. Goarany K, Kulczycki G, Blake MB, editors. Mining social tags to predict mashup patterns. Proceedings of the 2nd international workshop on Search and mining user-generated contents; 2010: ACM.

[3] 3. Tapia B, Torres R, Astudillo H, Ortega P, editors. Recommending APIs for mashup completion using association rules mined from real usage data. 30th International Conference of the Chilean Computer Science Society (SCCC),; 2011: IEEE.

[4] 4. Liang QA, Chung J-Y, Miller S, Ouyang Y, editors. Service pattern discovery of web service mining in web service registry-repository. IEEE International Conference On E-business Engineering, 2006 ICEBE'06; 2006: IEEE.

[5] 5. Lee C-H, Hwang S-Y, Yen I-L, editors. A service pattern model for flexible service composition. IEEE 19th International Conference on Web Services (ICWS),; 2012: IEEE.

[6] 6. Vollino B, Becker K. Usage profiles: A process for discovering usage patterns over web services and its application to service evolution. International Journal of Web Services Research (IJWSR). 2013;10(1):1-28.

[7] 7. Bayati S, Nejad AF, Kharazmi S, Bahreininejad A, editors. Using association rule mining to improve semantic web services composition performance. 2nd International Conference on Computer, Control and Communication, 2009 IC4 2009; 2009: IEEE.

[8] 8. Shafiq O, Alhajj R, Rokne JG, editors. Reducing search space for web service ranking using semantic logs and semantic FP-tree based association rule mining. IEEE International Conference on Semantic Computing (ICSC), 2015; 2015: IEEE.

[9] 9. Khanh Dam H. Predicting change impact in Web service ecosystems. International Journal of Web Information Systems. 2014;10(3):275-90.

[10] 10. Rong W, Liu K, Liang L, editors. Personalized web service ranking via user group combining association rule. IEEE International Conference on Web Services, 2009 ICWS 2009; 2009: IEEE.

[11] 11. Shafiq O, Alhajj R, Rokne JG, editors. Reducing search space for web service ranking using semantic logs and semantic FP-tree based association rule mining. IEEE International Conference on Semantic Computing (ICSC),; 2015: IEEE.

[12] 12. Huang K, Fan Y, Tan W, editors. An empirical study of programmable web: A network analysis on a service-mashup system. IEEE 19th International Conference on Web Services (ICWS),; 2012: IEEE.

[13] 13. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2012;42(4):463-84.

[14] 14. Wu X, Zhang C, Zhang S. Efficient mining of both positive and negative association rules. ACM Transactions on Information Systems (TOIS). 2004;22(3):381-405.

[15] 15. Kunegis J, Preusse J, Schwagereit F, editors. What is the added value of negative links in online social networks? Proceedings of the 22nd international conference on World Wide Web; 2013: ACM.

[16] 16. Kunegis J, Lommatzsch A, Bauckhage C, editors. The slashdot zoo: mining a social network with negative edges. Proceedings of the 18th international conference on World wide web; 2009: ACM.

[17] 17. Ni Y, Fan Y, Tan W, Huang K, Bi J. NCSR: negative-connection-aware service recommendation for large sparse service

network. IEEE Transactions on Automation Science and Engineering. 2016;13(2):579-90.
[18] 18. Meng S, Dou W, Zhang X, Chen J. KASR: A Keyword-Aware Service Recommendation method on MapReduce for big data applications. IEEE Transactions on Parallel and Distributed Systems. 2014;25(12):3221-31.
[19] 19. Yu Q. CloudRec: a framework for personalized service Recommendation in the Cloud. Knowledge and Information Systems. 2015;43(2):417-43.
[20] 20. Wang Y, He Q, Yang Y, editors. Qos-aware service recommendation for multi-tenant saas on the cloud. IEEE International Conference on Services Computing (SCC); 2015: IEEE.
[21] 21. Jung G, Mukherjee T, Kunde S, Kim H, Sharma N, Goetz F, editors. Cloudadvisor: A recommendation-as-a-service platform for cloud configuration and pricing. Services (SERVICES), 203 IEEE Ninth World Congress on; 2013: IEEE.
[22] 22. Zhang M, Ranjan R, Menzel M, Nepal S, Strazdins P, Wang L. A cloud infrastructure service recommendation system for optimizing real-time QoS provisioning constraints. arXiv preprint arXiv:150401828. 2015.
[23] 23. Guo L, Zheng X, Ding C, Mu D, Li Z, editors. Cloud service recommendation: state of the art and research challenges. 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid); 2015: IEEE.
[24] 24. Patiniotakis I, Verginadis Y, Mentzas G, editors. Preference-based cloud service recommendation as a brokerage service. Proceedings of the 2nd International Workshop on CrossCloud Systems; 2014: ACM.
[25] 25. Ganchev I, Ji Z, O'Droma M, editors. A cloud-based service recommendation system for use in UCWW. 11th International Symposium on Wireless Communications Systems (ISWCS); 2014: IEEE.
[26] 26. Gui Z, Yang C, Xia J, Huang Q, Liu K, Li Z, et al. A service brokering and recommendation mechanism for better selecting cloud services. PloS one. 2014;9(8):e105297.
[27] 27. Aznoli F, Navimipour NJ. Cloud services recommendation: Reviewing the recent advances and suggesting the future research directions. Journal of Network and Computer Applications. 2017;77:73-86.
[28] 28. Ma H, Hu Z, editors. Cloud service recommendation based on trust measurement using ternary interval numbers. International Conference on Smart Computing (SMARTCOMP); 2014: IEEE.
[29] 29. Jayapriya K, Mary NAB, Rajesh R. Cloud Service Recommendation Based on a Correlated QoS Ranking Prediction. Journal of Network and Systems Management. 2016;24(4):916-43.
[30] 30. Sumbe S, Jadhav S, Pawar M. A Survey of Modern Approach for Cloud Service Recommendation based on Security. International Journal of Computer Applications (0975–8887) Volume. 2015.
[31] 31. Shafiq O, Alhajj R, Rokne JG, editors. Reducing search space for Web Service ranking using semantic logs and Semantic FP-Tree based association rule mining. IEEE International Conference on Semantic Computing (ICSC); 2015: IEEE.
[32] 32. Rong W, Liu K, Liang L, editors. Personalized web service ranking via user group combining association rule. IEEE International Conference on Web Services (ICWS); 2009: IEEE.
[33] 33. Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics. 2000;28(2):337-407.
[34] 34. Mangayarkkarasi K, Chidambaram M. RANK-BASED WEIGHTED ASSOCIATION RULE MINING TECHNIQUE FOR SECURE CLOUD COMPUTING ENVIRONMENT. Global Journal of Engineering Science and Research Management. 2017;6(3):13-20.
[35] 35. Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering. 2005;17(6):734-49.
[36] 36. Sarwar B, Karypis G, Konstan J, Riedl J, editors. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web; 2001: ACM.