Vol: 06 Issue: 06
International Journal of Data Science and Advanced Analytics (ISSN: 2563-4429)

Balancing Accuracy and Efficiency: A Comparative Analysis of Collaborative Filtering Algorithms for Job Recommendation Systems

Arash Khosravi[1], Ahmad Azarnik[2]
Faculty of Engineering, Mahallat Institute of Higher Education, Mahallat, Iran
[1] Khosravi.280@gmail.com, [2]ahmad.azarnik@gmail.com

Corresponding author email: ahmad.azarnick@gmail.com

Abstract**—** Recommender systems are commonly used to suggest relevant items to users, like movies or products. The digital transformation of the business sector has led to a surge in online job opportunities. This shift necessitates effective job recommendation systems to connect qualified candidates with relevant positions. This study evaluates the performance of four collaborative filtering algorithms for a job recommender system: Singular Value Decomposition (SVD), SVD++ (SVDPP), co-clustering, and Non-Negative Matrix Factorization (NMF). We employ error rate, training time, and cross-validation performance as key evaluation metrics. Our findings reveal a trade-off between accuracy and efficiency. The co-clustering approach achieves the lowest error rates, indicating its effectiveness in recommending relevant jobs. However, this benefit potentially comes at the cost of increased training time compared to other methods. Conversely, the NMF-based model demonstrates significantly faster training times, making it computationally efficient.

Keywords**— Recommender System, E-recruitment, Singular Value Decomposition, Machine Learning, Collaborative-Filtering.**

1. Introduction

Recommender system (RS) or recommender -the word system is sometimes replaced by synonyms such as "platform" or "engine"- is a subset of the information filtering system that seeks to predict the "score" or "priority" that the user will give items[1] – e.g. data, information, and goods. In recent years, recommender systems have become very common and have been used in various fields, some of its popular domains of application [2] include movies, music, news, books, research papers, traveling, search queries, social tags, and product dominance. In e-commerce, recommender systems are initially based on factors: demographics, history of purchasing, and behavior of buyers [3]. In tourism [4] the important items are demography, details of travel, user's information, details of destination, and user's feedback. [5] used demographic data for the recommendation system LIFE Finder. Common recommendation methods are collaborative filtering (CF)[6-8], content-based (CBF) [9-10], Demographic filtering (DF), and Hybrid.

Collaborative Filtering: It is considered the most popular method in recommender systems. Collaborative filtering or CF works based on the user's past decisions and is an effective method to solve the information overload problem where each user is associated with a set of ranking scores on the item set. CF algorithms are based on the hypothesis that users will rate and act similarly on other items if users have similarly rated items or behave similarly. In other words, these systems suggest items by considering users' tastes, in terms of item's interests, and assuming that users will be interested in things that similar users have rated highly [11]. The basic assumption of these recommender systems is that if users x and y have evaluated the same item, or have similar behaviors -for example, shopping, watching TV, listening to music-, then on other items will evaluate or act similarly. Collaborative techniques use a database of users' preferences (or interests) about items to predict additional topics or products that an active user might like [1].

Content-based recommender systems: These systems recommend items based on the content of users' information, assuming that a user will like items similar to one other users have previously been interested in. User profiles can be created by building a model of user preferences using descriptions and types of items that a user is interested in, or a history of user interactions stored in the system e.g. Purchase history. It means that a Content-based recommender system tries to suggest items that are similar to what a user liked in the past. In fact, matching the attributes of a user profile, in which preferences and interests are stored, with the attributes of a content object is the basic process performed by this technique [10].

Demographic filtering: RSs are integrated with concepts such as followers, friend lists, posts, and tags. These social data have been of interest to researchers with three primary goals: improving the quality of predictions and recommendations, providing and creating new RS, and clarifying the relationships between these data and collaborative processes [12].

Memory-based: Memory-based methods predict based on the similarities between users or items and work only on the users' rating matrix and use the similarity criterion to determine the distance between two users or two data items based on ratio coefficient e.g. user-to-user or item-to-item methods and the combination of these two methods. It should be noted that most business systems, such as Amazon's, are memory-based.

Model-based: Model-based methods make predictions based on a mathematical model. Models such as "Bayesian neural network" classifiers, "matrix factorisation" and fuzzy systems. One of the most accurate methods is based on the "factorisation matrix" model. Matrix factorisation (MF) uses the integration of users' information and their feedback into items -such as clicks, purchases, and downloads- to improve accuracy in older recommender systems. In simpler words, this model uses ranking. Its purpose is to model user interactions with the latest features related to items in the system. In fact, it maps users and items in a low-dimensional hidden space to determine the similarity between them. In addition, MF can be represented by both a mathematical formula and a graphical probabilistic model. The conditional Boltzmann machine is one of the MF models with a combination of neighborhood method.

Session-based recommender systems: These recommender systems use user interactions in a session [13], and these systems are used on YouTube and Amazon [14]. These are especially useful when user history –e.g. previous clicks, purchases- of the user is not available or relevant in the current session. Most of the proposed session-based systems rely on the sequence of recent interactions in a session without the need for additional details (historical, demographic) of the user. Session-based recommendation techniques mainly employ generative sequential models such as neural networks [15], and transformers [16].

Hybrid recommender systems: Both content-based RS and pure collaborative refinement RS have problems, for this reason, hybrid methods for recommender systems have been proposed, which are combinations of content-based and refinement. There are different ways to combine collaborative and content-based methods:

- Implementing collaborative then the results are sent to content-based method[17][18].
- First, the CF and CBF are combined together, next the results are sent to DF[19].
- Combination of CF with methods such as clustering[20], or Fuzzy logic [21].

Several studies empirically compare the performance of hybrid with robust and content-based methods to show that hybrid methods can provide more accurate recommendations than pure methods. These methods can also be used to overcome some common problems in RSs such as cold-star and data sparsity as well as knowledge engineering bottlenecks in knowledge-based approaches [12].

2. RELATED WORKS

The information system (IS) has been supporting companies and people through resource management such as storing and tracking of employees and candidates. The applicant data has been reviewed through applicant management systems supporting internal work processes and communication processes between the human resource management group and other departments. Recently, the use of these systems has increased. The amount of digital information and the emergence of electronic commerce reforms the way companies do business in many ways. At first, simple solutions are applied, such as posting recruitment ads in the job section of the corporate website.

Currently, the proposed RS frameworks are used to address the problem of information overload in any domain and enable clients to focus on important information in their domain of interest. An area where such recommendation systems can play an important role is helping university graduates achieve their dreams by offering them jobs based on their interests and skills. Nowadays, there are a lot of websites that provide a wealth of information about job opportunities, but this is overwhelming for students as they need a lot of information to find the ideal job. At the same time, existing job recommendation systems (JRS) consider users' interests, characteristics, and skills, therefore, these systems can recommend jobs tailored to the user. Researchers examine the existing job recommendation system and highlight the drawbacks of these systems such as cold-start, and scalability and fragmentation. In addition, the proposed implementations of the job recommendation system using machine learning have been researched in order to identify how recommender systems introduce the features of security, reliability, and transparency in the job recommendation process, and these are beyond the scope of this research.

[22] stated that a recommendation system is a technique that provides users with information that they may have been interested in or had access to in the past. The online recruitment platform or Internet-based recruitment platform is one of the most successful business progress that is changing the way companies hire candidates[23]. These platforms have expanded in recent years, as hiring the right person is a challenge that most companies face, and also the unavailability of certain people in some skill areas has long been recognised as a major obstacle to the success of companies[24].

Channels such as an Internet job portal, social media programs, or a career website have taken into account a company's career development plans. While companies have posted job positions on these portals, job seekers use them to publish their profiles. For every job posted, thousands of resumes are received by the system. As a result, companies receive a huge amount of job descriptions and candidate resumes online. This vast amount of information gives a great opportunity to increase the quality of matching between people and jobs in the company. This potential of search functionality has not been used in recruitment applications, which are mainly limited to Boolean method searches. Therefore, the need to use JRS technologies that can help recruiters increases gradually[25]. However, the job recommender system is still a challenging domain and a growing research area, in order to support this research area.

Recruitment steps include working with a program from candidates and pre-selection of them. However, the best match between jobs and candidates depends on fundamental aspects that are difficult to measure. These fundamental aspects are a significant reason why information systems are widely involved in personnel selection. For a long time, IS technology has been used for the pre-selection of applicants, based on the Boolean search method. The method contains a combination of keywords that determine the skill requirements in order to determine those candidates who match the search criteria, and such skills must be matched in many e-recruitment applications [25].

However, as mentioned above, simple filtering techniques such as the Boolean search method cannot be adequate to understand the complexity of job proportionality[26]. Decisions often depend on fundamental characteristics such as personal characteristics or social skills that cannot be easily tapped. In addition, understanding the requirements, in terms of mandatory and optional skills should be considered [27].

Among the techniques of the proposed systems, it is possible to solve the problem of information overload by prioritizing and analyzing the information for each user based on their learned preferences[28]. In addition, the success of personalization depends entirely on the existence of a comprehensive plan. User profiles that precisely capture people's interests [29]and are fully compatible with the methods under review. In addition, the RSs can use the ranking information to find the type of work required and to determine which type of candidate's profile has been considered by the potential recruiter in the past for a positive ranking[30].
The information can then be used to predict matches between jobs and previously unranked candidates. The need to use recommender systems is a selection process method that can be created from different motivations and include different perspectives. While we are interested in how people choose a suitable occupation, other researchers are interested in how people effectively cooperate[31].
[32] stated that the dynamics of the labor market and the job forming are constantly evolving. Career mobility is not evident, and providing effective advice in this area is particularly challenging. [32] presented the Work Market Explorer interactive dashboard, which enables job seekers to explore the job market in a personalized way based on their skills and abilities. Through a user-centric process with job seekers and career brokers, a dashboard is proposed to enable job seekers to discover career recommendations and the qualifications they need, as well as how to map these qualifications to their profile.
[33] stated that the implicit and explicit feedback signals that they can collect are rare occurrences that complicate the evaluation task. Online evaluation (A/B testing) is usually the most reliable way to measure the results of their experiments, but it is a slow process. In contrast, the offline evaluation process is faster, but it is very important to ensure that it informs our decision to make new improvements in production. [34] used the neighborhood method and PSO optimization algorithm for information processing in networks, especially networks with a high number of users. In this research, their method was data clustering which could be optimized with the PSO optimization algorithm. The advantages of this method are high reliability, low time, and error in information processing. [35] examined the prediction of users' behavioral models and the use of data mining methods, which, as the author of the article has pointed out, in large databases, this method cannot have high reliability. [36] proposed a hybrid framework called USG for location recommendation in location-based social networks. This recommender obtains the probability of registering the user's position in a place based on three probabilities obtained from three user-based group filtering models, the social influence model and spatial influence model.

2.1. Steps in employments

The process of hiring people is valuable for a company. There are two perspectives in this field that are mutually exclusive: job seekers are hired by companies by determining a set of requirements and restrictions on skills, expertise level, and grades. On the other hand, job seekers create their own CVs and after specifying their educational background, work experiences, and skills[37][38] go after their applications.
[25]state that the relationship between recruitment tasks and the recruitment process can be divided into two main stages: the recruitment stage and the selection stage, both stages include a planning and an implementation part. The planning part determines the overall strategy and actual actions to attract valuable employees as well as explicit selection methods. Implementation includes employer branding activities that include all long-term activities and marketing activities that attract qualified candidates. The attraction phase is aimed at creating a description for the market, and the job openings in the selection phase begin by re-screening resumes and other submissions. Then, the final selection of candidates begins according to the comparison between people who have been filtered in the previous stages. Finally, applicant management serves as a secondary function, and it includes contacting applicants, managing applicant data, and related processes such as directing applications to organizational members involved in decision-making in the hiring process.

2.2. E-recruitment systems

E-recruitment is a system to quickly reach a large number of potential job seekers. E-recruitment is attractive and its growth started in the late 1990s when the world experienced rapid economic growth. Recent research shows that the increasing demand for IS technologies for human resource management in general processes and recruitment in particular has been of interest. Most companies focus on e-recruitment systems as the main recruitment channels. Advertisements are automatically logged into the job portal as soon as they are published.
The applicant creates a profile to apply to one of the mentioned job positions. The user's profile is stored in the system allowing the applicant to reuse it for another job position, this is also a feature of these systems. The latest functionality allows companies to identify applicants. Therefore, the companies achieve success and are able to create a uniform view for all applicant data in one portal. This portal is used by the recruitment department to be able to find the applicant's documents. Applicants with good documents are directed to HR departments for further processing. In addition, the system also supports all required communication processes as it tracks the status of the applicant in the application process[31] Electronic recruitment platforms are usually based on search and Boolean filtering techniques that cannot adequately understand the complexity of a person's job and make decisions for selecting people[39].
These changes created a huge demand for qualified people. Electronic recruitment platforms such as company homepages and job portals (e.g. monster.com) show the progress of the International Association of Recruitment Websites includes more than 40,000 job sites that serve job seekers and recruiters. While companies post job openings for these jobs on portals, job seekers use them to post their profiles, this has created a large number of job descriptions relevant to candidates[40].
The adoption of these e-recruitment operating systems has been cost-saving, effective, and suitable for both parties, i.e.

recruiters and job seekers [41]. Some of these systems have weaknesses that result in a large number of candidates missing employment opportunities. The research results show that this type of search is inadequate to achieve the right fit between the candidate's talents and job conditions[42]. [41]presented several categories of recruitment resources:

- General-purpose job sites (eg, Monster.com, HotJobs.com) that offer complete online recruiting. While job seekers search for jobs based on categories such as experience, location, education, or any combination of these attributes, recruiters search databases for applicants by skill, experience, preference, education, salary, or any combination of these attributes.
- Job markets (for example, Dice.com, Erexchange.com) serve specialized markets such as a specific job, industry, education or any combination of specializations
- Electronic recruitment application service providers – e.g. RecruitUSA, PeopleClick- offer a range of services like recruitment software, recruitment process management, training, and education.
- A combination of recruitment service providers in the traditional sense that provide services, e.g. magazines and journals.
- Electronic Recruitment Consortium is a search engine that directs job traffic directly to a member's career-like website, for example DirectEmployers.com; NACElink.com
- A corporate career website is a recruiting resource typically used by 500 fortune companies whereas the use of a corporate career website is one that regularly implements e-commerce programs.

In this context, it is usually necessary to refer to person's occupation, person, and organization of appropriate people[43]. We explain the requirements in th following.
Requirements: In job recommendation systems, it should be stated that the system must have the following features:

- The matching of people with jobs depends on their skills and education.
- Recommending people is a two-way process that needs to consider not only the desired preferences but also the conditions of the recruiter and the job candidate should be understood bilaterally.
- Recommendations should be based on the candidate's characteristics as well as personality traits.
- The individual is considered unique.

One of the problems is a two-way recommendation between the job seeker and the job. The recommendation process can be divided into two parts: job recommendation and job applicant recommendation. The design idea of these two parts is almost the same [44][31]. For a job seeker, a job with a higher matching degree should be recommended. Likewise, for a job, the job seeker with a higher degree of match should be recommended[44]. In general, there are ranking items or top "n" people that best fit the desired job or the top job. [38] stated that the adaptation of skill needs should distinguish between the requirements in the adaptation process and the requirements in the restrictions that must be applied by the applicant.
Candidates must be matched to a job based on job performance indicators. In the choice theory, the information available in a specific case at the time of choice and decision is called predictive data, which consists of individual attributes. Process forecasting refers to the evaluation of metrics using predictive data and a specific method of composite data. In order to build candidate profiles, metadata extracted from existing resumes is reviewed and analyzed. [45] presented a system that creates user profiles in the recruiting environment directly by analyzing the behaviors of web users. In this system, user profiles and their information can be accessed by identifying and checking users' information. [46]used input data for their Curriculum Vitae (CV) recommender: demographic data, educational data, work experience, language skills and IT skills, honors, publications, and so on. In general, the candidate's profile consists of three factors:

- Personal information about employees, such as first name, last name and location
- Information about the current and past careers of the positions that the candidate has chosen. This section may contain the company name, position, company description, job start date, and end date. The company description section may contain more information about the company, e.g. number of employees and industry.
- Information related to educational experiences, such as the name of the university degree, fields of study, begin and end.

In addition, for cooperation, candidates may be asked to rate the job profiles from 1 to 5 whether their preferred profiles are relevant to their career prospects and plans. From these metadata, several features can be considered for training and recommendation. In other words, the job profile should be made to describe the requirements and list all the relevant skills of the employee.

2.3 Data mining stages of recommender systems: In this section, we explain required stages in processing the data.

- **Business Understanding:** This stage is the most important stage of the process. First of all, the problem should be understood to carry out the data mining project. Secondly, the influencers on the project should be determined.
- **Data selecting:** Data selection is done in two parts. One is when we reduce the number of attributes and the other is when we select the data by reducing the observations which can be done through three ways: sampling, intelligent sampling, and learn to ignore.
- **Data Understanding:** This part is related to the concept of data. It includes the following steps: collecting primary data, describing data, exploring data, and indicating data quality.

Data Preparation: This section is related to data preparation and includes the following stages: data cleaning, data transformation, and data integration so that data coding and naming have the same standard.

3. Proposed Model

According to related works and requirements, we propose a method for evaluating job recommender system. The data preparation stage utilizes a dataset containing 459 jobs rated by 2,883 users. We extract user search data from the search table and construct a user-job interaction matrix. This matrix is populated by incrementing the count for each job a user has searched for. Jobs not searched for by a particular user are assigned a value of zero. The resulting matrix, representing user-job interactions, is then converted into a standard format suitable for using by some machine learning algorithms. Finally, the processed data is stored in a memory-resident data frame for efficient model training (Fig 1).

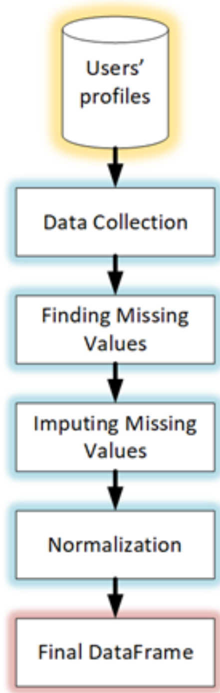

Fig. 1. Preprocessing steps adopted in this study

The sample output of this data structure can be seen in Table 1.

Table 1. Sample Data used in this study

Worker	Guard	Hardware - Internet advertising	supermarket -fast food	Accountants	Electrical industries
5	1	0	0	0	0
0	0	13	1	1	5
0	0	0	0	0	0
0	0	1	0	0	0
0	0	0	0	0	0
2	0	0	0	0	0
0	0	0	0	0	0

In the following we explain the steps in details:

3.1. Preprocessing steps:

The user data employed in this recommender system encompasses all user interactions, including searches, saved profiles, and final selections. This comprehensive approach captures various user behaviours that can indicate interest in a job. For example, a user might search for two welders, save the profiles of two additional welders, and ultimately visit the profile of one welder before making a selection. In this scenario, the user would be considered to have three implicit ratings for welders. To facilitate the application, the user data is structured into a three-column table. This table represents a user-job interaction matrix, where each row corresponds to a user, and the columns represent:

- *item (job)*: The unique identifier for the job the user interacted with.
- *rating*: An implicit rating value assigned based on the user's interaction type (e.g., search = 1, save = 2, visit = 3).
- *userID*: The unique identifier for the user.

This transformation process reduces the data dimensionality while preserving essential information about user-job interactions.

Missing Value Imputation: Our approach addresses missing values in the user-job interaction data through imputation. This is crucial to mitigate two key concerns: data distortion and model bias. Significant missing data can distort the distribution of variables within the dataset. For example, missing entries might inflate or deflate the representation of specific job categories (Data Distortion). Missing data can introduce bias into the dataset, leading the model to produce inaccurate analyses. In our case, the dataset size is relatively small, and losing any data points could significantly impact the final model's performance. Therefore, we prioritize data completeness.

Imputation Strategy: We employ imputation to fill missing values. Here, we replace missing values with zeros (0). This approach assumes that a missing value signifies a lack of interaction between a user and a particular job. However, it's important to acknowledge that a zero value can also represent an actual interaction with a rating of zero.

Data Normalization: Data normalization is a crucial step that helps to improve the training process and achieve accurate results. It ensures that features within the dataset are on a similar scale, preventing features with larger ranges from dominating the model's learning process.

3.2. Employed Collaborative Filtering algorithms

In this section we explain (Fig 2):

The user-based filtering approach involves the following steps:

- *User Similarity*: The system calculates the similarity between the active user (the user for whom recommendations are generated) and other users based on their historical interactions with items.
- *Rating Prediction*: The model predicts item ratings for the active user by leveraging the ratings provided by similar users from step 1.

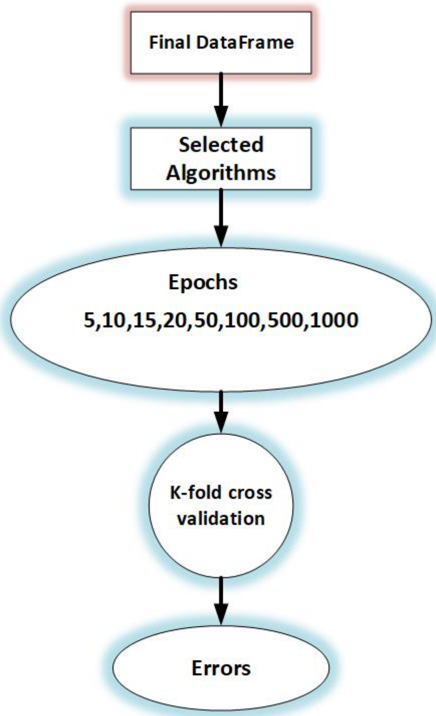

Fig. 2. The structure of training and test

SVD: Singular Value Decomposition (SVD) is a matrix factorization technique commonly used in recommender systems for dimensionality reduction and recommendation generation. SVD is a powerful tool for recommender systems due to its ability to reduce data complexity, extract meaningful features, and generate recommendations. SVD decomposes a user-item interaction matrix (containing user ratings or interactions with items) into three matrices:

U: Represents user latent factors, capturing user preferences for different underlying factors.

Σ: Contains the singular values, indicating the importance of each latent factor.

V^T: Represents item latent factors, capturing the characteristics of each item in terms of the latent factors.

SVD simplifies the original data by representing it in a lower-dimensional space, making it easier to analyze and handle large datasets. SVD identifies latent factors that capture underlying patterns in user-item interactions, revealing hidden relationships and insights into user preferences.

The recommender systems must take advantage of all available interactions, both explicit (e.g. numerical ratings) and implicit e.g. likes, purchases, rejected, flagged. For this purpose, SVD++ is also designed to consider implicit interactions. Compared to other algorithms, SVD also considers user bias. The predicted user rating that you will give to the item is calculated in equation 1:

1) $\tilde{r}_{ui} = \mu + b_i + b_u + \sum_{f=0}^{n\,factors} H_{u,f} W_{f,i}$

SVD++(SVDPP) aims to improve upon the standard SVD algorithm by incorporating both explicit and implicit feedback data.

- *Explicit feedback*: This refers to user-provided ratings (e.g., star ratings on movies).
- *Implicit feedback*: This includes any user interaction data that can infer user preferences, such as clicks, purchases, browsing history, etc.

By considering both explicit and implicit interactions, SVD++ provides a more comprehensive understanding of user preferences, potentially leading to more accurate recommendations. SVD++ takes into account individual user biases, meaning it recognizes that users might have different rating tendencies compared to others. This helps to personalize recommendations further. SVD++ is not a model-based method, this means that if a new user is added, the algorithm will not be able to model it unless the entire model is retrained. Even though the system may have collected some interactions for that new user, its latent factors are not available and therefore no recommendations can be calculated. This is an example of a cold-start problem, in the sense that the recommender cannot deal with new users or new cases effectively, and special strategies must be considered to solve this problem. One possible way to solve this problem is to modify SVD++ to become a model-based algorithm, thus allowing easy handling of new cases and new users.

In SVD++, we do not have new users' latent agents, so we need to represent them in a different way. Latent factors of the user indicate the preference of that user over the latent factors of the relevant item, so the latent factors of the user can be estimated through past user interactions. If the system is able to collect some interactions for the new user, its latent factors can be estimated. This doesn't completely solve the cold-start problem, as the recommender still needs reliable interactions for new users, but at least it doesn't need to recalculate the entire model every time.

Non-negative matrix factorization (NMF) is a technique used in recommender systems for dimensionality reduction and feature extraction. NMF is a valuable tool for recommender systems due to its ability to reduce data complexity, extract meaningful features, and provide interpretable insights into user preferences, leading to potentially more accurate and personalized recommendations.

NMF decomposes a user-item interaction matrix (containing user ratings or interactions with items) into two lower-dimensional matrices

W: Represents hidden user features or latent factors. Each row in W captures the user's preferences for different latent factors.

H: Represents hidden item features or latent factors. Each column in H captures the characteristics of each item in terms of the latent factors. NMF simplifies the original data by representing it in a lower-dimensional space, making it easier to analyze and handle large datasets. NMF identifies latent factors that capture underlying patterns in user-item interactions, revealing hidden relationships and insights into user preferences.

NMF is a group of algorithms in multivariate analysis and linear algebra in which an X matrix is factored into two P and Q matrices, with the characteristic that all three matrices have no negative elements. Let the input data matrix contain a set of "n" data vectors as columns. We consider form factors in equation 2:

$$2)\ where, X \epsilon R^{N\times M}, P \epsilon R^{D\times N}\ and\ Q \epsilon R^{D\times M}, X \approx PQ^T$$

Co-clustering: Co-clustering is a technique used in recommender systems that involves clustering both users and items simultaneously. This allows for the identification of groups (co-clusters) where users with similar preferences are associated with items that share certain characteristics. Co-clustering offers a valuable approach for recommender systems by identifying groups of users with similar preferences and items with shared characteristics. This allows for more targeted and potentially more accurate recommendations.

3.3. Model Training and Evaluation

This section details the process of training and evaluating the recommender system model.

Hyperparameter Tuning: The model's performance is highly dependent on the chosen hyperparameters. In this work, we focus on two key hyperparameters: the number of epochs and the number of folds for cross-validation. We explore a range of epochs (5, 10, 15, 20, 50, 100, 200, 500, and 1000) to determine the optimal training duration for the model. Additionally, we employ k-fold cross-validation with 2, 5, and 10 folds to assess the model's generalizability on unseen data (Fig. 2).

Evaluation Metric: We evaluate the model's performance using Root Mean Square Error (RMSE). RMSE measures the difference between the predicted ratings by the model and the actual user ratings. Lower RMSE values indicate better model performance. The formula for calculating RMSE is provided below:

$$RMSE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_i - x_i{'})^2}$$

4. RESULTS AND ANALYSIS

This section presents an empirical evaluation of the proposed recommender system model benchmarked against established techniques: Singular Value Decomposition (SVD), Sparse SVD++ (SVDPP), and co-clustering. The evaluation focuses on three key metrics: error, training time, and cross-validation performance.

4.1. Training Time and *Cross-Validation Analysis:*

Our analysis, visualized in Fig. 3, reveals a clear trend between the proposed NMF-based model and the number of cross-validation folds. NMF consistently exhibits the shortest training times across all cross-validation configurations (2, 5, and 10 folds). Conversely, SVDPP demonstrates the longest training times in all scenarios.

For instance, with 2-fold cross-validation, NMF achieves the best training time. This pattern persists with 5-fold and 10-fold cross-validation, where NMF maintains the lowest training time (0.026928 and 0.027227, respectively) compared to SVDPP's significantly higher times (94.678282 and 112.732624, respectively).

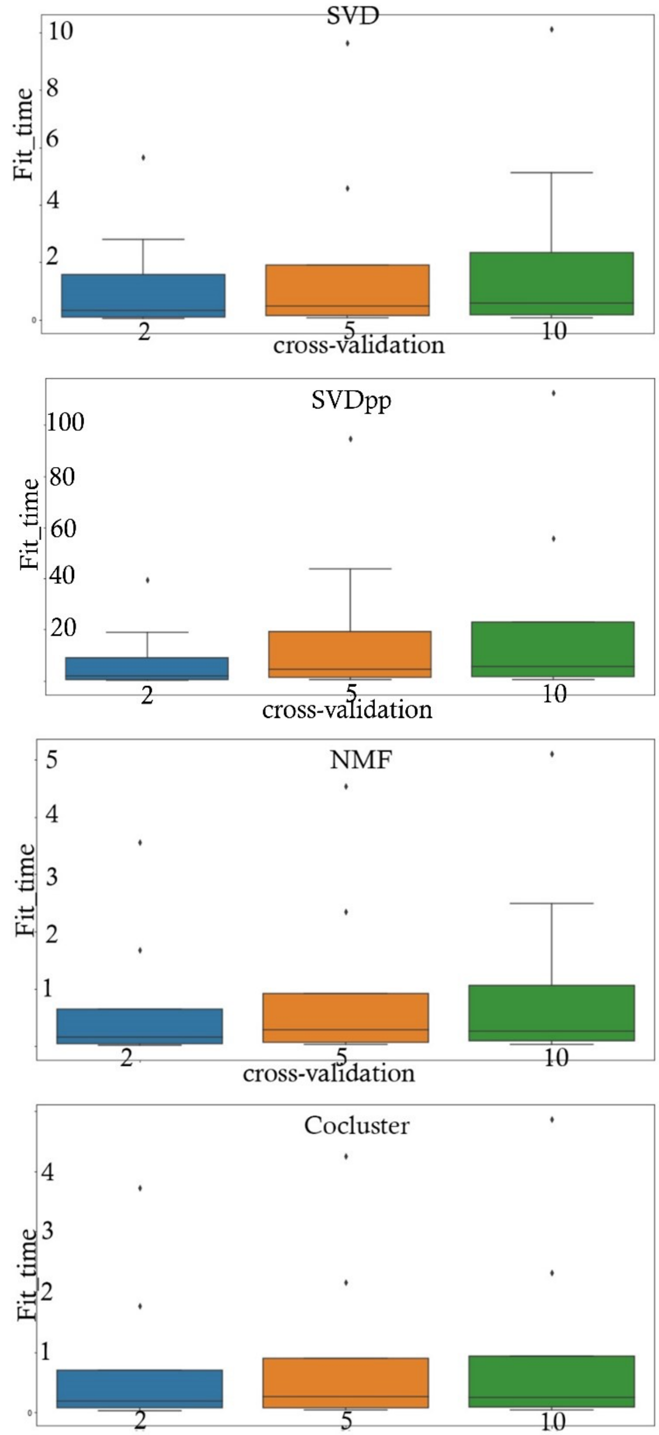

Fig. 2 .Training time of algorithms

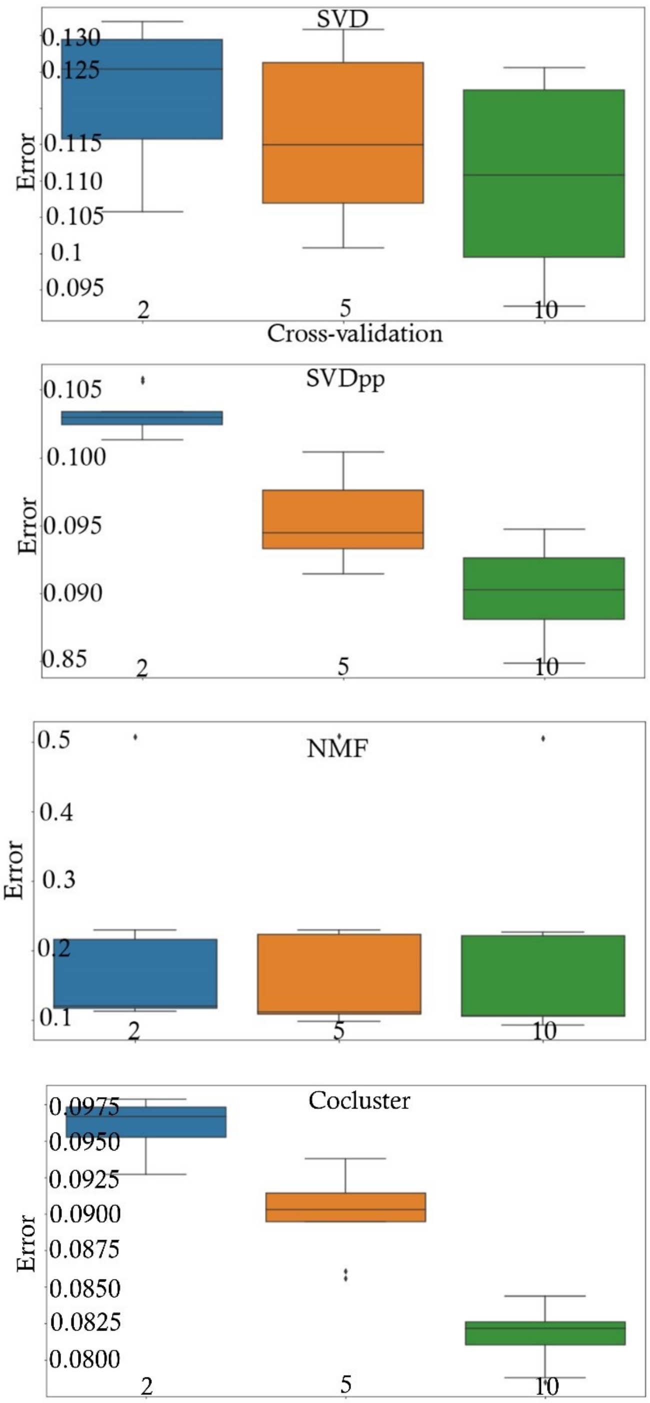

Fig. 3. Results of Error time regarding to cross-validation

4.2. Error and Cross-Validation Analysis

Fig. 4 depicts the relationship between error and the number of cross-validation folds. While co-clustering consistently achieves the lowest error rates across all folds (0.092706, 0.085571, and 0.078475 for 2, 5, and 10 folds, respectively), the NMF-based model exhibits higher error rates (0.506724, 0.508318, and 0.505258).

Trade-off Consideration: It's important to acknowledge a potential trade-off between error and training efficiency. As discussed in the previous section, NMF demonstrated significant advantages in training time. Here, we observe that co-clustering achieves lower error rates, but potentially at the cost of increased training time (further investigation is needed to confirm this).

4.3. Impact of Epochs on Training Time

Fig. 5 showcases the influence of epochs (training iterations) on training time for the evaluated models. Consistent with previous findings, NMF consistently exhibits the fastest training times across all tested epoch values (5, 10, 15, etc.). Conversely, SVDPP demonstrates the slowest training times in all scenarios. For example, with only 5 epochs, NMF achieves the best training time (0.015443) compared to SVDPP's significantly higher time (0.016257). This trend strengthens as the number of epochs increases. By 15 epochs, NMF maintains the lowest training time (0.010971) while SVDPP exhibits a much larger training time (1.624648). The NMF-based model demonstrates a significant advantage in training efficiency, requiring less time to converge compared to the benchmark methods, particularly SVDPP. This efficiency becomes more pronounced as the number of epochs increases.

4.4. *Error and Epochs Analysis*

Fig. 5 explores the relationship between error rates and the number of epochs (training iterations) for the evaluated models. While co-clustering achieves consistently lower error rates across all epoch configurations (5, 10, 15, etc.), the NMF-based model exhibits higher error rates. For example, with just 5 epochs, co-clustering demonstrates a significantly lower error (0.078475) compared to NMF (0.508318). This trend persists as the number of epochs increases. As observed in previous sections, co-clustering achieves lower errors, but potentially at the expense of training time. Here, we see co-clustering maintaining lower errors with increasing epochs. However, a more comprehensive analysis is needed to determine if this advantage comes at a significant cost in training efficiency for co-clustering compared to NMF.

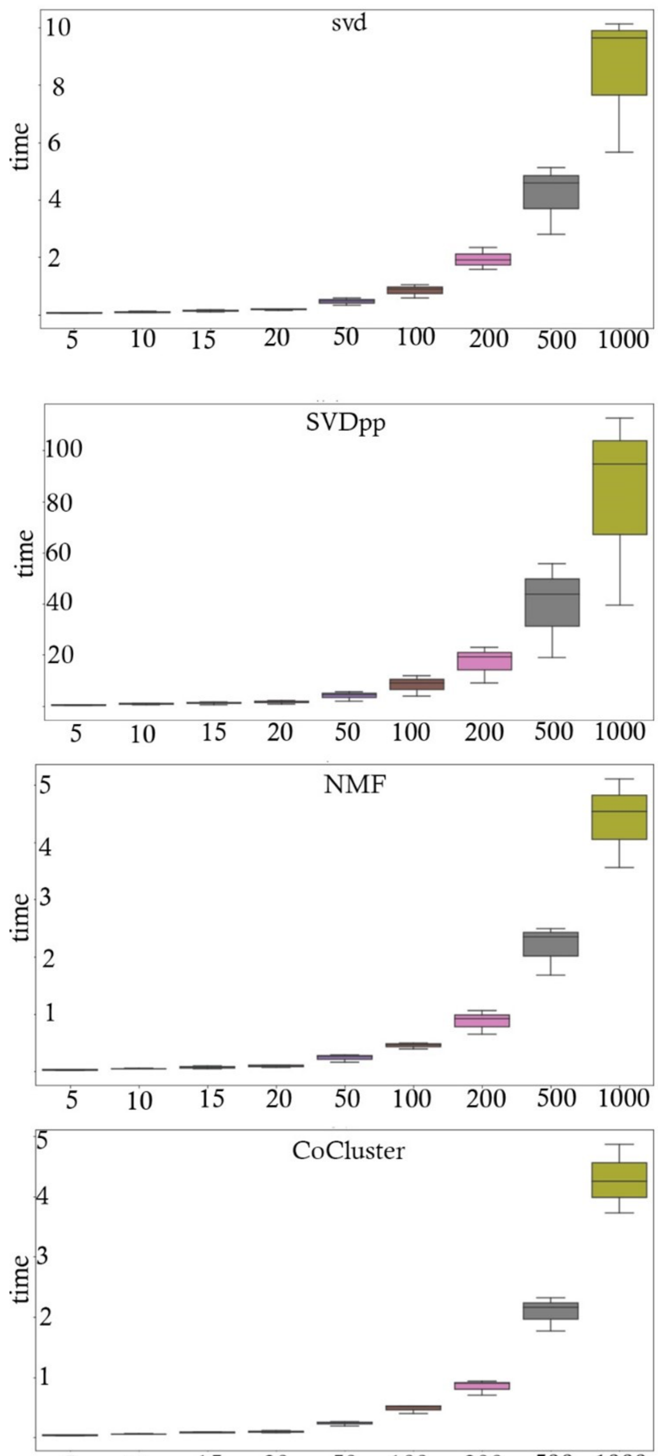

Fig. 4. Training time of algorithms in different epochs

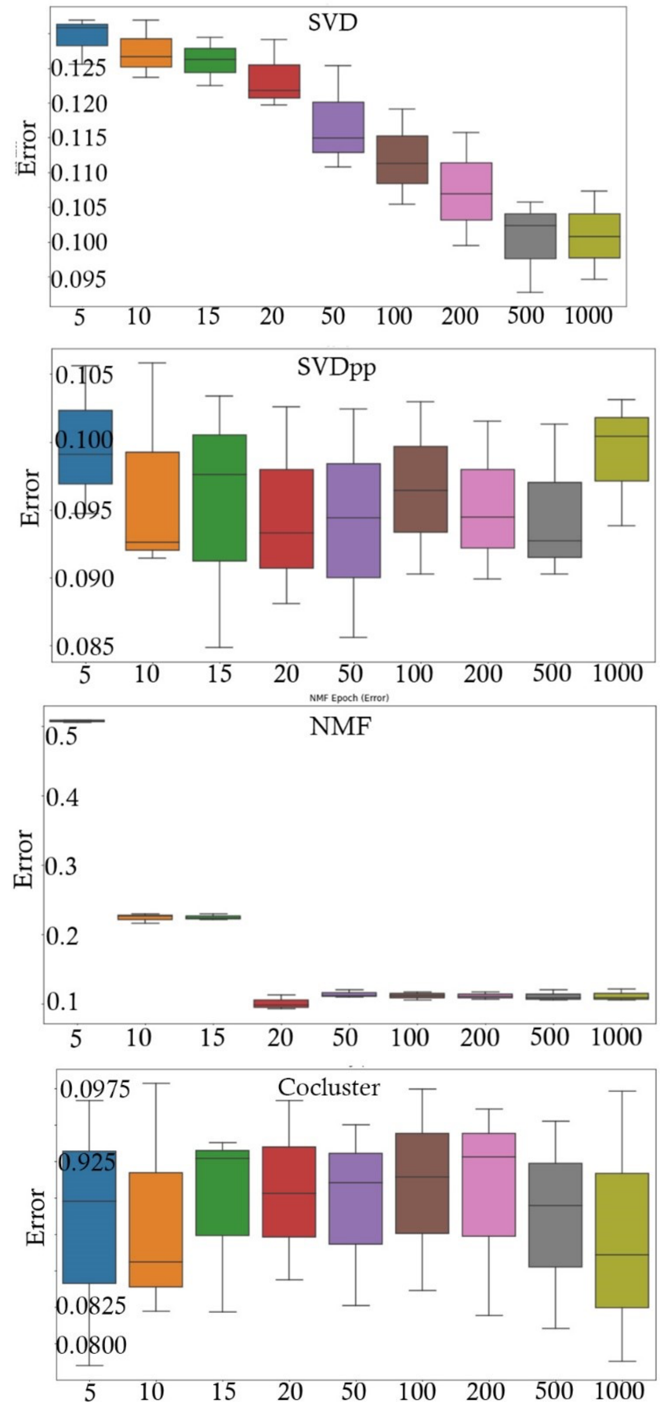

Fig. 5. RMSE in different epochs

5. Discussion

5.1. *Performance* Comparison *of Recommender System* Techniques*:*

This study presents a comparative analysis of several recommender system models: Non-negative Matrix Factorization (NMF), Singular Value Decomposition (SVD), Sparse SVD++ (SVDPP), and co-clustering. The evaluation focuses on three key metrics: training time, cross-validation performance, and error rates.

5.2. *Training* Time and *Cross-Validation*:

NMF consistently exhibits the shortest training times across all cross-validation configurations (2, 5, and 10 folds). This efficiency advantage becomes more pronounced as the number of folds increases. Conversely, SVDPP demonstrates significantly longer training times in all scenarios.

5.3. Error and Cross-Validation:

Co-clustering consistently achieves the lowest error rates across all cross-validation folds, indicating its superior performance in accurately predicting user preferences. The NMF-based model exhibits higher error rates compared to co-clustering, suggesting a potential trade-off between training efficiency and accuracy.

5.4. Trade-off Considerations:

While NMF demonstrates significant advantages in training time, co-clustering achieves lower error rates. Further investigation is needed to quantify the exact cost of this trade-off in terms of training time for co-clustering compared to NMF.

5.5. Impact of Epochs:

The trends observed in training time and error rates persist with varying numbers of epochs (training iterations). NMF maintains its training time advantage across all epoch configurations, while co-clustering continues to achieve lower error rates.

5.6. Overall Findings:

The NMF-based model offers a clear advantage in training efficiency, significantly reducing training time compared to benchmark methods, particularly SVDPP. Co-clustering demonstrates superior accuracy in predicting user preferences, as evidenced by its consistently lower error rates. A more in-depth analysis is necessary to precisely quantify the trade-off between training time and accuracy for co-clustering compared to NMF.

5.7. Future Research Directions:

Further investigation is needed to explore potential optimizations for co-clustering that could improve its training efficiency while maintaining its accuracy advantage. Additionally, research could delve into hybrid approaches that combine the strengths of NMF and co-clustering to potentially achieve both efficient training and high accuracy.

6. CONCLUSION

This study explored the effectiveness of four collaborative filtering algorithms (SVD, SVDPP, co-clustering, and NMF) for a job recommender system. Our evaluation focused on three key metrics: error rate, training time, and cross-validation performance. The results revealed a trade-off between accuracy and efficiency. Co-clustering is the most accurate but takes the longest to set up. This means it recommends the best jobs but takes more time to get started. NMF is the fastest to set up but isn't quite as accurate. This makes it a good choice if you need recommendations quickly, even if they aren't perfect. The best method depends on what you need most. If accuracy is crucial, go with co-clustering. If speed is more important, NMF is a good option. The researchers plan to explore ways to improve NMF's accuracy while keeping it fast. Future research will delve deeper into the error-efficiency trade-off and explore techniques to enhance NMF's accuracy while preserving its efficiency advantage.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Syst.*, 2013, doi: 10.1016/j.knosys.2013.03.012.

[2] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, 2015, doi: 10.1016/j.dss.2015.03.008.

[3] Z. Huang, W. Chung, and H. Chen, "A graph model for e-commerce recommender systems," *J. Am. Soc. Inf. Sci. Technol.*, 2004, doi: 10.1002/asi.10372.

[4] A. Moreno, A. Valls, D. Isern, L. Marin, and J. Borràs, "SigTur/E-Destination: Ontology-based personalized recommendation of Tourism and Leisure Activities," *Eng. Appl. Artif. Intell.*, 2013, doi: 10.1016/j.engappai.2012.02.014.

[5] B. Krulwich, "Lifestyle finder: Intelligent user profiling using large-scale demographic data," *AI Mag.*, 1997.

[6] H. Papadakis, A. Papagrigoriou, C. Panagiotakis, E. Kosmas, and P. Fragopoulou, "Collaborative filtering recommender systems taxonomy," *Knowl. Inf. Syst.*, 2022, doi: 10.1007/s10115-021-01628-7.

[7] M. Nilashi, K. Bagherifard, O. Ibrahim, H. Alizadeh, L. A. Nojeem, and N. Roozegar, "Collaborative filtering recommender systems," *Res. J. Appl. Sci. Eng. Technol.*, 2013, doi: 10.19026/rjaset.5.4644.

[8] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Collaborative filtering recommender system," 2019. doi: 10.1007/978-3-030-11928-7_30.

[9] U. Javed, K. Shaukat, I. A. Hameed, F. Iqbal, T. M. Alam, and S. Luo, "A Review of Content-Based and Context-Based Recommendation Systems," *Int. J. Emerg. Technol. Learn.*, 2021, doi: 10.3991/ijet.v16i03.18851.

[10] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," in *Recommender Systems Handbook*, 2011. doi: 10.1007/978-0-387-85820-3_3.

[11] J. Zhao, F. Zhuang, X. Ao, Q. He, H. Jiang, and L. Ma, "Survey of Collaborative Filtering Recommender Systems," *Journal of Cyber Security*. 2021. doi: 10.19363/J.cnki.cn10-1380/tn.2021.09.02.

[12] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intelligent Data Analysis*. 2017. doi: 10.3233/IDA-163209.

[13] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A Survey on Session-based Recommender Systems," *ACM Comput. Surv.*, 2022, doi: 10.1145/3465401.

[14] Y. Ma, B. M. Narayanaswamy, H. Lin, and H. Ding, "Temporal-Contextual Recommendation in Real-Time," 2020. doi: 10.1145/3394486.3403278.

[15] B. zs Hidasi and A. Karatzoglou, "Recurrent neural networks with Top-k gains for session-based recommendations," 2018. doi: 10.1145/3269206.3271761.

[16] W. C. Kang and J. McAuley, “Self-Attentive Sequential Recommendation,” 2018. doi: 10.1109/ICDM.2018.00035.

[17] L. M. De Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, “Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks,” *Int. J. Approx. Reason.*, 2010, doi: 10.1016/j.ijar.2010.04.001.

[18] S. H. Choi, Y. S. Jeong, and M. K. Jeong, “A hybrid recommendation method with reduced data for large-scale application,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, 2010, doi: 10.1109/TSMCC.2010.2046036.

[19] I. Christensen and S. Schiaffino, “A hybrid approach for group profiling in recommender systems,” *J. Univers. Comput. Sci.*, 2014.

[20] L. Liu, N. Mehandjiev, and D. L. Xu, “Multi-criteria service recommendation based on user criteria preferences,” 2011. doi: 10.1145/2043932.2043950.

[21] L. H. Son, “HU-FCF: A hybrid user-based fuzzy collaborative filtering method in Recommender Systems,” *Expert Syst. Appl.*, 2014, doi: 10.1016/j.eswa.2014.05.001.

[22] R. Mishra and S. Rathi, “Efficient and Scalable Job Recommender System Using Collaborative Filtering,” 2020. doi: 10.1007/978-981-15-1420-3_91.

[23] P. Banerjee and R. Gupta, “Talent Attraction through Online Recruitment Websites: Application of Web 2.0 Technologies,” *Australas. J. Inf. Syst.*, 2019, doi: 10.3127/ajis.v23i0.1762.

[24] E. Parry and S. Tyson, “An analysis of the use and success of online recruitment methods in the UK,” *Hum. Resour. Manag. J.*, 2008, doi: 10.1111/j.1748-8583.2008.00070.x.

[25] F. Färber, T. Weitzel, and T. Keim, “An Automated Recommandation Approach to Selection in Personnel Recruitment,” *Amcis*, 2003.

[26] F. Ramadhani and M. Zarlis, “Analysis of e-Recruitment System Design,” *Int. J. e-Education, e-Business, e-Management e-Learning*, 2019, doi: 10.17706/ijeeee.2019.9.1.38-45.

[27] K. Ghazzawi and A. Accoumeh, “Critical Success Factors of the E-Recruitment System,” *J. Hum. Resour. Manag. Labor Stud.*, 2014.

[28] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung, “Melu: Meta-learned user preference estimator for cold-start recommendation,” 2019. doi: 10.1145/3292500.3330859.

[29] A. M. Rashid *et al.*, “Getting to know you: Learning new user preferences in recommender systems,” 2002.

[30] E. Faliagka *et al.*, “On-line consistent ranking on e-recruitment: Seeking the truth behind a well-formed CV,” *Artif. Intell. Rev.*, 2014, doi: 10.1007/s10462-013-9414-y.

[31] J. Malinowski, O. Wendt, T. Keim, and T. Weitzel, “Matching people and jobs: A bilateral recommendation approach,” 2006. doi: 10.1109/HICSS.2006.266.

[32] F. Gutiérrez, S. Charleer, R. De Croon, N. N. Htun, G. Goetschalckx, and K. Verbert, “Explaining and exploring job recommendations: A user-driven approach for interacting with knowledge-based job recommender systems,” 2019. doi: 10.1145/3298689.3347001.

[33] A. Mogenet, T. A. N. Pham, M. Kazama, and J. Kong, “Predicting online performance of job recommender systems with ofline evaluation,” 2019. doi: 10.1145/3298689.3347032.

[34] Y. Charband and N. Jafari Navimipour, “Online knowledge sharing mechanisms: a systematic review of the state of the art literature and recommendations for future research,” *Inf. Syst. Front.*, 2016, doi: 10.1007/s10796-016-9628-z.

[35] G. Liu *et al.*, “TOSI: A trust-oriented social influence evaluation method in contextual social networks,” *Neurocomputing*, 2016, doi: 10.1016/j.neucom.2015.11.129.

[36] H. Li, R. Hong, S. Zhu, and Y. Ge, “Point-of-interest recommender systems: A separate-space perspective,” 2016. doi: 10.1109/ICDM.2015.27.

[37] V. Gatteschi, F. Lamberti, A. Sanna, and C. Demartini, “A Semantic Matchmaking System For Job Recruitment,” 2010.

[38] M. Fazel-Zarandi and M. S. Fox, “Semantic matchmaking for job recruitment: An ontology-based hybrid approach,” 2009.

[39] J. Malinowski, T. Weitzel, and T. Keim, “Decision support for team staffing: An automated relational recommendation approach,” *Decis. Support Syst.*, 2008, doi: 10.1016/j.dss.2007.05.005.

[40] J. Martinez-Gil, “An overview of knowledge management techniques for e-recruitment,” *Journal of Information and Knowledge Management*. 2014. doi: 10.1142/S0219649214500142.

[41] I. Lee, “An architecture for a next-generation holistic e-recruiting system,” *Commun. ACM*, 2007, doi: 10.1145/1272516.1272518.

[42] M. Maree, A. B. Kmail, and M. Belkhatir, “Analysis and shortcomings of e-recruitment systems: Towards a semantics-based approach addressing knowledge incompleteness and limited domain coverage,” *J. Inf. Sci.*, 2019, doi: 10.1177/0165551518811449.

[43] T. Sekiguchi, “Person-Organization fit and Person-Job fit in employee selection: A review of the literature,” *Osaka Keidai Ronshu*, 2004.

[44] H. Yu, C. Liu, and F. Zhang, “Reciprocal recommendation algorithm for the field of recruitment,” *J. Inf. Comput. Sci.*, 2011.

[45] R. Rafter and B. Smyth, “Passive Profiling from Server Logs in an Online Recruitment Environment,” 2001.

[46] S. Guo, F. Alamudun, and T. Hammond, “RésuMatcher: A personalized résumé-job matching system,” *Expert Syst. Appl.*, 2016, doi: 10.1016/j.eswa.2016.04.013.