2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM) | 978-1-6654-2678-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/IMCOM53663.2022.9721629

# An Efficient Algorithm for Ranking Candidates in E-Recruitment System

Abdul Hanan Minhas, Mohammad Daniyal Shaiq, Saad Ali Qureshi, Musa Dildar Ahmed Cheema,
Shujaat Hussain, Kifayat Ullah Khan
*Knowledge Discovery & Data Science (KDD) Lab*
*Department of Computer Science*
*National University of Computer and Emerging Sciences*
Islamabad, Pakistan
i202062@nu.edu.pk, i191794@nu.edu.pk, i181288@nu.edu.pk, i191765@nu.edu.pk,
shujaat.hussain@nu.edu.pk, kifayat.alizai@nu.edu.pk

***Abstract*— Over the last decade, the growth of e-recruitment has resulted in the expansion of web channels dedicated to candidate recruitment, making it easy to find and apply for jobs. However, as a result, today's human resource managers are inundated with applications for each job opening. This leads to the production of significant number of documents, referred to as resumes or curriculum vitae (CV). Optimal processing of this data is necessary from a Human Resource strategic and economic perspective, where cost and time effectiveness is paramount. We propose an efficient ranking algorithm to overcome the high time and cost complexity associated with the pairwise comparison of candidates in the state-of-the-art Multi-Criteria Decision Making (MCDM) based ranking algorithm. This algorithm is integrated with matrix sorting and pruning based solution to enhance its scalability. Our proposed algorithm was tested on three different datasets: real-world recruitment, simulated DBLP, and synthetic datasets. Our algorithm shows promising results, which makes it effective and efficient on real-world resume ranking processes.**

***Index Terms*—Resume, curriculum vitae, recommendation system, e-recruitment, ranking, Multi-Criteria Decision Making, Analytic Hierarchical Process**

## I. Introduction

In the modern world, the internet is the biggest platform being used for hiring the right candidate for a particular job. Multiple companies are now using web resources to better utilize the available data for recruiting the suitable candidates. The process to hire a potential candidate for a vacant post utilizing electronic means is referred to as e-recruitment or online recruiting [1]. This automated system covers the full recruiting cycle, including screening, sourcing, and employing candidates that meet the job requirements in a more effective and efficient manner [2]. Although this automated system has had a significant positive effect on electronic Human Resource Management (HRM), however, with the advent of globalization, the barrier to access global job advertisements has decreased substantially. Hence, a single job post from a reputable organization, might receive a large influx of resumes from the interested applicants. Moreover, the differences between any two consecutive resumes can be marginal. Therefore, finding the best-suited candidate against a particular job post has become more difficult than ever, for proficiency-intensive organizations [3]. This leads towards the need of an efficient e-recruitment solution.

For this purpose, new tools are developed for e-recruitment system to help identify professionals with respects to job-posts, while also decreasing recruitment time and cost. Moreover, recommending most relevant candidates at the top-N list is another major challenge. Studies reveal that automating the recruitment process can have a significant impact over the employment industry [4]. One study observed a 44% decrease in operational costs as well as a drop-in average time needed to fill a vacancy from 70 to 37 days [4].

To develop an e-recruitment ranking system, the fundamental step is to match every user profile or CV with the job description. However, this step is primarily suffered with high matching complexity in past research. In recent years, there have been many techniques deployed in creating most efficient e-recruitment systems to match the best candidates for a specific job. Recently, L2R-based methodologies have produced state-of-the-art results in relevant information retrieval [5]. These models use semi or supervised machine learning to construct and optimize ranking models for information retrieval [6]. Its training data has some partial ranked lists specified for items which it uses to re-rank the new items. These L2R-based models assume that the underlying dataset preserve the semantic information. However, this assumption doesn't translate well into the real-world scenarios, where a single entity might be represented by a multitude of equally valid replacements; hence, leading towards increased ambiguity. To overcome the problem of ambiguity among entities, Semantic-based ranking [7] is employed; however, these modules themselves suffer from domain incompleteness [8].

Another very-well used method in e-recruitment for ranking is MCDM [9]. The state-of-the-art MCDM technique includes the Analytic Hierarchy Process (AHP). The methods of rating alternatives and aggregating to find the most relevant candidates are combined in this technique. The technique is used to rank a set of alternatives or to choose the best from a set of alternatives. The ranking is based on a broad objective that

*Funded by Higher Education Commission (HEC) Pakistan and Ministry of Planning, Development and Special Initiatives under National Center of Artificial Intelligence*

is broken down into a collection of criteria [10]. AHP suffers from the time and cost complexity due to the generation of separate pairwise comparison matrix for every feature [11]. Another shortcoming of AHP includes human intervention in intermediate steps of the algorithm [12]. Fig. 1 depicts the hierarchy structure of AHP for e-recruitment process.

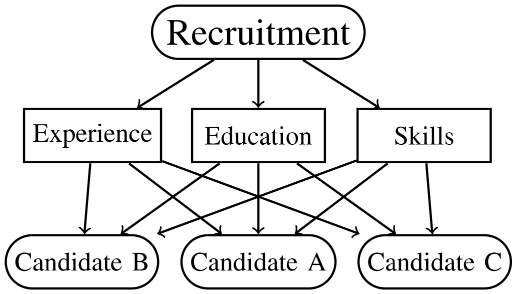


Fig. 1: AHP Hierarchy for E-Recruitment Process

This study presents an efficient ranking algorithm using the principles of MCDM methods. It is a problem-solving method that considers multiple criteria, simultaneously. In the current case, it awards weights according to the job requirements. The objective is to obtain a list of most relevant candidate with respect to each job description. Our focus is to optimize the matrix complexity issues of AHP algorithm using matrix sorting and pruning based techniques. Furthermore, our proposed ranking algorithm uses the dynamic scaling to overcome the human intervention without affecting the efficiency of ranking process.

The following sections are arranged as follows; next section presents the related work on ranking in e-recruitment system. Section III explains the methodology of our proposed ranking algorithm. In section IV, we report the experiments and results, and section V concludes the work with the summary of our findings.

## II. Related Work

In the past few years, e-recruitment played its major role to minimize the burden of the hiring process while compromising on the end results. To further enhance the systems, researchers are trying to improve the efficiency of the ranking module of e-recruitment systems through multiple techniques and algorithms. Research in ranking methodologies can be divided into three main approaches. These approaches include Learning to-rank (L2R), semantic based, and MCDM.

### A. *Learning-to-Rank (L2R)*

Recently, L2R-based methodologies have produced state-of-the-art results in data retrieval [13]. Furthermore, L2R-based models can be divided into three main approaches; i.e. Pairwise, Point-wise and List-wise. Kouadria et al. [14] in their study employed a rank aggregation approach by merging complementary L2R models into a single hybrid model. The authors overcame the drawback of aggregation, which was known to suffer from overlooking important details, via generation of partially ranked lists. However, each base-learner has an equal representation towards the final score, hence, a relevant base learner might be overlooked.

Similarly, Braun et al. [13] presented a comparison between the different L2R approaches. They recommend an efficient process for automating the ranking of CVs corresponding to job vacancies. Furthermore, they compared four algorithms, each taken from one type of approach employed in L2R methods, namely; Gradient Boosted Regression Trees (GBRT), a Lambda MART, Smooth Rank model, and the Best Single Feature (BSF) model. Although, the results of their work showed the best among the 4 algorithm, the results were biased towards two recommended algorithms (GBRT and Lambda Mart) due to the limited amount of training and testing data. Bruch et al. [15] argued that the nature of L2R models that employ machine learning is covetous and compromises the robustness of models. The objective of their approach was to generate robust models through better optimization of loss function. Their approach focused on concrete distribution for the scores and planned on working with other distributions, while also adhering to the effectiveness-efficiency trade-off.

The work proposed by Shafiq et al. [16] presented a novel L2R methodology that applied data mining techniques to extract personality traits of the users before applying ranking itself. Support Vector Machine (SVM) was used to classify the personalities of the candidates. However, their proposed ranking approach only utilized a single criteria, which introduce biasness in the ranked list. Furthermore, the CV extraction process required users to enter their data and information in a web form. The L2R based models employed in the above research only tried to enhance the ranking of documents by either increasing the robustness of the models or by incorporating more features in the training set. Moreover, L2R-based models assumed that the underlying dataset preserves the semantic information; however, this assumption doesn't translate to the real-world scenarios, where a single entity might be represented by multiple valid replacements; hence, leading towards increased ambiguity.

### B. *Ontology Based Ranking*

Semantic methodologies employ graph-based representation to overcome the problem of equally valid entities. Hence, semantic similarity between alternative entities can be calculated using ontology mappings. In the work by Mohamed et al. [17], the authors proposed a ranking system using semantics by applying resume ontology to check the similarity between the resumes and the job description. The evaluation of the system ranking in comparison to the manual ranking resulted in accurate recommendations of candidates, however, the use of ontology itself increased the complexity of the whole system. Moreover, the addition of new criteria required building of the whole ontology again from the start.

Researchers in another study [18], presented a method for sorting resumes that relied solely on resumes rather than job descriptions. They attempted to address the issue of ranking when no dataset is provided. Unlike existing state-of-the-art algorithms, their suggested approach does not use job offers or any other semantic resource to rank resumes. Inter-

Resume Proximity (IPR) is the term they use to describe their proposed solution. The language resemblance between the resumes provided in response to the same job posting is referred to as IPR. The non-relevance of data with respect to numerous job descriptions is the research's shortcoming.

### C. Multi-Criteria Decision Making (MCDM)

Another widely employed method for ranking is MCDM. It is a sub-discipline of operations research that looks at how to make decisions based on multiple conflicting criteria [19]. Faliagka et al. [11] was the first work to propose a ranking method using MCDM. They used AHP method to develop an automated system for e-recruitment where people's web presence and technological advances of the time could be used to achieve that goal. The work mapped users CVs as HR-XML input to the AHP model to generate a rank for the candidates. It evaluated the candidates based on personality, education, skills, experience, and loyalty. The work generated accurate rankings, but the limitation of the whole process depended on a predefined format of input through which the AHP method takes features and candidates data. Another problem of AHP approach is the pairwise comparison matrix generation of candidates for every feature. Furthermore, this approach generated a new matrix for every feature, which make this approach unscalable.

In another research proposed by Faliagka et al. [10], AHP was applied to an integrated e-recruitment system for automated personality mining and applicant ranking. This work brought innovation in the previous work by adding an efficient version of personality mining from the internet. The study showed promising results, however, the system was unable to map the domain specific work experience to the requirement of job description. Nawzad et al. [12] is one of the latest work that proposed an MCDM method in e-recruitment using AHP. The work used the original working of AHP method to propose the best CV among three. The work claimed that the proposed system would generate efficient ranking for CVs even more than 100 but the increase in complexity of the pairwise comparisons refutes this claim. Another work by [20] applied The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) MCDM to rank 20 candidates for personnel selection in the domain of logistics. The work showed that MCDM methods are much more feasible for the recruiters in personnel selection than most methods but the work itself has some limitations.

Due to the generation of a separate pairwise comparison matrix for each feature, AHP method has high algorithm complexity. Furthermore, it faces the problem of human intervention or lack of automaticity, which sometimes generate biasness in the final ranked candidates. Also, it has the problem of manual scaling, which add the biasness in the ranking process. Moreover, these techniques could not provide any novel approach in the existing methods regarding research. To conclude, these approaches are independent and are capable to work in different scenarios, however, amendments in its framework can produce promising ranking solution. In this paper, we will optimize AHP algorithm by overcoming the issues mentioned above. Next section describes the overall methodology of our proposed solution.

## III. SLASHRANK, THE PROPOSED ALGORITHM

For e-recruitment system, specifically for ranking candidates based on job posts, we propose the algorithm named SlashRank. SlashRank belongs to MCDM which is an operation research sub-discipline that examines multiple conflicting criteria in decision-making [19]. The objective of SlashRank algorithm is to automate the process of ranking by reducing human intervention, and optimizing the algorithm concerning time and space complexity. Furthermore, this proposed algorithm aims to solve the issues observed in AHP based MCDM technique as discussed in Section II. Fig. 2 represents the overall pipeline of proposed SlashRank algorithm.

In further sections, the framework of proposed ranking algorithm is explained in detail.

### A. Input

SlashRank algorithm leverages the structured format of both input documents that is job descriptions and resumes. Moreover, algorithm also requires an explicit input feature priority vector, delineating the priority for each input feature. Proposed algorithm undertakes job description covering all the requirements of the recruiter regarding a job post such as education, experience, and skills etc., whereas the resumes consist of the personal information, academic history, experience, and skills of a candidate. Furthermore, the feature priority vector contains the importance of the features like experience, education, and skills normalized between 0 and 1. The proposed ranking solution uses these inputs to rank the candidates for a specific job position.

### B. Data Preprocessing

To rank the candidates, we generated the score of the candidates by comparing their feature values with the job description. The values of the dataset attributes are in the form of string for instance education, skills, and location. As a result, the string-matching component induced high complexity in our algorithm. To overcome the problem of time complexity, we did some preprocessing in the dataset. As the complexity of string matching is high, we mapped the strings to unique numeric entities. This step optimized our algorithm with respect to the time complexity.

### C. Dynamic Scale Generation

In the SlashRank algorithm, a dynamic scale is generated for every feature based on the job description input, to compute each candidate's score for ranking purposes. The highest value on the scale maps to the value of the feature in the job description. Afterwards, the scale declines based on descending order of the specific feature's values. For example, if the requirement of the job post for the education feature is Master's (MS) degree, then our algorithm generate the scale from MS to the lowest level of education such as Bachelor's (BS) degree in a

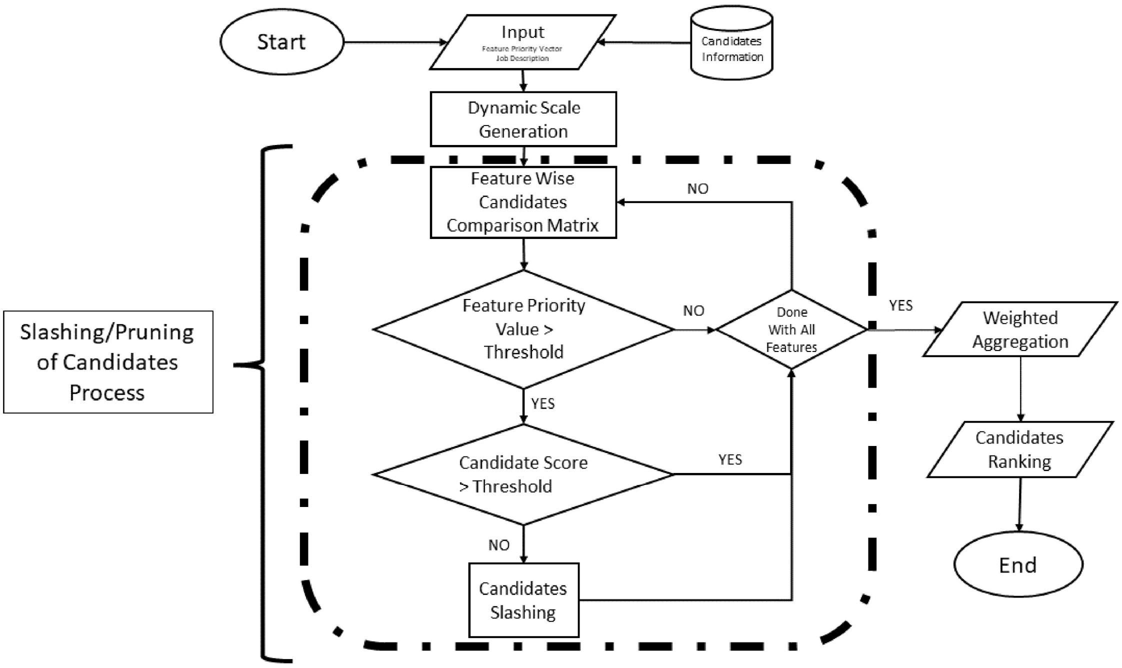


Fig. 2: Flowchart of Slash Rank

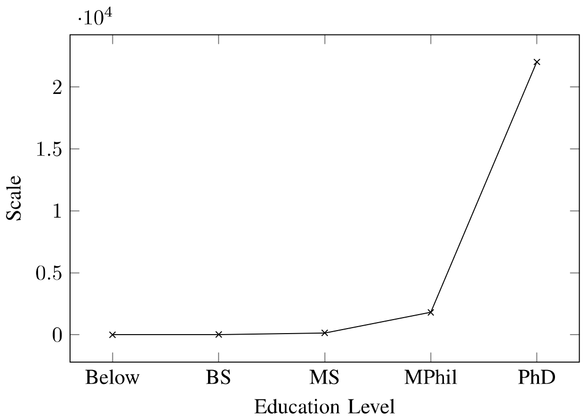


Fig. 3: Scale for education having requirement of PhD in the job description

descending manner. Any candidate who has a degree of 'MS' or above gets the maximum value on the scale and others get the lower values according to their degree order in the scale. Similarly, this method is adopted for all other features. Fig. 3 depicts the scale for education, having requirement of PHD in the job description.

### *D. Candidate Comparison*

This section explains the process of candidate's comparison with respect to every feature. Initially, algorithm compares all the candidates based on their value for the highest priority feature. For example, if the recruiter gives the highest priority to the education feature for the job position, then all the candidates are compared with each other based on their respective

TABLE I: Case study on candidate comparison process

| **Education** | **C1** | **C2** | **C3** | **C4** |
|---|---|---|---|---|
| C1 | 1 | 10 | 10 | 10 |
| C2 | 0.1 | 1 | 0.1 | 6.66 |
| C3 | 10 | 10 | 1 | 10 |
| C4 | 0.1 | 0.15 | 0.1 | 1 |

Normalization

| **Education** | **C1** | **C2** | **C3** | **C4** | **Eigen Vector** |
|---|---|---|---|---|---|
| C1 | 0.08 | 0.47 | 0.89 | 0.36 | 0.45 |
| C2 | 0.008 | 0.04 | 0.008 | 0.28 | 0.07 |
| C3 | 0.89 | 47 | 0.08 | 0.36 | 0.45 |
| C4 | 0.008 | 0.007 | 0.008 | 0.04 | 0.015 |

Sorting

| **Education** |
|---|
| C1 = 0.45 |
| C3 = 0.45 |
| C2 = 0.07 |
| C4 = 0.02 |

Pruning

| **Experience** | **C1** | **C3** | **C2** |
|---|---|---|---|
| C1 | 1 | 0.13 | 10 |
| C3 | 10 | 10 | 1 |
| C2 | 7.5 | 1 | 0.1 |

score for the education first. This score is originated from the scale generated in the Section III-C. Afterwards, candidate comparison matrix is normalized based on the generated scores. In the next step, all the candidates are sorted based on their score for the highest priority feature. Furthermore, the step of slashing is integrated to determine whether a candidate qualify for the next step specially generation of its score with respect to the next most important feature. This process is repeated for all the features used for the ranking purpose, for example education, experience, and skills. Table I explains the case study on how the process of candidates' comparison works in our proposed algorithm.

**Algorithm 1 Pseudocode of Pruning** ($C_{m,n}$ ,$P_n$ ,$T_p$, $T_s$)

**Input:**
$C_{m,n}$ : Candidate's data, where m and n are number of candidates and features respectively
$P_n$ : A feature priority vector having the importance of all the features
$T_p$ : Threshold for feature priority
$T_s$ : Threshold for the candidates feature score

```
IF Pn > Tp
    FOR EACH candidate ∈ Cm,n DO
        IF candidate score > Ts
            Prune candidate
        ELSE
            Pass
ELSE
    ITERATE next feature
```

**Output:**
$R_c$ : Ranked candidates list

### E. Slashing/Pruning of Candidates

The process of pruning to reduce the complexity of candidate's comparison with respect to each feature is explained in this section. In this step, the algorithm checks if the feature has a priority value greater than a given threshold. In case it's greater than a threshold, the algorithm slashes or prunes the candidates based on the threshold applied to their feature score. Afterwards, the same steps are repeated for the remaining features, however, with each passing feature, the number of candidates are reduced according to the distribution of the data. In return, it reduces the matrix size for the next feature which leads towards complexity optimization.

Algorithm 1 explains the pruning step in SlashRank algorithm along with the pseudocode to better understand its functionality.

### F. Global Summation

Once the candidate's comparison step is over, the scores of all the candidates who survived the process of slashing are compiled against all the features. Afterwards, the algorithm takes the product of the candidate feature score $V_f$ with its respective feature priority score $P_f$, thus calculating the final score $S_c$ for each candidate. Moreover, the final score is sorted, and algorithm output the list of candidates ranked from highest to lowest in relevance with the job post.

$$S_C = \sum_{f=1}^{n} (V_f X P_f) \quad (1)$$

Where S is score of a candidate, n is total number of features involved in ranking, V is score of a candidate for feature f and P is the priority of a feature f.

## IV. Experiments

The major experimental results of our suggested approach, SlashRank, are presented in this section. We assessed the performance of our algorithm on three different datasets with respect to effectiveness and efficiency of the algorithm. All our algorithms and experimental techniques are written in Python 3.8 and run on a Windows PC with a 3.6GHz Intel i9 10th gen 10850K 10 cores CPU and 128GB of RAM.

### A. Experimental Datasets

We performed the experimental studies on three different datasets: Real World Recruitment[1], DBLP[2], and synthetic[3] datasets. The details of these datasets are mentioned in the below sections.

TABLE II: Datasets Details

| Datasets | Records | Features |
|---|---|---|
| Real-World Recruitment | 96 | Experience, skills, education |
| DBLP | 1700000 | Experience, number of papers, subjects |
| Synthetic | 10000 | Experience, skills, education |

*1) Real World Recruitment Dataset:* This was a local real-world recruitment dataset. This dataset includes 9 real-world recruitment processes having job descriptions and the resumes information against them in a structured format. These recruitment processes belong to multiple domains especially accounting, HR, information technology etc. Table III refers to the job titles along with the number of candidates applied for specific post.

*2) DBLP Dataset (To Simulate E-Recruitment Process):* The DBLP Bibliography data downloaded in June 2021 is included in this data set. The DBLP computer science bibliography is a free online database of key computer science journals and proceedings. There are four categories of information for each author: name, experience, area, and conference name. Area specifies the domain of author's specialization. Experience is the total number of years the author spends in the publication field. This dataset was used for the evaluation of our proposed ranking algorithm by simulating e-recruitment ranking process.

[1]local dataset
[2]https://www.kaggle.com/daniyalshaiq/dblp-authors-dataset
[3]https://www.kaggle.com/daniyalshaiq/synthetic-erecruitment-user-dataset

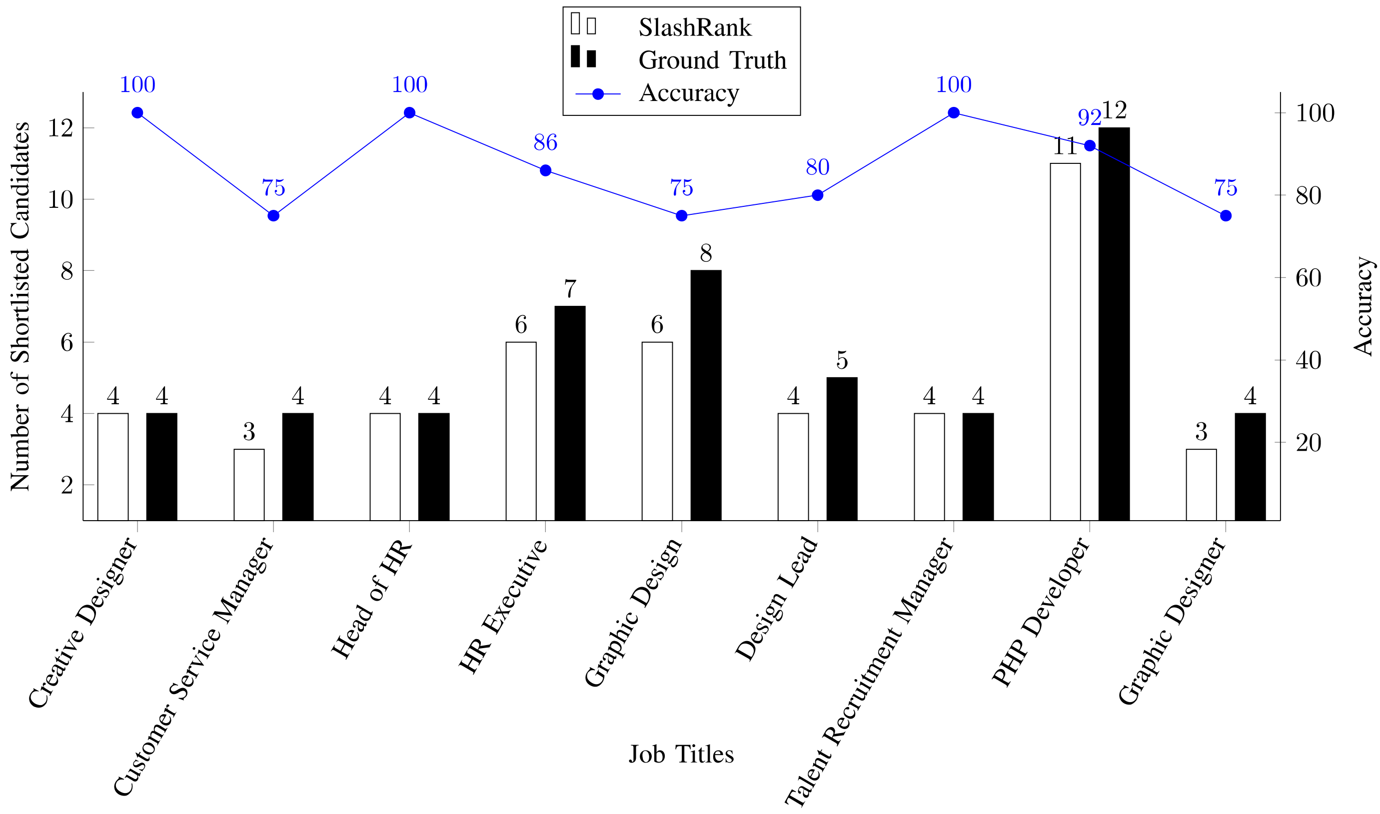


Fig. 4: Evaluation of SlashRank on Real-World Recruitment Dataset

TABLE III: Real World Dataset Details

| **Job Title** | **Number of Candidates** |
|---|---|
| Creative Designer | 5 |
| Customer Success Manager | 10 |
| Head of HR | 6 |
| HR Executive | 15 |
| Graphic Design | 14 |
| Design Lead | 15 |
| Talent Recruitment Manager | 6 |
| PHP Developer | 15 |
| Graphic Designer | 10 |

*3) Synthetic Dataset:* Third dataset consists of a corpus of 10000 resumes. We synthesize this dataset to test our ranking solution. We built this dataset by following the industrial data model for shortlisting the candidates. We also published this dataset on Kaggle for the purpose of reproducibility for the future research related to the ranking in e-recruitment. The attributes of this dataset includes education, experience, and skills as they covers all the information needed to evaluate the candidate against a specific job post.

### *B. Effectiveness Evaluation*

On the above-mentioned datasets, we first assess the effectiveness of our proposed ranking algorithm SlashRank as a powerful e-recruitment decision-support tool. In this section, we present some interesting findings by performing experiments on all three datasets.

*1) Real World Dataset:* First experiment was performed on the real-world recruitment dataset. A total of 9 job descriptions were used for the testing of our proposed ranking solution. Fig. 4 shows the overall performance of our algorithm on the multi-domain real-world job posts.

It is evident from Fig. 4 that for most of the job descriptions, the ground truth shortlisted candidates were included in top-k ranked list generated by our algorithm. For instance, all 4 shortlisted candidates for the post of creative designer were ranked at top 4 positions by our algorithm. However, in some cases, our algorithm didn't perform well due to multiple limitations associated with this real-world dataset. These limitations include biasness due to human intervention in the process of shortlisting candidates, and other external factors involved in the process of real-world recruitment.

*2) DBLP Dataset:* For the purpose of effectiveness evaluation, we performed another experiment on the DBLP dataset,

| Author_ID | author | experience | num_of_papers | Publication_Venue | subjects | Scores |
|---|---|---|---|---|---|---|
| 504 | Brendan O'Flynn | 16 | 60 | ['SENSYS', 'ECCTD', 'IPSN', 'LCN', 'ISCAS', 'B... | ['Computer Hardware†', 'Other', 'Information S... | 15.0962 |
| 735 | Claude D'Amours | 20 | 51 | ['ITA', 'PIMRC', 'WCNC', 'EUSIPCO', 'CCECE', '... | ['Information Systems†', 'Computer Hardware†',... | 15.0954 |
| 167 | Alessandro D'Alconzo | 12 | 30 | ['CONEXT', 'GLOBECOM', 'VTC', 'NETWORKING', 'I... | ['Other', 'Rank: B', 'Automotive Engineering†'... | 9.2361 |
| 359 | Antonio A. D'Amico | 12 | 16 | ['ICC', 'MCSS', 'VTC', 'MOBILIGHT', 'CAMSAP', ... | ['Other', 'Automotive Engineering†', 'Rank: B'] | 7.9407 |
| 891 | Daniel O'Neill | 8 | 18 | ['GLOBECOM', 'VTC', 'INFOCOM', 'ICC', 'WCNC', ... | ['Rank: B', 'Automotive Engineering†', 'Distri... | 7.1052 |
| 138 | Ala'a Al-Momani | 4 | 8 | ['VNC', 'VTC', 'SP', 'IVS'] | ['Other', 'Automotive Engineering†', 'Computer... | 5.8349 |
| 705 | Chuan'ao Jiang | 4 | 6 | ['VTC', 'APNOMS', 'SINC', 'GLOBECOM', 'IWCMC'] | ['Automotive Engineering†', 'Distributed Compu... | 5.7201 |
| 67 | Adaildo G. D'Assunção | 4 | 1 | ['VTC'] | ['Automotive Engineering†'] | 5.6805 |
| 54 | Abdullahi Abubakar Mas'ud | 4 | 1 | ['VTC'] | ['Automotive Engineering†'] | 5.6805 |
| 194 | Alexander O'Neil | 4 | 1 | ['VTC'] | ['Automotive Engineering†'] | 5.6805 |
| 147 | Ala'eddin Masadeh | 4 | 5 | ['VTC', 'GLOBECOM', 'ICC', 'ICCCN', 'WCSP'] | ['Automotive Engineering†', 'Rank: B', 'Other'... | 5.6750 |
| 768 | Colman O'Sullivan | 4 | 2 | ['VTC'] | ['Automotive Engineering†'] | 5.6335 |

Fig. 5: Candidates Ranked List (DBLP Dataset)

which was simulated for e-recruitment ranking process. We randomly took a row from the dataset and declared it as a job description (JD). Algorithm ranked all the other data points with respect to that target JD. Table IV represents the target JD from DBLP dataset.

TABLE IV: Job Description (DBLP)

| Experience | No. of Papers | Publication Venue | Subjects |
|---|---|---|---|
| 4 | 2 | ISCAS | Other |

After taking job description as an input, our algorithm performed the ranking process according to the details mentioned in the methodology section. Afterwards, SlashRank generated the output in the form of list of ranked candidates based on the similarity of all the other data points with respect to the target data point as job description.

From Fig. 5, it can be depicted that our algorithm ranked those authors at top of the ranking list who were closer to the target JD in terms of feature values. For instance, the experience of Author_ID 735 is 20 years, however it is ranked below the Author_ID 504 with an experience of 16 years. This is due to the influence of other features like number of papers, which has highest priority as compared to the experience feature. Our algorithm trade off the ranking based on the feature priorities. Author_ID 504 fulfilled the requirements of rest of the features whose priorities are higher as compared to the number of papers feature. Therefore, his score is higher than Author_ID 735. That's how our algorithm assign scores to all the candidates and at the end ranked them based on these scores.

*3) Synthetic Dataset:* For testing on the synthetic dataset, the data point referred in table V was randomly picked as a target job description. This job description was used to rank all of the other candidates.

| Resume_ID | education | experience | skills | scores |
|---|---|---|---|---|
| 837 | PHD | 9 | [Project Management, Unit testing, JavaScript,... | 5.9844 |
| 492 | PHD | 9 | [Adobe PhotoShop, Spring boot, AWS, JavaScript... | 5.9844 |
| 596 | MPHIL | 10 | [NoSQl, jquery, Node js, Project Management, J... | 5.2654 |
| 350 | PHD | 8 | [jquery, JavaScript, Adobe illustrator, Unit t... | 5.1913 |
| 786 | PHD | 6 | [flutter, Software Design Cycle, jquery, Java,... | 4.4599 |
| 122 | MS | 10 | [Software Design Cycle, NoSQL, JavaScript, jqu... | 4.4247 |
| 768 | PHD | 5 | [jquery, tensorflow, AZURE, JavaScript, NoSQl,... | 4.2621 |
| 508 | MPHIL | 9 | [JavaScript, tensorflow, Unit testing, Project... | 4.1137 |
| 479 | MPHIL | 9 | [Node js, JavaScript, jquery, SQL, NoSQL, Adob... | 4.1137 |
| 491 | PHD | 4 | [OLAP, AZURE, Spring boot, JavaScript, jquery,... | 4.0712 |
| 301 | PHD | 3 | [Project Management, SQL, JavaScript, jquery, ... | 3.8864 |
| 52 | PHD | 3 | [jquery, JavaScript] | 3.8864 |
| 78 | PHD | 2 | [jquery, JavaScript] | 3.7806 |
| 880 | PHD | 2 | [jquery, JavaScript, Java, Software Design Cyc... | 3.7806 |

Fig. 6: Candidates Ranked List (Synthetic Dataset)

TABLE V: Job Description (Synthetic)

| Education | Experience | Skills |
|---|---|---|
| PHD | 10 | Jquery, JavaScript |

Fig. 6 shows the ranked candidates list with respect to the job post. It is evident from the results that algorithm performed well by ranking those candidates up who were close to the target job description. Some of the authors for instance ID 596 and ID 122 in the ranked list have experience of 10 years as required by the JD, however, they don't meet the PhD requirement of the JD. Therefore they ranked down as compared to other candidates.

### C. Efficiency Evaluation

We examine the efficiency of our proposed ranking algorithm in this section. We tested our algorithm in different scenarios and compared it efficacy with the Analytical Hierarchy Process (AHP). We compared the time taken by our algorithm with the AHP algorithm for each job description in a logarithmic scale on DBLP dataset.

In Fig. 7, it is evident that SlashRank performed well with respect to time taken for ranking candidates against each JD. Also, our proposed algorithm is scalable due to the pruning step based on the candidate scores and feature priorities. Furthermore, the proposed algorithm is proved to be stable and efficient on multiple datasets.

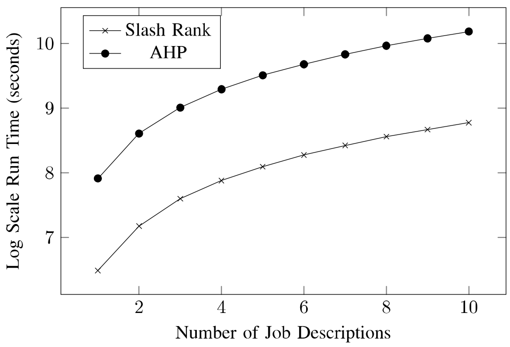


Fig. 7: Time Comparison (AHP vs Slash Rank)

We evaluated our proposed ranking algorithm with respect to time complexity. We compared its complexity with the AHP algorithm. The complexity of AHP is high due to the pairwise comparison of all the candidates with respect to every feature. The complexity of our algorithm drops due to the integration of candidate pruning step. This decrease in number of candidates drops the complexity due to less number of comparisons as compare to the AHP method.

## V. Conclusion and Future Work

We proposed a candidate ranking algorithm SlashRank in this research work, a multi-criteria decision making based candidate recommendation system that can assist recruiters in finding more suitable applicants for their job posts. SlashRank uses MCDM based technique with optimized scalable way of comparing the candidates with respect to the job post by using matrix sorting and pruning steps. The findings indicated that the suggested method is effective in real-world resume ranking processes. Furthermore, it provides scalable and efficient recommendations than state of the art MCDM method. In the future, we endeavor to find important facts about job recruiting and extract as much information as possible to develop an accurate and optimized applicant ranking system.

## Acknowledgments

This research work was funded by Higher Education Commission (HEC) Pakistan and Ministry of Planning, Development and Special Initiatives under National Center of Artificial Intelligence

## References

[1] B. Akila, S. Vasantha, and P. Thirumagal, "Effectiveness of e-recruitment for man power selection process," *Journal of Critical Reviews*, vol. 7, no. 5, p. 2020, 2019.

[2] M. Baum and R. Kabst, "Websites in the recruitment context: A conceptual model," *Proceedings of the Third European Academic Workshop on electronic Human Resource*, vol. 570, pp. 128–144, 2010.

[3] M. N. Freire and L. N. de Castro, "e-recruitment recommender systems: a systematic review," *Knowledge and Information Systems*, vol. 63, no. 1, pp. 1–20, 2021.

[4] E. Faliagka, L. Iliadis, I. Karydis, M. Rigou, S. Sioutas, A. Tsakalidis, and G. Tzimas, "On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed cv," *Artificial Intelligence Review*, vol. 42, no. 3, pp. 515–528, 2014.

[5] R. Nimbekar, Y. Patil, R. Prabhu, and S. Mulla, "Automated resume evaluation system using nlp," in *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pp. 1–4, IEEE, 2019.

[6] D. X. Sousa, S. Canuto, M. A. Goncalves, T. C. Rosa, and W. S. Martins, "Risk-sensitive learning to rank with evolutionary multi-objective feature selection," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, pp. 1–34, 2019.

[7] V. Senthil Kumaran and A. Sankar, "Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (expert)," *International Journal of Metadata, Semantics and Ontologies*, vol. 8, no. 1, pp. 56–64, 2013.

[8] 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," *Journal of Information Science*, vol. 45, no. 6, pp. 713–735, 2019.

[9] H.-C. Lee and C.-T. Chang, "Comparative analysis of mcdm methods for ranking renewable energy sources in taiwan," *Renewable and Sustainable Energy Reviews*, vol. 92, pp. 883–896, 2018.

[10] E. Faliagka, A. Tsakalidis, and G. Tzimas, "An integrated e-recruitment system for automated personality mining and applicant ranking," *Internet research*, 2012.

[11] E. Faliagka, K. Ramantas, A. K. Tsakalidis, M. Viennas, E. Kafeza, and G. Tzimas, "An integrated e-recruitment system for cv ranking based on ahp.," in *WEBIST*, pp. 147–150, 2011.

[12] S. Nawzad and C. Top, "Using ahp for the recruitment system: A case study at lafargeholcim company in kurdistan region of iraq," *International Journal of Economics, Commerce & Management*, vol. 7, no. 6, pp. 183–194, 2019.

[13] H. Braun, "Applying learning-to-rank to human resourcing's job-candidate matching problem: A case study.," 2017.

[14] A. Kouadria, O. Nouali, and M. Y. H. Al-Shamri, "A multi-criteria collaborative filtering recommender system using learning-to-rank and rank aggregation," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 2835–2845, 2020.

[15] S. Bruch, S. Han, M. Bendersky, and M. Najork, "A stochastic treatment of learning to rank scoring functions," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 61–69, 2020.

[16] W. Rc, Y. Munas, K. Cs, Fern, and N. Vithana, "Personality based e-recruitment system," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, 2017.

[17] A. Mohamed, W. Bagawathinathan, U. Iqbal, S. Shamrath, and A. Jayakody, "Smart talents recruiter-resume ranking and recommendation system," in *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pp. 1–5, IEEE, 2018.

[18] L. A. Cabrera-Diego, M. El-Bèze, J.-M. Torres-Moreno, and B. Durette, "Ranking résumés automatically using only résumés: A method free of job offers," *Expert Systems with Applications*, vol. 123, pp. 91–107, 2019.

[19] Ž. Stević, D. Pamučar, A. Puška, and P. Chatterjee, "Sustainable supplier selection in healthcare industries using a new mcdm method: Measurement of alternatives and ranking according to compromise solution (marcos)," *Computers & Industrial Engineering*, vol. 140, p. 106231, 2020.

[20] O. KORKMAZ, "Personnel selection method based on topsis multi-criteria decision making method," *Uluslararası İktisadi ve İdari İncelemeler Dergisi*, no. 23, pp. 1–16, 2019.