Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **1**, February **2025**

Personalized Intelligent Job Recommendation Based on Deep Learning in Elasticsearch Environment

Xiu-Yan Cui*

Hebei Software Institute, Baoding 071000, P. R. China
Hebei Province University Intelligent Interconnection Equipment
and Multimodal Big Data Application Technology Research and Development Center
ziyuerushi@126.com

David Hwang

College of Engineering
St. Paul University Philippines, Tuguegarao 3500, Philippines
hv0084@163.com

*Corresponding author: Xiu-Yan Cui
Received February 18, 2024, revised May 29, 2024, accepted August 28, 2024.

Abstract. *Aiming at the issue that the jobs predicted by the existing job recommendation methods cannot be precisely matched with the job seekers, this paper designs a personalized intelligent job recommendation method based on deep learning in the Elasticsearch environment. Firstly, to deal with the data sparsity and cold start issue of traditional collaborative filtering algorithms, the similarity calculation formula of Pearson's correlation in the algorithm is modified to enhance the accuracy of similarity calculation. Secondly, on the ground of the history of different job seekers, the keywords of job seekers are constructed in terms of various weights, and input into ElasticSearch to recall the set of jobs that are strongly related to the job seekers. Then adopting Convolutional Neural Network (CNN) to extract the text data features of job seekers and positions, and meanwhile, integrating the attention mechanism to get the auxiliary features of job seekers and positions, and finally, integrating the implicit feedback data with other attribute features of job seekers and positions as the input of the improved collaborative filtration algorithm, so that we can realize the intelligent recommendation of jobs for job seekers. Simulation outcome indicates that the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE) and Root Mean Square Percentage Error (RMSPE) of the suggested method are 0.0681, 0.0519, 0.0834 and 0.0628 respectively, which are lower than those of the comparison method, and the accuracy of the recommendation is effectively enhanced.*
Keywords: Job recommendation; deep learning; ElasticSearch; attention mechanisms; convolutional neural networks

---

1. **Introduction.** With the increasing number of users of online recruitment platforms and the continuous release of job information by enterprises, the relevant data and information have increased exponentially [1]. The huge amount of job seekers and job information makes the information management in the field of online recruitment increasingly complex, and job seekers face difficulties in choosing jobs. However, at present, many job boards search according to the keywords entered by job seekers and recommend the matched results to job seekers, which simply matches the keywords of the jobs and ignores the education and salary information of job seekers [2, 3, 4]. There are also

some recruitment websites that only push some hot jobs to job seekers, but not according to the user's own job interests and preferences for recommendation, the degree of personalization is generally low [5]. Therefore, how to effectively carry out job personalization recommendation is a key measure for the talent market to alleviate employment difficulties. The practical application scenario of employment recommendation realized by intelligent learning technology includes two aspects: (1) personalized job recommendation realized by user portrait analysis and (2) forecasting recruitment demand.

1.1. **Related work.** Personalized intelligent job recommendation is an important research direction in the field of human resources service and recruitment. Its core purpose is to improve the accuracy and efficiency of matching between jobs and job seekers through algorithms and technical means.

Thorat et al. [6] proposed a recommendation model based on collaborative filtering approach to recommend items with similar characteristics to the target user's historical preferred items. Koren et al. [7] proposed to combine Latent Factor Model (LFM) with collaborative filtering, but the recommendation accuracy is not high. Kethavarapu and Saraswathi [8] extracted feature keywords from job seekers' CVs and matched them with job information, and finally recommended the results to target users. Mhamdi et al. [9] combined clustering and association rules to construct an effective two-way recommendation model for human resources. Yang et al. [10] adopted a content-based recommendation method to complete two-way recommendation of online recruitment and job searching, but the computation is slow.

Recently, deep learning has gained world-renowned achievements in image, video, and speech processing, which pushes the application of deep learning to the field of job recommendation to become a new research hotspot [11]. Compared with other collaborative filtering recommendation algorithms, deep learning can automatically extract useful feature representations from original data through multi-layer neural network structure, reducing the need for artificial feature engineering. Reusens et al. [12] introduced the time factor into the job recommendation model, and at the same time, used the gradient Boosting regression tree to train the features. Mishra and Rathi [13] used job feature information and user's historical behavior records as inputs to screen out user-related jobs from a large number of jobs as recommendations. Roy et al. [14] calculated the similarity of jobs based on the history of job seekers' interaction matrix to generate a list of job recommendations. Xue et al. [15] suggested a deep collaborative filtering recommendation algorithm based on multidimensional feature crossover, but ignored the dynamic interests of job seekers. Chen et al. [16] combined a resume model with a decision tree in order to match the best comparable candidates to assign ranking points. Kethavarapu and Saraswathi [17] utilized Elasticsearch in conjunction with traditional systematic filtering algorithms to obtain web content, which the user can then recommend based on the associated content while searching. Chipps et al. [18] combine the features of gradient boosting tree into recommendation by transforming them through deep learning methods to realize recommendation of talents, but the recommendation accuracy is not high. Bendechache et al. [19] proposed an intelligent job recommendation system by using Elasticsearch search engine to do search and combining resume and job requirements. Jiechieu and Tsopze [20] use Convolutional Neural Network (CNN) to realize the intelligent job recommendation function through its self-learning and personalized preference analysis. Mao et al. [21] combine RNN model and graph attention mechanism model to apply to job recommendation, but there is a cold-start problem. Qin et al. [22] apply RNN model with graph attention mechanism model to recommend jobs, but there is a cold-start problem.

1.2. **Contribution.** The above job recommendation method only extracts information from the interaction history of job seekers and uses it to realize job recommendation. With the drastic increase of data, problems such as cold start and sparse data occur, resulting in unsatisfactory recommendation effect. The cold start problem refers to the lack of user and job data on a newly developed recruitment platform, which makes it difficult for the recommendation system to generate effective recommendation results in the initial stage, thus affecting users' satisfaction and willingness to use the platform. Focusing on the above issues, this article designs a personalized intelligent job recommendation method based on deep learning in Elasticsearch environment, which is compared with other job recommendation methods to verify the efficiency.

(1) The traditional collaborative filtering algorithm is optimized, and the similarity calculation formula of Pearson's correlation coefficient in the algorithm is modified to enhance the accuracy of similarity calculation. Secondly, constructing keywords relied on the history of job seekers and input them into ElasticSearch to recall the set of jobs that are strongly related to the job seekers.

(2) Two parallel CNN incorporating the attention mechanism are adopted to extract the text features related to job seekers and jobs, and the extracted text features and other auxiliary features are adopted as inputs to the enhanced collaborative filtering algorithm to realize the intelligent recommendation of jobs for job seekers.

2. Theoretical analysis.

2.1. **Traditional recommendation algorithm.** Collaborative filtering algorithm constructs user-item scoring matrix through information of user's historical behavioral data, selects users with high similarity as target user's nearest neighbor set through the calculation of similarity between users, and finally carries out the scoring prediction to recommend the users [23].

(1) Construct the user-item rating matrix. Construct a user-item rating matrix $A_{n\times m}$ for the set of n users $U : \{u_1, u_2, ..., u_n\}$ and the set of m items $I : \{i_1, i_2, ..., i_m\}$. a_{ij} represents the ratings of user i on item j. The rating matrix A is as bellow.

$$A = \begin{bmatrix} a_{11} & \cdots \\ \vdots & a_{nm} \end{bmatrix} \tag{1}$$

(2) User similarity calculation. User similarity calculation is done on the rating matrix, which essentially calculates the distance between user rating vectors. Because the standard deviation of variables is involved in the calculation of Pearson correlation coefficient, it is not affected by dimensions. This means that even if two variables have different scoring ranges or dimensions, their correlation can still be compared. Currently, a variety of similarity calculation methods are used in recommender systems, the most used is Pearson correlation coefficient [24], which is computed as follows.

$$sim(X, Y) = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{m}}{\sqrt{\frac{\sum x_i^2 - (\sum x_i)^2}{m} \frac{\sum y_i^2 - (\sum y_i)^2}{m}}} \tag{2}$$

where X and Y denote the rating vectors of user a and user b.

(3) Scoring prediction. After calculating the similarity between users, according to the Top-N principle, the top users are taken as the nearest neighbors of the target users, and the predicted scores of the project are computed as bellow.

$$P_{u,i} = \overline{S_u} + \frac{\sum\limits_{w \in H} sim(u,w) \times (S_{wi} - \overline{S_w})}{\sum\limits_{w \in H} sim(u,w)} \tag{3}$$

where $P_{u,i}$ denotes the predicted rating of item i by target user u, $\overline{S_u}$ and $\overline{S_w}$ are the mean ratings of all items by users u and w, S_{wi} is the rating of item i by user w, and H denotes the set of nearest neighbors of the target user u who have rated item i.

2.2. **Attention mechanism.** When people choose a small portion of useful information from this large amount of information and ignore other information, this ability is called "attention". Attention is essentially the ability to obtain new input features b, c, and d by performing different specifications of convolutional operations on different input features a at different locations of the same sample [25]. By analyzing the correlation between the different locations of the samples, feature extraction and matrix multiplication are performed. After the weighted sum operation, the feature extraction is completed to obtain the output feature e. The basic network framework of the attention mechanism is indicated in Figure 1.

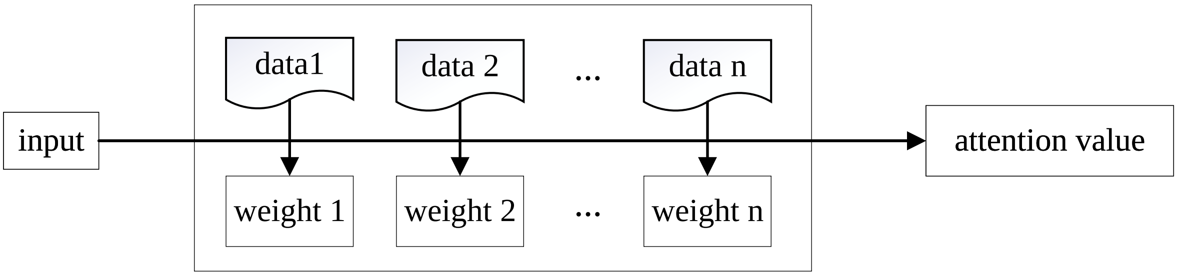

Figure 1. The basic network framework for attentional mechanisms

Let the number of times that have gone through the convolution operation be M. y_{ij} denotes the correlation between i-th and j-th spatial locations, which is computed as bellow.

$$y_{ij} = \frac{\exp(b_i * c_j)}{\sum\limits_{j=1}^{M} \exp(b_i * c_j)} \tag{4}$$

y_{ij} is computed for all channels and then a matrix Y is constructed and the final output is obtained as follows, where μ denotes the weight constant.

$$e_j = \mu \sum_{i=1}^{M} (y_{ij} d_i) + a_j \tag{5}$$

3. **Improvement of collaborative filtering algorithm based on user features.** Intending to the data sparsity problem and cold start issue of traditional collaborative filtering algorithm, the similarity calculation formula of Pearson's correlation coefficient in the algorithm is reformed to reduce the interference of user activity, project heat and rating on the similarity calculation results, and improve the accuracy of similarity calculation, for the sake of enhancing the precision of the subsequent job recommendation. Main steps are as bellow.

3.1. **Construction of user preference matrix.** The preference matrix between users and user features is calculated using the user-item rating matrix based on the association coefficient $rel_{i,f} = \frac{N_{i,f}}{N_i}$ between the item and the user feature.

$$pref\_user_{u,f} = \frac{\sum\limits_{i \in I_u} R_{u,i} \times rel_{i,f}}{len(I_u)} \tag{6}$$

where $pref\_user_{u,f}$ denotes the preference score of user u for user feature f, I_u denotes the set of scored items of user u, $len(I_u)$ denotes the size of set I_u, and $R_{u,i}$ denotes the score of user u on item i.

3.2. **Item preference matrix construction.** The scoring matrix is first populated using Equation (7) and then calculated to obtain the preference matrix $pref\_proj_{i,f}$ between items and item features.

$$R'_{u,i} = \frac{\sum\limits_{f \in I_i} avg\_proj_{u,f} \times pref\_proj_{u,f}\ len(F_{u,i})}{\sum\limits_{f \in I_i} pref\_proj_{u,f}\ len(F_i)} \tag{7}$$

where I_i is the set of item features that item i has, $avg\_proj_{u,f}$ is the mean value of user u's ratings on feature f, $pref\_proj_{u,f}$ is the preference score of user u on item feature f calculated in the previous section, F_i is the set of item features that item i has, and $F_{u,i}$ is the intersection of the set of all item features.

3.3. **Calculation of similarity.** On the ground of the constructed preference matrix, the similarity of item preference and user preference is calculated respectively, and the Pearson correlation coefficient is chosen as the calculation method of similarity, as indicated in Equation (8).

$$sim\_proj(u,v) = \frac{\sum\limits_{f \in F_{u,v}} (p_{u,f} - \overline{p_u})(p_{v,f} - \overline{p_v})}{\sqrt{\sum\limits_{f \in F_{u,v}} (p_{u,f} - \overline{p_u})^2} \sqrt{\sum\limits_{f \in F_{u,v}} (p_{v,f} - \overline{p_v})^2}} \tag{8}$$

$$sim\_user(u,v) = \frac{\sum\limits_{f \in F_{u,v}} (p_{u,f} - \overline{pu}_u)(p_{v,f} - \overline{pu}_v)}{\sqrt{\sum\limits_{f \in F_{u,v}} (p_{u,f} - \overline{pu}_u)^2} \sqrt{\sum\limits_{f \in F_{u,v}} (p_{v,f} - \overline{pu}_v)^2}} \tag{9}$$

where $sim\_proj(u,v)$ is the similarity between user u and user v in terms of item feature preferences, P denotes the item feature preference score, $F_{u,v}$ is the set of item feature preferences shared by user u and user v, $p_{u,f}$ is the scoring of user u's preference for item feature f, and $\overline{pu}_u$ is the scoring mean.

$sim\_user(u,v)$ is the similarity of user feature preferences, pu is the user feature preferences, $F_{u,v}$ is the set of user feature preferences, and $pu_{u,f}$ is the scoring of user u's preference for user feature f. $pu_{v,f}$ is the same as $pu_{u,f}$.

The scoring mean $\overline{pu}_u$ of user feature preferences, $\overline{pu}_v$, is the same as $\overline{pu}_u$.

(4) Combining Equation (8) and Equation (9), and integrating the two relied on the similarity weight coefficient η, the improved similarity calculation method incorporating the project features and user features is finally obtained.

4. **Personalized intelligent job recommendation based on deep learning in Elasticsearch environment.**

4.1. **Keyword construction for job seekers in an ElasticSearch environment.** To address the issue that existing job recommendation methods cannot effectively recommend job seekers or newly posted jobs, this article designs a personalized intelligent job recommendation method on the ground of deep learning in Elasticsearch environment.

First, keywords are constructed relied on the history of job seekers and input into ElasticSearch to recall the job sets that are strongly related to job seekers. Then two parallel convolutional neural networks are adopted to extract the textual data features of job seekers and jobs, and the attention mechanism is integrated to gain more accurate modeling of textual features to enhance the recommendation effect.

Through this process, the relevant text features of job seekers and positions are achieved as an auxiliary feature of each, and finally, the implicit feedback data is fused with other attribute features of job seekers and positions as the input of the improved collaborative filtering algorithm to realize the intelligent recommendation of jobs for job seekers. The flow structure is shown in Figure 2.

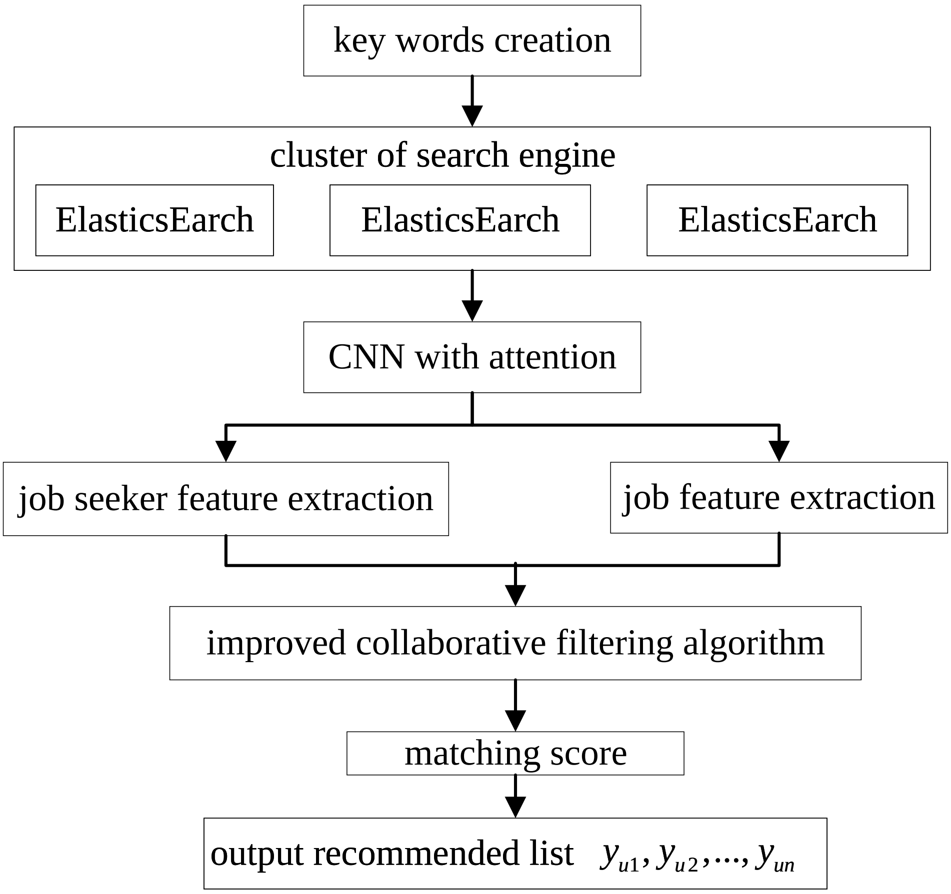

Figure 2. Flow of the suggested algorithm

Using the TFIDF [?] keyword construction algorithm, a probabilistic model is implemented by noting all the history records of a job seeker as N and a particular record as $J = \{j_1, j_2, \ldots, j_m\}$, where j is a word in the record. Then, the frequency of word j in the current record of the job seeker is $v_{jl} = n/m$, and $TF = \sum_1^N v_{jl}$ is the likelihood, i.e., conditional probability of word j in the entire record of the job seeker, i.e., the conditional probability $P(j|UserLiked)$. Then, the frequency of the inverse document for word j in the entire set is indicated as below.

$$IDF = \log \frac{D}{d_j, j_l \in d} \tag{10}$$

where D denotes the set of job seekers.

In terms of the Bayesian equation, $TF \times IDF$ is then proportional to $P(j|UserLiked)$. The posterior probability of word j in the record is obtained, and then all the words are ranked in terms of probability, and in terms of the principle of maximizing the posterior probability, the top N is taken as the set of keywords for job seekers.

After extracting the keyword set using the probabilistic model, the keyword set can be passed to ElasticSearch as the personalized input of the job seeker, and a part of the job set can be recalled from ElasticSearch, which has strong correlation with the preference of the current job seeker.

4.2. **Attention mechanism based feature extraction.** After correlating the job seekers' preferences and job sets, two parallel CNN is adopted to extract features from the job seeker word vector matrix and the job word vector matrix, respectively. The network structure mainly contains convolutional level, pooling level, attention level, and fully connected level. The steps in detail are as follows.

(1) Convolutional level. Each convolutional level has s neurons. The word vector matrix A_u of the job applicant is adopted as the input to the level, and the features of A_u are extracted after convolutional operations on it. Each neuron i has a filter $filt_i$, of size $t \times r$, where t is the dimension of the features produced by A_u and r is the size of the filter. Each convolution kernel performs a convolution operation and outputs a feature map map_i:

$$map_i = f(A_u * filt_i + b_i) \tag{11}$$

where f is the activation function $relu$, $*$ is the convolution operation, and b_i is the corresponding bias.

(2) Pooling level. The maximum values in the corresponding region of each pooling layer filter are selected, and then the maximum values in the corresponding region are combined to extract the main features from the feature map map_i output from the convolutional level. The feature map map_i' achieved after the maximum pooling operation is indicated below:

$$map_i' = MaxPooling(map_i) \tag{12}$$

(3) Attention level. The l-th keyword vector of the word vector matrix A_u of job seeker u is g_l', and its corresponding attentional weight a_l is as follows.

$$a_l = \frac{\exp(h_l)}{\sum_{j=1}^{r} \exp(h_j)} \tag{13}$$

where $h_l = relu(A_g g_l'' + b_1)$, A_l are the weight parameter matrices, and b_1 is the bias.

The attentional weights of all word vectors are concatenated according to the word order of the original word vector matrix to obtain the attentional weight matrix $A(u) = (a_1, a_2, ..., a_s)$ corresponding to A_u. $A(u)$ is multiplied by the corresponding A_u of job applicant u to get the updated word vector matrix $A_u^a = A(u)A_u$.

(4) Full connectivity level. Taking map_i and A_u^a as inputs to the fully connected layer, multiply them with the weight matrix W and add the bias b. The output, $output_u$, is adopted as the eigenvector of the job seeker, as below.

$$output_u = f(W * (map_i' + A_u^a) + b) \tag{14}$$

Similarly, adopt the above steps to extract the word vector matrix A_j of the job, and finally get the feature vector $output_i$ of the job.

4.3. **Personalized intelligent job recommendation based on deep learning in Elasticsearch environment.** (1) **Characterization Input.** The input is divided into two parts, one is the job seeker feature information and the other is the job feature information. The text data feature $output_u$ is extracted by the method in the previous section, and the above auxiliary features are fused to get the auxiliary features of job seekers. Finally, the job seeker features and auxiliary features are fused to obtain the job seeker feature information, which is adopted as the input of the model. The details are indicated as below.

$$y_u = F_u \oplus F_J \tag{15}$$

where y_u denotes the job seeker feature information, F_u denotes the job seeker's feature data, and F_J denotes the job seeker's auxiliary features. The same procedure is used to obtain the joint feature information $y_j = F_j \oplus J$ of the desired job and the actual job, where F_j denotes the feature data of the job and J denotes the auxiliary features of the job.

(2) **Attention-based representation learning collaborative filtering.** The attention mechanism is introduced to distinguish the importance of different historical interactions of job seekers, which takes into account the different importance of different interaction positions of job seekers to the final recommendation outcome.

$$\begin{cases} y_u' = Softmax(W_u y_u \odot W_J y_j) \odot W_u y_u \\ y_j' = Softmax(W_u y_u \odot W_J y_j) \odot W_J y_j \end{cases} \tag{16}$$

where $\odot$ represents the Hadamard product operation, W represents the weight matrix, and $Softmax(\cdot)$ function calculates the weights of each dimension by normalization. The idea of the attention mechanism is utilized to differentiate the importance of different interaction history positions of job seekers in predicting the target position, and the outputs y_u' and y_j' of the attention mechanism are the inputs of the learning network for the subsequent representation.

Then the potential vector representation of learning job seekers is as below.

$$P_u = relu(W_x a_{x-1} + b_x) \tag{17}$$

where W_x denotes the weight matrix, b_x denotes the bias, a_{x-1} denotes the feature weights of the job applicant, $relu(\cdot)$ is the activation function, and P_u denotes the potential vector representation of the job applicant. In the same way, the potential vector representation of the job is obtained as q_j. The final prediction vector of the learning part of the attention-based representation is denoted as $c^f l_\gamma$, as indicated in Equation (18).

$$c^f l_\gamma = P_u \oplus q_j \tag{18}$$

(3) Learning collaborative filtering based on attention-based matching function. A linear embedding layer is first used to learn the latent vector representations of job seekers and jobs, and then the latent vector representations are used to obtain the weights of each dimensional feature through Equation (19).

$$A_{out} = Softmax(a_0) \odot a_0 \tag{19}$$

where P_u and q_i are spliced to obtain the input a_0 of the attention network, and A_{out} is the input of the subsequent matching function learning network. The final prediction vector is denoted as c_γ^{nl}, as indicated in Equation (20).

$$c_\gamma^{nl} = relu(W_\gamma q_{\gamma-1} + b_\gamma) \tag{20}$$

At last, the predictive vector representations obtained from the two parts are fused to get the final list of job recommendations for the whole model, as indicated as below.

$$y_{ui}^{\wedge} = \delta\left(W_{out}\begin{bmatrix} c_\gamma^{fl} \\ c_\gamma^{nl} \end{bmatrix}\right) \tag{21}$$

5. Performance testing and analysis.

5.1. Comparison experiment. For the purpose of evaluating the recommendation performance of the personalized intelligent job recommendation method based on deep learning in the Elasticsearch environment designed in this paper, experiments and analyses are carried out in this paper using real datasets. The dataset used in this paper is the data collected from an online recruitment platform, and through a series of data preprocessing work, the job data warehouse applicable to recommendation is obtained. The scale of the experimental data is shown in Table 1. The dataset contains 4692 applicants, 15000 jobs and 170844 user behavior records. In the experiment, the dataset is randomly divided into training set, testing set and validation set according to 7:2:1 on the dimension of job application.

Table 1. The size of the dataset in the experiment

Designation	Job applicant	Workplace	User behavior record
Quantity	5718	20000	182374

To facilitate the analysis, the comparison model in the literature [17] is denoted as CBDC, the comparison model in the literature [20] is denoted as SPMR, the comparison model in the literature [21] is denoted as AJRM, and the algorithm in this article is denoted as PIDL. All the experiments are carried out under the Linux operating system and the Python v3.7 programming environment. The convolutional kernel size is set to 3 in the experiments, and to prevent the risk of overfitting, a dropout strategy is used by setting the dropout value to 0.5. The learning rate is set to 0.001, the Adam optimization function is used, and the number of iterations is set to 20.

In this experiment, the performance of the design model is evaluated using hit radio (HR@k) [27], normalize discount cumulative gain (NDCG@k) index, where k represents the length of the recommendation list. Table 2 indicates the comparison of NDCG and HR for recommendation list lengths of 10 and 20, respectively.

Table 2. Experimental outcome of various algorithms on experimental datasets

Method	NDCG@10	NDCG@20	HR@10	HR@20
CBDC	0.4627	0.4752	0.6435	0.7715
SPMR	0.4916	0.5238	0.6889	0.8216
AJRM	0.4102	0.4316	0.5907	0.7208
PIDL	0.5528	0.5803	0.8102	0.8924

The suggested method PIDL achieves the best performance on NDCG@10, NDCG@20, HR@10 and HR@20 indexes, which is 19.47%, 22.12%, 25.91%, and 12.45%, 10.79%, 25.91% and 17.61% better than CBDC method, and AJRM method, respectively, 15.67%, and 34.76%, 34.45%, 37.16%, 23.81% than the AJRM method, respectively. It can be seen that the PIDL method suggested in this article is significantly better than CBDC, SPMR and AJRM methods.

This is because the AJRM method only applies the traditional collaborative filtering algorithm to the job recommendation and does not consider the feature information of the job seeker and the job, which leads to a much worse recommendation effect than the other three deep learning-based recommendation models. The CBDC method only recommends the most popular jobs and does not consider the job seeker's preference and the internal features of the job, so its recommendation performance is weaker than that of the SPMR method. The SPMR method is based on CNN and collaborative filtering algorithm to recommend jobs, but it does not improve the collaborative filtering algorithm and does not extract the feature information of the job, which results in its recommendation effect is not as good as the PIDL method. The PIDL improves the similarity calculation of the collaborative filtering algorithm, and utilizes the attention mechanism to extract the preference features of the applicant and the job, which is conducive to improving the accuracy of the job recommendation, so the recommendation performance is weaker than the PIDL method. Recommendation accuracy, so the recommendation performance performs best.

A comparison of the recall of the PIDL method proposed in this paper with the other three methods is indicated in Figure 3. When the length of the recommendation list is 100, the recall rates of CBDC, SPMR, AJRM and PIDL methods are 0.81, 0.86, 0.73 and 0.92, respectively, and the performance of the PIDL method is better than that of CBDC, SPMR and AJRM methods. Since the PIDL method improves the accuracy of similarity calculation by improving the similarity calculation formula of Pearson's correlation number in collaborative filtering algorithm and introduces the attention mechanism to further mine the characteristics of job seekers and jobs, it improves the accuracy of recommendation, whereas the CBDC, SPMR, and AJRM methods don't mine the preference characteristics of job seekers and jobs, which significantly reduces the quality of the recommendations, whereas the CBDC, SPMR, and AJRM methods don't mine the preference characteristics of job seekers and jobs, which leads to the poor recommendation effect. The improvement of HR, NDCG, and Recall performance metrics is verified to alleviate the problems of cold start and data sparsity that most job recommendation algorithms may have.

5.2. **Comparison and analysis of recommended accuracy of different methods.** To estimate the forecasting accuracy of different recommended methods, the experimental results were measured by five metrics: correlation coefficient (R), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Root Mean Square Percentage Error (RMSPE). The outcome of the four method evaluation metrics are given in Table 3 and the results are plotted on a visual bar comparison chart as indicated in Figure 4.

Table 3. Comparison of recommended accuracy of different methods

Method	**R**	**MAE**	**MAPE**	**RMSE**	**RMSPE**
CBDC	0.8059	0.2097	0.2269	0.2854	0.2109
SPMR	0.8512	0.1261	0.1352	0.2038	0.1297
AJRM	0.7216	0.3168	0.3517	0.4029	0.3524
PIDL	0.9327	0.0681	0.0519	0.0834	0.0628

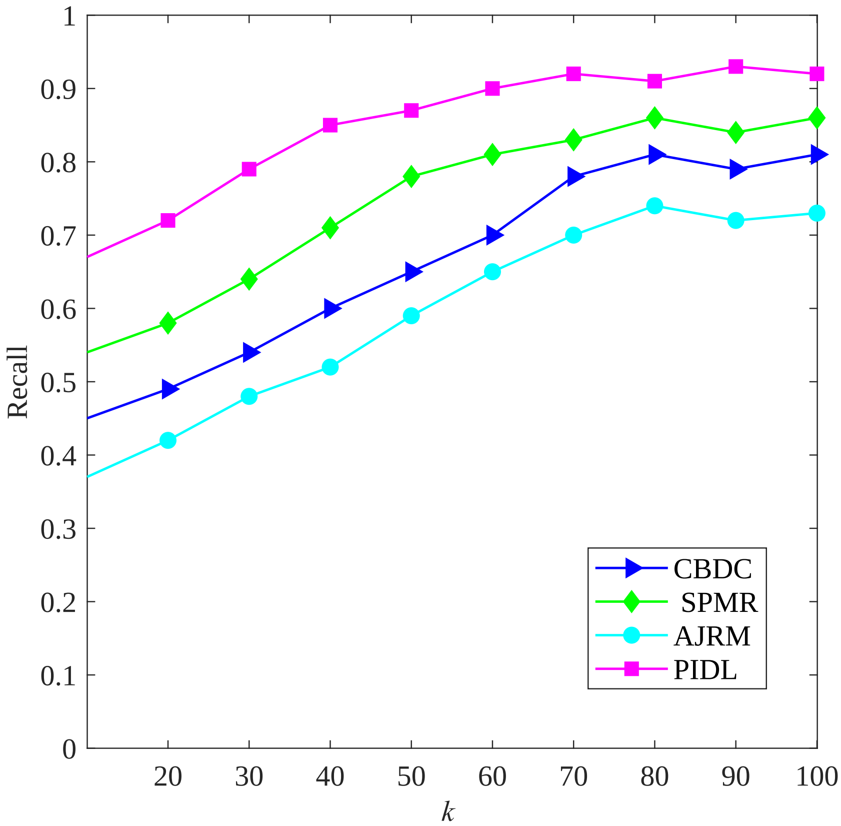

Figure 3. Recall comparison of various algorithms

As can be seen from Table 3 and Figure 4, the accuracy evaluation indexes of PIDL are significantly better than those of CBDC, SPMR, and AJRM. The MAE of the PIDL method was 0.0681, which was reduced by 0.1416, 0.058, and 0.2487 compared to the CBDC, SPMR, and AJRM methods, respectively; the RMSPE of the PIDL method was 0.0628, which was reduced by 0.1481, 0.0669, and 0.2626 compared to the CBDC, SPMR, and AJRM methods, respectively. In addition, in the recommendation results, the MAPE value and RMSE value of the PIDL model are both lower than 0.1, which reinforces the fact that PIDL, a personalized intelligent job recommendation method based on deep learning in the Elasticsearch environment, is more suitable for solving the job recommendation problem.

Comparing the correlation coefficients R, it can be seen that the R-value of the PIDL method is 0.9327, the R-value of the CBDC method is 0.8059, the R-value of the SPMR method is 0.8512, and the R-value of the AJRM method is 0.7216, which is an improvement of 15.73%, 9.57%, and 29.25%, respectively, as compared to the CBDC, SPMR, and AJRM methods. This indicates that the introduction of the attention mechanism to extract the features of job seekers and job preferences is effective, and the improvement of the similarity calculation in the collaborative filtering algorithm makes the recommendation effect better than the comparison model, and has a better fitting effect on the job recommendation and job seekers' preferences. Therefore, since the CEFB method utilizes ElasticSearch for strong recall of job seeker preferences and current jobs, and introduces the attention mechanism for feature lifting, it makes the PIDL method have better fitting effect and recommendation accuracy.

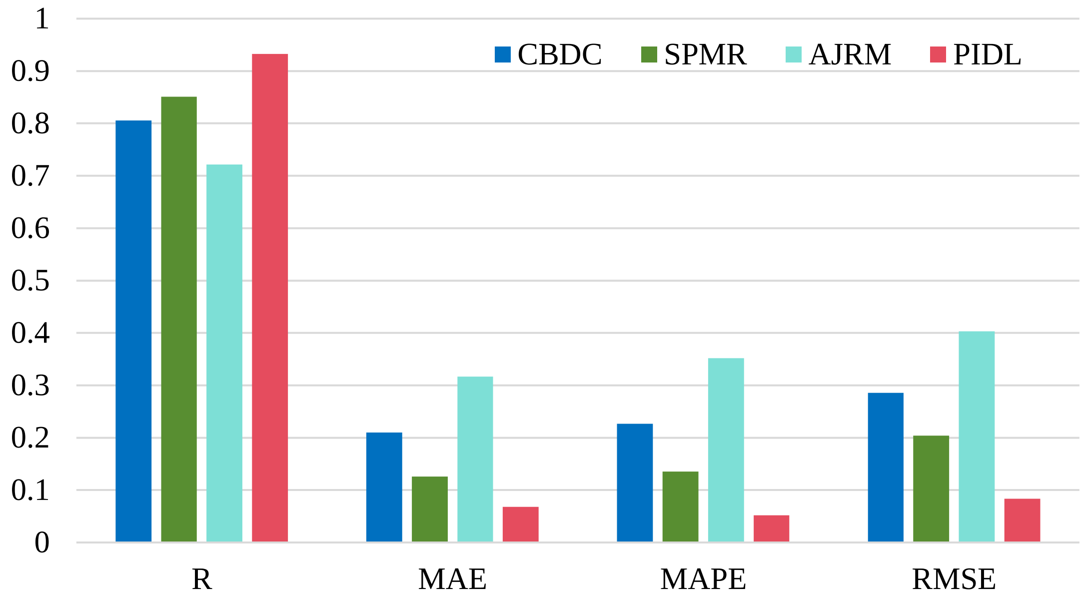

Figure 4. Recommended accuracy comparison

6. **Conclusion.** Intending to the issue that the existing job recommendation methods cannot recommend jobs efficiently, this article suggests a personalized intelligent job recommendation method based on deep learning in Elasticsearch environment. First, the similarity calculation formula of Pearson correlation coefficient in collaborative filtering algorithm is modified to improve the accuracy of similarity calculation. Secondly, we construct keywords based on the history of job seekers and input them into ElasticSearch to recall the set of jobs that are strongly related to job seekers. Then two parallel CNN incorporating the attention mechanism are adopted to extract the text features related to job seekers and jobs, and the extracted text features and other auxiliary features are used as inputs for the improved collaborative filtering algorithm to realize the intelligent recommendation of jobs for job seekers. The experimental outcome indicates that compared with the existing job recommendation methods, the method suggested in this article has lower MAE, MAPE, RMSE and RMSPE, and can be better applied to job recommendation.

Acknowledgment. This work is supported by the Funded by Science Research Project of Hebei Education Department (No. QN2023142).

REFERENCES

[1] R. G. Belsare, B. Prmitr, and V. Deshmukh, "Employment Recommendation System: A Review," *International Journal of Science and Engineering Applications*, vol. 7, no. 5, pp. 64-67, 2018.
[2] R. Liu, W. Rong, Y. Ouyang, and Z. Xiong, "A hierarchical similarity based job recommendation service framework for university students," *Frontiers of Computer Science*, vol. 11, pp. 912-922, 2017.
[3] M. F. Koh, and Y. C. Chew, "Intelligent job matching with self-learning recommendation engine," *Procedia Manufacturing*, vol. 3, pp. 1959-1965, 2015.
[4] M. Reusens, W. Lemahieu, B. Baesens, and L. Sels, "Evaluating recommendation and search in the labor market," *Knowledge-Based Systems*, vol. 152, pp. 62-69, 2018.
[5] S. A. Alsaif, M. Sassi Hidri, H. A. Eleraky, I. Ferjani, and R. Amami, "Learning-based matched representation system for job recommendation," *Computers*, vol. 11, no. 11, 161, 2022.
[6] P. B. Thorat, R. M. Goudar, and S. Barve, "Survey on collaborative filtering, content-based filtering and hybrid recommendation system," *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31-36, 2015.

[7] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30-37, 2009.

[8] U. P. K. Kethavarapu, and S. Saraswathi, "Concept based dynamic ontology creation for job recommendation system," *Procedia Computer Science*, vol. 85, pp. 915-921, 2016.

[9] D. Mhamdi, R. Moulouki, M. Y. El Ghoumari, M. Azzouazi, and L. Moussaid, "Job recommendation based on job profile clustering and job seeker behavior," *Procedia Computer Science*, vol. 175, pp. 695-699, 2020.

[10] S. Yang, M. Korayem, K. AlJadda, T. Grainger, and S. Natarajan, "Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach," *Knowledge-Based Systems*, vol. 136, pp. 37-45, 2017.

[11] R. Ravita, and S. Rathi, "Inductive Learning Approach in Job Recommendation," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 2, pp. 242-251, 2022.

[12] M. Reusens, W. Lemahieu, B. Baesens, and L. Sels, "A note on explicit versus implicit information for job recommendation," *Decision Support Systems*, vol. 98, pp. 26-35, 2017.

[13] R. Mishra, and S. Rathi, "Enhanced DSSM (deep semantic structure modelling) technique for job recommendation," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 7790-7802, 2022.

[14] P. K. Roy, S. S. Chowdhary, and R. Bhatia, "A Machine Learning approach for automation of Resume Recommendation system," *Procedia Computer Science*, vol. 167, pp. 2318-2327, 2020.

[15] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 3, pp. 1-25, 2019.

[16] W. Chen, P. Zhou, S. Dong, S. Gong, M. Hu, K. Wang, and D. Wu, "Tree-based contextual learning for online job or candidate recommendation with big data support in professional social networks," *IEEE Access*, vol. 6, pp. 77725-77739, 2018.

[17] U. P. K. Kethavarapu, and S. Saraswathi, "Concept based dynamic ontology creation for job recommendation system," *Procedia Computer Science*, vol. 85, pp. 915-921, 2016.

[18] B. E. Chipps, L. B. Bacharier, J. R. Farrar, D. J. Jackson, K. R. Murphy, W. Phipatanakul, S. J. Szefler, W. G. Teague, and R. S. Zeiger, "The pediatric asthma yardstick: practical recommendations for a sustained step-up in asthma therapy for children with inadequately controlled asthma," *Annals of Allergy, Asthma & Immunology*, vol. 120, no. 6, pp. 559-579, 2018.

[19] M. Bendechache, S. Svorobej, P. T. Endo, A. Mihai, and T. Lynn, "Simulating and Evaluating a Real-World Elasticsearch System Using the Recap Des Simulator," *Future Internet*, vol. 13, no. 4, 83, 2021.

[20] K. F. F. Jiechieu, and N. Tsopze, "Skills prediction based on multi-label resume classification using CNN with model predictions explanation," *Neural Computing and Applications*, vol. 33, pp. 5069-5087, 2021.

[21] Y. Mao, S. Lin, and Y. Cheng, "A Job Recommendation Model Based on a Two-Layer Attention Mechanism," *Electronics*, vol. 13, no. 3, 485, 2024.

[22] C. Qin, H. Zhu, T. Xu, C. Zhu, C. Ma, E. Chen, and H. Xiong, "An enhanced neural network approach to person-job fit in talent recruitment," *ACM Transactions on Information Systems (TOIS)*, vol. 38, no. 2, pp. 1-33, 2020.

[23] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19-46, 2024.

[24] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121-2133, 2022.

[25] C. M. Chen, Q. Miao, S. Kumar, and T.-Y. Wu, "Privacy-preserving authentication scheme for digital twin-enabled autonomous vehicle environments," *Transactions on Emerging Telecommunications Technologies*, vol. 34, e4751, 2023.

[26] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF* IDF, LSI and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758-2765, 2011.

[27] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, "Soft cache hits: Improving performance through recommendation and delivery of related content," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1300-1313, 2018.

[28] H. Yoganarasimhan, "Search personalization using machine learning," *Management Science*, vol. 66, no. 3, pp. 1045-1070, 2020.

[29] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, e623, 2021.