StudieMe: COLLEGE RECOMMENDATION SYSTEM

Vidish Sharma, Tarun Trehan, Rahul Chanana
Computer Science & Engineering Department
Jaypee Institute of Information Technology
Noida Sector 62, Uttar Pradesh, India.
{vidishsharma1311, taruntrehen, rahulchanana1998}
@gmail.com

Suma Dawn
Computer Science & Engineering Department
Jaypee Institute of Information Technology
Noida Sector 62, Uttar Pradesh, India.
{suma.dawn@jiit.ac.in}

Abstract—In this work, we present a novel web platform for a college selection process. Having a recommendation system as a helping hand to give them detailed information about the options they have and the best options to choose from according to their caliber is a huge requirement. In this paper, we worked on a designing a recommendation system that could understand the skill set and interest of a user through the data from the User's Profile to suggest recommended options of colleges for the users to select. We have developed the college recommendation system as a web platform which gives the result as top matched colleges for a particular user.

Keywords—Cosine Similarity, TF-IDF Vectorisation, Recommendation System, Web Architecture, Web App, Website, Design, React, Python, Machine Learning, Document Similarity, Web Crawler.

I. INTRODUCTION

In today's world, we look around undergraduates struggling for their post graduation. They face problems in finding appropriate colleges or finding their interests also as they don't really know what they want to pursue. They problems they face like increasing competition level, sometimes they have fewer grades, building peer pressure, etc. leads to tensions of getting enrolled in the best. Also they have to think about the projects they have done till now, the internships they have worked for creates an existing skill set and area of interest for them. The college they take should be according to their interest area and should be matched equally with their skill set. To tackle this complex problem faced by students we came up with an idea of 'StudieMe' a college recommendation system which takes into account all the major factors on which the choice of the right college depends. This will help students in getting an aggregate presentation of their requirements.

The system will help the students in selecting the right college for them as important factor such as score, preferred study subjects / domain of study, areas and countries and other parameters are considered for presenting the best matches. Also we have some additional options for the students who want to pursue P.H.D as these students want to pursue their degree under some professor. So, we also provide the students with the most matched faculties (according to their subject of interest) with the student's profile. This also helps students in constraining their search criteria. Further, the system is equipped with crawling university sites at regular interval so that the users and be presented with wholesome and up-to-date information. The design principles used in the presentation allow for simplicity and brevity, hence data is presented with minimal navigation.

Section II presents a brief study of various literature and comparison with existing websites. In Section III, the system architecture, including design, workflow, and methodology are shown. Results and discussion is present in section IV. Section V concludes the work presented in this paper.

II. LITERATURE SURVEY

- Semantic Cosine Similarity - Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi.
- Document Similarity for Texts of Varying Lengths via Hidden Topics - Hongyu Gong, Tarek Sakakini, Suma Bhat, Jinjun Xiong.
- Calculating Semantic Similarity between Academic Articles - Ming Liua , Bo Langa and Zepeng Gua.

Papers [1], and [2] explain the various methods that can be used for measuring similarities in documents and various semantics. Enhancement of cosine similarity measurement may be implemented by incorporating semantic checking between dimensions of two-term vectors from. This technique expects to expand the closeness esteem between two term vectors which contain semantic connection between their measurements with various sentence structure. Paper [3] presents a document matching approach to bridge this gap, by comparing the texts in a common space of hidden topics. Here, we are evaluating the similarity algorithm on two tasks and find that it consistently performs strong baselines. We likewise feature the advantages of the domain knowledge to text matching.

Further, existing websites like Shiksha.com, CollegeDunia.com, Career360.com, Yocket.com etc presents the ranked list of universities based only on a single factor of GRE score / or an entrance score. Also, all the user data has to be fed manually. So, that's too much work for a user to do for getting the desired results.

978-1-7281-2068-3/19/$31.00 ©2019 IEEE

III. SYSTEM ARCHITECTURE: Design Principles, Workflow and Methodology

The web architecture is used to provide a User-centric dashboard which uses the implemented algorithms and give the user a smooth and easy to use interface. This is divided into 3 servers-

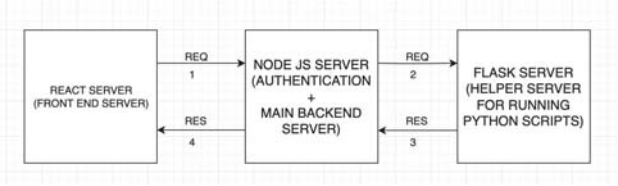


Figure 1: Basic Server Query Architecture

- React Server — React Js is a front-end framework for making single-page applications. We are using React Js for providing all GUI functionalities on the user side. It is directly in contact with the Node Js server. The frontend responds to the user actions and requests or sends data from/to node Js server.
- Node JS Server — Node Js is an open-source, cross-platform JavaScript run-time environment which executes JavaScript code outside the browser. We are using Express with Node Js for creating a request-response server. The express server is used for tasks divided into 3 categories, namely, (i) providing data from database - It takes req from the react server, fetch the data from database (PostgreSQL) and return it to frontend; (ii) Authentication - It does the authentication and serialization of the user when called from frontend on the authentication route: (iii) Calling Flask Server - When a Request is made by the frontend that needs Results from python script on a particular dataset, it request results from the flask and returns it to the frontend.
- Flask Server — Flask Server act as a helper server to the Node Js backend. The role of this server is to take test data, run python scripts and provide results accordingly. It is only reachable by Node Js server and not by frontend directly to avoid security breach of data.

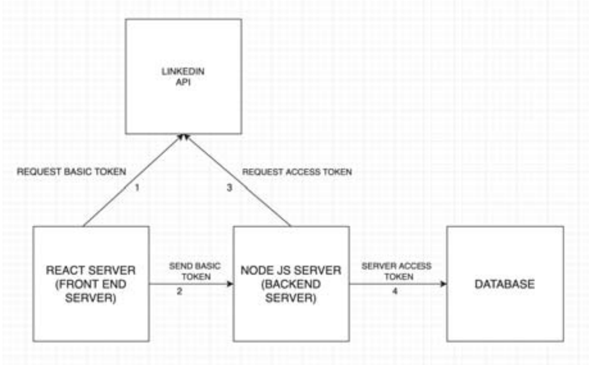
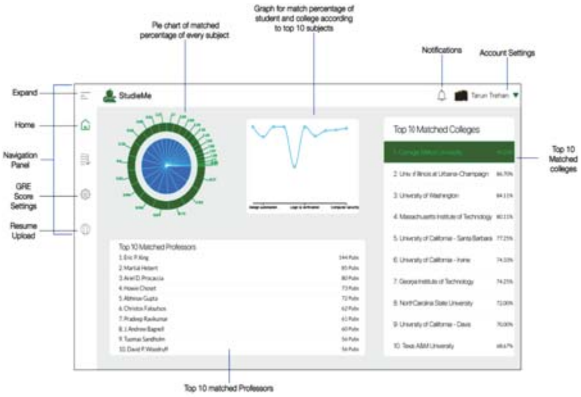


Figure 2: JWT based 3rd party Authentication Architecture

Data Collection can happen either by online entry or by picking up the LinkedIn profile of individuals. The LinkedIn authentication takes place in 2 steps and on frontend and node js server. First the react server calls the LinkedIn API which redirects the user and asks permission from the user to allow our app do authentication through his LinkedIn account also to use his basic information to provide accurate results in the app functions. After a successful grant from the user, LinkedIn provides a basic authentication token which could be used to get access token from LinkedIn API. This basic token is then sending to the Node Js server. Node Js then request the LinkedIn API with the basic token to get access token. After successful grant of the token from LinkedIn API, the LinkedIn Access token is saved in the database and a new Javascript web token is signed and created which is further used in authentication of the user. The JWT is sent to frontend as a sign of successful completion of the authentication process and as a further means of data retrieval. The access token is used to fetch user details and basic LinkedIn user profile from the LinkedIn API.

(a) Design Principles


Pie chart of matched percentage of every subject
Graph for match percentage of student and college according to top 10 subjects
Notifications
Account Settings
Expand
Home
Navigation Panel
GRE Score Settings
Resume Upload
Top 10 Matched colleges
Top 10 matched Professors
StudieMe
Tarun Trehan

Top 10 Matched Colleges

1. Carnegie Mellon University [illegible]
2. Univ of Illinois at Urbana-Champaign 86.70%
3. University of Washington 84.11%
4. Massachusetts Institute of Technology 80.11%
5. University of California - Santa Barbara 77.25%
6. University of California - Irvine 74.33%
7. Georgia Institute of Technology 74.25%
8. North Carolina State University 72.00%
9. University of California - Davis 70.00%
10. Texas A&M University 68.67%

Top 10 Matched Professors

1. Eric P Xing 144 Pubs
2. Martial Hebert 93 Pubs
3. Ariel D Procaccia 80 Pubs
4. Howie Choset 73 Pubs
5. Abhinav Gupta 72 Pubs
6. Christos Faloutsos 62 Pubs
7. Pradeep Ravikumar 61 Pubs
8. J Andrew Bagnell 60 Pubs
9. Tuomas Sandholm 56 Pubs
10. David P Woodruff 56 Pubs

Figure 3: Design and explanation of various sections in user dashboard.

This platform aims to develop Web-based College recommendation architecture with the interactive and graphical content display, it should follow some principles.

- Serve for the Recommendation dashboard. This platform mainly provides the detailed information through graphs and charts about colleges and best-suited options to choose from, so it should follow the principle of thorough and clear information display and accurate recommendation is the key idea of developing this platform.
- Simplicity and brevity. The structure of the platform, color and font should be simple and brief, the navigation of the web should be clear and definite, an instructive map should be detailed, the style of whole pages should be unified.
- Easily operated. The user may have a low skill of educational technology, so the plat-

form should be simple and easily operated, and they don't need the training to use the platform. The platform should run fast, provide the detailed and appropriate prompt message.

- Safe, stable and easily maintained. The platform should run stable, and assure the safety of the content, and the platform can be easily updated and extended because of the technology.

(b) Methodology

The first step is to take up the user's resume data from their LinkedIn profile. LinkedIn is the best option for taking up the users profile and thus get to know about the user's knowledge and ability. All these resumes are later on put up as a data set. Resumes contain skills, knowledge, qualifications, projects and job experience. The other dataset consists of the data of colleges of the USA that provides a master degree in the subjects provided in the subject dataset. This is required for the matching of both the dataset. This matching is very important to determine which college a user can get with the help of his abilities and skills which leads to the best choice. These colleges are ranked according to 26 different subjects that are offered. All these colleges have their GRE cutoffs (out of 340) that we have scraped from the web. GRE is another important factor needed to determine the college as most of the colleges in the USA take their GRE score as a necessary selection process. The next step is to use faculty dataset. It is necessary to help the user know which faculty is the best they are able to work under. This will work by finding the similarities in resumes and colleges dataset and thus will suggest them the best faculty to work under. We look for the top 10 faculties of the 26 subjects that are being offered and the number of publications made by them. By taking into context the number of publications, we give the faculties their ranking and determine the top faculty profile. Subject raw data description is made and college is ranked according to their faculty publications. This will increase the abstract level to increase accuracy. Similarities in resume and college will help determine the faculty under which they should work.

(c) Algorithms used

- TF-IDF Vectorization — TF-IDF stands for term frequency and inverse document frequency.

Term frequency determines the count of numbers of words occurring in a given document. The Inverse Document Frequency is the number of times a word is occurring in the corpus of documents.

The formula for the term frequency is stated below:-

$TFi,j = ni,j$

$\Sigma k\ ni,j$

The formula for the inverse document frequency is stated below:-

$idf(w) = \log(n/dft)$

TF-IDF is the algorithm to determine the importance of the words by weighting them. The words withthe less occurrence are weighed less while the words occurring more have higher weight.

$wi,j = tfi,j * \log(N/dfi)$

We used TF-IDF for finding the weight in the subject dataset, resume dataset and faculty dataset.

- Cosine Similarity —Cosine similarity between two sentences can be found as a dot product of their vector representation.

There are various ways to represent sentences/paragraphs as vectors.

$similarity = \cos(\theta) = A.B\ /\ ||A||.||B||$

We got the weights by the TF-IDF for subject dataset, resume dataset and faculty dataset. By getting the weights, we compared the subject dataset with resume dataset and resume dataset with faculty dataset and mapping is done. Mapping done is m to n onto mapping.

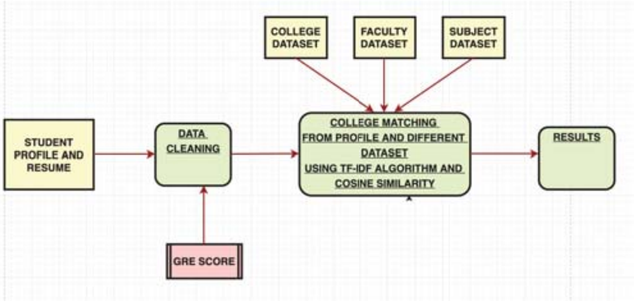

COLLEGE DATASET
FACULTY DATASET
SUBJECT DATASET
STUDENT PROFILE AND RESUME
DATA CLEANING
COLLEGE MATCHING FROM PROFILE AND DIFFERENT DATASET USING TF-IDF ALGORITHM AND COSINE SIMILARITY
RESULTS
GRE SCORE

Figure 4: Module Diagram

(d) Workflow

Dataset used:

There are three datasets-

- User Dataset - It contains the user's resume that contains their projects and achievements and the other important factor that is their GRE score. It also contains the user's LinkedIn Data.
- College Dataset - It contains the data of the colleges of the United States and the GRE score required to get enrolled in the respective subjects. This dataset consists of 20320 values of all the colleges and the subjects that are offered by the institute with the GRE score required in each subject. The database is updated using web crawlers at regular intervals.
- Faculty Dataset - It contains the data about the faculties that an institute offers to work under different subjects along with their number of publications. The dataset consists of 1237 faculties from all the institutes along with the number of publications done by them.

User dataset will be in text format while the college and the faculty dataset will be in the CSV format and will be implemented in python using Pandas data frame. The population of the college and the faculty datasets were done by the web crawler (from websites like QSworld.com, shiksha.com, collegedunia.com, etc).

Input and User Authentication: The very first step is user authentication using LinkedIn so that we can get the user’s information like his internships or his subjects or his projects, etc. Then the user will upload his resume also as we need to collect the maximum data about the user as we can. After this the user will input his GRE score for further college selection.

```
Function #1: Input Function
Resume=open(r”/Users/vidishshsarma/Documents/Minor-
Project/Data/Resumes/vidish.txt”)
doc = str(resume.read())
gre = 320 #This GRE Score will be user input.
de = ds.loc[ds['GRE Score'] >= gre]
```

Data Cleaning: The next step of the workflow is data cleaning. It is done for three datasets, i.e user resume dataset, subject raw description dataset and faculty profile dataset by removing stop words, lemenization, stemming which are done through the inbuilt library of NLTK.

```
Function # 2: Data Cleaning
from nltk.corpus import stopwords
stopWords = stopwords.words('english')
tfidf_vectorizer = TfidfVectorizer(stop_words=stopWords)
Output: Cleaned dataset of the user, subjects, and faculty.
```

Using Matching Algorithms (Function - 3): Then comes the process of matching. Matching is done among the three datasets i.eUser’s information dataset, subject dataset, and the colleges' dataset by applying the method of cosine similarity and TF-IDF vectorisation.
Result- It returns the subject that matched with the highest probability. Also names the top 20 colleges ranked according to the subjects and the highest GRE score.

Next step is perform matching between subject and user data. The initial training dataset is said to contain almost 26000 entries about various subjects for matching with user data. The choice of subject is found using cosine similarity between the user vector and subjects.

```
Function # 3: Matching Algorithm
train_set = [doc1.. doc26000]
tfidf_vectorizer = TfidfVectorizer(stop_words=stopWords)
tfidf_matrix_train =
tfidf_vectorizer.fit_transform(train_set)
result = cosine_similarity(tfidf_matrix_train[0:1] , tfidf_-
matrix_train)
sub_rank = []
for x in top_5:
   sub_rank.append(result_3.index(x))
subs = []
for x in sub_rank:
   subs.append(subjects[x + 1])
```

Calculating Score: Moving onto the next step we calculate scores of these 20 colleges and sort them and pick the top 15 from them. These top15 are picked according to the minimum score using inbuilt libraries of numpy and pandas data-frame.

```
Function # 4:Sorting college data for individual user.
frames = [dv_0,  .. dv_15]
dv = pd.concat(frames)
average = []
for x in my_colleges:
   rank_1 = []
   rank_1.append(dv.loc[dv['Colleges'] == x, 'Rank'])
    dc = dv.loc[dv['Colleges'] == x, 'Rank']
   rank_arr = dc.values
   average.append(np.average(rank_arr))
average_college = []
for x in average_rank:
   average_college.append(my_colleges[x])
#Output- Top 15 colleges for the user.
```

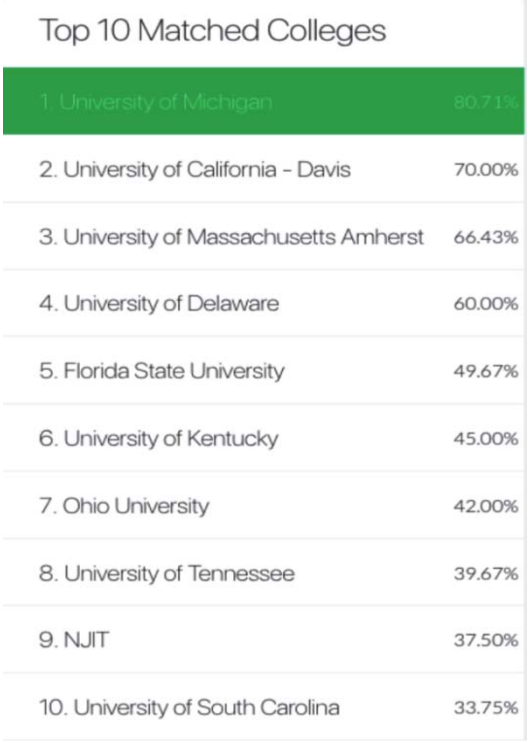
Final Matching and giving Results: In this step, matching is done between the faculties that we get according to the top subjects (calculated in the function 2) of the top 15 colleges that we get from step 3 along with the user resume by the method of cosine similarity and TF-IDF vectorisation. The matching of faculties to subjects and user preference is based on the publication details of the students.

```
Function -#5: #Facutly matching-
faculty = []
for x in average_college:
   faculty.append(da.loc[da['College'] == x , 'Faculty'])
   faculty_arr = np.asarray(faculty)
publication = []
for x in average_college:
     publication.append(dx.loc[dx['Colleges'] == x , 'Publi-
cations'])
   publication_arr = np.asarray(publication)
#Output: Display of the best 10 colleges from the above
#15 colleges according to the college ranking and their top
#10 faculties matched as per the subject choices.
```

IV. RESULTS & DISCUSSION

Using the above-presented method, users can get the top 10 best-matched colleges for which she can apply for Masters. The dashboard also displays the chosen institute's top 10 best-matched faculties for the user under whom she can pursue her studies. The user's skills and knowledge matching to different subjects gives her dominant subject interests as the most match subject. While existing websites like Shiksha, College Dunia, Career 360, etc present the choice of universities based on GRE score or a particular entrance exam, this framework considers multiple parameters for the choice including user preference of subjects and faculties. Further, all user data can be fetched from the LinkedIn profile itself and the manual entry of data is not required. Also, user's LinkedIn profile or the resume profiles are considered for better matching of the institutes/university of study as well as for getting the accurate results. The various outputs are shown in figures 5 – 7. The final dashboard is depicted in Figure 3.

Initially, the ten best-matched colleges for Users profile data (LinkedIn Data). The percentage at the side of the college names shows us how much is the similarity between the college and the user. This is depicted in figure 5.



Top 10 Matched Colleges

1. University of Michigan	80.71%
2. University of California - Davis	70.00%
3. University of Massachusetts Amherst	66.43%
4. University of Delaware	60.00%
5. Florida State University	49.67%
6. University of Kentucky	45.00%
7. Ohio University	42.00%
8. University of Tennessee	39.67%
9. NJIT	37.50%
10. University of South Carolina	33.75%

Figure 5: Top 10 Institute Listing

Secondly, the user's preferred match with different subjects in whom she can pursue masters is depicted in figure 6. So, the number on the circumference of the graphs shows which is the how much a subject matched with the user profile and hence will tell us the best-matched subject for that user.

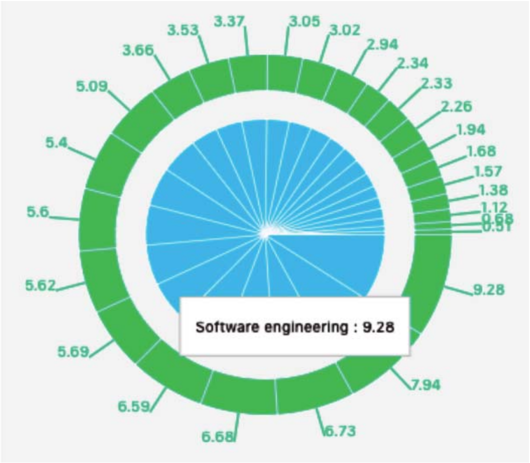
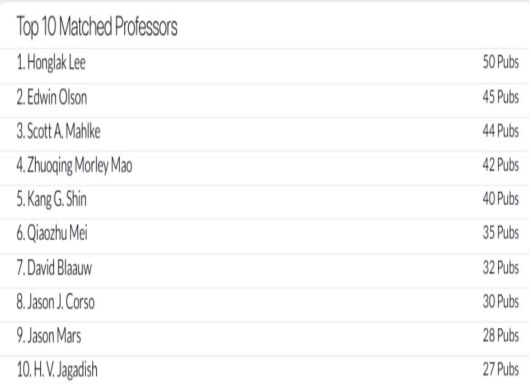


Figure 6: Subject Matching

Figure 7 shows the best-matched instructors or faculties for the preferred subject. This is based on the publication data of the instructors.



Top 10 Matched Professors

1. Honglak Lee	50 Pubs
2. Edwin Olson	45 Pubs
3. Scott A. Mahlke	44 Pubs
4. Zhuoqing Morley Mao	42 Pubs
5. Kang G. Shin	40 Pubs
6. Qiaozhu Mei	35 Pubs
7. David Blaauw	32 Pubs
8. Jason J. Corso	30 Pubs
9. Jason Mars	28 Pubs
10. H. V. Jagadish	27 Pubs

Figure 7: Top 10 matched Instructors

V. CONCLUSION

StudieMe, the college recommendation system can be useful for every student who wishes to pursue higher studies. The user data can be aggregated from their profiles or from their LinkedIn profiles. The choice of institutes that is recommended is based on the user's preference of subjects and their individual scores in exams such as GRE or other exams. This system can suggest colleges to users without requiring the latter to search for them elsewhere. It is a very effective and convenient tool for applicants. As only a handful of existing recommenders designed for finding appealing colleges have been developed in the past, this system enriches the searching capabilities while simplifying its navigation using proper design principles.

In this work, we have proposed a college recommendation system which applies TF-IDF vectorisation and cosine similarity to predict the best-matched subject and colleges for the profile of the user. To improve the matching we use the college's faculties and their publications.

Our recommendation model is a unique one-step solution to applicants who are looking for colleges. We claim that our recommender is effective and advantageous as the suggestions generated by our recommendation system are relevant and accurate. The system can be used for graduates and undergraduates who wish to pursue higher studies without the hassle of visiting multiple sites.

REFRENCES

1. Huang, A. (2008, April). Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand* (Vol. 4, pp. 9-56).
2. Rahutomo, F., Kitasuka, T., & Aritsugi, M. (2012, October). Semantic Cosine Similarity. In *The 7th International Student Conference on Advanced Science and Technology ICAST*.
3. Gong, H., Sakakini, T., Bhat, S., & Xiong, J. (2019). Document Similarity for Texts of Varying Lengths via Hidden Topics. *arXiv preprint arXiv:1903.10675*.
4. Liu, M., Lang, B., & Gu, Z. (2017). Calculating Semantic Similarity between Academic Articles using Topic Event and Ontology. *arXiv preprint arXiv:1711.11508*.
5. Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1-35). Springer, Boston, MA.
6. Denley, T. (2013). Degree compass: A course recommendation system. *EDUCAUSE Review Online*.
7. Reddy, M. Y. S., & Govindarajulu, P. (2018). College Recommender system using student'preferences/voting: A system development with empirical study. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY*, *18*(1), 87-98.
8. Haruna, K., Ismail, M. A., Damiasih, D., Sutopo, J., & Herawan, T. (2017). A collaborative approach for research paper recommender system. *PloS one*, *12*(10), e0184516.
9. Hasan, M., Ahmed, S., Abdullah, D. M., & Rahman, M. S. (2016, May). Graduate school recommender system: Assisting admission seekers to apply for graduate studies in appropriate graduate schools. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)* (pp. 502-507). IEEE.