Journal of Theoretical and Applied Information Technology
15th December 2019. Vol.97. No 23

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

DESIGN OF A CONTEXT-AWARE RECOMMENDER SYSTEMS FOR UNDERGRADUATE PROGRAM RECOMMENDATIONS

V.VAIDHEHI[*1], R.SUCHITHRA[2]
[1] Ph.D. Scholar, Jain University, Bangalore-29, India
[2] Associate Professor, Department Of Computer Science, Jain University, Bangalore-29, India
E-mail: vaidhehi.v @christuniversity.in [1], r.suchithra@jainuniversity.ac.in [2],
Corresponding Author: Phone: +09845256910*

ABSTRACT

There are a variety of undergraduate programs available in the education system. Therefore, designing a recommender system based on academics performance alone may not be always helpful. Thus, there is a high demand for knowing other contextual information of the user which can influence the efficiency of the recommender system. Intelligent systems can be designed to predict the contextual information about the user. Selecting the most appropriate undergraduate program by considering different contextual parameters is highly needed for students who have passed class 12. This research work is to design a context-aware recommender system, which recommends under-graduate programs to students of class 12 based on the academic performance and with contextual parameters like financial background, Knowledge level, Group, interested-subject and interested-profession using collaborative filtering approach. This research paper proposes a novel method for creating a rating matrix and the identification and processing of contextual information in an efficient manner. This Context-aware Recommender system is designed based on the predictive values for the various contextual parameters using a contextual modeling approach. Implicit ratings are calculated using the collaborative approach. The results indicate that context-aware recommender engine is more efficient in generating the recommendations thereby improving the user satisfaction level.

Keywords: *Recommender System, Contextual Parameters, Context-Aware Recommender System, Contextual Modeling, Rating Matrix*

1. INTRODUCTION

Higher education institutions across the globe offer a variety of programs in various disciplines for the students after class 12. When it is for students, to choose the right program like BA Music, BSc Physics, and B.Com, after class 12, it is all the more difficult and an important step in one's career. Program selection is essential for every student to decide the career. Therefore, it is highly needed for the student to choose the best program based on some of the factors like marks, personal interests, family details, etc. Recommendations can be generated by using recommender systems (RS).

Recommender systems are widely used in various domains and their use in program selection will enhance the performance of the students effectively. There are various types of recommender systems and this paper is based on the collaborating filtering recommendation system. User satisfaction can be enhanced by using Context-aware recommender systems (CARS) by generating effective recommendations by understanding the user to the specific one or more contextual information. The recommender system is designed by considering students (users), courses (items) and their marks scored as represented by ratings. Recommendations are generated based on academic performance. But, there are many factors that can influence the preference of a user. A user might prefer the LLB program as his parents are lawyers. These situational properties (interested-subject, interested-profession, family background, financial background, Knowledgelevel, Group,etc.) can change the items in the recommendation list. Furthermore, using context in recommender systems provides more trust in recommendations. For recommender systems, the contextual information is considered as the additional information that may be relevant for making a list of recommendations.

With the rise of multiple educational options after class 12, recommendation systems became the basis for modern course advisory systems. The general approaches for implementing recommendation engines, such as content and collaborative filtering, are capable of generating the recommendation list of items for a given user. While these general approaches work well with large sets of user and item information, they lack a general approach for including highly dynamic context-information for generating personalized recommendations list. This paper proposes an approach to improve the performance of the system using context-information for generating personalized recommendations. This research work examined the application of such a system in recommending a suitable undergraduate program. This research is based on multi-criteria based recommendations where the list of recommendations are generated based on contextual parameters. As the final recommendations will depend largely on the contextual information of an individual, this research work is to identify and process contextual information in an effective way. The capabilities of the system are evaluated successfully for prediction, top-N recommendation, and user satisfaction.

The recommender model is built based on a rating matrix. The implicit rating matrix is derived from the academic details for each student. In this research work, the recommender engine is designed using academic details and contextual information The contextual information is predicted by considering the information of the student. The proposed recommender system generates high-quality personalized recommendations for various undergraduate programs in light of the contextual situation. The outline of this paper is indicated as follows: Section 2 elaborates the review of various research activities carried out in this domain. Section 3 explains the various steps involved in this research work. The results are discussed in section 4.

2. LITERATURE REVIEW

Recommender systems are widely used in various domains for their efficiency in recommending appropriate items for their users. Generally, recommender systems are used to recommend movies, music, products, news articles, etc. based on the users' historical data and requirements. Recommender systems can be beneficial in the education sector as the students will have a dilemma in their selection of subjects, programs, and courses. Various research has been done regarding the applications of Recommender systems in other domain whereas its usage in education is comparatively less [1-3]. Thus this research work is to study the usage of recommender systems in education.

The recommender system has three basic components namely users, items and ratings. Recommender engine is the core of the recommender system which works based on the design of user profile and item profile. Recommender systems in the education sector [1] and [4] are designed by mapping students as users, courses/programs as items and student feedback about the course/program as ratings. There exists a variety of design techniques to build a user profile and item profile. There are different types of recommender systems which include a content-based recommender system, a collaborative recommender system, a knowledge-based recommender system[5-6], a demographics based recommender system, hybrid recommender system, constraint-based recommendation [7-8] and context-aware recommender systems [9].

Various techniques are involved in generating recommendations. The collaborative filtering approach works in such a way that users with similar preferences in the past will tend to have similar preferences also in the future. Collaborative filtering methods fall into two categories: memory-based algorithms and model-based algorithms. In memory-based techniques, the value of the unknown rating is computed as an aggregate function of the ratings. Model-based collaborative techniques provide recommendations by estimating parameters of statistical models for user ratings.

Jose Aguilar et.al [10] proposed a general framework for designing knowledge-based recommender systems using Fuzzy Cognitive Maps. Sunita and Lobo [11] proposed a prototype for building a recommender system based on machine learning algorithms in an e-Learning environment using Moodle data. A framework to recommend courses based on the clustering technique is discussed in [12]. Amin Y. Noamana and Fekry Fouad Ahmed [13] proposed a framework for academic advising with an Administrative perspective. Sunita and Lobo [14] proposed a framework for recommending online courses based on Apriori rules. Salam and Qusai [15] proposed a framework for personalized recommender systems in e-Learning based on learning objectives using domain ontology. Onah and Sinclair [16] designed a

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

framework based on collaborative filtering for recommender systems in open online courses. A framework for the design of case-based recommender systems is discussed in [17].

Karzan Wakil et.al [18] designed web-based recommendation systems for universities in IRAQ using a neural network and decision tree classifiers and other relevant attributes to enhance the student selection process. Kongsakun and Fung [19] designed an intelligent recommendation system based on grades using data mining algorithms like neural networks, Support Vector Machine, decision tree and association rules for Universities in Thailand. The design of a course recommender system using linear classifiers is discussed in [20]. Esraa and Mona [21] proposed a web-based model to facilitate the data accessibility of different users of the academic advising system.

Subba and Govindarajulu [22] designed a college recommender system based on student's preferences using recommender system techniques and database querying approaches. Aditi Bhide et.al [23] highlights different recommender mechanisms based on data mining and neural networks for university admissions. Alexander Felfernig et.al [24] proposed a general design approach for building recommender systems for different applications. Janusz Sobecki [25] implemented web-based recommender systems using different types of recommendation techniques and consensus theory. Prem and Vikas [26] elaborated on the design of the recommender system for students using grades. The design of a rule-based recommender system for elective courses is discussed in [27]. The design of a content-based and collaborative based recommendation system using ratings for online courses is discussed in [28].

Amer and Jamal [29] designed a recommender system by using collaborative filtering and association rules to recommend elective courses. The usage of contextual information to create an intelligent recommender engine is elaborated in [30]. G. Adomavicius et.al [30] elaborates on the various ways of identifying the contextual information and incorporating them into the recommender engine. G. Adomavicius et.al [30] introduce three different approaches like contextual pre-filtering, post-filtering, and modeling for incorporating contextual information into the recommendation process. A detailed survey of context-aware recommender systems for technology-enhanced learning is discussed in [31]. K. Verbert 4-dimensional [31] evaluates the recommender systems used in education and outlines future research activities in this field. The use of genetic algorithms to calculate ratings and to design a personalized context-aware recommender system to recommend places of interest are studied thoroughly in [32]. Jiang et.al [33] designed CARS from a cooperative learning perspective. The usage of CARS in recommending a suitable restaurant in multiple platforms like web-based application and android application is discussed in [34]. The general framework for context-aware recommender systems is discussed in [35].

In literature review has given a concise overview of multiple commonly used techniques and implementations. Traditionally, recommendation systems only use the historical data and preferences of users to recommend the undergraduate program. This research paper aims to design a context-aware recommendation system, which uses academic details along with context information financial background, Knowledge level, Group, interested subject and interested-profession to recommend the undergraduate program. The proposed design approach improves the prediction results and enhances the efficiency of the system by generating better recommendations. We specifically focus on the identification of contextual information and the processing of contextual information. Another goal is to make use of the implicit rating matrix instead of the explicit data.

In the existing literature, contextual parameters are considered to be static whereas this research work considers both static and dynamic contextual parameters. This research work identifies the contextual information by using inferential methods where machine learning models are used to identify the given contextual attribute. As the given problem is a multi-criteria based decision problem, this research considers multiple contextual parameters. In the design of the recommender engine, the rating matrix is considered as significant input. The existing recommender system considers the rating matrix as an explicit input from the user. This research proposes an implicit way of creating a rating matrix.

3. METHODOLOGY

The proposed approach is based on the user's context information (financial background, Knowledge level, Group, interested subject and interested-profession), academic performance, and item-based information to filter the item list in a

Journal of Theoretical and Applied Information Technology
15th December 2019. Vol.97. No 23

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

personalized way. The design of context-aware recommender engine is based on the contextual modeling approach, preprocessing and filtering of raw data, context extraction, and creating the rating matrix in an implicit way. The overview of the research work is shown in Figure 1.

3.1 User Profile Acquisition

Questionnaire-based data collection is carried out. 3421 student's marks in various subjects in class 12 and class 10 is considered. Based on the student's data, 53 undergraduate programs are identified. Data preprocessing tasks are carried out to make the data ready for building tasks. Data is cleaned by removing the incomplete information. Instances for which program name is not specified is completely removed from the dataset. When data from multiple students are integrated, uniform naming conventions are followed. Data transformation tasks like normalization and uniform representation of marks are implemented.

The fundamental step in creating a personalized recommender system is to acquire and learn user preferences. The purpose of the user profile is to identify individual users. In the real-world, understanding the user on various attributes involves fuzzy information or knowledge. The challenge of making personalized recommendations is to determine the user profile effectively. Personal information about the user is more important in personalized RS. The user profile is directly proportional to the efficiency of RS. Therefore, the efficient user profile building will improve the efficiency of the RS.

The user profile is not static. It changes periodically. The user profile is also updated periodically by getting feedback (implicit) and explicit feedback is obtained through ratings. Implicit feedback is implemented by observing users' actions. There are different types of maintaining the user profile in a personalized recommender system. They are the individual user profile and Group user profile. Individual user profile provides for only one user's interest/preferences whereas Group user profile describes the common interests or of goals of a group of users.

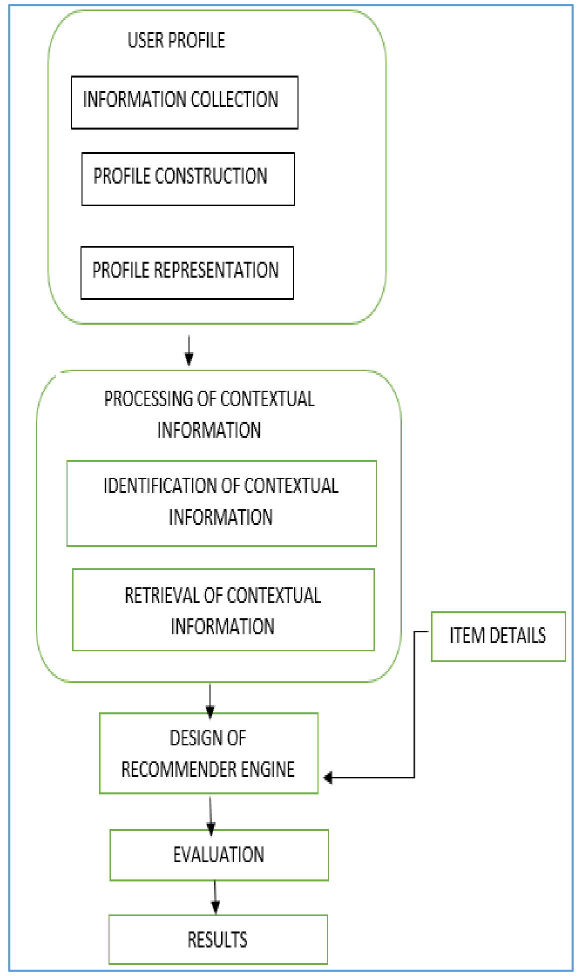

Figure 1 Research Overview

The various steps for user profile acquisition are information collection, profile construction, and representation to provide personalized services. Information collection of the user can be collected in various ways such as explicit user information and implicit user information. Explicit user information can be collected by using the Registration form, Questionnaires, Users are asked to rate items and tracking user queries. Explicit Information includes the following Demographic information (gender, education background, age, location, occupation), Data about interest & preferences (the topic of interest, taste, preferred products, brand preferences, etc), Opinion based information (reviews, comments, feedback) and Explicit ratings. The advantages of having explicit user information are that it is easy to collect and data is with less noise. The drawback of explicit user information collection is that the users have to invest time and effort and privacy concerns.

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

Implicit user information is based on behavior. It can be collected through different sources like Web page logs, Clickstream, Purchase records, Browsing histories and Content or structural information from visited web pages. Implicit information includes Implicit feedback, Implicit ratings, Images, Videos, Posts, Clickstream, User logs, and Web content includes textual content, multimedia content, network information like tags, review, comments, posts, pictures, tweets, videos, audio clips, and social networking information. The advantages of using implicit user information are that it does not involve extra effort, automatically data is collected when the user interacts with the system, and it provides easy and continuous access to data. The drawback of the implicit information collection is that it is very difficult to convert user behavior into user preferences as the accuracy depends on whether the user behavior is interpreted correctly and also involves the privacy issues.

The user profile is constructed in two ways, Knowledge-based and behavior-based. The knowledge-based profile gives explicit knowledge about items and implicit knowledge about users. This approach uses rules based on proposing items by exactly matching the rules with users. Decision rules are used to classify the user's interest based on demographic characteristics. The behavior-based approach uses models to construct a profile and discover useful patterns of the user. Machine learning model, frequency pattern, sequential pattern, neural network, and graph models are the various models used to construct a user profile. The user profile can be represented as a set of weighted keywords, topics, concepts, and ratings. Once the user profile is constructed, it could be used to generate personalized services like personalized search, queries and design context-aware RS.

In this research work, student details are collected through a questionnaire, from various people who finished / pursuing / employed with an undergraduate degree in regular education. The various attributes pertaining to different categories like performance, eligibility, skill set information, current-role, education background, family details, financial status, locality details, personal details, learning style, challenges, and preferences are considered in the questionnaire. A pilot study of the questionnaire is conducted and the questionnaire is updated based on the suggestions. Once the questionnaire is successfully tested, it is used for data collection. Out of 3500 records, 3421 records are considered for mining. 79 student instances had incomplete information. Out of 3421 records, 1653 are male and 1768 are female. The age of people is between 18 to 34 with a mean of 24.7. Data from the questionnaire is represented in .csv format.

Marks in 26 subjects named indicate the marks scored by the student. *Tavg* indicates the average marks in the tenth standard. The minimum value of *Tavg* is 40% whereas the maximum value is 98%. The average value of *Tavg* is 77.86. *Tboard* indicates the board of education in the tenth standard. It includes State-board, CBSE and ICSE. *TSubjects* indicates the subject codes studied in the tenth standard. *XSL* indicates the second language of the student in the tenth standard. *PUAverage* indicates the average marks in twelveth / Pre University. The minimum value of *PUScore* is 40% and the maximum value is 98%. The average value of *PUScore* is 72.38. *PUSubjects* indicates the subject codes studied in twelveth / preuniversity. PU*board* indicates the board of education in class 12. PU*SL* indicates the second language of the student in class 12.

Classroom-participation indicates the classroom participation (participative, listening and not listening) nature of the student. 1738 instances have rated as participative, 833 instances have rated as listening and 850 instances have rated as Not-listening. *Attendance-pattern* indicates the attendance pattern (regular/Irregular) of the student. *Learning-group* indicates the learning group participation (Notworking, working with the group, Hardworking, listening, initiative, participative and contributive) characteristics of the student. *Entranceperformance* indicates student performance in the entrance examinations for the courses. It takes different values as poor, average, above average, good, very good, excellent and NA (not applicable). *Eligibility-criteria* indicates whether the student is eligible to take the concerned course or not. *Cocurricular-extracurricular-activities* indicate the most interesting cocurricular-extracurricular activity in which the student had participated. In the case of multiple interesting activities, a student can give the most preferred one. *Additional-achievements* indicates whether the student has additional achievements or not. 1706 instances had additional achievements. *Knowledge-level* indicates the knowledge level of the students. *English-proficiency* indicates the English proficiency level of the students. *Cognitive-skill* indicates the cognitive skill of the students. *Academic-strength* indicates the academic strength of the students. The attributes, *Knowledge-level*, *Academic-strength*, *English-*

proficiency, and *Cognitive-skill* can take different values as Average, above average, good, very good and excellent as different indicators for the corresponding attribute.

Medium indicates the medium of education (local language medium, English) of the students. 1071 instances had local language medium and 2350 instances had English medium. *Diff-yop* indicates the difference in year of the passing of class 12 and class 10. *Breakinstudies* indicates whether the student had undergone a study break or not. Only 281 instances underwent a study break in the sample. *Additional-qualification* indicates the additional certifications of the student like Diploma, Certificate or No (nothing). Before joining the degree course, 1167 students had completed their diploma, 1128 students had completed the certificate course. *Multiple-Attempts* indicates whether the student had taken more than one chance to pass the examinations. 345 instances used more than one chance to clear the examination. *Fathersqualification* indicates the qualification of the father of the student. *Mothersqualification* indicates the qualification of the mother of the student. Different levels of parental qualification considered are School education, UG, PG, and HigherPG. *Sibling* indicates the number of siblings (0, 1, 2 or more than 2) of the student. *AnnualIncomestatus* indicates the annual income in the family of the student. *Location* indicates the locality(rural, semi-urban and urban) of the student. 1094 instances are based on rural, 1166 instances are based on suburban and 1161 instances are based on urban location.

Interestedprofession indicates the interested profession of the student. *Interestedsubject* indicates the interested subject of the student. For these attributes, the students recorded their most preferred interests. *Learning-challenges* indicates whether the student is a normal or having any physical challenge or learning disability. The sample includes 45 students with a learning disability and 44 students with physical challenges. *CourseMatching* attribute indicates whether the student is satisfied with the course or not. This attribute is a 3 scale measure that specifies 1551 students reported as satisfied 899 students reported as not satisfied and 971 students gave neutral value. *Financial-bk* indicates the financial background of the student. *Group* indicates the group of the student as arts, commerce, and science. *Current-status* indicates the current status of the individual. The population includes 1587 students, 1188 employed people, 394 unemployed people, and 252 homemakers. The population of the study includes people from different parts of the country. All the samples considered for analysis are eligible to take the concerned course in a regular classroom mode of education. *The course* indicates various courses. Student information is collected from 53 undergraduate courses. The minimum number of instances in a course is 52 and the maximum is 70. In this research work, rule-based profile construction and vector space profile representation are followed.

3.2 Contextual Information

The satisfaction of the user can be improved by using contextual information in recommending the items. This paper considers the contextual information as "situational parameters" which may have an effect on the list of items generated in recommendations. In a recommender system, the contextual information plays a pivot role in generating the list the recommendation In general, the context data is classified as fully observable, partially observable and unobservable contextual information. Fully observable contextual information can be completely obtained from the user details. Partially observable contextual information can be (to some extent) obtained from the user details. Unobservable contextual information cannot be obtained from the user details. Based on how the contextual information changes over time, it is classified as static and dynamic. Static contextual information remains unchanged whereas dynamic contextual information changes over time.

Various methods to retrieve contextual information are explicit, implicit and inferential methods. In the explicit method, questionnaire or interview mechanisms are used to collect explicit contextual information. In the implicit method, the past history of the user is used to collect the implicit contextual information. In an inferential method, predictive models are used to infer the contextual information of the user. This method is used in the applications where the past transactions of the same user do not exist. The inferential method is more powerful because of the predictive models used to infer the contextual information.

3.2.1 Identification Of Contextual Information

Contextual information is obtained by selecting the appropriate attributes. Attribute selection in WEKA is a twofold process. In the first step, the attributes are evaluated towards its class label. In the second step, the search method specifies the selection procedure to select the subset of attributes. In WEKA, the attribute selection is implemented based on correlation, information gain, gain ratio, symmetrical uncertainty, repeated sampling and evaluating an attribute with respect to the base-classifier (One R). The different functions to evaluate the attributes are CorrelationAttributeEval, GainRatioAttributeEval, InfoGainAttributeEval, OneRAttributeEval, SymmetricalUncertAttributeEval and, ReliefFAttributeEval. The search method plays a crucial role in selecting attributes. The Ranker search method provides the rank value for each attribute during the evaluation of the attributes. All the above mention evaluation methods use the ranker search method. These rank values are significant for attribute selection.

Therefore the attributes ranked from 1 to 23 are considered as significant contextual parameters. The selected attributes are *InterestedSubject, interested-profession, Pusubjects, Academic-strength, Group, cognitive-skill, knowledge-level, Tsubjects, PU-Average, TAVG, entrance-performance, English-proficiency, PUSL, Medium, PUboard, learning-group, Tboard, break-in-studies, Sibling* and *Location.* The results of attributes selection are categorized as significant contextual information for building recommender models. This

Sl.no	Input	Output
1	fathers-qualification, mothers-qualification, sibling, annual-income-status, location	Financial Background
2	Student details	Interested Subject
3	FQ, MQ, Sibling, Annual-income, location,	Group
4	Academic-strength, co-curricular, add-qualification, add-achievements, entrance-performance, location	Knowledge-level

is shown in table 1.

Table 1 Types of Contextual Information

Type of Contextual information	Fully observable	Partially observable	Unobservable
Static	*Pusubjects,* ***Group****, Tsubjects, PU-Average, TAVG, PUSL, Medium, PUboard, Tboard, Sibling* and *Location*	***knowledge-level****, entrance-performance, English-proficiency*	*learning-group, break-in-studies,* ***financial-background***
Dynamic	*Academic-strength*	***InterestedSubject, interested-profession***	*cognitive-skill*

3.2.2 Retrieval Of Contextual Information

From table 1, *InterestedSubject, interested-profession, Group, knowledge-level, and financial-background* are considered as contextual information in this implementation. *Interestedprofession* is obtained through the explicit method in the questionnaire. The other contextual information *InterestedSubject, Group, knowledge-level, and financial-background* are inferred through predictive models. These models are built as stated in Table 2.

3.3 Item Details

For each undergraduate program details to specify the eligibility criteria for various subjects for each program, the entrance examination status of each program, the various subject combination at tenth class, various subject combination of subjects at the PU level, professional outcome of each program, program fee, type of program as ARTS, COMMERCE, and science, targeted students interest each program, and the academic prerequisite are maintained.

Table 1Predictive Models for Contextual Information

3.4 Design Of Personalized Recommender System

3.4.1 Collaborative Filtering based Recommender System (CFRS)

CFRS automates the process of "word of mouth approach" of recommendations. This method is based on values or ratings of items assigned by other people with similar interests. Here, the user

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

expresses his/her preferences by rating items. Ratings are used to build a user profile. The system then matches these ratings against the ratings submitted by all other users of the system. The result is the set of users' neighbors. This formalizes the concept of people with similar tastes. This is the technique for matching people with similar interests. The major advantage of this method is to enhance the personalized recommendations with improved user satisfaction and high accuracy. One of the major drawbacks of CFRS is that it requires a rating matrix to be collected from huge participants in an explicit way. Therefore, this research addresses the creation of a rating matrix in an implicit way and then using contextual information for generating recommendations to improve the personalized recommendations. This paper considers the academic performance of the students and contextual parameters as ratings in various subjects and using this implicit information the preferences are computed. In this research work, the most appropriate number of neighbors (k value) for neighborhood calculation is set as 5. The different recommender models (user-based and item-based) for the student data set with different similarity measure (cosine and Pearson) is studied and based on the results the user-based model with cosine similarity is implemented in this research.

3.4.2 Proposed Personalized Recommender System

The personalized recommender system is designed in this research work by using contextual information. There are different types of approaches to design context-aware recommender systems. This research uses the contextual modeling approach to design the personalized recommender system. Context-aware recommender systems are one of the efficient ways to design a personalized recommender system. The general recommender system like the user-user collaborative recommender system can recommend items to the user in a generic way. The recommendations are normally generated based on ratings(R) which are taken from User details (U) and item details (I). The recommender system is defined as <User, Item, Ratings> which is stated as R-→U*I.

Contextual Information plays an anchor role in generating a list of top – N recommendations. Contextual data is used to design a personalized recommender system. A personalized touch can be given to the list of recommendations by having an efficient user profiling system. One of the efficient user profiling systems is to infer the characteristics of the user by having a predictive model. Thus, the contextual information is retrieved from the user data. In context-aware recommendation systems, the recommendations are generated based on ratings(R) which are taken from User details (U), item details (I) and Context details (C). The recommender system is defined as <User, Item, Context, Ratings> which is stated as R-→U*I*C.

There are different modeling approaches to design a Context-aware recommender system. They are Contextual pre-filtering, contextual post-filtering, and contextual modeling. In contextual pre-filtering, the contextual information is used as a pre-filtering mechanism to generate the recommendations. In contextual post-filtering, the recommendations are generated in the usual way and the contextual information is used as a post-filtering mechanism to filter the recommendations. In both these approaches, the contextual information is used to filter the recommendations where the recommendations are generated in the usual way by using any of the approaches. But in contextual modeling, the contextual information is used to update the rating matrix and accordingly, the personalized recommendations are generated. Also, this approach supports the generation of recommendations in a multidimensional way whereas, the other approaches generate the recommendations in a two dimensional way. In this research work, five contextual parameters are represented as c1,c2,c3,c4, and c5 which represent financial background, Knowledge level, Group, interested-subject, and interested-profession respectively. The design of the contextual modeling approach is shown in Figure 2. The design of the proposed recommender engine is shown in Figure 3.

The user-based collaborative filtering recommender model with cosine similarity for neighborhood computation is designed by considering both academic and contextual parameters using a contextual modeling approach (named as CARS-UBCF-COS). User-based collaborative recommender model with cosine similarity for neighborhood computation by considering academic performance and not considering contextual parameters is named as UBCF_COS. The proposed personalized recommender engine considers the contextual information *InterestedSubject, Group, knowledge-level and financial-background* which are inferred through predictive models. The other contextual information *Interestedprofession* is collected in the explicit method.

The proposed approach is based on context modeling as the rating matrix will be updated based on the contextual information. Also, multidimensional recommendations can be generated based on each of the contextual parameters. The algorithm used in the design of personalized recommender engines using context data is given in algorithm 1.

Every Context data has a weight value in every program. Program wise CI computation is elaborated in algorithm 2. Based on the list of parameters for every CI, the weight value for all the programs is calculated.

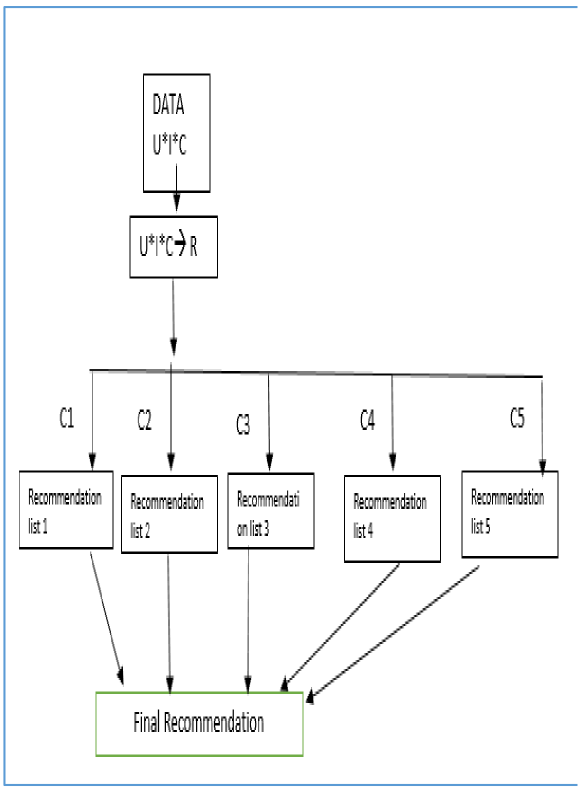

Figure2 Contextual Modeling approach

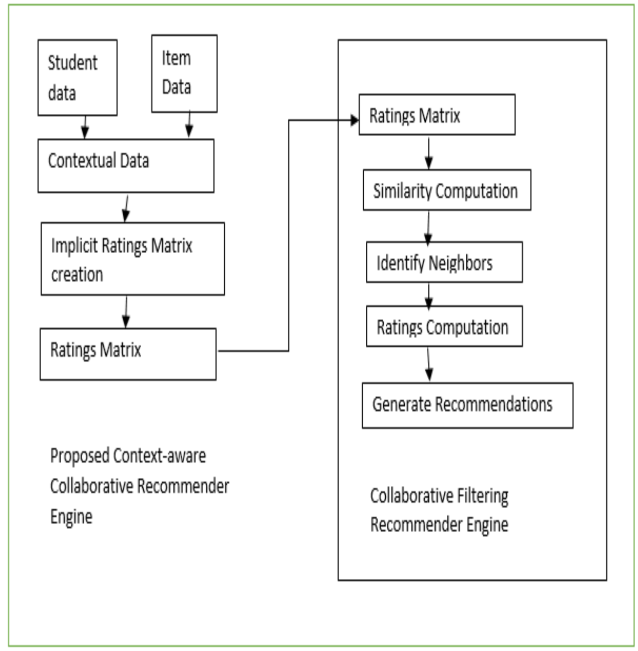

Figure 3 Design Of Proposed Context-Aware Recommender Engine

This creates weight for every parameter of each CI as explained in algorithm 3. The rating matrix is created as explained in algorithm 4. The rating matrix is then preprocessed so that it contains rating value which is from 1 to 5. The rating matrix is then binarized by taking the threshold of 3. Any ratings with a value of 3 and above are taken as 1, otherwise, it is zero. This approach improves the number of ratings in the rating matrix and hence the performance of the recommendation model is also improved.

Algorithm 1 PCARS
//input context data
//output personalized recommendations
//m is the total number of student instances
//cn is the total number of context data
//CI is the array of context data
Step 1; initialize P={p1,p2,….pn1} //set of programs
Step 2: for each CIi
Step 3: Compute program-wise CIs
  Compute weights for each CI
  Compute ratings for each CI
  Normalize the ratings for each CI
Step 4: Compute the final rating as the average of all ratings of all Cis
Step 5: Design Collaborative filtering recommender engine
Step 6: Define the evaluation scheme, training, and test data split
Step 7: Generate recommendations

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

Step 8: Evaluate the recommender model

Algorithm 2 Programwise CI computation

```
//input historical student data
//output list of program-wise parameter (CIP) for each CI
//n is the total number of programs
//m is the total number of student instances
Step 1; initialize P={p1,p2,….pn1} //set of programs
Step 2: for each pi
Step 3: Initialize CIP of each CI as {}
        For each student s in m
        and CI where course=pi
        If CIPi of CI is not added to List, then
        Add CIPi of CI
```

Algorithm 3 weights for each CI

```
// input historical data of the student and program details
// output n2*n1 matrix with weights for each CI
Step 1; initialize P={p1,p2,….pn1} //set of programs
Step 2:initialize CIPs = {pa1,pa2-----pan2} for each CI
Step 3: n1=total number of programs
Step 4:n2 =total number of CIPs for each CI
Step 5 : Intialize CIP[i] as {}
Step 6: Initalize CIP-count[i] as zero
Step 7 j=0
Step 8: for every program pi
for every CIP j from CIP
        tw=0
        n=number of CIP j for Pi
        n1=number of CIPs for each CI
        for every CIP j from CIP
        w( pi, subj)= n/n1
        tw=tw+w(pi, subj)
        x=1-tw
        tso=number of other CIPs not listed
        for every CIP not in CIP of Pi
                w( pi, subj)= x/tso
```

Algorithm 4 Creation of rating matrix

```
// input: the weight of CI for each student (m*n) where m is the total number of students; n is the total number of subjects
//Input program weights as P(n*o) where n is the total number of CIPs and o is the total number of programs
// output m*o rating matrix ---normalized rating matrix
Step 1: initialize variables
Step 2 : //Compute RM
For i= 1 to m
        For j = 1 to o
                Rm(I,j)=0
                For k= 1 to n
                        Rm(I,j)=rm(I,j)+s(I,k)*p(k,j)
Step3://Normalize RM
// x is the maximum rating
        For I = 1 to m
                For j=1 to o
                        Rm(I,j)=round(5*rm(I,j)/x,0)
                        //normalized value between 1 to 5
```

4. RESULTS AND DISCUSSION

Contextual parameters are identified and processed in WEKA. The various multiclass classifiers like J48 and Naïve Bayes are designed to retrieve the contextual information in an inferential way using predictive models. J48 is a decision tree based predictive model. Naïve Bayes predictive model works with a probabilistic approach. The performance of the predictive models is evaluated based on stratified K-fold cross-validation. Accuracy, Precision, Recall, F-measure and ROC area are studied for the predictive models. The learning time of the predictive model is evaluated based on Time taken to build model, Kappa statistic, MAE, RMSE, RAE and Root Relative Error. This approach helps to build the user profile in an efficient way which will improve the design of recommender systems and increase the efficiency of the recommendations generated by the system. The performance of the predictive model is tabulated in Table 3 and the learning rate of the predictive model is tabulated in Table 4.

The context-based recommender system is implemented in R. Recommenderlab library in R has more features to design and implement recommender systems. The aggregated rating matrix created by using all context data is represented in .csv format. In this experiment, the implicit approach for creating the rating matrix is implemented. The implicit way of creating the rating matrix increases the number of ratings in the rating matrix. This approach can be used in situations where the users cannot rate all the items. The academic strength alone is used in the earlier experiment. In this implementation, contextual information is used. Specifically, a Context-aware recommender system based on the inferential model of predicting the contextual information that uses implicit ratings is proposed in this chapter. The results are empirically tested and compared with other non-contextual recommender techniques. The recommender system is generally evaluated for Prediction and Recommendation.

Table 2 Performance of Predictive model

MODEL	Algorithm	Accuracy	Precision	Recall	F-measure	ROC area
financialbac kground	Naïve Bayes	100%	1	1	1	1
	J48	100%	1	1	1	1
Knowledgel evel	Naïve Bayes	95.5861	0.956	0.956	0.955	0.995
	J48	98.86	0.989	0.989	0.988	0.999
Group	Naïve Bayes	100	1.000	1.000	1.000	1.000
	J48	100	1.000	1.000	1.000	1.000
Interestedsu bject	Naïve Bayes	85.004	0.858	0.850	0.850	0.993
	J48	99.7662	0.998	0.998	0.998	1.000

Table 4 Learning rate of the predictive models

MODEL	Algorithm	Time to build the model	Kappa	MAE	RMSE	RAE	Root Relative Error
financialbac kground	Naïve Bayes	0.03	1	0	0	0	0
	J48	0.01	1	0	0	0	0
Knowledge level	Naïve Bayes	0.05	0.9487	0.017	0.0772	14.7826%	32.1992%
	J48	0.085	0.9868	0.0029	0.0383	2.5137%	15.9919%
Group	Naïve Bayes	0.03	1	0.0128	0.0328	3.7415%	7.9536%
	J48	0.08	1	0	0	0%	0%
interestedsu bject	Naïve Bayes	0.05sec	0.8461	0.0108	0.0697	25.0041%	47.4009%
	J48	0.06 sec	0.9976	0.0001	0.0091	0.3261%	6.256%

4.2 Evaluation Of Top – N Recommendations

The user-based collaborative filtering recommender model with cosine similarity for neighborhood computation is designed by considering both academic and contextual parameters using a contextual modeling approach (named as CARS-UBCF-COS). User-based collaborative recommender model with cosine similarity for neighborhood computation by considering academic performance and not considering contextual parameters is named as UBCF_COS.

The contextual approach shows the enhancement in the performance of the recommender system which is shown in Table 5.

Table 5 Comparison of CARS-UBCF-COS and UBCF-COS

Proposed Models	Number of Recom mendati ons	TP	FP	FN	TN	Precisio n	Recall	TPR	FPR
CARS-UBCF-COS	1	0.99998	0.00001	0.00001	52.9999	0.99999	0.99999	0.99999	0.000001
UBCF-COS	1	0.99855	0.00145	0.00145	52.9986	0.99855	0.99855	0.99855	2.73E-05

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

User-based evaluation metrics like Novelty, Diversity, Unexpectedness, coverage, and Serendipity. In this research work for evaluating the recommendation list level novelty, random sampling of 100 instances chosen from 3421 instances. CARS-UBCF-COS recommender model is used to generate different recommendation lists varying from n=1 to n=3. All the items within the recommendation list for 100 test instances for the different number of recommendation items varying from the list size of one to three are found to be unique. Therefore, count (users) = 0. Thus,

Recommendation list level novelty = $\frac{100-0}{100}$ which is found to be 1 which specifies that novelty is achieved in the recommendation list.

In this research work for evaluating the recommendation list level novelty, random sampling of 100 instances chosen from 3421 instances. CARS-UBCF-COS recommender model is used to generate different recommendation lists varying from n=1 to n=3. As it is an undergraduate program recommendation, redundancy of lists in the list is checked to compute the similarity. All the items within the recommendation list for 100 test instances for the different number of recommendation items varying from the list size of one to three are found to be unique. Therefore, count (users) = 0. Thus, Diversity is found to be 1 which specifies that the list of recommendations generated by the system is diverse in nature.

In this research work for evaluating the primitive recommender based unexpectedness, random sampling of 100 instances chosen from 3421 instances. CARS-UBCF-COS recommender model is used to evaluate the Primitive recommender based unexpectedness in the recommendation lists by considering the number of recommendations as 1 for 100 users. Various predictive models are designed as mentioned earlier. Only for 2 users, the predicted and recommended item is found to be the same. Therefore the unexpectedness is evaluated to be 98% which signifies that the performance of the recommender system is found to be stable.

In this research work for evaluating the coverage, random sampling of 100 instances chosen from 3421 instances. CARS-UBCF-COS recommender model is used to evaluate the item space coverage by considering the number of recommendations as 1 for 100 users. 49 out of 53 items are found on the recommendation list. Thus item space coverage is calculated to be 0.9245 which signifies that the performance of the recommender system covers a wide range of items.

In order to evaluate user space coverage, out of 899 instances where the chosen program is not satisfied with their interest, 100 instances are chosen for evaluation. Only 2 recommendations where matching with the program studied. $User\ Space\ coverage = \{Recommended\ value\} - \{Actual\ value\}$. Since the users are not satisfied with the chosen program, the effectiveness of the recommender system is found to be 98%. Genre coverage is similar to diversity which is found to 1 which is highly diverse in nature. The proposed CARS-UBCF-COS recommender model is evaluated on offline mode for user evaluations on different parameters like Novelty, Diversity, Unexpectedness, item coverage and user coverage. The results are tabulated in Table 6.

Table 6 CARS metrics

Sl.no	Metric	Test case	Results
1.	Novelty (Recommendation list level)	100 random samples; N=1 to N=3	No redundant items found
2.	Diversity	100 random samples; N=1 to N=3	Unique
3.	Unexpectedness (Primitive Recommender based Unexpectedness)	100 random samples; N=1	98%
4.	Coverage (item coverage)	100 random samples; N=1	49 out of 53 items recommended 92.45%
5.	Coverage (User space Coverage)	899 (100)	2 matches 98%

4.3 Evaluation Of Prediction

Prediction accuracy metrics such as mean absolute error, root mean squared error measure the accuracy with which the recommender system can predict the user's ratings of the test items. This is applicable to the domain where users are providing explicit/implicit ratings. The item-based collaborative filtering recommender model with

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

cosine similarity is designed by using a contextual modeling approach that is named CARS-IBCF-COS. The recommender system is evaluated for its performance in predicting the ratings. It is tabulated in Table 7.

Table 7 Prediction of ratings using CARS

Recommender Model	RMSE	MSE	MAE
CARS-UBCF-COS	0.2123524	0.04509356	0.1065010
CARS-IBCF-COS	0.6492441	0.42151795	0.5201597

5. CONCLUSION

This work is to help the preuniversity students to select the appropriate undergraduate program. The objective of this work is to support parents, teachers, counselors, and higher education institutions to have the right students for the right courses. The primary focus is to minimize the dropouts in the institutions. The idea is not to replace a human being with the system but to assist people in taking effective decisions. This research work is focused to improve the academic performance of the students by enabling the students to study appropriate programs in higher education institutions. This work will help the students to improve their academic performance by studying the appropriate course based on individual skillset. In a classroom environment, the teaching-learning process will be improved as the right students are there for a course. Finally, parents will have a room of relief.

This research shows the design of a contextual information based recommender system using the collaborative approach. The rating matrix is crucial for designing the recommender system. This implicit approach of designing the rating matrix using domain knowledge of the application along with contextual information enhances the performance of the recommendation models. The proposed method for creating a rating matrix is found to be effective as it would not be possible to get feedback from the students in an explicit way. The proposed method of identifying and processing contextual information to design the recommender engine is found to be efficient in its performance.

As of now 53 courses in different disciplines of the undergraduate program are considered. As a future advancement, the performance of the system can be analyzed for more number of courses in different disciplines like medical, paramedical courses. Also, the size of the dataset can be enhanced for each category of courses. Recommendations can be further enhanced by having recommender models using deep learning. This paper considers five contextual parameters only. But the performance of the recommender engine can be further improved by considering more contextual parameters and building appropriate predictive models to predict the contextual parameter. Also, deep recommender engines can be designed based on multiple recommender models.

REFERENCES:

[1] H. Drachsler, K. Verbert, O. C. Santos, and N. Manouselis, “Panorama of Recommender Systems to Support Learning,” *Recomm. Syst. Handb.*, pp. 421–451, 2015.
[2] D. Gui-Qin, Z. Yan-Song, and H. Yu-Min, “Research on selection system based on Bayesian recommendation model,” *Adv. Mechatron. Syst. (ICAMechS), 2011 Int. Conf.*, pp. 35–38, 2011.
[3] K. N. Rao, “Application Domain and Functional Classification of Recommender Systems — A Survey,” vol. 28, no. 3, pp. 17–35, 2008.
[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Syst.*, vol. 46, pp. 109–132, 2013.
[5] S. Bouraga, I. Jureta, S. Faulkner, and C. Herssens, “Knowledge-Based Recommendation Systems: A Survey,” *Int. J. Intell. Inf. Technol.*, vol. 10, no. 2, pp. 1–19, 2014.
[6] R. Burke, “Knowledge-based recommender systems”
[7] A. Felfernig and R. Burke, “Constraint-based recommender systems: technologies and research issues,” *Proc. 10th Int. Conf. Electron. Commer. ICEC ’08*, vol. 8, no. 5, pp. 1–10, 2008.
[8] A. Felfernig, G. E. Friedrich, D. Jannach, and M. Zanker, “Developing Constraint-based Recommenders,” *Recomm. Syst. Handb.*, pp. 187–215, 2011.
[9] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval, “Context-Aware Recommender Systems for Learning: A Survey and Future Challenges,” *Learn. Technol. IEEE Trans.*, vol. 5, no. 4, pp. 318– 335, 2012.
[10] J. Aguilar, P. Valdiviezo-Díaz, and G. Riofrio, “A general framework for intelligent recommender systems,” *Appl. Comput. Informatics*, vol. 13, no. 2, pp. 147–160, 2017.

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

[11] Matthay ER," Counseling for college: A professional's guide to motivating, advising and preparing students for higher education" (2nd ed.) Princeton
[12] Feghali, T., Zbib, I., & Hallal, S. A Web-based Decision Support Tool for Academic Advising. Educational Technology & Society, ogy14 (1), 82–94., 2011
[13] A. Y. Noaman and F. F. Ahmed, "A New Framework for e Academic Advising," Procedia Comput. Sci., vol. 65, no. Iccmit, pp. 358–367, 2015.
[14] S. B. Aher and L. L.M.R.J., "A Framework for Recommendation of courses in E-learning System," Int. J. Comput. Appl., vol. 35, no. 4, pp. 21–28, 2011.
[15] S. Fraihat and Q. Shambour, "A Framework of Semantic Recommender System for e-Learning," J. Softw., vol. 10, no. 3, pp. 317–330, 2015.
[16] O. D.F.O. and J. E. Sinclair, "Collaborative filtering recommendation system: a framework in massive open online courses," Proc. 9th Annu. Int. Technol. Educ. Dev. Conf., pp. 1249–1257, 2015.
[17] F. Lorenzi and F. Ricci, "Case-based recommender systems: A unifying view," Lect. Notes Comput. Sci. (Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 3169 LNAI, pp. 89–113, 2005.
[18] K. Wakil, B. Akram, N. Kamal, and A. Safi, "Web Recommender System for Private Universities ' Admission in Iraq : UHD Case Study," vol. 4, no. 5, pp. 329–341, 2014.
[19] K. Kongsakun and C. C. Fung, "Neural network modeling for an intelligent recommendation system supporting SRM for universities in Thailand," WSEAS Trans. Comput., vol. 11, no. 2, pp. 34–44, 2012.
[20] Q. E. Booker, C. Business, M. State, and M. Hall, "A student course recommender system using linear classifiers : an experimental system."
[21] E. Afify and M. Nasr, "A Proposed Model for a Web-Based Academic Advising System," Int. J. Adv. Netw. Appl., vol. 3361, pp. 3345–3361, 2017.
[22] Y. S. Reddy and P. P. Govindarajulu, "College Recommender system using student preferences voting : A system development with an empirical study," vol. 18, no. 1, pp. 87–98, 2018.
[23] A. Bhide, K. Karnik, P. Badge, V. Joshi, and V. M. Lomte, "A Review on Recommender Systems for University Admissions," vol. 5, no. 12, pp. 141–146, 2016.
[24] A. Felfernig, M. Jeran, G. Ninaus, F. Reinfrank, S. Reiterer, and M. Stettinger, "Basic approaches in recommendation systems," Recomm. Syst. Softw. Eng., pp. 15–37, 2014.
[25] J. Sobecki, "Implementations of Web-based Recommender Systems Using Hybrid Methods," Int. J. Comput. Sci. &amp, vol. 3, no. 3, pp. 52–64, 2006.
[26] P. M. V.Sindhwani, "Recommender systems," Commun. ACM, pp. 1–21, 2010.
[27] Y. Jiun and K. Wen, "CareerMaker : An Adaptive Elective Courses Recommendation System," pp. 1118–1125, 2013.
[28] J. M. Reddy and T. Wang, "Online Study and Recommendation System," 2014.
[29] A. Al-badarenah and J. Alsakran, "An Automated Recommender System for Course Selection," vol. 7, no. 3, pp. 166–175, 2016.
[30] G. Adomavicius, B.Mobasher, F. Ricci and A. Tuzhilin, "Context-Aware Recommender Systems," Association for the Advancement of Artificial Intelligence, pp. 67–80, 2011.
[31] K. Verbert et al., "Context-aware recommender systems for learning: A survey and future challenges," IEEE Trans. Learn. Technol., vol. 5, no. 4, pp. 318–335, 2012.
[32] S. A.El-Moemen, T. Hassan, and A. A.Sewisy, "A Context-Aware Recommender System for Personalized Places in Mobile Applications," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 3, pp. 442–448, 2016.
[33] N. Jiang, C.-W. Tan, W. Wang, H. Liu, and J. Gu, "Understanding Cooperative Learning in Context-Aware Recommender Systems: A User-System Interaction Perspective," ICIS 2017 Transform. Soc. with Digit. Innov., pp. 0–11, 2018.
[34] S. Saeedi, X. Zou, M. Gonzales, and S. Liang, "Design and Implementation of a Context-aware Personalized Recommendation System : Mobile and Web Application," no. September, pp. 1–14, 2017.
[35] W.Beer and W.Hargassner, " General Framework for Context-Aware Recommendation of Social Events,"Proc.2nd International Conference Intelligent Systems Applications, 141-146, 2013