Optimizing Academic Pairings in Smart Campuses: A Recommendation System for Academic Communities

Elvandi da Silva Junior[1][a], Gabriel Casanova[1][b], Daniel Youssef de Hollanda Lopes[1][c], Ana Paula Militz Dorneles[1][d], Renan Bordignon Poy[1][e], José Palazzo M. de Oliveira[2][f] and Vinícius Maran[1][g]

[1]*Laboratory of Ubiquitous, Mobile and Applied Computing (LUMAC), Polytechnic School, Federal University of Santa Maria, Av. Roraima, 1000, Santa Maria, Brazil*

[2]*Informatics Institute, Federal University of Rio Grande do Sul, Porto Alegre, Brazil*

Keywords: Academic Pairings, Hybrid Filtering, Recommendation System, Smart Campus.

Abstract: Collaborating across disciplines can advance research fields by offering new ways of addressing complex problems and fostering integration and acknowledging similarities among different fields. Pairing preferences, such as teaching the same or different content areas, grade spans, or buildings, are also crucial in mentorship programs, as they can significantly impact the effectiveness of the partnership. This necessary academic pairing can be difficult to a set of factors, as cultural, geographical or personal. This study addresses the challenge of academic pairings, emphasizing the need to publicize university resources and projects to promote interconnection between professionals from different areas within the same academic environment. The research describes the *"Unified Recommendation System"* - a recommendation system for academic communities. This is a hybrid recommendation system and was developed to recommend relevant projects of interest to the user, in addition to being easy to access through an application for students, teachers and the general community. Thus, the developed prototype demonstrated significant potential as a relevant tool in the context of smart campuses, with user interest recommendation rates of more than 81% in the evaluated scenario.

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

Smart campuses are part of a broad category of smart solutions designed to enhance human experiences in a variety of ways, such as fostering connectivity, efficiency, and personalization (Sneesl et al., 2022b), while also contributing to a more sustainable planet (Clark and Eisenberg, 2008).

This category strongly emphasizes the integration of innovative tools into education (Huang et al., 2019), striving to equip students, teachers and the campus community with access to resources that hold the potential to revolutionize the educational environment into a more digitized and contemporary setting (Yin, 2014).

One challenge that smart campuses educational tools are well-equipped to address is Academic Pairings. This involves a collaborative process where academics and practitioners coexist in a shared space, jointly generating knowledge (Stahl and Hesse, 2009). This process can also foster interprofessional collaboration (Davoli and Fine, 2004), positively impacting both the projects and the individuals involved (Chiocchio et al., 2011; Karam et al., 2018; Green and Johnson, 2015). Furthermore, these smart resources can be useful for promoting ongoing university initiatives to the academic community.

Academic communities refer to groups of individuals who are involved in academic pursuits, such as students, faculty, researchers, and administrators, and who share common goals and interests related to the pursuit, exchange, and validation of knowledge (Wakeling et al., 2019). An effective strategy involves the use of Recommendation Systems (RS), which are commonly divided into the types: content-based fil-

[a] https://orcid.org/0000-0001-8783-9578
[b] https://orcid.org/0009-0009-5420-7334
[c] https://orcid.org/0009-0006-4189-3812
[d] https://orcid.org/0000-0002-8177-3763
[e] https://orcid.org/0009-0004-7301-6609
[f] https://orcid.org/0000-0002-9166-8801
[g] https://orcid.org/0000-0003-1916-8893

Silva Junior, E., Casanova, G., Lopes, D., Paula Militz Dorneles, A., Poy, R., M. de Oliveira, J. and Maran, V.
Optimizing Academic Pairings in Smart Campuses: A Recommendation System for Academic Communities.
DOI: 10.5220/0012630200003693
In *Proceedings of the 16th International Conference on Computer Supported Education (CSEDU 2024) - Volume 1*, pages 124-134
ISBN: 978-989-758-697-2; ISSN: 2184-5026
Copyright © 2024 by Paper published under CC license (CC BY-NC-ND 4.0)

tering (CBF), collaborative filtering (CF), and hybrid filtering (HF) (Adomavicius and Tuzhilin, 2005).

These systems have the capability to suggest items to users (Aamir and Bhusry, 2015), including projects and professors. In this context, this study aims to address the problem of Academic Pairings at universities.

Thus, an intermediary–a recommendation system must be established between users and potential educational items of interest, through an application that employs a hybrid recommendation system (Maruyama et al., 2023). To adress this, we developed a recommendation system, considering the specifics of educational items related to academic pairings. We developed a prototype of this recommendation system and evaluated it in the UFSM[1] university.

The paper is structured as follows: Section 2 presents a background on the subjects of Recommender Systems and Smart Campuses; Section 3 provides an overview of related work; Section 4 describes the recommendation system built and its software architecture; Section 5 discusses the data acquisition and evaluation process in testing the recommendation system and Section 6 summarizes the conclusionss of this paper.

2 BACKGROUND

This section provides a comprehensive review of recommender systems, focusing on the most frequently employed algorithms in recommendation systems. Additionally, it explains the concept of smart campuses, describes their various types, and different application areas.

2.1 Recommender Systems

Recommender systems, which emerged in the 90s, are filtering tools designed to improve the quality of recommendations and are extensively utilized across the internet (Batmaz et al., 2018). Currently, these systems are prevalent on various online platforms, such as e-commerce, where they assist customers in locating products, and social networks,where they aid users in identifying content of interest, as well as other applications, such as banking.

These systems have become increasingly integral to users' daily lives, due to their personalized suggestions and their effectiveness surpassing that of keyword-based search engines (Bai et al., 2020). Whithin an academic context, recommender systems can prove to be exceptionally useful for a variety of reasons. They function as facilitators of learning and teaching tasks, enable students to make more appropriate and informed decisions, and assist organizations in gaining a better understanding of users (Verbert et al., 2012; Hagemann et al., 2019; Zhou et al., 2019).

Recommendation systems in a smart campus can employ various techniques to provide relevant recommendations to members of the academic community. Below, some of these techniques and some of the algorithms associated with each of them will be presented.

2.1.1 Content-Based Filtering

Content-Based Filtering utilizes detailed information about academic items and user preferences to provide relevant recommendations. "Items" in the academic context include courses, lectures, learning resources, academic articles, events, and other elements related to the academic environment. Content-Based Filtering algorithms analyze the characteristics of the items and create user profiles based on previously demonstrated preferences (Banik, 2018). Among these algorithms, we can mention:

- **TF-IDF (Term Frequency-Inverse Document Frequency).** TF-IDF is a widely used algorithm for text analysis. It assesses the importance of keywords in documents based on how frequently they appear relative to a set of documents. In the academic context, TF-IDF can be applied to recommend study materials, academic articles, and resources based on keywords and topics related to the user's interests. The algorithm ranks items based on how well their characteristics (e.g., keywords) match the user's profile (Karabiber, 2020).
- **Word Embedding.** Word embedding algorithms, such as Word2Vec and GloVe, map words to real-number vectors. These vectors represent the semantic meaning of words and, by extension, academic items. In the academic context, these vectors can be used to calculate semantic similarity between items and suggest those with similar semantic content to the user's interests (Kenyon-Dean et al., 2020).

2.1.2 Collaborative Filtering

Collaborative filtering is based on the interactions and behaviors of members of the academic community to make recommendations. It considers how students, professors, researchers, and other users interact with each other, participate in academic activities, and col-

[1] https://www.ufsm.br

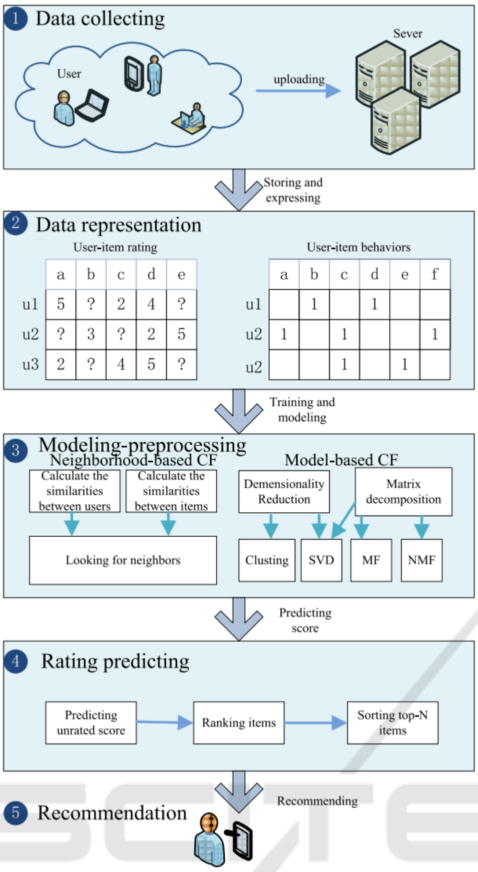

Figure 1: The framework of collaborative filtering-based RS (Chen et al., 2018).

laborate on research projects (Banik, 2018) (Figure 1).

This method can be user-based or item-based, as described below:

- **User-Based Collaborative Filtering:** In this method, user similarity is calculated based on their past behaviors and preferences. The k-Nearest Neighbors (KNN) algorithm is commonly used to find similar users. Once similar users are identified, the system recommends items based on the choices of these similar users that have not yet been evaluated by the current user (Boehmke and Greenwell, 2019).
- **Item-Based Collaborative Filtering:** In this approach, item similarity is calculated based on user preferences. The KNN algorithm is an example, and it calculates the similarity between items based on user ratings. Then, it recommends items similar to those that a user has already liked (Boehmke and Greenwell, 2019).

The rationale of user-based CF and item-based CF is shown in Figure 2.

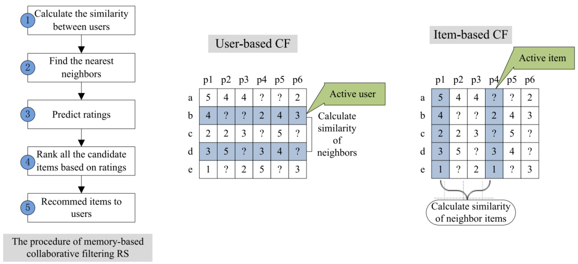

Figure 2: The rationale of user-based CF and item-based CF (Chen et al., 2018).

2.1.3 Hybrid Filtering

Hybrid filtering combines elements of content-based filtering and collaborative filtering to provide more accurate and diverse recommendations. This approach can be executed in various ways, including integrating results from both approaches or switching between them based on specific criteria. The algorithms used in hybrid filtering can include any combination of content-based and collaborative filtering algorithms, depending on the adopted strategy (Banik, 2018).

There are some techniques and models that can be employed to achieve better results in hybrid filtering. Below are some of the most common types of hybrid recommendations (Kharsa and Al Aghbari, 2023):

- **Weighted Models:** In hybrid systems, the results of content-based filtering and collaborative filtering techniques are combined using a weighted model. This model assigns weights to recommendations generated by each technique based on user context and preferences. The weighted combination of results helps provide balanced and personalized recommendations.
- **Technique Switching:** In this approach, the system alternates between content-based filtering and collaborative filtering based on user needs and contextual characteristics. For example, it may use collaborative filtering to recommend study groups based on users' past interactions and then apply content-based filtering to suggest study materials related to the topics discussed in the groups.
- **Recommendation-Level Machine Learning:** Machine learning algorithms, such as neural networks, can be used to automatically learn how to combine the results of different filtering techniques in a hybrid recommendation model. This enables dynamic adaptation to user preferences and interaction patterns, resulting in highly personalized recommendations (Kharsa and Al Aghbari, 2023).

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Considering that the proposal of this work is categorized as Recommendation-Level Machine Learning, in the next sections we will present important concepts that were used in developing the proposal: Cosine Similarity and the KNN algorithm.

2.1.4 Cosine Similarity

Cosine similarity plays an important role in collaborative filtering and content-based recommendation, helping to identify similar users or items to create personalized recommendations (Chen et al., 2018). It can be applied both to measure similarity between users and between items in recommendation systems:

- **Cosine Similarity for User:** In this case, each user is represented as a vector of item ratings. Cosine similarity is calculated between the rating vectors of two users to measure how similar their likes and preferences are. High cosine similarity between two users indicates that they have similar preferences, which can be used to make recommendations based on the interactions of similar users.
- **Cosine Similarity for Item:** In this case, each item is represented as a vector of user ratings. Cosine similarity is calculated between the rating vectors of two items to measure how similar the items are in terms of user preferences. High cosine similarity between two items indicates that they are similar in terms of how they are rated by users, which can be used to suggest similar items when a user interacts with a specific item.

2.1.5 KNN Algorithm

The k-Nearest Neighbors (KNN) algorithm is a supervised machine learning method used for classification and regression (Khoa et al., 2013). It operates based on the idea that similar objects tend to be close to each other in a feature space (Zhang et al., 2018). KNN is a simple and intuitive approach (Boehmke and Greenwell, 2019):

- **Training.** During the training phase, the algorithm stores all examples from the training dataset in a multi-dimensional space, where each example is represented by a feature vector. Each example is also associated with a class or target value.
- **Classification/Regression.** When making predictions for a new example, KNN searches for the k nearest examples in the feature space. Proximity is typically measured using a metric like Euclidean distance. The k nearest examples either vote for the class of the example or contribute to calculating a weighted average for regression.
- **Choosing k.** The choice of the k value is crucial in KNN. A small k value (e.g., 1) makes the algorithm sensitive to noise and outliers, while a large k value smoothens the decision boundary, making it less sensitive to local data variations.
- **Weight Function.** To address different weights for nearest neighbors, you can assign weights based on distance. Closer neighbors may have larger weights than farther ones.
- **Classification vs. Regression.** KNN can be used for both classification and regression tasks. In classification, the example is assigned to the most frequent class among the k nearest neighbors. In regression, a weighted average of the target values of the k nearest neighbors is calculated.
- **Performance.** KNN's performance can be influenced by the appropriate choice of distance metric, data preprocessing, and selecting an optimal k value. It's also important to have a representative training dataset.

2.1.6 Cold Start

The Cold start in recommendation systems refers to situations in which the system needs to make recommendations without relevant historical data. This challenge is particularly common when dealing with new users or items and requires the application of creative strategies to provide a useful experience to users, even when information is limited.

User-related cold start occurs when a new user registers in the recommendation system and hasn't had significant interactions yet. Given that the system lacks information about this user's preferences, making personalized recommendations becomes a challenge. On the other hand, item-related cold start happens when a new item is introduced into the recommendation system and lacks a history of interactions with users. In this scenario, the system has no data on how users have reacted to that specific item (Joy et al., 2021).

2.2 Smart Campuses

In the realm of technological advancements, a concept that has gained significant prominence is that of a Smart Campus. It can be characterized as an entity that strategically employs technology and infrastructure with the main goal of improving its processes, thus improving their utility for individuals (Sánchez-Torres et al., 2018).

The realization of this concept can be achieved through the deployment of a diverse array of tools. These include IoT (Internet of Things) technology, sensors, cloud computing, user interfaces, blockchain and a variaty of other applications (Gilman et al., 2020; Fernández-Caramés and Fraga-Lamas, 2019). Each of these components can facilitate the creation and implementation of Smart Campus, contributing unique benefits to the system.

Some of the benefits include the enhanced quality of life for inhabitants of these environments; enhanced user experience for students, staff, and estate managers, optimized space utilization, improved management of resources such as computer labs and air conditioners and energy conservation, among others (Villegas-Ch. et al., 2019; Sutjarittham et al., 2018; Wang, 2014).

In spite of these benefits, smart campuses also face significant challenges. These include limitations of perspectives that rely entirely on data, the integration of individuals' practices and the importance of considering ethics and domestication (Bates and Friday, 2017). Additional challenges involve the absence of a technology adoption model, perceived fees, perceived trust and perceived value (Sneesl et al., 2022a).

3 RELATED WORK

This section provides an overview of related research in recommender systems and smart campuses, highlighting recommended item types and filtering techniques employed in each work.

(Ibrahim et al., 2019) introduced a framework targeting the academic domain, recommending various courses for students. Their hybrid system combines collaborative and content-based filtering with ontology for information extraction.

(Bai et al., 2019) explored a collaborative recommendation system for researchers with similar interests. The model incorporates collaborative filtering, content-based filtering, and social network-based filtering, forming a hybrid approach.

(Mrhar and Abik, 2019) proposed a recommendation system for online course platforms, offering personalized suggestions based on user profiles. The author utilizes content-based filtering and deep learning to enhance algorithm accuracy.

(Xiao et al., 2018) presents a personalized recommendation system that tailors suggestions to individual users based on their interests and learning history. The system employs both content-based and collaborative filtering techniques to recommend didactic materials and essential learning resources.

This work aims to develop a platform that recommends a wide variety of items based on previous studies. Table 1 summarizes the types of recommended resources and the methods used for processing data.

The proposed system adapts precisely to individual user preferences. It offers a personalized and profile-based recommendation experience that accurately aligns with user-defined preferences during item acquisition. This is achieved through the integration of advanced algorithms and recommendation elements, named as *"Unified Recommendation System"*.

4 A RECOMMENDATION SYSTEM FOR ACADEMIC COMMUNITIES

The *"Unified Recommendation System"* serves as a software architecture for smart campuses, offering a variety of recommender systems and techniques tailored to different item types.

4.1 Data Aquisition

Obtaining data is a crucial step in optimizing recommendation systems, providing the essential foundation for filtering algorithms to analyze a wide range of information. In the world of recommendation systems, data is the lifeblood that yields valuable insights into user preferences, behaviors, and trends. This process entails not only the collection of diverse data but also ensuring that recommendation algorithms have a robust basis for generating accurate and pertinent suggestions.

A Python script was used to extract data from the UFSM projects and professors system, throught a web scrapping process. In this way, a total of 56865 items were obtained, divided between projects, professors and events.

From the data obtained, it was necessary to to analyze the collected resources, separating them into different categories and topics, so that the filtering algorithm could obtain greater precision in the search and recommendation. To accomplish this task, we used the ChatGPT 3.5 Large Language Model (LMM)(OpenAI, 2023) to categorize items based on a series of frequently used generic categories. This step resulted in 9540 categories and subcategories.

4.2 Software Architecture Definition

The software architecture to support recommendation processes is segmented into three main modules, which is as shown in Figure 3: The User Access Module (UAM), the Recommendation Management mod-

Table 1: Studies and Characteristics of Recommendation Systems.

Work	Type of Recommended Item	Filtering Strategy
(Ibrahim et al., 2019)	Undergraduate and post-graduate courses	Ontology-based filtering, Content-based filtering, Collaborative filtering
(Bai et al., 2019)	Collaborating researchers	Collaborative filtering, Content-based filtering, Social network-based filtering
(Mrhar and Abik, 2019)	Online courses	Content-based filtering, Deep Learning
(Xiao et al., 2018)	Didactic materials for learning	Collaborative filtering, Content-based filtering
(Hoang et al., 2022)	Courses in specialized fields	Ontology-based filtering, Content-based filtering, Collaborative filtering
This research	Educational resources (courses, mini-courses, video lessons, teaching materials, e-books, lectures, events, scientific articles, thesis, similar user profiles, other educational platforms)	Collaborative filtering, Content-based filtering

ule (RMM), and the Administrative and Persistence Module (APM).

The modules of the software architecture are presented below:

4.2.1 User Access Module (UAM)

In the user access module, users engage with the material provided through recommendations, by using their personal devices such as cell phones, computers, and alternative devices. This section allows users to interact with resources, direct themselves to other university portals, and communicate their interests.

The module is composed by three submodules: (i) User Interface Module, responsible for presenting recommendations appropriately to the user, (ii) Control Module: Responsible for capturing user usage information and sending usage statistics to the recommendation and administration modules, and (iii) Persistence and Communication module, responsible for storing information collected in the recommendation process and make requests to other modules of the architecture (using the REST standard).

UAM was prototyped using the Ionic framework (Yusuf, 2016), which allows the generation of web and mobile applications.

4.2.2 Administrative and Persistence Module (APM)

The Administrative and Persistence Module is the responsible for the communication with university external systems. It is also responsible to manage and store user profile and navigation information.

The module is composed by four submodules: (i) Communication module, responsible for make requests to other modules of the architecture (using the REST standard), (ii) Persistence module, responsible for the storage of the data collected in the recommendation process, and to retrieve items to present to users in a recommendation, (iii) User Navigation Module, responsible to store and retrieve user data given by the UAM module, and (iv) User Profile Processig Module, responsible for the management of user profile data and his/her topics of interest.

APM was prototyped using Java SpringBoot framework (Yusuf, 2016) using JDBC driver to the communication with PostgreSQL database. APM stores resource descriptions, user interests, interacted items, and recommendation history. All recommendations are saved in this section and, when requested, sent to the recommendation management module to initiate the filtering and recommendation process.

4.2.3 Recommendation Management Module (RMM)

The recommendation management module contains all the code blocks and functions of the recommendation system. Here, new codes are added, edited and corrected, building the core of the personalized recommendation process for each user. This environment processes user information, sends requests of the database, and operates as an indirect approach, reflecting the user's actions and interactions.

One of the key strengths of the platform's recommendation system lies in its utilization of a hybrid approach, seamlessly blending both content-based and collaborative filtering techniques.

By incorporating these two distinct filtering methods, the platform optimizes the diversity and freshness of recommendations. Content-based filtering allows the system to understand the intrinsic character-

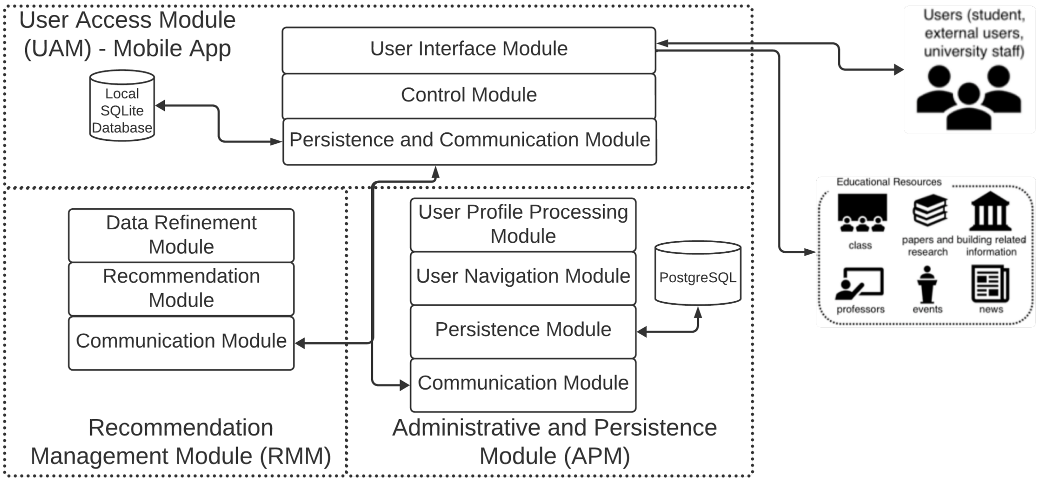

Figure 3: *"Unified Recommendation System"* software architecture definition.

istics of educational resources and match them with the user's preferences. On the other hand, collaborative filtering leverages collective user behavior to suggest resources that align with the preferences of users with similar tastes. Changing between these filters guarantees that whenever a user asks for recommendations, they receive new suggestions.

The algorithm encompassing Content-Based Filtering adheres to a sequence of operations for generating recommendations. The first step begins by cataloging the user's interests into tables, which are subsequently associated with the user's topics. Following this, educational resources that align with these topics are then listed based on the the user-topic relationship.

Each resource undergoes a process known as 'bag-of-words', which isolates keywords, quantifies their occurrences, and compiles a list accordingly. An analogous process is conducted for user-favorite items. Cosine similarity is then employed to calculate the similarity between text features. Ultimately, the features bearing the highest similarity are recommended to the user and and can be seen later displayed on the screen.

Having users previously declared their interests, the algorithm employs Collaborative Filtering by seeking out other users with comparable interests, creating a list for each. The interests of both users are combined, with a numerical increment for each shared interest. The algorithm then selects the recommended resources for the users with the highest similarity and compiles them into a list. This list is then shuffled and ultimately presented to the user.

RMM was prototyped in Python language, with Scikit (Kramer and Kramer, 2016) and Surprise (Hug, 2020) libraries.

5 EVALUATION

5.1 Evaluation Scenario

In order to assess the accuracy of resource recommendations, a profile representing a student from the Polytechnic School of UFSM who has a broad interest in the field of technology has been created, which is as shown in Figure 4. This profile was created with the intention of delving deeper into the analysis of the effectiveness of recommendations provided to users, considering their preferences and affinities with the technology field as a whole.

In this situation, topics of interest such as Technology, Applied Technology, Assistive Technology, Automotive Technology, Banking Technology, BIM Technology, and Bluetooth Technology (as shown in Figure 5a) were selected. This approach enabled the system to search for relevant data and information related to these specific subjects and to identify other users who shared the same interests. Thus, a dynamic platform was created that promotes interconnection among individuals with diverse technology-related affinities, enriching the user experience and fostering knowledge exchange and collaboration.

The evaluation of each recommended resource occurs by sliding it to the side of the screen, which is as shown in Figure 5b, where the left represents disinterest and the right signifies that the user wants more recommendations of that type. This simple and intuitive interaction provides users with effective control over the suggested content, allowing them to further customize their experience according to their preferences. As users interact with the system, they provide

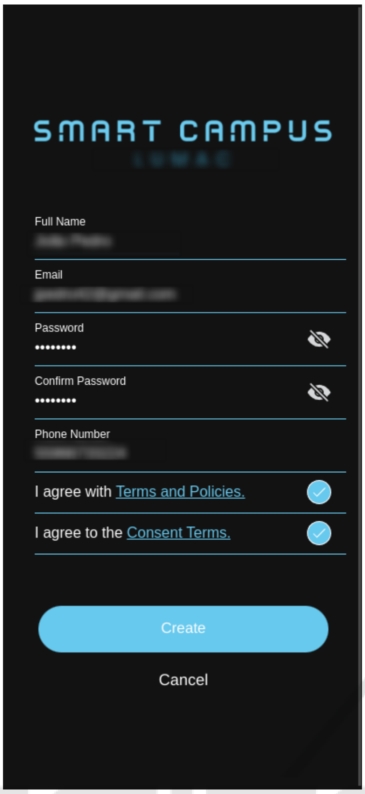

Figure 4: New user registration screen.

valuable feedback that continually enhances the accuracy of recommendations, making the learning and discovery environment even more tailored to their individual needs.

5.2 Results and Discussion

In the proposed test scenario, the recommendation system returned a sample consisting of 58 recommended items in about 2 seconds, as shown in Figure 6. Among these items, we noticed that 47 of them piqued the user's interest, while 11 did not generate interest, resulting in an accuracy rate of 81% in recommendations related to this specific profile.

The accuracy of the algorithm tends to improve as users interact more with the system and accumulate a more substantial data history. This enhancement becomes even more noticeable as we add more users and data to our database. This growth results in an increase in the number of users with similar interests, which, in turn, leads to more refined recommendations aligned with each user's individual preferences.

The recommendation process, based on user preferences and interests, has proven to be an effective tool for presenting projects aligned with their selected topics of interest. The recommendations have highlighted projects directly related to specific areas of interest, as well as professors in those fields or who were directly involved in a project.

The presence of projects that piqued the interest of users with similar profiles added an additional layer of relevance, suggesting that the recommendations not only met individual criteria but also reflected common trends and preferences within the community of users with similar interests. Furthermore, the identification of professors with expertise and involvement in the areas of interest enriched the recommendations by providing a direct connection to the academic and practical knowledge of these professionals.

However, it is important to note that projects with similar keywords were recommended, even if they did not align with the user's specific interest. This occurred, for example, when projects related to Language Technologies or Methodologies for Social Sciences were suggested due to the broad search for "Technology," even if the user had not explicitly selected these categories.

The similarity in keywords between the recommended resources and the selected categories can also lead to confusion during the evaluation process. Misinterpretation of an item due to shared project titles or similar elements can result in inaccurate assessments. This inaccuracy in evaluation can, in turn, lead to errors in recommendations, underscoring the importance of enhancing recommendation accuracy to avoid misunderstandings and optimize the user experience.

As users continue to use the system and provide more feedback, the algorithm becomes more proficient in understanding and anticipating their needs and preferences. This process initiates an ongoing cycle of improvement, making the system progressively more effective in suggesting relevant and pertinent content.

Furthermore, as our user community grows, and more people share similar opinions and interests, the algorithm also benefits from collective wisdom. This further contributes to the increased accuracy of recommendations. Therefore, the continuous enhancement of the algorithm's accuracy is a dynamic process that benefits all users, enriching their experience and making it more personalized.

6 CONCLUSION

The implementation of innovative technologies has brought significant modernization to the field of e-learning, especially in smart academic environments. This has resulted in the creation of a new scenario

(a). Interface to select topics of interest by the user (in Portuguese).

(b). Recommendation evaluation screen – the user can swipe to right or left, see more details about the item or mark it as favorite.

Figure 5: Example of interfaces provided by the mobile application.

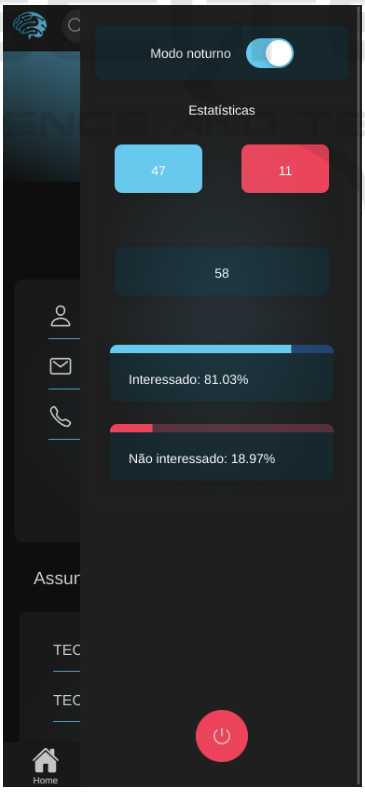

Figure 6: Interest statistics screen.

where communication among various mobile devices plays a fundamental role.

This study highlights the effectiveness of a recommendation system that is based on individual user interests to address the challenges that teachers, students, and the academic community as a whole face in finding compatible academic peers and facilitating connections within the academic community. However, it also recognizes the importance of improving the accuracy of recommendations in order to avoid potential inaccuracies, enhance the user experience, and increase the efficiency of the proposed solution.

One innovative feature of this study is its approach to connecting the academic community with projects aligned with their interests, revealing opportunities that might otherwise have gone unnoticed. The most notable aspect is that the recommendation process uses multiple techniques to recommend items to academic communities.

However, it is important to note that this study has limitations, such as the evaluation process being conducted at a single university, the absence of testing with a large number of users, and the lack of longitudinal data to track changes in user interests over time. Possible areas for future research include expanding testing to other universities, enhancing system accuracy, incorporating Deep Learning and Neural Network techniques, conducting performance tests on different hardware and platforms, adding georeferencing features for points of interest in the application, and utilizing ontologies for even more personalized recommendations.

ACKNOWLEDGEMENTS

This research is supported by CNPq/MCTI/FNDCT n. 18/2021 grant n. 405973/2021-7 and CNPq/MCTI Nº 10/2023 - UNIVERSAL grant n. 402086/2023-6. The research by José Palazzo M. de Oliveira is partially supported by CNPq grant 306695/2022-7 PQ-SR. The research by Vinícius Maran is partially supported by CNPq grant 306356/2020-1 DT-2, CNPq PIBIC and PIBITI program and FAPERGS PROBIC program.

REFERENCES

Aamir, M. and Bhusry, M. (2015). Recommendation system: State of the art approach. *International Journal of Computer Applications*, 120:25–32.

Adomavicius, G. and Tuzhilin, A. (2005). 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*, 17:734–749.

Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., and Xia, F. (2019). Scientific paper recommendation: A survey. *IEEE Access*, 7:9324–9339.

Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., and Xia, F. (2020). Scientific paper recommendation: A survey. *IEEE Access*, 7:9324–9339.

Banik, R. (2018). *Hands-On Recommendation Systems with Python: Start building powerful and personalized, recommendation engines with Pytbhon*. Packt Publishing.

Bates, O. and Friday, A. (2017). Beyond data in the smart city: Repurposing existing campus iot. *IEEE Pervasive Computing*, 16:54–60.

Batmaz, Z., Yurekli, A., Bilge, A., and Kaleli, C. (2018). A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, 52:1–37.

Boehmke, B. and Greenwell, B. (2019). *Hands-On Machine Learning with R*. Chapman & Hall/CRC The R Series. CRC Press.

Chen, R., Hua, Q., Chang, Y.-S., Wang, B., Zhang, L., and Kong, X. (2018). A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access*, 6:64301–64320.

Chiocchio, F., Forgues, D., Paradis, D., and Iordanova, I. (2011). Teamwork in integrated design projects: Understanding the effects of trust, conflict, and collaboration on performance. *Project Management Journal*, 42:78 – 91.

Clark, W. and Eisenberg, L. (2008). Agile sustainable communities: On-site renewable energy generation. *Utilities Policy*, 16:262–274.

Davoli, G. W. and Fine, L.-J. (2004). Stacking the deck for success in interprofessional collaboration. *Health Promotion Practice*, 5:266 – 270.

Fernández-Caramés, T. and Fraga-Lamas, P. (2019). Towards next generation teaching, learning, and context-aware applications for higher education: A review on blockchain, iot, fog and edge computing enabled smart campuses and universities. *Applied Sciences*.

Gilman, E., Tamminen, S., Yasmin, R., Ristimella, E., Peltonen, E., Harju, M., Lovén, L., Riekki, J., and Pirttikangas, S. (2020). Internet of things for smart spaces: A university campus case study. *Sensors (Basel, Switzerland)*, 20.

Green, B. and Johnson, C. D. (2015). Interprofessional collaboration in research, education, and clinical practice: working together for a better future. *The Journal of chiropractic education*, 29 1:1–10.

Hagemann, N., O'Mahony, M. P., and Smyth, B. (2019). Visualising module dependencies in academic recommendations. *Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion*.

Hoang, V. N., Le Thanh, P., My, L. O. T., Vinh, L. C., and Xuan, V. T. (2022). Towards a novel architecture of smart campuses based on spatial data infrastructure and distributed ontology. In *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 3*, pages 662–673. Springer.

Huang, L.-S., Su, J., and Pao, T. (2019). A context aware smart classroom architecture for smart campuses. *Applied Sciences*.

Hug, N. (2020). Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5(52):2174.

Ibrahim, M. E., Yang, Y., Ndzi, D. L., Yang, G., and Al-Maliki, M. (2019). Ontology-based personalized course recommendation framework. *IEEE Access*, 7:5180–5199.

Joy, J., Raj, N. S., and Renumol, V. G. (2021). Ontology-based e-learning content recommender system for addressing the pure cold-start problem. *Journal of Data and Information Quality*, 13.

Karabiber, F. (2020). Tf-idf—term frequency-inverse document frequency.

Karam, M., Brault, I., Durme, T. V., and Macq, J. (2018). Comparing interprofessional and interorganizational collaboration in healthcare: A systematic review of the qualitative research. *International journal of nursing studies*, 79:70–83.

Kenyon-Dean, K., Newell, E., and Cheung, J. C. K. (2020). Deconstructing word embedding algorithms. *arXiv preprint arXiv:2011.07013*.

Kharsa, R. and Al Aghbari, Z. (2023). Leveraging association rules in feature selection for deep learning classification. *SN Computer Science*, 5(1):112.

Khoa, N. M., Viet, D. T., and Hieu, N. H. (2013). Classification of power quality disturbances using wavelet transform and k-nearest neighbor classifier. In *2013 IEEE International Symposium on Industrial Electronics*, pages 1–4.

Kramer, O. and Kramer, O. (2016). Scikit-learn. *Machine learning for evolution strategies*, pages 45–53.

Maruyama, M. H. M., Silveira, L. W., de Oliveira, J. P. M., Gasparini, I., and Maran, V. (2023). Hybrid recommender system for educational resources to the smart university campus domain.

Mrhar, K. and Abik, M. (2019). Toward a deep recommender system for moocs platforms. *Proceedings of the 3rd International Conference on Advances in Artificial Intelligence*.

OpenAI (2023). Chatgpt 3.5. https://chat.openai.com. Accessed: 2023-12-20.

Sánchez-Torres, B., Rodríguez-Rodríguez, J. A., Rico-Bautista, D. W., and Guerrero, C. D. (2018). Smart campus: Trends in cybersecurity and future development. *Revista Facultad de Ingeniería*, 27(47):104–112.

Sneesl, R., Jusoh, Y. Y., Jabar, M. A., and Abdullah, S. (2022a). Conceptualizing iot-based smart campus adoption model for higher education institutions: A systematic literature review. *2022 Applied Informatics International Conference (AiIC)*, pages 7–12.

Sneesl, R., Jusoh, Y. Y., Jabar, M. A., and Abdullah, S. (2022b). Revising technology adoption factors for iot-based smart campuses: A systematic review. *Sustainability*.

Stahl, G. and Hesse, F. (2009). Paradigms of shared knowledge. *International Journal of Computer-Supported Collaborative Learning*, 4:365–369.

Sutjarittham, T., Gharakheili, H., Kanhere, S., and Sivaraman, V. (2018). Realizing a smart university campus: Vision, architecture, and implementation. *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6.

Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnić, I., and Duval, E. (2012). Context-aware recommender systems for learning: A survey and future challenges. *IEEE Transactions on Learning Technologies*, 5:318–335.

Villegas-Ch., W., Molina-Enriquez, J., Chicaiza-Tamayo, C., Ortiz-Garcés, I., and Luján-Mora, S. (2019). Application of a big data framework for data monitoring on a smart campus. *Sustainability*.

Wakeling, S., Spezi, V., Fry, J., Creaser, C., Pinfield, S., and Willett, P. (2019). Academic communities: The role of journals and open-access mega-journals in scholarly communication. *Journal of Documentation*, 75(1):120–139.

Wang, H.-I. (2014). Constructing the green campus within the internet of things architecture. *International Journal of Distributed Sensor Networks*, 10.

Xiao, J., Wang, M., Jiang, B., and Li, J. (2018). A personalized recommendation system with combinational algorithm for online learning. *Journal of ambient intelligence and humanized computing*, 9:667–677.

Yin, F. (2014). Construction and application of wisdom campus. *Computer Programming Skills & Maintenance*.

Yusuf, S. (2016). *Ionic framework by example*. Packt Publishing Ltd.

Zhang, S., Li, X., Zong, M., Zhu, X., and Wang, R. (2018). Efficient knn classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1774–1785.

Zhou, X., Zhang, J., and Zhang, Y. (2019). The 1st international workshop on context-aware recommendation systems with big data analytics (cars-bda). *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS