FINN:Feature Interaction Neural Network for Person-Job Fit

1st Miao He
Department of Military Installations
Army Logistics University of PLA
Chongqing, China
hemiao@nudt.edu.cn

2nd Tao Wang
College of Systems Engineering
National University of Defense Technology
Changsha, China
wangtao@nudt.edu.cn

3rd Yuanyuan Zhu
College of Marxism
Chongqing College of Electronic Engineering
Chongqing, China
alisa625@126.com

4th Yingguo Chen
College of Systems Engineering
National University of Defense Technology
Changsha, China
ygchen@nudt.edu.cn

5th Feng Yao
College of Systems Engineering
National University of Defense Technology
Changsha, China
yaofeng@nudt.edu.cn

6th Ning Wang
Unit 32375
People's Liberation Army of China
China
270959009@qq.com

2021 7th International Conference on Big Data and Information Analytics (BigDIA) | 978-1-6654-2466-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIGDIA53151.2021.9619599

***Abstract*—Person-Job Fit is one of the most critical steps in the recruitment chain. Existing studies on the Person-Job Fit problem either heavily depend on manually generated features or only focus on learning good representations of resumes and jobs based on the textual features. To this end, with the comprehensive consideration of numerical, categorical, and semantic features, we propose a novel end-to-end Feature Interaction Neural Network (FINN) to predict the matching scores between jobs and resumes. The key idea is to learn the latent vector of interacting signals in the type of categorical and semantic features. Specifically, we design several special operators to capture the interacting signals of categorical features between jobs and resumes. Meanwhile, the convolutional neural networks are introduced to model the interactions based on a word-level similarity matrix. We validate the effectiveness of model FINN by comprehensive experiments on a large-scale real-world dataset, which consists of 452,214 jobs and 651,711 resumes. Compared with several classical supervised learning methods, the proposed model can achieve an improvement of 1.96%-24.46% in the metric accuracy, F1 and AUC.**

***Index Terms*—Person-Job Fit, Feature Interaction, Interacting Operator, Similarity Matrix, Convolutional Neural Network**

I. Introduction

With the rapid growth of the Web in recent years, a vast amount of job postings are publicly available on the Internet. Many job seekers exploit online recruitment platforms such as LinkedIn[1], and Xing[2] to search for job positions that they are interested in. In addition to job seekers, often recruiters need to browse the user profiles and select the most fitted candidates for particular positions. Despite the abundance of available job opportunities and employee candidates, it is a struggle for job seekers (or recruiters) to navigate through theses information to find the appropriate positions (or candidates). While the manual process of recruitment for both job seekers and recruiters remains cost-consuming and time-consuming. Hence, it is necessary and advantageous to construct an effective and automatic matching model to predict the fitness between resumes and jobs, which is called Person-Job Fit.

Machine learning methods have been developed for the Person-Job Fit problem. In earlier, this problem is formulated as a job recommendation problem. On the one hand, job recommendation system can collect a list of jobs for a job seeker based on his preferences, on the other hand, it can also provide a list of candidate resumes for recruiters based on the position's requirements[1], [2], [3]. Though traditional machine learning models[4] are successful for matching score prediction in job recommendation, these models heavily rely on hand-craft features generated with domain expert knowledge. Generally, these models constructed with human-based features suffer from a few drawbacks, including inaccurate, inefficient, and subjective[5]. In addition, human-based features directly restrict the scalability of machine learning approaches applied on more and more data with new type features.

Recently, advances in deep learning have brought significant progress to the Person-Job Fit problem with several deep matching models proposed. The power of deep learning models lies in the ability to learn distributed representations from the raw data in an automatic fashion. Moreover, deep learning models can easily avoid many limitations of hand-crafted features utilized in machine learning models. Researchers find that it is effective to predict the person-job fitness by projecting the content of both jobs and resumes into a shared latent representation space[6]. Along this line, several different deep matching structures[7], [8], [9], [10], [5], [11], [12] are proposed for the Person-Job Fit problem, based on the effective representation learning of jobs and resumes through their textual descriptions.

However, these models mentioned above only focus on the free text listed in the job description and work experience, and overlook other structured features(e.g., career level, education, region). As a result, some important information that is helpful in handling the complexity of the Person-Job Fit problem might be missed without these fields. Generally, people would

[1] www.linkedin.com
[2] www.xing.com

978-1-6654-2466-0/21/$31.00 ©2021 IEEE

focus on some key words and key phrases(e.g., company name, skill, title, tag) while exploring the textual information of jobs or resumes. One major weakness of these representation-based models is the failure to capture local lexical matching signals[13], since a fixed dimensional vector cannot summarize all the information of raw text in jobs or resumes.

To overcome these issues, we propose a novel Feature Interaction Neural Network (FINN) for Person-Job Fit. Our proposed model can capture the comprehensive interaction information based on all pairwise categorical feature interactions and semantic entity interactions given a job post and a user resume. Specifically, the common categorical features of the job and resume are embedded into low-dimensional spaces, which reduces the dimension of input features. Meanwhile, all semantic entities of the job and resume are also embedded into another fixed dimensional latent vectors. Afterward, we develop two interaction modules to learn the local relationships of categorical features and semantic entities between jobs and resumes respectively. For the latent vectors of common categorical features, we aggregate the pair-wise feature representations in various ways in the interaction layer. For the semantic entity representations, we construct a matching matrix to represent the word-level matching signals between each job and resume. Then, Convolutional Neural Networks (CNNs) with different kernels are used to model the matching relations between the job and resume in the field of semantic entities. Finally, extensive experiments on a large-scale real-world dataset validate the effectiveness of our proposed FINN model compared with baseline models.

In summary, this article makes the main contributions as follows:

- We propose a novel model named FINN for Person-Job Fit by learning feature interaction signals in several fields.
- We propose two modules to learn the interaction signals of categorical feature fields and textual feature fields respectively.
- We construct extensive experiments on a large real-world dataset, the results have demonstrated the effectiveness of model FINN.

This paper is organized as follows. We discuss related literature in section II. In section III, we formally define the person-job fit problem and give an overview of our proposed model FINN. The overview and technical details of model FINN are introduced in section IV. We introduce the experiments in section V. In section VI,we give the conclusion of this paper.

II. Related Work

In this section, the researches that are related to our work will be reviewed. Specifically, we identify two lines of research: job recommendation and person-job fit.

A. Job Recommendation

Job recommendation is a particular recommendation problem with some specials. Generally, a popular item (e.g., book, movie, or product) can be recommended to a lot of different users for consumption in the classic recommender systems, while a job position of a company can give offers to only a few job seekers[14]. In addition, there are two directions of job recommendation: one is to collect a list of relevant jobs for a user, while the other is to recommend some qualified users to a given job position according to its requirements. The two directions of this problem also can be combined as a bilateral recommendation system by considering the preference of both recruiters and candidates[1], [15].

The problem of recommending jobs for a given user has been researched with the typical recommender methods (e.g., collaborative filtering, content-based filtering, and hybrid combinations of both). For example, CASPER is a job recommendation system that applies a collaborative filtering method to recommend jobs to users based on what similar users have previously interacted with[16]. Zhang et al. propose an item-based collaborative filtering method to help students find jobs quickly[17]. Collaborative filtering often suffers from the challenges of cold start and data sparsity which can be addressed by content-based methods. A content-based recommender system is developed to retrieve relevant jobs for Facebook and LinkedIn users based on user profiles and job information[3]. It would be helpful to consider the user's historical positions when recommending jobs to him. Hence, researches about career path prediction are introduced into job recommendation systems[18], [19], [20]. Some researchers construct hybrid approaches for job recommendation by combining both methods of collaborative filtering and content-based filtering[21]. In addition, the ACM RecSys Challenge 2016 is to build a job recommendation system to predict those jobs that a given user will be interested. Some machine learning models are applied to generate the recommendations on XING (a career-oriented social networking site) [22], [23], [24]. Recently, an end-to-end deep learning model has been constructed to jointly learn the representations of jobs through three types of information networks[25].

As another direction, recommending personnel for job positions also has been extensively studied. Nikolaos et al. propose a content-based mdoel to calculate the matching score between a candidate and the given job position[26]. In 2017, another RecSys Challenge is organized by XING, the challenge is finding those users that are appropriate candidates for the given jobs. Solutions[27], [28], [29] for this job recommendation challenge heavily rely on feature engineering. This problem is also regarded as personnel selection or talent searching[30], [31] which directly affect the core competencies of the company. A data mining framework is constructed based on the decision tree for the problem of personnel selection [32]. PROSECT is designed as a decision support system to help screen candidates for each job posting based on the meta-data and information extracted from the resumes[33]. Song et al. propose a model to recruit suitable volunteers for nonprofit organizations (NPOs) through the profiles collected from multiple social networks[34], [35]. Along this line, Jia et al. present a deep learning framework with source confidence and consistency regularization to learn a joint representation of a user's interests, behaviors, and personnel traits[36]. Also, deep learning models with representation learning approaches are developed for talent search systems at LinkedIn[37].

According to the literature reviewed above, the core idea of the job recommendation task is to predict the matching score between jobs and resumes, which is also the main challenge of the Person-Job Fit problem.

B. Person-Job Fit

The core step of job recommendation and personnel selection is Person-Job Fit, whose goal is to measure the matching score between the job position and resume. Most existing studies treat the problem as a binary classification task input with a given job-resume pair. Zhu et al. propose a data-driven model to project resumes and jobs into a shared latent space with two Convolutional Neural Networks (CNNs) respectively and measure the matching score between them with cosine similarity[6]. Along this line, Qin et al. develop an ability-aware framework for person-job fit problem[7]. Afterward, a refinement strategy based on historical interaction data is designed to enhance the model performance. This enhanced model is explored into two applications including job recommendation and personnel selection[12]. A siamese adaptation of a convolutional neural network is developed to capture the underlying semantics of resumes and jobs, which enabling to encode fitted person-job pairs closer to each other, and make dissimilar pairs distant from each other[38]. Khatua and Nejdl propose another model based on siamese architecture to predict matching scores between recruiters and job seekers on manually annotated data extracted from Twitter[39].

Generally, both employers and employees have preference on each other. For example, employers may tend to select talents with experience in a particular company or project even if they all qualify for a given position. To this end, Yan et al. propose two interactive memory profiling modules to learn the latent preference of recruiters and job seekers respectively. The matching degree of the person-job pair is measured by a multi-layer perceptron(MLP) fed with the concatenation of final representations of job and resume[10]. In order to balance the different demands of employers and talents, a multi-task learning model is established to address the person-job fit problem[9]. Luo et al. design an adversarial learning network to model the representations of job and resume with the consideration of consistency between talent's experiences and skills[5].

Moreover, the emergence of different deep learning techniques provides some novel perspectives for the problem of Person-Job Fit. As usual, researchers treat person-job fit problem as a supervised text match problem, which relies on a considerable number of labeled samples to train the matching model. The amount of available labeled person-job pairs vary in different job categories. Hence, Bian et al. introduce the domain adaption to study the Person-Job Fit problem, that utilizing the knowledge learned from a source domain with sufficient labeled instances to improve the prediction performance in another domain with limited labeled data[8]. In addition, Bian et al. propose a co-teaching framework with two mutually helpful components to solve the problem of data sparsity. The final representations of jobs and resumes can be enhanced by the two components that capture the semantic information in different views[11].

We also propose a deep learning model to address the Person-Job Fit problem just as the prior studies. However, different to the prior approaches which focus on the representations learning of the textual data in the jobs and resumes, our proposed model additionally considers the influence of structural features in jobs and resumes just as [40] to fuse the representations of structural and textual data. Nonetheless, we learn the feature interactions between resumes and jobs instead of in just one profile (job or resume).

III. Problem Formulation

In this paper, we aim at studying the Person-Job Fit problem, with a goal of predicting the matching score between a job posting and a resume based on their multi-type of features. We will give the problem formulation in this section.

Specifically, to formulate the Person-Job Fit problem, we denote the job posting set as $P = \{p_1, p_2, ..., p_m\}$ and the resume set as $R = \{r_1, r_2, ..., r_n\}$, where the m and n is the total number of jobs and resumes respectively. Both job posting p_i and resume r_j contain three types of features, i.e., continuous features, categorical features and semantic features, denoted as $p_i = \{p_{i,1}, p_{i,2}, p_{i,3}\}$ and $R = \{r_{j,1}, r_{j,2}, r_{j,3}\}$. For resume r_j, the l-th ($l = 1, 2, 3$) type of features $r_{j,l}$ may consist of n_l fields, denoted as $r_{j,l} = \{r_{j,l,1}, r_{j,l,2}, ..., r_{j,l,n_l}\}$. For instance, *the number of CV entries that a candidate has listed as work experiences* is one field of the 1-th type features (i.e., continuous features) in the resume r_j. Specially, for semantic features, the l-th ($l = 3$) type of features, consist of n_3 key words that extracted from the resumes (e.g., terms in the user's current job titles). Similarly, we use $p_{i,l} = \{p_{i,l,1}, p_{i,l,2}, ..., p_{i,l,m_l}\}$ denote the m_l fields of the l-th type of features in job posting p_i. Please note that the job posting p_i and resume r_j have common fields in the l-th($l = 2$) type of features.

Finally, all possible pairs of the resumes and jobs are denoted as the set of $M = \{p_i, r_j, y_{i,j}\}$. The label $y_{i,j} \in \{0, 1\}$ represents the matching result between job p_i and resume r_j, i.e., $y_{i,j} = 1$ means a matched pair of person-job, while $y_{i,j} = 0$ means a failed one. Hence, the target to study the Person-Job Fit problem is to predict the label $y_{i,j}$.

IV. The proposed model FINN

In this section, we will give an overview of our model FINN, which aims to predict the matching results between jobs and resumes. Then, we introduce the details of model FINN.

A. Overview

As illustrated in Fig.1, the proposed FINN model takes three types of features (i.e., numerical features, categorical features, and semantic entities) extracted from resumes and jobs as inputs. Followed by a feature embedding module that mapping the categorical data and semantic entities into two different hidden spaces respectively. Meanwhile, the numerical data are normalized with standard normal distribution. Next, the embedding of categorical and semantic features are fed into two different local interacting modules, which are utilized to learn the hidden representations of local interaction in two types of features. Finally, we concatenate the numerical features with these two hidden representations of job and resume, and fed them into a Multi-Layer Perceptron (MLP) to predict the matching score with a softmax function.

B. Feature Embedding Module

We first construct a bag of words that are extracted from all resumes and jobs. Then, we project the categorical features and semantic entities into one-hot vectors. Therefore, each job and resume is represented as a sparse vector by concatenating all types of features. Specifically,

$$\begin{aligned} \boldsymbol{X}^R &= [x^R_{1,1}; x^R_{1,2}; ...; x^R_{1,n_1}; ...; \boldsymbol{x}^R_{l,1}; \boldsymbol{x}^R_{l,2}; ...; \boldsymbol{x}^R_{l,n_l}] \\ \boldsymbol{X}^P &= [x^P_{1,1}; x^P_{1,2}; ...; x^P_{1,m_1}; ...; \boldsymbol{x}^P_{l,1}; \boldsymbol{x}^P_{l,2}; ...; \boldsymbol{x}^P_{l,m_l}] \end{aligned} \quad (1)$$

where m_l and n_l denote the number of fields in the l-th type features of jobs and resumes, and $l \in \{1, 2, 3\}$. $\boldsymbol{x}^P_{l,k}$ and $\boldsymbol{x}^R_{l,k}$ are the processed feature of the k-th field of the l-th type in jobs and resumes respectively. $\boldsymbol{x}^P_{l,k}$ and $\boldsymbol{x}^R_{l,k}$ are scalar values when $l = 1$. Otherwise, they are one-hot vectors in the type of categorical and semantic features.

Next, the one-hot vectors of each field are projected into a low-dimensional space, i.e.,

$$\begin{aligned} \boldsymbol{e}^R_{l,k} &= \boldsymbol{V}_{l,k}\boldsymbol{x}^R_{l,k} \\ \boldsymbol{e}^P_{l,k} &= \boldsymbol{V}_{l,k}\boldsymbol{x}^P_{l,k} \end{aligned} \quad (2)$$

where $\boldsymbol{V}_{l,k}$ is a embedding matrix of the k-th field of the l-th type of features in both jobs and resumes.

C. Local Interacting Module

After getting the feature embedding of resumes and jobs, we further extract the feature interactions in different feature types. For the categorical features, we design several operations to get the interactions. Please note that the number of fields in categorical features of resumes and jobs is equal. The mathematical definition is as follows:

$$\boldsymbol{h}_{l,k} = \boldsymbol{W}_{l,k}[\boldsymbol{e}^R_{l,k}; \boldsymbol{e}^P_{l,k}; \boldsymbol{e}^R_{l,k} \diamondsuit \boldsymbol{e}^P_{l,k}] + \boldsymbol{b}_{l,k} \quad (3)$$

where $\boldsymbol{h}_{l,k}$ denotes the hidden interaction representation of the k-th field in categorical features between jobs and resumes, and the $\diamondsuit$ indicates operators (i.e., plus, minus, element-wise product), just as shown in Fig 1. The $\boldsymbol{W}_{l,k}$ and $\boldsymbol{b}_{l,k}$ are parameters to train. Then, we concatenate all the representations learned in different fields of the categorical features, i.e.,

$$\boldsymbol{h}_l = \boldsymbol{h}_{l,1} \oplus \boldsymbol{h}_{l,2} \oplus ... \oplus \boldsymbol{h}_{l,m_l} \quad (4)$$

where $l = 2$ and m_l is the total number of common fields in categorical features between jobs and resumes. $\oplus$ indicates the concatenation among the hidden vectors. $\boldsymbol{h}_l$ is the representations of interaction between jobs and resumes based on the categorical features. Particularly, the $l = 2$ in equation 4 means the features are categorical.

Correspondingly, we construct another module to learn the interactions between resumes and jobs based on the semantic features consist of several key words or key phrases. For the semantic features, $\boldsymbol{e}^R_{l,k}$ and $\boldsymbol{e}^P_{l,k}$ are the embedding vector of the k-th word in resumes and jobs, and the l is 3 now. We represent the interaction between jobs and resumes with the embedding of words as a matching matrix $\boldsymbol{M} \in \mathbb{R}^{d \times d}$, with each element $\boldsymbol{M}_{i,j}$ standing for the basic interaction, i.e. similarity between word embedding $\boldsymbol{e}^R_{l,i}$ and $\boldsymbol{e}^P_{l,j}$.

$$\boldsymbol{M}_{i,j} = \boldsymbol{e}^R_{l,i} \otimes \boldsymbol{e}^P_{l,j} \quad (5)$$

where $\otimes$ indicates the cosine operator in this paper. The dimension of each word embedding is d.

Then, we utilize several one-dimensional convolutional layers with different sizes of kernels on the matching matrix $\boldsymbol{M}$ to capture neighbor feature interactions between jobs and resumes. A feature $\boldsymbol{c}_i$ is generated from a sub-matrix of $\boldsymbol{M}$:

$$\boldsymbol{c}_i = ReLU(\boldsymbol{W} \cdot \boldsymbol{M}[i : i + h - 1] + b_i) \quad (6)$$

where $i \in [0, d - h + 1]$, $\boldsymbol{W} \in \mathbb{R}^{d \times 1}$ and $b_i \in \mathbb{R}$ are the parameters. We use the non-linear activation function ReLU to process the result. Then a feature map is produced with this kernel:

$$\boldsymbol{C}^{(d-h+1)\times d} = [\boldsymbol{c}_1; \boldsymbol{c}_2; ...; \boldsymbol{c}_{d-h+1}] \quad (7)$$

For each feature map, we utlize the max-pooling strategy to extract features in the space. We can obtain several features with different window size h. These features are concatenated to a single vector denoted as $\boldsymbol{h}_l$, where $l = 3$.

D. Classification Module

With the processing of the Local Interacting Module, we can obtain the hidden vectors of the interactions between jobs and resumes based on different types of features. After that, we utilize a MLP fed with the concatenation of the normalized numerical features and the hidden vectors. We concatenate the normalized numerical features in resumes and jobs respectively:

$$\begin{aligned} \boldsymbol{x}^R_1 &= x^R_{1,1} \oplus x^R_{1,2} \oplus ... \oplus x^R_{1,n_1} \\ \boldsymbol{x}^P_1 &= x^P_{1,1} \oplus x^P_{1,2} \oplus ... \oplus x^P_{1,m_1} \end{aligned} \quad (8)$$

where $\oplus$ represents the operator of the concatenation. m_1 and n_1 are the number of numerical features in jobs and resumes respectively. Then, the concatenated single vector which is also the input of MLP can be formulated as:

$$\boldsymbol{z}^{(0)} = \boldsymbol{x}^R_1 \oplus \boldsymbol{x}^P_1 \oplus \boldsymbol{h}_2 \oplus \boldsymbol{h}_3 \quad (9)$$

where $\oplus$ represents the operator of the concatenation. $\boldsymbol{h}_2$ and $\boldsymbol{h}_3$ are the hidden vectors of the interactions between jobs and resumes based on categorical features and semantic features respectively.

The output of each layer in the MLP can be calculated through the following equation:

$$\boldsymbol{z}^{(l)} = ReLU(\boldsymbol{W}^{(l)}\boldsymbol{z}^{(l-1)} + \boldsymbol{b}^l) \quad (10)$$

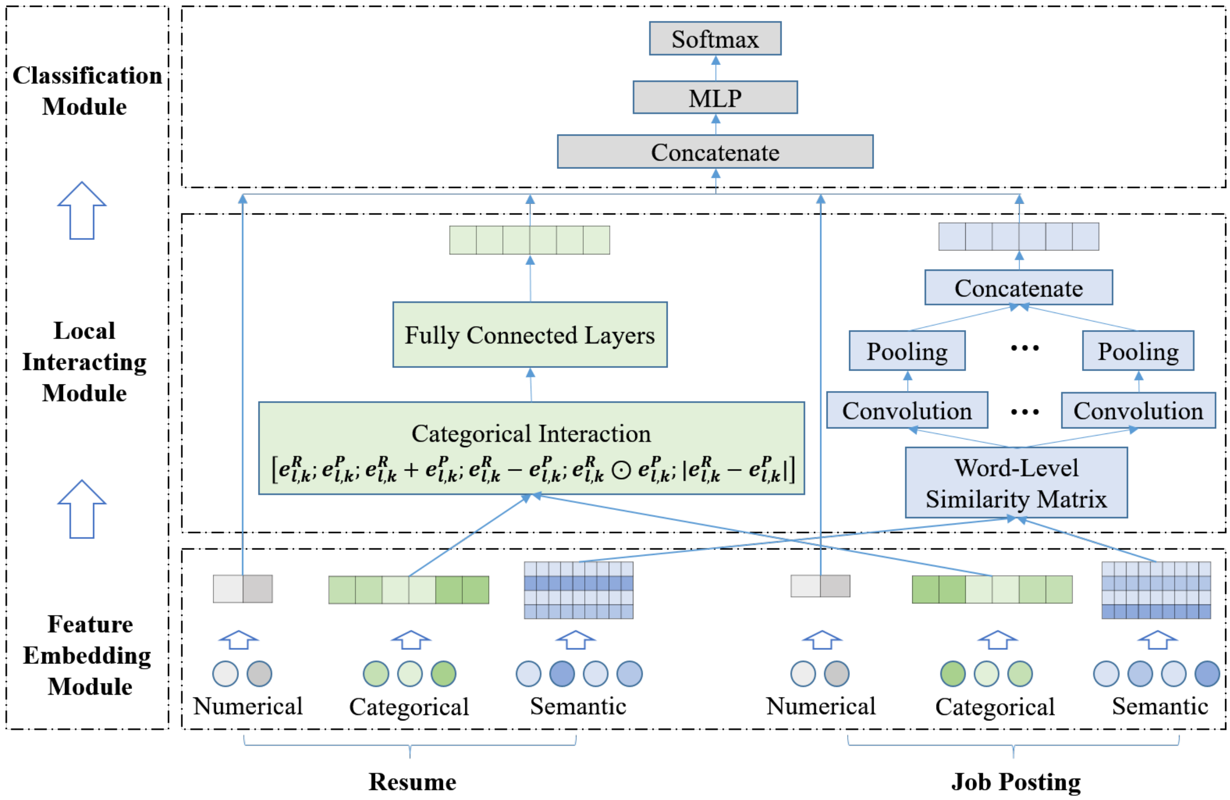

Fig. 1. An illustration of the our proposed model FINN, which consists of three components: Feature Embedding Module, Local Interacting Module, Classification Module. Meanwhile, we design two different parts to learn the local interacting representations of categorical features and semantic features between resume and job respectively.

where $\boldsymbol{z}^{(l)}$ indicates the output of the l-th layer, and the activation function is ReLU. $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^l$ are the parameters to train.

Following the previous studies for the Person-Job Fit problem[6], [7], [12], we also formulate the task as a binary classification. The matching score between jobs and resumes can be obtained as a two-dimensional vector:

$$(s_0, s_1)^\top = \boldsymbol{W}^{(L+1)}\boldsymbol{z}^{(L)} + \boldsymbol{b}^{(L+1)} \tag{11}$$

where L is the depth of the MLP.

The softmax function is applied to output the probability of belonging to each class:

$$p_k = \frac{e^{s_k}}{e^{s_0} + e^{s_1}} \tag{12}$$

where $k = 0, 1$.

Finally, to optimize the model, we set the objective function as cross-entropy loss:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \ln p_1 + (1 - y_i)\ln p_0) \tag{13}$$

where y_i is the matching label of the i-th training instance, while N is the total number of instances for training.

V. Experiments

In this section, to validate the effectiveness of model FINN, we will introduce the experiment details on a real-world dataset provided by XING published through the RecSys Challenge 2017. Specifically, we mainly answer the following research questions:

- **Question 1.** How does the model FINN perform compared with the baseline methods?
- **Question 2.** How does the two type of feature interaction modules affect the classification effectiveness?
- **Question 3.** How does the model perform under different parameter settings?

A. Data Description

This dataset includes three files, i.e., resumes of users, job postings, and historical interaction records between users and jobs. We give the details of the resumes in Table I. Besides, most of the feature fields in jobs are the same as resumes (i.e., the bold fields listed in Table I), except that jobs have additional location, type of employment, job title, and concepts extracted from tags, skills, and company name. The location information about a job is represented as latitude and longitude. The type of employment is a categorical feature, while job title and concepts are described as numeric IDs are regarded as semantic features just like the job roles in resumes. The interaction file contains four fields of features, i.e., resume id, job id, timestamp, and the type of interaction. Each interaction is one of six actions:{*impression,click,bookmark,reply,delete,recruiter interest*}. The first five types are generated by the job seekers, while the last one represents that the recruiter's interest in the resume.

TABLE I
FIELDS OF THE RESUMES

Type	Fields	Comments
numerical	experience-1	identifies the number of work experiences in resume
numerical	experience-2	the total work years of the user
numerical	experience-3	the work years in current job of the user
categorical	**id**	anonymized ID of the resume
categorical	**career level**	career level ID, e.g., Beginner, Manager
categorical	**discipline**	anonymized IDs represent disciplines such as Consulting
categorical	**industry**	anonymized IDs represent industries
categorical	**country**	describes the country in which the user is currently working
categorical	**region**	specifies for some users who work in Germany currently
categorical	**premium**	the user subscribed to XING's payed premium membership
categorical	wtcj	the user's willingness to change jobs
categorical	edu_degree	the university degree of the user
semantic	edu_fields	fields of studies that the user studied
semantic	job roles	list of terms extracted from the user's current job titles

In total, this dataset consists of 1.3M job postings, 1.5M resumes and over 322M person-job interaction records over a span of 94 days. We remove those incomplete resumes and jobs (e.g., jobs without title or location information). Finally, we get the pruned dataset consists of 452,214 job postings, 651,711 resumes, and 4,576,489 positive interactions. Following the idea of prior study[41], we also adopt the strategy of random sampling to generate the negative instances that are equal to the number of positive ones. In this article, a positive sample means that user and recruiter have some positive action on each other, such as *click, bookmark, reply, and recruiter interest*. The time distribution of positive interactions between jobs and resumes is shown in Fig 2(a). Clearly, most of the positive interactions are in November of 2016 and January of 2017. In addition, Fig 2(b), 2(c), and 2(d) summarize the words distribution of resumes and jobs respectively.

We utilize a time-based split on the dataset to get train, valid, and test sets. Specifically, we select the interaction pairs from the last 14 days as the test set, and use the interactions from the first 66 days for training. The remaining interaction pairs are utilized as the valid set.

B. Experimental Setup

In this part, we will introduce the details of our experiments, including the parameters of our model, baseline methods, as well as evaluation metrics.

First, we explain the parameters of our model. From Fig 2(b), and 2(c), we can observe that most of the words number of job roles and job title are less than 10. Hence, the maximum number of words is set to 10. As illustrated in Fig 2(d), about 95% of jobs' tags have a length with less than 40 words, we set the maximum words number as 40 for tags. In the feature embedding module, both categorical data and semantic data (i.e., a number of words) are transferred to a dense vector representation. The length of word vectors is set to 20, and we set the size of categorical feature embedding as 10. In addition, we set the learning rate and batch size as 0.001 and 1024 respectively.

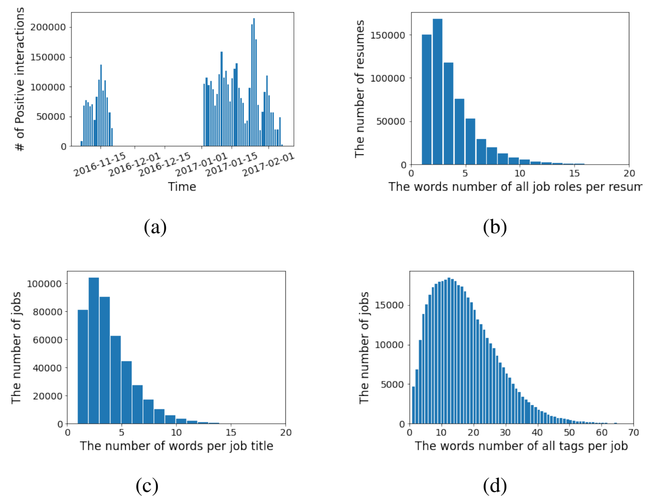

Fig. 2. (a):The time distribution of positive interactions between jobs and resumes. (b):The words distribution of job roles in resume. (c):The words distribution of job title. (d):The words distribution of job tags.

To validate the performance of FINN, we select several classic supervised learning methods as the baselines, including Decision Tree (**DT**), Multinomial Naive Bayes (**MultiNB**), Logistical Regression (**LR**), Random Forests (**RF**), Gradient Boosting Decision Tree(**GBDT**), Extreme Gradient Boosting(**XGBoost**), and Light Gradient Boosting Machine (**LightGBM**).

We quantify the prediction performance of each model with following evaluation metrics: **Accuracy**, **F1-measure**, and **AUC**.

C. Experimental Results

Overall Performance Comparisons (RQ1). We give the summarization of the overall performance in Table II, where the bold numbers are the best results in terms of different evaluation metrics. Please note that all the results of our model shown in this paper are the average values of 20 independent runs. According to the results, our proposed model FINN outperforms all the baseline methods. It seems that our designed two local interaction modules can improve the performance with a better prediction of matching scores between jobs and resumes. The improvements of our model compared with the worst one are 15.80%, 14.92%, and 24.46% in terms of Accuracy, F1-measure, and AUC, respectively. Besides, the FINN outperforms the best supervised model LightGBM by a boost of 3.50%, 1.96%, and 2.52% for the three metrics, respectively.

The Effectiveness of Two Local Interacting Modules (RQ2). To measure the contribution of each local interacting module, we construct variants of FINN as follows:

- *FINN_cat* is the variant that only use the local interacting module based on the categorical features.

TABLE II
THE OVERALL PERFORMANCE OF ALL METHODS

Methods	ACC	F1	AUC
DT	0.7140	0.7044	0.7142
MultiNB	0.7180	0.7082	0.7958
LR	0.7581	0.7481	0.8260
RF	0.7550	0.7528	0.8226
GBDT	0.7849	0.7792	0.8653
XGboost	0.7850	0.7788	0.8654
LightGBM	0.7988	0.7940	0.8810
Our Model	**0.8268**	**0.8095**	**0.9032**

- *FINN_word* is the variant that only user the local interacting module base on the semantic features.

The results of different variants of FINN are shown in Fig.3(a). We can observe that gradually removing the local interacting module significantly reduces the performance. *FINN_cat* that only using the local interacting module based on categorical features reduces the Accuracy, F1, and AUC about 4.97%, 6.00%, and 4.47%, respectively. FINN gets a boost of 10.53%, 15.76%, and 10.95% in terms of the metric Accuracy, F1 and AUC than the *FINN_word*. It shows the effectiveness of involving both of the local interacting modules by several interacting operators and a convolutional neural network.

Impact of Different Parameter Settings (RQ3). To validate the performance of FINN with different hyper-parameter settings, we take the learning rate as an example. We try several learning rates to train the model FINN. According to the results illustrated in Fig. 3(b), 3(c), and 3(d), the performance of our model improves with the reducing of learning rate, when the value is bigger than 0.001. When learning rate $lr < 0.001$, the performance of FINN does not change stably. In summary, the value of all evaluation metrics get best when $lr = 0.001$.

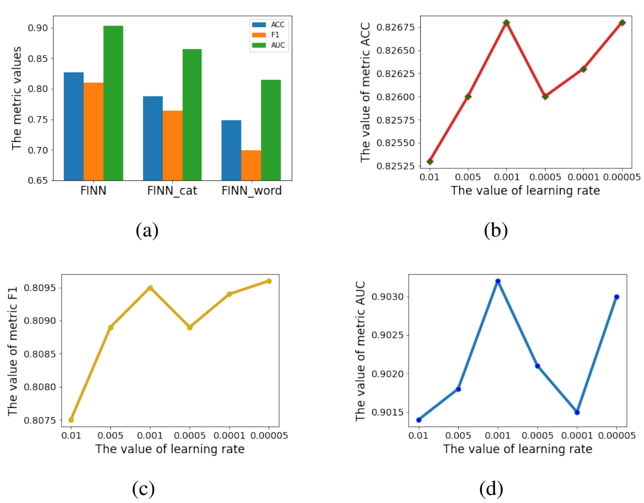

Fig. 3. (a):The performance of FINN and its variants. (b): The value of metric accuracy w.r.t. the learning rate. (c): The value of metric F1 w.r.t. the learning rate. (d): The value of metric AUC w.r.t. the learning rate.

VI. CONCLUSION

In this paper, we proposed a model called **FINN** as an abbreviation for **F**eature **I**nteraction **N**eural **N**etwork for Person-Job Fit problem. FINN transforms the Person-Job Fit to a binary classification task. The categorical and semantic features were projected into two different low dimensional spaces with two parameter matrices, respectively. Then, we designed two different modules to capture the feature interactions between jobs and resumes based on the categorical and semantic features, respectively. We further concatenated the normalized numerical features with these two hidden vectors output from local interaction modules as a single vector. Moreover, we utilize the MLP with a softmax activation function to measure the mathing scores between jobs and resumes. Finally, we validated the effectiveness of FINN based on a real-world dataset that has been widely used on the ACM RecSys Challenge 2017.

ACKNOWLEDGEMENT

This work was supported by the National Social Science Found of China (number: 2020-SKJJ-C-103).

REFERENCES

[1] J. Malinowski, T. Keim, O. Wendt, and T. Weitzel, "Matching people and jobs: A bilateral recommendation approach," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, vol. 6. IEEE, 2006, pp. 137c–137c.

[2] D. H. Lee and P. Brusilovsky, "Fighting information overflow with personalized comprehensive information access: A proactive job recommender," in *Third International Conference on Autonomic and Autonomous Systems (ICAS'07)*. IEEE, 2007, pp. 21–21.

[3] M. Diaby, E. Viennet, and T. Launay, "Toward the next generation of recruitment tools: An online social network-based job recommender system," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2013.

[4] C. Ozcaglar, S. Geyik, B. Schmitz, P. Sharma, A. Shelkovnykov, Y. Ma, and E. Buchanan, "Entity personalized talent search models with tree interaction features," in *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, vol. 2, 2019, pp. 3116–3122.

[5] Y. Luo, Y. Wen, H. Zhang, and X. Zhang, "Resumegan: An optimized deep representation learning framework for talent-job fit via adversarial learning," in *Proceedings of International Conference on Information and Knowledge Management*, 2019, pp. 1101–1110.

[6] C. Zhu, H. Zhu, H. Xiong, C. Ma, F. Xie, P. Ding, and P. Li, "Person-Job Fit: Adapting the Right Talent for the Right Job with Joint Representation Learning," *ACM Transactions on Management Information Systems*, vol. 9, no. 3, pp. 1–17, 2018.

[7] C. Qin, H. Zhu, T. Xu, C. Zhu, L. Jiang, E. Chen, and H. Xiong, "Enhancing person-job fit for talent recruitment: An ability-aware neural network approach," in *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*, 2018, pp. 25–34.

[8] S. Bian, W. X. Zhao, Y. Song, T. Zhang, and J. R. Wen, "Domain adaptation for person-job fit with transferable deep global match network," in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019, pp. 4810–4820.

[9] R. Le, T. Zhang, W. Hu, D. Zhao, Y. Song, and R. Yan, "Towards effective and interpretable person-job fitting," in *Proceedings of the 28th International Conference on Information and Knowledge Management*, 2019, pp. 1883–1892.

[10] R. Yan, T. Zhang, R. Le, X. Zhang, Y. Song, and D. Zhao, "Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 914–922.

[11] S. Bian, X. Chen, W. X. Zhao, K. Zhou, Y. Hou, Y. Song, T. Zhang, and J.-r. Wen, "Learning to Match Jobs with Resumes from Sparse Interaction Data using Multi-View Co-Teaching Network," in *Proceedings of the 29th International Conference on Information and Knowledge Management*, 2020.

[12] 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*, vol. 38, no. 2, p. 33, 2020.

[13] W. Guo, X. Liu, S. Wang, H. Gao, A. Sankar, Z. Yang, Q. Guo, L. Zhang, B. Long, B. C. Chen, and D. Agarwal, "DeText: A Deep Text Ranking Framework with BERT," in *International Conference on Information and Knowledge Management, Proceedings*, 2020, pp. 2509–2516.

[14] K. Kenthapadi, B. Le, and G. Venkataraman, "Personalized job recommendation system at LinkedIn: Practical challenges and lessons learned," in *Proceedings of the 11th ACM Conference on Recommender Systems*, 2017, pp. 346–347.

[15] T. Mine, T. Kakuta, and A. Ono, "Reciprocal recommendation for job matching with bidirectional feedback," in *Proceedings - 2nd IIAI International Conference on Advanced Applied Informatics, IIAI-AAI 2013*, 2013, pp. 39–44.

[16] R. Rafter, K. Bradley, and B. Smyth, "Automated collaborative filtering applications for online recruitment services," in *AH 2000: Adaptive Hypermedia and Adaptive Web-Based Systems*, vol. 1892, 2000, pp. 363–368.

[17] Y. Zhang, C. Yang, and Z. Niu, "A research of job recommendation system based on collaborative filtering," in *2014 Seventh International Symposium on Computational Intelligence and Design*, vol. 1, 2014, pp. 533–538.

[18] B. Heap, A. Krzywicki, W. Wobcke, M. Bain, and P. Compton, "Combining career progression and profile matching in a job recommender system," in *Pacific Rim International Conference on Artificial Intelligence PRICAI 2014: Trends in Artificial Intelligence*, vol. 8862, 2014, pp. 396–408.

[19] M. He, D. Shen, Y. Zhu, R. He, T. Wang, and Z. Zhang, "Career trajectory prediction based on cnn," in *2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 2019, pp. 22–26.

[20] W. Chao, Z. Hengshu, H. Qiming, X. Keli, and X. Hui, "Variable Interval Time Sequence Modeling for Career Trajectory Prediction : Deep Collaborative Perspective," in *WWW '21: Proceedings of the Web Conference 2021*, 2021, pp. 612–623.

[21] 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*, p. S095070511730374X, 2017.

[22] W. Xiao, X. Xu, K. Liang, J. Mao, and J. Wang, "Job recommendation with Hawkes process: an effective solution for RecSys Challenge 2016," in *ACM International Conference Proceeding Series*, 2016, pp. 1–4.

[23] V. Leksin and A. Ostapets, "Job recommendation based on factorization machine and topic modelling," in *ACM International Conference Proceeding Series*, 2016, pp. 5–8.

[24] A. Pacuk, P. Sankowski, K. Wegrzycki, A. Witkowski, and P. Wygocki, "RecSys Challenge 2016: Job recommendations based on preselection of offers and gradient boosting," in *ACM International Conference Proceeding Series*, vol. 098, no. 327 003, 2016, pp. 3–6.

[25] V. S. Dave, M. Al Hasan, B. Zhang, K. AlJadda, and M. Korayem, "A combined representation learning approach for better job and skill recommendation," in *International Conference on Information and Knowledge Management*, no. August, 2018, pp. 1997–2006.

[26] N. D. Almalis, G. A. Tsihrintzis, and N. Karagiannis, "A content based approach for recommending personnel for job positions," in *IISA 2014 - 5th International Conference on Information, Intelligence, Systems and Applications*, 2014, pp. 45–49.

[27] M. Volkovs and T. Poutanen, "Content-based Neighbor Models for Cold Start in Recommender Systems," in *Proceedings of the Recommender Systems Challenge 2017*, 2017.

[28] H. Lu and M. Zhang, "How Integration helps on Cold-Start Recommendations," in *Proceedings of the Recommender Systems Challenge 2017*, 2017, pp. 0–5.

[29] J. Lian, M. Hou, and H. Wang, "Practical Lessons for Job Recommendations in the Cold-Start Scenario," in *Proceedings of the Recommender Systems Challenge 2017*, 2017.

[30] V. Ha-Thuc, G. Venkataraman, M. Rodriguez, S. Sinha, S. Sundaram, and L. Guo, "Personalized expertise search at LinkedIn," in *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*, 2015, pp. 1238–1247.

[31] V. Ha-Thuc, Y. Xu, S. P. Kanduri, X. Wu, V. Dialani, Y. Yan, A. Gupta, and S. Sinha, "Search by Ideal Candidates:Next Generation of Talent Search at LinkedIn," in *www'16*, 2016, pp. 195–198.

[32] C. F. Chien and L. F. Chen, "Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry," *Expert Systems with Applications*, vol. 34, no. 1, pp. 280–290, 2008.

[33] A. Singh, R. Catherine, K. Visweswariah, V. Chenthamarakshan, and N. Kambhatla, "PROSPECT: A system for screening candidates for recruitment," in *International Conference on Information and Knowledge Management*, no. October 2017, 2010, pp. 659–668.

[34] X. Song, L. Nie, L. Zhang, M. Akbari, and T. S. Chua, "Multiple social network learning and its application in volunteerism tendency prediction," in *SIGIR 2015 - Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 213–222.

[35] X. Song, Z. Y. Ming, L. Nie, Y. L. Zhao, and T. S. Chua, "Volunteerism Tendency Prediction via Harvesting Multiple Social Networks," *ACM Transactions on Information Systems*, vol. 34, no. 2, 2016.

[36] Y. Jia, X. Song, J. Zhou, L. Liu, L. Nie, and D. S. Rosenblum, "Fusing social networks with deep learning for volunteerism tendency prediction," in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016, pp. 165–171.

[37] R. Ramanath, H. Inan, G. Polatkan, B. Hu, Q. Guo, C. Ozcaglar, X. Wu, K. Kenthapadi, and S. C. Geyik, "Towards deep and representation learning for talent search at LinkedIn," in *International Conference on Information and Knowledge Management, Proceedings*, 2018, pp. 2253–2262.

[38] S. Maheshwary and H. Misra, "Matching Resumes to Jobs via Deep Siamese Network," in *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018*, 2018, pp. 87–88.

[39] A. Khatua and W. Nejdl, "Matching Recruiters and Jobseekers on Twitter," in *Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2020*, 2020, pp. 266–269.

[40] J. Jiang, W. Wang, J. Xu, and X. Luo, "Learning Effective Representations for Person-Job Fit by Feature Fusion," in *Proceedings of the 29th International Conference on Information and Knowledge Management*, 2020.

[41] J. Lian, F. Zhang, M. Hou, H. Wang, X. Xie, and G. Sun, "Practical lessons for job recommendations in the cold-start scenario," in *Recommender Systems Challenge*, 2017, pp. 1–6.