
Research on Deep Learning-Based Algorithm and Model for Personalized Recommendation of Resources

Yu'e Liu

Yunnan University Dianchi College, Kunming, Yunnan, China, 650228
liuyue@ynudcc.cn*
* corresponding author

(Received January 11, 2023 Revised February 20, 2023 Accepted February 25, 2023, Available online March 1, 2023)

Abstract

Resource recommendation systems have become increasingly important in modern resource management as they can provide high-quality services to customers and achieve accurate marketing. These systems use personalized information to solve business needs such as customer consultation and product recommendation. This paper focuses on the algorithm and model of resource personalized recommendation based on deep learning and uses human resource recommendation as an example. The paper discusses the importance of resource recommendation systems in modern resource management and the role they play in providing high-quality services and achieving accurate marketing. It then provides an overview of deep learning-based algorithms and models used for resource personalized recommendation. The paper examines the challenges involved in developing personalized recommendation systems and highlights the benefits of deep learning-based algorithms. The study explores the process of building a personalized recommendation system for human resources using deep learning-based algorithms. The paper presents the steps involved in collecting and preprocessing data, designing and training the model, and evaluating the performance of the recommendation system. It provides a detailed explanation of the model architecture and the optimization techniques used to improve the accuracy of the recommendation system. In conclusion, this paper presents a study of the algorithm and model of resource personalized recommendation based on deep learning, with human resource recommendation as an example. The study shows the potential of deep learning-based algorithms in building effective recommendation systems and highlights the importance of personalized recommendations in modern resource management. The paper also provides insights into the challenges and benefits of developing personalized recommendation systems and presents a framework for building such systems.

Keywords: HR Recommendation, Deep Learning, Sdae, Data Sparsity

1. Introduction

The rapid development of information technology has brought about significant changes in the way people access and consume information. Online information has become an essential part of people's daily lives, and universities need to keep up with this trend. It is the responsibility of the university to understand and analyze the news and hot issues for the first time after the occurrence of relevant online public opinion. Therefore, there is a need to establish a data mining-based online public opinion monitoring system for universities [1,2]. Online public opinion monitoring is essential for universities to keep up-to-date with the latest developments in their field and respond to them in a timely manner. With the help of a data mining-based online public opinion monitoring system, universities can track the latest news and trends related to their field, analyze the public sentiment towards them, and respond accordingly.

Personalized recommendation systems have become increasingly popular and necessary as the amount of online information continues to grow rapidly. The recommendation of resources based on user preferences and interests has been one of the most critical issues in personalized recommendation systems. Deep learning has emerged as a promising technique for developing personalized recommendation systems because it can learn from large-scale and heterogeneous data with high accuracy. Researchers have been developing various deep learning-based algorithms and models to provide users with personalized recommendations of resources such as movies, music, books, and products. These models utilize deep neural networks to extract and learn the underlying features of the resources and users and to predict users' preferences for the resources. The research on deep learning-based algorithms and models

for personalized recommendation has the potential to significantly improve the accuracy and effectiveness of recommendation systems and enhance the user experience.

The goal of the research on deep learning-based algorithms and models for personalized recommendation of resources is to design and develop effective and efficient personalized recommendation systems that can adapt to users' changing preferences and interests. The research focuses on developing novel deep learning-based algorithms and models that can extract and learn rich features from user data and resource data to improve recommendation accuracy. The research also investigates how to integrate contextual information, such as time and location, into the recommendation process to provide more relevant and timely recommendations. The research on deep learning-based algorithms and models for personalized recommendation has the potential to contribute to various application areas, such as e-commerce, social media, and entertainment, and to benefit users by providing them with more personalized and satisfactory experiences.

The data mining-based online public opinion monitoring system can be used to collect and analyze large volumes of data from various sources such as social media, blogs, and news articles. The system can use advanced data mining techniques to extract relevant information and provide insights into the public sentiment towards a particular topic. The online public opinion monitoring system can also help universities to identify potential issues and crises and respond proactively [4-6]. For example, if negative sentiment towards a particular issue is detected, the university can take appropriate measures to address the issue before it becomes a major problem.

In conclusion, the establishment of a data mining-based online public opinion monitoring system is necessary for universities to keep up with the latest developments and respond to them in a timely manner. The system can help universities to track the latest news and trends related to their field, analyze the public sentiment towards them, and respond accordingly. It can also help universities to identify potential issues and crises and respond proactively. With the help of an online public opinion monitoring system, universities can stay ahead of the curve and continue to provide high-quality education and research.

2. Relevant technology and theoretical analysis

2.1. The basic workflow of recommendation system

Recommendation systems have become ubiquitous in our lives, with companies like Amazon and Netflix relying on them to suggest products and content [5]. The basic workflow of a recommendation system consists of several stages, including data collection, data preprocessing, model training, and evaluation. The first stage in the workflow is data collection. The recommendation system needs data about users, items, and interactions between users and items. This data can come from a variety of sources, such as user behavior on a website, purchase history, or explicit ratings given by users. Once the data is collected, it needs to be preprocessed. This involves cleaning and transforming the data into a format that can be used for training the recommendation model. This stage may also involve feature engineering, where relevant features are extracted from the data to improve the accuracy of the model.

The next stage is model training, where the recommendation system learns to make predictions based on the data. The most common approach is collaborative filtering, which relies on finding similarities between users and items to make recommendations [6-9]. Other approaches include content-based filtering, which uses features of the items to make recommendations, and hybrid approaches that combine collaborative and content-based filtering. After the model is trained, it needs to be evaluated to ensure that it is performing well [10,11]. This involves testing the model on a held-out dataset to see how well it can predict user-item interactions. Metrics such as precision, recall, and F1 score can be used to evaluate the model's performance. Finally, the recommendation system can be deployed and used to make recommendations for users. The system can provide personalized recommendations based on a user's past behavior, such as items they have purchased or rated, or items that are similar to those they have shown interest in.

In conclusion, the basic workflow of a recommendation system consists of data collection, data preprocessing, model training, evaluation, and deployment. Each stage plays an important role in the accuracy and effectiveness of the

system. With the increasing availability of data and advancements in machine learning techniques, recommendation systems will continue to play an important role in helping users discover new products and content. The goal of the recommendation system is to filter out the objects that may be of interest to the user from a large number of objects to be recommended, and the main research objects are the system user, the items to be recommended and the recommendation algorithm [1]. The basic workflow of the recommendation system is shown in Figure 1 below.

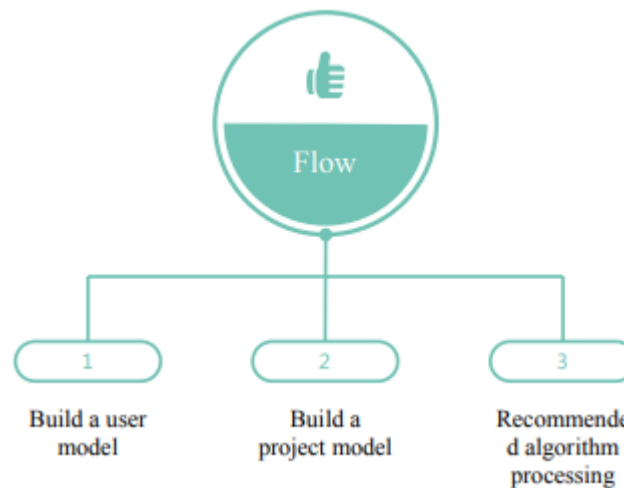


Figure. 1. Basic workflow

2.2. Introduction of related algorithms

There are several related algorithms used in computer science and engineering to solve complex problems. These algorithms are often designed to optimize a particular metric or solve a specific problem [12-15]. Some of the most commonly used algorithms include sorting algorithms, search algorithms, and machine learning algorithms. Sorting algorithms are used to sort a collection of data elements in a particular order. The most commonly used sorting algorithms include bubble sort, quicksort, and merge sort. These algorithms are used in a variety of applications, including database management, data analysis, and programming.

Search algorithms are used to find a particular element or value in a collection of data. The most commonly used search algorithms include binary search, linear search, and depth-first search [16,17]. These algorithms are used in a variety of applications, including text processing, image processing, and data analysis. Machine learning algorithms are used to build predictive models based on historical data. These algorithms use statistical techniques to identify patterns in data and use these patterns to make predictions about future data [18]. Some of the most commonly used machine learning algorithms include decision trees, support vector machines, and neural networks [19]. These algorithms are used in a variety of applications, including speech recognition, image classification, and fraud detection. Other related algorithms include optimization algorithms, graph algorithms, and cryptographic algorithms. Optimization algorithms are used to find the optimal solution to a problem, while graph algorithms are used to analyze and manipulate graphs. Cryptographic algorithms are used to secure data and communications, and include encryption and decryption algorithms.

In conclusion, there are several related algorithms used in computer science and engineering to solve complex problems. These algorithms are designed to optimize a particular metric or solve a specific problem. Some of the most commonly used algorithms include sorting algorithms, search algorithms, and machine learning algorithms. Other related algorithms include optimization algorithms, graph algorithms, and cryptographic algorithms. Each algorithm has its own unique characteristics and applications, making them an essential tool for solving complex problems in various fields.

The first one is the UBCF algorithm [2]. The UBCF algorithm has two main tasks, the first step is to query the nearest neighbors, to find the nearest neighbors mainly by measuring the similarity between users, the higher the

similarity, the closer the distance between users. The similarity between user a and user b is denoted as $sim(a,b)$. The similarity between users can be expressed as the similarity between two vectors. The common similarity calculation methods include cosine similarity, Pearson similarity and modified cosine similarity. The second step is to generate the corresponding recommendation list. Let the set of nearest neighbors of user a be NNa , then the predicted rating $P_{a,i}$ of user a for item i can be obtained by using the known ratings in the set NNa , which is calculated as follows.

$$P_{a,i} = R_a + \frac{\sum_{b \in NNa} sim(a,b) * (R_{a,i} - R_{b,i})}{\sum_{b \in NNa} |sim(a,b)|} \tag{1}$$

Where $sim(a,b)$ represents the similarity between user a and user b, $R_{a,i}$ and $R_{b,i}$ represent the average ratings of user a and b for all items, respectively, and $R_{b,i}$ represents the rating of user b for item i. The above method predicts the user's ratings of all unrated items, and then recommends the top N items with the highest ratings to the user, which is called top-N recommendation.

Second, the IBCF algorithm [3,20]. The basic idea of this algorithm is to find the set of nearest neighbors of a target item, and since the user's ratings of these nearest neighbors are close to the ratings of the target item, the user's predicted ratings of the target item can be obtained based on the ratings of these nearest neighbors. Based on this principle, the ratings of the remaining unrated items can be found, and then the top N items with the highest ratings are selected and recommended to the users.

Third, the MBCF algorithm [4,21]. This algorithm mainly uses existing methods in the fields of machine learning, data mining and information retrieval to process and model user data offline to make predictions. The commonly used models are plain Bayesian classification model, linear regression model, implicit semantic model and probabilistic correlation model, among which Latent Factor Model, LFM is the most popular collaborative filtering model in the field of recommender systems in recent years, which has several essentially similar variants, such as Latent Dirichlet Allocation, probabilistic Latent Semantic Analysis, Probabilistic Matrix Factorization, PMF, all of which can be used in personalized recommender systems. The implicit semantic model calculates the rating of item i by user a by the following equation.

$$Rating(a, b) = R_{a,i} = U_a T_{vi} \tag{2}$$

In addition, the evaluation metrics of recommendation algorithms are shown in Figure 2 below.

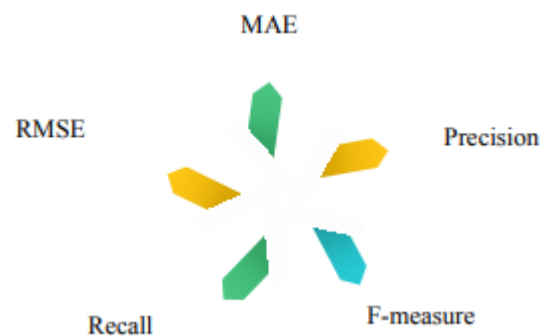


Figure. 2. Evaluation metrics of recommendation algorithms

3. Deep learning-based HR recommendation algorithm design

3.1. General requirements of the algorithm

Algorithms are a set of instructions or procedures that solve a particular problem or perform a specific task. Algorithms have become essential in various fields, from computer science to engineering and mathematics [22]. However, to ensure that algorithms are effective and efficient, they must meet certain general requirements. One of the most important requirements of an algorithm is correctness [23]. The algorithm must be able to solve the problem

or perform the task it was designed for without errors or inaccuracies. To achieve correctness, the algorithm must be well-designed, well-implemented, and well-tested. Another important requirement is efficiency. The algorithm must use minimal resources such as time and memory to solve the problem or perform the task. The efficiency of an algorithm is usually measured by its time and space complexity, which determine the amount of time and memory required for the algorithm to execute.

Usability is also a crucial requirement for an algorithm. The algorithm must be easy to use and understand by its intended users. Usability includes factors such as ease of use, user-friendliness, and the ability to handle different inputs and situations [23-25]. Another requirement is scalability, which refers to the ability of the algorithm to handle large-scale problems or tasks. As the size of the problem or task increases, the algorithm must be able to handle it without compromising efficiency or accuracy. Finally, the algorithm must be reliable and robust. The algorithm must be able to perform consistently under different conditions and inputs, and handle errors and exceptions gracefully. Robustness includes the ability of the algorithm to recover from errors and exceptions and continue executing the task or problem. In conclusion, algorithms play a vital role in solving problems and performing tasks in various fields. However, to ensure that algorithms are effective and efficient, they must meet certain general requirements such as correctness, efficiency, usability, scalability, and reliability. Meeting these requirements is crucial to ensure that algorithms can be used effectively in different applications and situations.

In the HR recommendation system, there are mainly two types of data: one is the data related to candidates, including their basic personal information, education experience, skills and specialties, work experience, the type of positions they are interested in, their treatment requirements, and the information they give in the process of using the recommendation system. This category is collectively referred to as user data. The other category is job-related data, including job types, information of recruiting companies, job content, salary, benefits, basic requirements for applicants, etc. This category is collectively referred to as project data [5-6].

In a realistic job site, the system will first ask the candidate to fill in basic personal information, which generally has fixed options to choose from and is formatted information. In addition, the system will also provide a richer information filling section such as self-introduction, educational background, work experience, self-evaluation, etc., which have no fixed options and belong to unformatted text information. On the other hand, when companies publish job requirements on the recruitment website, in addition to providing basic formatted information such as salary range, working hours, benefits, etc., they may also add textual information such as job content description and recruitment conditions as a supplement. In a more professional recruitment process, companies even value personalized resumes provided by candidates, where the textual information is more relevant and professional than the customized templates in the recruitment system. For both the candidate and the company, text descriptions provide a more flexible and effective way to describe their own content characteristics, and text information is richer than structured information. However, textual information is different from structured information, which needs to be text processed and transformed into structured feature vectors [7] before it can be used in recommendation algorithms. In particular, for Chinese text processing, the utilization of Chinese text information needs to go through two more processes, Chinese word separation and text feature vectorization, because there is no formal delimiter for words. Meanwhile, text information is used to construct text feature vectors by TF-IDF or counting, both of which require preconstructed keyword sets (also called vocabularies or dictionaries) and feature vectors (bag-of-words vectors) by means of address mapping. And since the vocabularies need to cover most of the keywords in the corpus in order to retain the information in the corpus as completely as possible and to characterize the text features more fully. Therefore, vocabularies are generally required to be large enough, and the dimensionality of the resulting text feature vectors reaches thousands or tens of thousands of dimensions, which increases nearly a hundred times compared with structured features that are often only tens of dimensions, greatly increasing the complexity of recommendation algorithms in performing similarity calculations. In addition, there are often only a few key content information in the text information that really determines the user's preference and evaluation of the item, which is sparsely distributed in the high-latitude vector and can easily be overwhelmed by other non-key dimensional information [8].

For these reasons, HR recommendation systems generally do not use textual information such as applicant's self-introduction, educational background and job description, or simply match some predefined keywords from the textual information as content features, and they mainly use simple and structured data. These data are manually filtered and customized for the characteristics of the HR recommendation field, which can basically cover the main factors of matching candidates and jobs, but cannot completely replace the role of textual information. To address the problem that it is difficult to extract and utilize the features of textual information in the HR recommendation field, it is most appropriate to adopt a strategy of fusing deep learning models with traditional collaborative filtering algorithms with the ability of deep learning to extract implicit features.

3.2. Overall design

The overall process scheme of the algorithm is as follows: the pre-preparation work has data acquisition and data pre-processing. Data pre-processing involves repeated data repair work, such as removing redundant data and filling in missing data according to certain rules, which is not a one-step process and requires continuous improvement of processing rules during the pre-processing process. After the data preparation is completed, the construction of the user-item rating matrix and the item text feature vector is started as the input of the hybrid depth collaborative filtering algorithm. The former is transformed from behavioral record information representing user preferences into ratings by certain rules, while the latter uses job description information to construct the textual feature vector of items. The probability matrix decomposition module then learns the implicit semantic vectors of users and items using both the low-dimensional feature vectors of items and the original rating matrix information. Subsequently, the predicted rating matrices of the non-cold-start items are calculated. The implicit semantic matrices U and V are the sets of implicit semantic vectors of all users and items obtained from the probability matrix decomposition, respectively, which are used to construct the predicted scoring matrices.

4. Deep learning-based algorithm for personalized recommendation of resources and model construction

4.1. Data collection

The data of job applicants include the gender, age, working years, job seeking intention, salary requirement, education background, working experience and self-presentation of job applicants, among which education background, working experience and self-presentation. The fields belong to long text fields. And the data of jobs include job category, number of views, number of favorites, working years requirement, salary, working hours, gender requirement, education requirement, benefits, job title and job description, etc., where job description belongs to long text field. The other part is the log system of the employment platform collects the data about the user's behavior record of using the system from the client, which is mostly log text data. In the process of using the recommendation system, users will click to browse the job information they are interested in, and may make further collections or applications for the job information if the details of the job match their job search needs. On the contrary, if the candidate is not interested in the browsed job, it may be marked as not interested [9]. The log text data collected in this paper records the behavior of these four major categories of candidates towards the post, which reflects the degree of candidates' preference for the post.

4.2. Data pre-processing

The data preprocessing work starts with extracting information about candidates' behaviors toward jobs from the unstructured log text files, including four types such as click to browse, favorite, apply and not interested. By analyzing the log records, it can be seen that the behavior log records are saved in a fixed format. The behavior log records in the text can be located by simple regular expression matching, and then the log record string is parsed according to the corresponding format to extract information such as applicant id, job id, behavior type, and occurrence time of the behavior. In addition, although the data collection work saves the original applicant and position information obtained from the collection into the database, these data still have problems such as duplicate redundancy, missing values and inconsistency. Therefore, the pre-processing stage also requires careful analysis of

these data and the design of corresponding pre-processing rules according to the specific problems existing in the data in order to clean and convert the data and thus improve the data quality.

4.3. Building HR data warehouse

Data preprocessing solves the problems of logical abnormalities, missing values and inconsistent data in the data after cleaning the human resource data, and the data saved in the database is already relatively pure at this time. After that, according to certain conversion and mapping rules, the numericalization of basic attribute fields in HR data can be realized. The mapping operation realizes the mapping from numerical values to string values by establishing dictionary tables for the non-numerical fields in the applicant and job information tables, and thus obtains the model of the data warehouse. The HR data warehouse consists of a three-part table structure with a user table, a project table and a user-project rating table.

The user table stores the basic attribute information of job applicants, eliminates useless fields such as social security number and name of job applicants, and establishes a dictionary table for mapping the fields such as target job category, salary range and education level. The item table stores the basic attribute information of job positions, and the job category, salary, education requirement, working time, and age requirement are mapped through the dictionary table respectively. The data preprocessing stage extracts the four types of behavior record information generated by the candidate for the job from the log text file, and this paper adopts a behavior-to-rating value mapping strategy [10-11] to transform the behavior record data into specific ratings to obtain the rating table, which stores the rating information of the candidate for the job and the occurrence time of the behavior corresponding to the rating. Since the type of candidate's behavior towards the job position reflects the degree of their preference for the position, which also represents the level of their rating for the job item. At this point, the HR data warehouse is constructed, using a snowflake structure, with `t_rating` as the fact table and other tables as dimension tables.

5. Conclusion

The development of the internet has led to the improvement of HR recommendation algorithms, and deep learning is now being used to process data more effectively. This paper examines the current state of collaborative filtering technology and explores the resource personalized recommendation algorithm and model using human resources as an example. The paper introduces theoretical methods such as deep neural networks and machine learning and analyzes the relationship between data preprocessing, data mining, and knowledge classification. The paper discusses the importance of personalized recommendations in HR management and highlights the role of collaborative filtering technology in achieving this goal. The authors present a detailed analysis of the different stages involved in building a personalized recommendation system, including data collection, preprocessing, and modeling. They also discuss the challenges involved in developing effective recommendation systems and the potential benefits of using deep learning-based algorithms.

The authors explore the application of deep neural networks in building personalized recommendation systems and provide a detailed explanation of the algorithm and model used in their study. They describe the data preprocessing techniques used to extract relevant features from the data and the machine learning algorithms used to build the model. They also present the evaluation metrics used to measure the performance of the recommendation system. The study demonstrates the potential of deep learning-based algorithms in building effective recommendation systems and highlights the importance of personalized recommendations in HR management. The authors discuss the limitations of their study and suggest future research directions to improve the accuracy and effectiveness of the recommendation system. In conclusion, this paper presents an analysis of the current state of collaborative filtering technology and explores the resource personalized recommendation algorithm and model using human resources as an example. The study highlights the potential benefits of using deep learning-based algorithms in building effective recommendation systems and the importance of personalized recommendations in HR management. The authors provide insights into the challenges involved in developing such systems and suggest future research directions to improve their accuracy and effectiveness.

References

- [1] C. Wang, L. Ma, R. Li, T. S. Durrani, and H. Zhang, "Exploring trajectory prediction through machine learning methods," *IEEE Access*, vol. 7, pp. 101441–101452, 2019.
- [2] G. George and A. M. Lal, "Review of ontology-based recommender systems in e-learning," *Comput. Educ.*, vol. 142, p. 103642, 2019.
- [3] Y. Bao, Y. Peng, and C. Wu, "Deep learning-based job placement in distributed machine learning clusters," in *IEEE INFOCOM 2019-IEEE conference on computer communications*, 2019, pp. 505–513.
- [4] W. Li *et al.*, "A comprehensive survey on machine learning-based big data analytics for IoT-enabled smart healthcare system," *Mob. networks Appl.*, vol. 26, pp. 234–252, 2021.
- [5] A. Crivellari and E. Beinat, "LSTM-based deep learning model for predicting individual mobility traces of short-term foreign tourists," *Sustainability*, vol. 12, no. 1, p. 349, 2020.
- [6] H. Luan and C.-C. Tsai, "A review of using machine learning approaches for precision education," *Educ. Technol. Soc.*, vol. 24, no. 1, pp. 250–266, 2021.
- [7] X. Zhou, W. Liang, I. Kevin, K. Wang, and S. Shimizu, "Multi-modality behavioral influence analysis for personalized recommendations in health social media environment," *IEEE Trans. Comput. Soc. Syst.*, vol. 6, no. 5, pp. 888–897, 2019.
- [8] I. Machorro-Cano, G. Alor-Hernández, M. A. Paredes-Valverde, L. Rodríguez-Mazahua, J. L. Sánchez-Cervantes, and J. O. Olmedo-Aguirre, "HEMS-IoT: A big data and machine learning-based smart home system for energy saving," *Energies*, vol. 13, no. 5, p. 1097, 2020.
- [9] V. Kamath and A. Renuka, "Deep Learning Based Object Detection for Resource Constrained Devices-Systematic Review, Future Trends and Challenges Ahead," *Neurocomputing*, 2023.
- [10] N. Sivaramakrishnan, V. Subramaniaswamy, A. Viloría, V. Vijayakumar, and N. Senthilselvan, "A deep learning-based hybrid model for recommendation generation and ranking," *Neural Comput. Appl.*, vol. 33, pp. 10719–10736, 2021.
- [11] M.-C. Chiu, J.-H. Huang, S. Gupta, and G. Akman, "Developing a personalized recommendation system in a smart product service system based on unsupervised learning model," *Comput. Ind.*, vol. 128, p. 103421, 2021.
- [12] A. Muniyasamy and A. Alasiry, "Deep learning: The impact on future eLearning," *Int. J. Emerg. Technol. Learn.*, vol. 15, no. 1, p. 188, 2020.
- [13] U. Gupta *et al.*, "The architectural implications of facebook's dnn-based personalized recommendation," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2020, pp. 488–501.
- [14] W. Juan, L. Yue-xin, and W. Chun-ying, "Survey of recommendation based on collaborative filtering," in *Journal of Physics: Conference Series*, 2019, vol. 1314, no. 1, p. 12078.
- [15] A. Bhoi, S. P. Pujari, and R. C. Balabantaray, "A deep learning-based social media text analysis framework for disaster resource management," *Soc. Netw. Anal. Min.*, vol. 10, pp. 1–14, 2020.
- [16] T. B. Lalitha and P. S. Sreeja, "Personalised self-directed learning recommendation system," *Procedia Comput. Sci.*, vol. 171, pp. 583–592, 2020.
- [17] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems," *Knowledge-based Syst.*, vol. 211, p. 106545, 2021.
- [18] S. Ahmadian, M. Ahmadian, and M. Jalili, "A deep learning based trust-and tag-aware recommender system," *Neurocomputing*, vol. 488, pp. 557–571, 2022.
- [19] K. Wang, T. Zhang, T. Xue, Y. Lu, and S.-G. Na, "E-commerce personalized recommendation analysis by deeply-learned clustering," *J. Vis. Commun. Image Represent.*, vol. 71, p. 102735, 2020.
- [20] S. S. Khanal, P. W. C. Prasad, A. Alsadoon, and A. Maag, "A systematic review: machine learning based recommendation systems for e-learning," *Educ. Inf. Technol.*, vol. 25, pp. 2635–2664, 2020.
- [21] Z. Zhong, M. Xu, M. A. Rodriguez, C. Xu, and R. Buyya, "Machine learning-based orchestration of containers: A taxonomy and future directions," *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1–35, 2022.
- [22] H. Gao, X. Qin, R. J. D. Barroso, W. Hussain, Y. Xu, and Y. Yin, "Collaborative learning-based industrial IoT API recommendation for software-defined devices: the implicit knowledge discovery perspective," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 6, no. 1, pp. 66–76, 2020.
- [23] M. F. Shakeel, N. A. Bajwa, A. M. Anwaar, A. Sohail, and A. Khan, "Detecting driver drowsiness in real time through deep learning based object detection," in *Advances in Computational Intelligence: 15th International Work-Conference on Artificial Neural Networks, IWANN 2019, Gran Canaria, Spain, June 12-14, 2019, Proceedings, Part I 15*, 2019, pp. 283–296.
- [24] S. Sharma, R. K. Dudeja, G. S. Aujla, R. S. Bali, and N. Kumar, "DeTrAs: deep learning-based healthcare framework for IoT-based assistance of Alzheimer patients," *Neural Comput. Appl.*, pp. 1–13, 2020.
- [25] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective," *Comput. Networks*, vol. 182, p. 107496, 2020.