A Feature Interaction Based Neural Network Approach: Predicting Job Turnover in Early Career Graduates in South Korea

Haewon Byeon
Department of AI-Software, Inje Medical Big Data Research Center, Inje University, Gimhae 50834, South Korea

***Abstract*—Predicting job turnover among early career university graduates is crucial for both employees and employers. This study introduced a Feature Interaction based Neural Network model designed to predict job turnover among university graduates in their 20s and 30s in South Korea within the first five years of employment. The FINN model leveraged the Graduates Occupational Mobility Survey dataset, which included detailed information on approximately 26,544 graduates. This rich dataset encompassed a wide range of variables, including personal attributes, employment characteristics, job satisfaction, and job preparation activities. The model combined an embedding layer to convert sparse features into dense vectors with a neural network component to capture high-order feature interactions. We compared the FINN model's performance against eight baseline models: Logistic Regression, Factorization Machines, Field-aware Factorization Machines, Support Vector Machine, Random Forest, Product-based Neural Networks, Wide & Deep, and DeepFM. Evaluation metrics used were Area Under the ROC Curve (AUC) and Log Loss. The results demonstrated that the FINN model outperformed all baseline models, achieving an AUC of 0.830 and a Log Loss of 0.370. The FINN model represents a significant advancement in predictive modeling for job turnover, providing valuable insights that can inform both individual career planning and organizational human resource practices. This research underscores the potential of advanced neural network architectures in employment data analysis and predictive modeling.**

Keywords*—Job turnover prediction; feature interaction based neural network; employment data analysis; predictive modeling; university graduates

I. Introduction

The advent of big data has revolutionized various sectors, including the labor market, where the analysis of employment data plays a crucial role in understanding workforce dynamics and predicting future trends [1,2]. This study focuses on the early career trajectories of university graduates in their 20s and 30s in South Korea, particularly their likelihood of job turnover within the first five years of employment. The accurate prediction of job turnover is paramount for both employees and employers [3-5]. Employees benefit by understanding potential career pitfalls, while employers can devise better retention strategies, thereby reducing recruitment and training costs [3,4].

Employees stand to gain significantly from insights into job turnover predictors. By identifying the factors that contribute to early job departure, graduates can better prepare themselves to meet the challenges of their initial employment experiences [6,7]. This knowledge empowers them to seek roles and environments that align more closely with their career aspirations and stability [8,9]. For employers, the implications of job turnover are substantial. High turnover rates can lead to increased recruitment costs, loss of organizational knowledge, and diminished productivity. By accurately predicting which employees are at risk of leaving, employers can implement targeted interventions to improve job satisfaction and engagement [10-12]. This proactive approach not only enhances employee retention but also fosters a more stable and committed workforce. The dual benefits of predicting job turnover—enhancing employee career stability and optimizing employer retention strategies—underscore the importance of this research. By leveraging advanced predictive models, this study seeks to provide valuable insights that can inform both individual career planning and organizational human resource practices.

Historically, regression analysis has been a popular method for studying job turnover. Regression models, such as logistic regression, have been widely used to identify factors influencing employee turnover by modeling the relationship between dependent and independent variables [13-16]. For instance, logistic regression can help in estimating the probability of an event (like job turnover) occurring, given a set of predictor variables (such as age, education level, job satisfaction, etc.) [13-16]. However, while regression analysis has its merits, it also has significant methodological limitations when applied to complex, high-dimensional datasets typical in employment studies [7]. One of the primary limitations of traditional regression analysis is its inability to effectively capture complex interactions between features [7]. In employment data, factors influencing job turnover are often interdependent. For example, the interaction between job satisfaction and work-life balance might significantly affect turnover rates, but such interactions can be challenging to model accurately using simple regression techniques. Regression models assume a linear or specific non-linear relationship between the independent and dependent variables, which may not hold true in real-world scenarios where relationships can be highly non-linear and intricate.

Moreover, regression models often require extensive feature engineering to improve their predictive accuracy [16]. Feature engineering involves the manual creation of new features from raw data to better represent the underlying patterns [17]. This process can be labor-intensive and requires domain expertise to identify meaningful interactions and transformations [18]. Despite these efforts, the performance of regression models may still be limited due to their inherent inability to model complex, higher-order interactions between features [19].

To address these limitations, there has been a growing interest in leveraging advanced machine learning techniques, particularly deep learning models, which can automatically learn feature interactions from raw data without extensive manual intervention. Among these, Feature Interaction based Neural Networks (FINNs) have shown great promise [20]. FINNs enhance the capabilities of traditional deep neural networks (DNNs) by explicitly modeling feature interactions, thereby improving predictive accuracy in complex datasets [21].

Deep neural networks have achieved remarkable success in various fields, such as image classification, natural language processing (NLP), and speech recognition, due to their ability to learn hierarchical feature representations [22]. In the context of employment data, DNNs can be particularly useful as they can capture complex, non-linear relationships between features. However, one of the challenges in applying DNNs to employment data is the sparsity and high-dimensionality of the data. Employment datasets often contain categorical variables, such as job title, industry, and education level, which are typically converted into high-dimensional sparse features using techniques like one-hot encoding. These sparse features need to be transformed into dense representations before being fed into the neural network.

FINNs address this challenge by employing a feature embedding layer that converts sparse categorical features into dense vectors [20]. These embeddings are then used to model pairwise interactions between features, capturing the complex dependencies that influence job turnover [20]. By incorporating a feature interaction layer, FINNs can learn both low-order and high-order interactions, providing a more comprehensive understanding of the factors driving employee turnover.

The need for FINNs is underscored by the limitations of traditional methods. For example, factorization machines (FMs) have been proposed to model feature interactions via the inner product of feature embeddings, but they primarily capture only second-order interactions [21]. While FMs have been successful in some applications, they may not fully exploit the higher-order interactions present in employment data. In contrast, FINNs can model both second-order and higher-order interactions, providing a more robust framework for predicting job turnover [21]. Furthermore, the integration of deep learning components in FINNs allows for the modeling of non-linear interactions and complex feature hierarchies, which are often present in employment data. This capability is particularly important for understanding the multifaceted nature of job turnover, where factors such as job satisfaction, career development opportunities, and organizational culture interplay in intricate ways.

FINNs work by first employing an embedding layer to transform high-dimensional sparse features into dense vectors [20,21]. This transformation is crucial for handling the sparsity issue inherent in employment data. Once the features are embedded, FINNs apply a feature interaction layer that captures pairwise interactions between the dense vectors. This layer can utilize operations such as inner product or element-wise product to model the interactions. By doing so, FINNs can effectively represent the complex relationships between features, which traditional regression models might miss. Moreover, FINNs extend the capability of simple interaction models by incorporating deep neural network components that can capture higher-order interactions [21]. This means that after modeling the basic pairwise interactions, the network can further process these interactions through multiple layers to extract more complex patterns. This deep architecture allows FINNs to model non-linear relationships and hierarchies among features, enhancing their predictive power. The novelty of FINNs lies in their ability to combine the strengths of traditional interaction models and deep learning. Unlike traditional models that require extensive manual feature engineering to capture interactions, FINNs can automatically learn these interactions from data. This reduces the need for domain expertise and manual intervention, making the modeling process more efficient and scalable.

While traditional regression methods have provided valuable insights into employee turnover, their limitations necessitate the adoption of more advanced techniques like Feature Interaction based Neural Networks. FINNs offer a powerful alternative by automatically learning complex feature interactions from high-dimensional data, thereby enhancing predictive accuracy and providing deeper insights into the factors influencing job turnover. This study aims to leverage FINNs to analyze the early career trajectories of university graduates in South Korea, with the goal of identifying key predictors of job turnover and informing strategies to improve employee retention. The remainder of this paper is organized as follows: Section II reviews related works. Section III describes the details of the proposed model. Section IV presents the experimental analysis. Finally, Section V concludes the paper.

II. Related Work

Logistic Regression (LR) is a fundamental technique widely used in classification tasks, including click-through rate prediction. It is a linear model that solves an unconstrained convex optimization problem, ensuring efficient convergence to a globally optimal solution via gradient descent [23]. The primary advantage of LR is its interpretability; by examining the weights assigned to each feature, one can understand the significance and impact of these features on the prediction outcome [23]. This transparency makes LR particularly valuable in fields where interpretability is crucial, such as field of employment. However, the linear nature of LR limits its ability to capture complex relationships between features [24]. To overcome this, extensive feature engineering is often required, including polynomial features and interaction terms, to improve the model's expressiveness.

Another shallow method worth mentioning is Polynomial Regression, which extends linear regression by considering polynomial terms of the features [25]. By including polynomial terms, the model can capture non-linear relationships between the features and the target variable. Polynomial regression can be particularly useful when the relationship between the features and the outcome is known to be non-linear. However, as the degree of the polynomial increases, the model becomes more complex and prone to overfitting, especially with limited data [25].

Decision Trees are another fundamental shallow method used for classification and regression tasks. A decision tree splits the data into subsets based on feature values, creating a tree-like model of decisions [26]. Each node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome. Decision trees are easy to interpret and visualize, making them useful for understanding the decision-making process. However, they can be prone to overfitting, especially with deep trees [27]. Techniques such as pruning, bagging, and boosting are often used to mitigate overfitting and improve performance.

Ensemble methods, such as Random Forests and Gradient Boosting Machines (GBM), build on the strengths of decision trees while addressing their limitations [28]. Random Forests create multiple decision trees using different subsets of the data and features, and then aggregate their predictions [29,30]. This approach reduces overfitting and improves generalization. GBM, on the other hand, builds trees sequentially, with each tree attempting to correct the errors of the previous ones [31]. This method is highly effective for both classification and regression tasks but can be computationally intensive.

Field-aware Factorization Machines (FFM) extend the capabilities of FM by introducing the concept of fields [32-34]. In FM, a feature interacts with other features using the same vector, whereas in FFM, a feature uses different vectors to interact with features from different fields [33]. This distinction allows FFM to model interactions more precisely, enhancing the model's expressiveness [34]. For instance, in a recommendation system, user-related features and item-related features can interact differently depending on their respective fields. However, the enhanced expressiveness of FFM comes at the cost of increased memory requirements, which can be a significant limitation when dealing with very large datasets [32].

In summary, shallow methods like Logistic Regression, Factorization Machines, and Decision Trees provide foundational techniques for modeling interactions and making predictions. While they offer interpretability and simplicity, their expressiveness is often limited, necessitating extensive feature engineering or the use of ensemble techniques to capture complex relationships. The ongoing development of these methods continues to enhance their applicability across various domains, from recommendation systems to employment analytics.

III. Proposed Method

Our main objective in this study is to model the feature interaction representation more effectively to predict job turnover among early career university graduates in South Korea. To achieve this, we propose a Feature Interaction based Neural Network (FINN) tailored for employment data analysis.

A. Sparse Input and Embedding Layer

Unlike image classification or speech recognition tasks, the input data in employment prediction tasks are usually non-contiguous and categorical. These raw input features are typically converted into high-dimensional sparse features via one-hot encoding. One-hot encoding is a process that transforms categorical variables into a binary vector representation, where only one element is "hot" (i.e., set to 1) and all other elements are "cold" (i.e., set to 0). For instance, consider the following categorical variables:

User ID = 001, 002, ...

Job Type = Engineer, Teacher, ...

Gender = Male, Female

Using one-hot encoding, an input instance can be transformed as follows:

User ID = 001 → [1, 0, 0, ...]

Job Type = Engineer → [1, 0, 0, ...]

Gender = Female → [0, 1]

The dimension of these features, particularly the user ID and job type, becomes large after encoding. For instance, if there are 550 job types, the dimension of the job type feature increases to 550, with only one of these values being effective. This results in extremely sparse feature vectors, where the majority of the elements are zero. The sparsity of the coded feature suggests that deep neural networks (DNNs) are not directly applicable because DNNs typically require dense input vectors to perform effectively.

Therefore, these sparse features are embedded into a continuous, dense, real-value vector space with lower dimensions. The embedding layer transforms these high-dimensional sparse vectors into dense vectors. This transformation is achieved by mapping each categorical value to a dense vector of fixed size. The embedding layer can be represented as:

$$\left[\mathrm{E} = [\mathrm{e}_1, \mathrm{e}_2, \dots, \mathrm{e}_\mathrm{I}, \dots, \mathrm{e}_\mathrm{m}] \right]$$

where (m) denotes the number of fields, (e_i\in \mathbb{R}^k) denotes the embedding vector of the (i)-th field, and (k) is the dimension of the embedding vector. The embedding process can be visualized as follows:

Each unique value in a categorical feature is assigned a unique dense vector.

These dense vectors are learned during the training process, allowing the model to capture semantic similarities between different categorical values.

The resulting embedded vectors are then concatenated to form a dense representation of the original sparse input.

For example, consider a feature vector with three categorical variables: user ID, job type, and gender. After one-hot encoding and embedding, the resulting dense vector might look like this:

$$\left[\{\text{User ID embedding}\} = [0.1,0.3,0.5] \right]$$

[{Job Type embedding} = [0.2,0.4,0.6]]

[{Gender embedding} = [0.3,0.7]]

These embeddings are then concatenated to form a single dense vector:

[E = [0.1,0.3,0.5,0.2,0.4,0.6,0.3,0.7]]

The dimension of the dense vector is much smaller than the original sparse vector, making it more suitable for input into a deep neural network. The embedding layer not only reduces the dimensionality of the input features but also captures latent relationships between different categorical values, which can be crucial for accurate prediction.

B. Feature-Interaction Layer

To improve prediction accuracy, it is crucial to model feature interactions after the embedding layer. The feature-interaction layer aims to model second-order feature relationships. Intuitively, we can represent the interaction of the (i)-th feature and the (j)-th feature using a vector (p_{ij}). However, due to data sparsity, training (p_{ij}) directly is impractical.

Instead, we use embedding vectors to calculate interaction vectors through methods like inner product and element-wise product. These methods are defined as:

$$\left[f_{\text{\textbackslash text\{inner\}}(x)} = \sum_{\{i,j \in X\}(\text{\textbackslash mathbf\{v\}}i \cdot \text{\textbackslash mathbf\{v\}}j)x_i x_j} \right]$$

$$\left[f\{\text{\textbackslash text\{element} - \text{wise\}}\}(x) = \textstyle\sum\{i,j \in X\}\left(\text{\textbackslash mathbf\{v\}}_i \circ \text{\textbackslash mathbf\{v\}}_j\right)x_i x_j \right]$$

where (X)is the set of features, (\mathbf{v}$_i$)and (\mathbf{v}$_j$)are the embedding vectors, and (\circ) denotes the element-wise product. These methods can be too simple to effectively calculate feature interactions, so we propose a more sophisticated method:

[\mathbf{p} {ij}= [p{ij}^1,p_{ij}^2,…,p_{ij}^l]]

[p_{ij}^u= \mathbf{v}_(i_j^(T\mathbf{W}^u\mathbf{v}))]

where (\mathbf{W}\in \mathbb{R}^{k \times k \times l}) is a three-dimensional tensor. Each slice (\mathbf{W}^u) represents a relation matrix.

C. Combination Layer and Deep Network

The interaction vectors (\mathbf{p})are concatenated and fed into a deep neural network (DNN). The combination layer merges the outputs of the feature-interaction layer:

[\mathbf{c}= [c_1,c_2,…,c_k]]

The deep network captures higher-order interactions between features. The fully connected layers are defined as:

$$\left[\text{\textbackslash mathbf\{h\}}^{\{(l)\}} = \sigma\left(\text{\textbackslash mathbf\{W\}}^{\{(l)\}\text{\textbackslash mathbf\{h\}}^{\{(l-1)\}}} + \text{\textbackslash mathbf\{b\}}^{\{(l)\}}\right) \right]$$

where (σ)is the activation function, (mathbf{W}$^{\{(l)\}}$)and (\mathbf{b}$^{\{(l)\}}$) are the weight matrix and bias vector of the (l)-th layer. The DNN captures high-order feature interactions through non-linear activation functions like ReLU, sigmoid, or tanh.

Finally, the output vector of the last neural network layer is used to calculate the prediction score:

$$\left[y_d = \sigma\left(\text{\textbackslash mathbf\{W\}}^{\{(L+1)\}\text{\textbackslash mathbf\{h\}}^{\{(L)\}}} + \text{\textbackslash mathbf\{b\}}^{\{(L+1)\}}\right) \right]$$

D. Output Layer and Learning

The overall formulation of the FINN model output is:

[y = σ (w_0+ ∑_{i=1}^n w_i x_i+ y_d)]

where (y) is the predicted probability of job turnover, (\sigma) is the sigmoid function, (n) is the number of features, and (w_i) are the weights of the sparse features. The loss function we aim to minimize is the binary cross-entropy loss:

[\text{loss}=

-∑_{x ∈X}[y_(i(x)\log(((┤))^{y}_i(x))))

+ (1 - y_i(x))\log(1 - () ^{y}_i(x))]]

where ($y_{i(x)}$)is the ground truth, ($\{y\}_{i(x)}$) is the predicted value,and (X)is the set of training instances.

To optimize the model, we use Mini-Batch Gradient Descent with the Adam optimizer. Adam combines RMSProp and momentum methods, adjusting the learning rate adaptively:

$$\left[\text{\textbackslash mathbf\{m\}t} = \beta_{1\text{\textbackslash mathbf\{m\}\{t-1\}}} + (1 - \beta_1)\text{\textbackslash mathbf\{g\}}_t \right]$$

$$\left[\text{\textbackslash mathbf\{v\}t} = \beta_{2\text{\textbackslash mathbf\{v\}\{t-1\}}} + (1 - \beta_2)\,\text{\textbackslash mathbf\{g\}}_t^2 \right] \left[(\frown)\{\text{\textbackslash mathbf\{m\}}\}_t = \text{\textbackslash frac\{\textbackslash mathbf\{m\}}_t\text{\}\{}1 - \beta_1^t \right]$$

$$\left[(\frown)\{\text{\textbackslash mathbf\{v\}}\}_t = \text{\textbackslash frac\{\textbackslash mathbf\{v\}t\}\{}1 - \beta_2^t \right]$$

$$\left[\theta_t = \theta\{t-1\} - \alpha\text{\textbackslash frac}\left\{ \hat{}\{\text{\textbackslash mathbf\{m\}}\}_t \right\} \left\{ \sqrt{\left\{ \hat{}\{\text{\textbackslash mathbf\{v\}}\}_t \right\}} + \epsilon \right\} \right]$$

where (β_1)and (β_2)are decay rates, (α)is the learning rate, (ϵ)is mathbf{g}$_t$)is the gradient.

We also apply dropout and batch normalization to prevent overfitting and stabilize training. Dropout randomly drops neurons during training with a probability (p), and batch normalization normalizes intermediate layer outputs.

IV. EXPERIMENTS

A. Dataset and Participants

This study utilizes data from the Graduates Occupational Mobility Survey (GOMS) conducted by the Korea Employment Information Service. The dataset includes approximately 5% of the 500,000 students who graduated from two-year and four-year colleges between August 2014 and February 2015, resulting in a sample size of 28,549 individuals. The survey was conducted in September and October 2016. The GOMS dataset is comprehensive, encompassing a wide range of variables that influence labor market entry and retention. These variables include academic background, current economic activity, job characteristics, job search activities, and individual demographics. The dataset's richness allows for a detailed analysis of the factors influencing job turnover among recent graduates. Participants were selected based on the following criteria: (1) they must have graduated after January 2014 and have secured their first job post-graduation. (2) Furthermore, only those employed in regular, full-time positions without fixed-term contracts were included in the study. This selection criterion ensures that the analysis focuses on stable employment scenarios, eliminating the variability introduced by temporary or part-time positions.

The GOMS survey provides detailed information on various aspects of the participants' careers and educational backgrounds. It includes data on the type of institution they graduated from, their graduation date, current employment status, details about their current job, and information about their first job. Additionally, the survey collects data on job search activities, vocational training experiences, language training, and certifications obtained. This comprehensive dataset allows for a nuanced analysis of the factors that influence job turnover among recent graduates. Table I is a summary of the dataset statistics and Table II is the variables measured in the study.

TABLE I. SUMMARY OF THE DATASET

Dataset	Instances	Categories	Fields	Positive Ratio
GOMS	28,549	50+	20+	0.27

TABLE II. VARIABLES MEASURED

Variable Category	Variable Name	Description
Personal Attributes	Gender	Male, Female
	Age	Age at the time of turnover
Employment Characteristics	Industry	Industry sector of the job
	Job Type	Specific job role
	Company Size	Number of employees
Working Conditions	Weekly Working Hours	Total hours worked per week
	Monthly Salary	Average monthly income
	Union Membership	Whether the employee is a union member
Job Satisfaction	Satisfaction Level	11 items on a 5-point Likert scale
	Job Fit	4 items measuring the alignment of job with skills and interests
Benefits	Social Insurance	Dummy variable indicating social insurance coverage
	Welfare Benefits	Dummy variable indicating availability of welfare benefits
Job Preparation	Work Experience	Employment experience during school
	Job Search Experience	Experience in job searching
	Vocational Training	Participation in vocational training
	Certification	Whether the individual holds any professional certifications
Academic Performance	GPA	Grade point average on a 5-point scale

B. Baseline Methods

We compare FINN with eight baseline models in our experiments (Table III), all implemented with TensorFlow and trained using the Adam optimization algorithm.

TABLE III. THE EIGHT BASELINE MODELS OF STUDY

Model	Description
Logistic Regression (LR)	A classical model in classification tasks that predicts the probability of positive samples. It is a linear model that uses the logistic function to model a binary dependent variable.
Factorization Machines (FM)	Models feature interactions by learning a feature vector for each feature and using the inner product of two feature vectors. FM is effective in capturing second-order interactions.
Field-aware Factorization Machines (FFM)	An extension of FM that considers the field information of each feature, allowing for more precise interaction modeling.
Support Vector Machine (SVM)	A supervised learning model used for classification tasks. SVM constructs a hyperplane or set of hyperplanes in a high-dimensional space to separate different classes. It is effective in high-dimensional spaces and for cases where the number of dimensions exceeds the number of samples.
Random Forest	An ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. It reduces overfitting by averaging multiple trees.
Product-based Neural Networks (PNN)	Uses product operations to perform pairwise interactions between features to capture interaction information. This model enhances the representation power by explicitly modeling feature interactions.
Wide & Deep	A hybrid model consisting of a single layer wide part and a multilayer deep part. The wide part captures memorization of feature interactions, while the deep part captures generalization.
DeepFM	Improves the Wide & Deep model by replacing the wide part with a factorization machine. DeepFM combines the strengths of FM and deep neural networks to capture both low-order and high-order feature interactions.

C. Evaluation Metrics

We use two primary evaluation metrics to assess model performance: Area Under the ROC Curve (AUC) and Log Loss.

- AUC: A widely used metric in binary classification that measures the ability of the model to distinguish between positive and negative samples. A higher AUC indicates better performance.

- Log Loss: Measures the distance between the predicted probabilities and the actual labels. Lower log loss values indicate better performance.

D. Data Processing and Experimental Setup

For data preprocessing, categorical features are converted into one-hot encoded vectors. Numerical features are discretized by equal-size buckets. We also apply negative down-sampling to address the issue of class imbalance, ensuring that the positive sample ratio is approximately 0.5.

We implement all models using TensorFlow and train them using the Adam optimization algorithm with a mini-batch size of 1000. The learning rate is set to 0.0001. For deep models, the depth of layers is set to 5, with ReLU activation functions. The number of neurons per layer is set to 700. We initialize the DNN hidden layers using Xavier initialization and the embedding vectors from uniform distributions. The experiments are conducted on two GTX 4060 Ti GPUs.

E. Performance Comparison and Analysis

We compare the performance of the FINN model with baseline models using the GOMS dataset. The results are summarized in Table IV.

TABLE IV. PERFORMANCE COMPARISON OF DIFFERENT MODELS

Method	AUC	Log Loss
LR	0.751	0.449
FM	0.783	0.417
FFM	0.792	0.408
SVM	0.770	0.430
Random Forest	0.765	0.435
PNN	0.801	0.399
Wide & Deep	0.813	0.387
DeepFM	0.820	0.380
FINN	0.830	0.370

F. Analysis of Results

The experimental results show that our proposed FINN model outperforms all baseline models in terms of both AUC and Log Loss. The superior performance of FINN can be attributed to its ability to effectively capture complex feature interactions through its feature interaction layer. Traditional models like Logistic Regression and Factorization Machines are limited in their ability to model higher-order interactions, which are crucial for accurate predictions in employment data.

Neural network-based models such as FFM, PNN, and DeepFM show better performance compared to traditional models, highlighting the importance of modeling feature interactions. Among these, DeepFM performs well due to its ability to capture both low-order and high-order interactions. However, FINN surpasses DeepFM by employing a more sophisticated feature interaction mechanism that extends beyond simple inner product or element-wise product operations.

To further analyze the effectiveness of FINN, we conduct additional experiments varying the size of the embedding vectors and the number of hidden layers in the DNN. The results, illustrated in Fig. 1 and 2 indicate that FINN consistently outperforms other models across different configurations, demonstrating its robustness and generalizability.

G. Parameter Study

In this subsection, we conduct hyper-parameter investigations for our model, focusing on the embedding part, the DNN part, and the feature interaction part. Specifically, we change the following hyper-parameters: (1) the dimension of embeddings, (2) the depth of DNN, and (3) the dimension of the feature interaction vector.

1) Embedding part: We change the embedding sizes from 10 to 50 and summarize the experimental results in Fig. 1 and Table V. As the dimension expands from 10 to 50, our model shows substantial improvement. We find that an embedding size of 30 yields the best performance on the GOMS dataset. Enlarging the embedding size increases the number of parameters in the embedding layer and the DNN part. The optimal embedding size balances model complexity and performance.

2) DNN part: We investigate the impact of different DNN depths by varying the number of hidden layers. Increasing the number of layers initially improves model performance; however, performance degrades if the number of layers continues to increase due to overfitting. Fig. 2 and Table VI shows that a depth of 5 layers provides the best balance between model complexity and performance.

3) *Feature interaction part:* We change the feature interaction vector sizes from 10 to 40. Fig. 3 and Table VII shows that a vector size of 10 provides the best performance on the GOMS dataset. The performance remains stable as we increase the vector size, indicating that the model is robust to this hyper-parameter.

TABLE V. EMBEDDING SIZE OF STUDY

Embedding Size	AUC (GOMS)	Log Loss (GOMS)
10	0.815	0.380
20	0.825	0.375
30	0.830	0.370
40	0.832	0.368
50	0.831	0.369

TABLE VI. DNN LAYERS OF STUDY

Number of Layers	AUC (GOMS)	Log Loss (GOMS)
3	0.828	0.373
5	0.830	0.370
7	0.831	0.369
9	0.830	0.371

TABLE VII. FEATURE INTERACTION VECTOR SIZE OF STUDY

Interaction Vector Size	AUC (GOMS)	Log Loss (GOMS)
10	0.830	0.370
20	0.828	0.372
30	0.829	0.371
40	0.829	0.371

H. Variable Importance Analysis

To identify the most important variables influencing job turnover, we use the feature importance scores from the FINN model. The top four variables are Monthly Salary, Job Satisfaction, Company Size, and Weekly Working Hours.

The importance scores indicate the relative impact of each variable on the prediction of job turnover. Table VIII and Fig. 4 illustrates the importance scores of these variables.

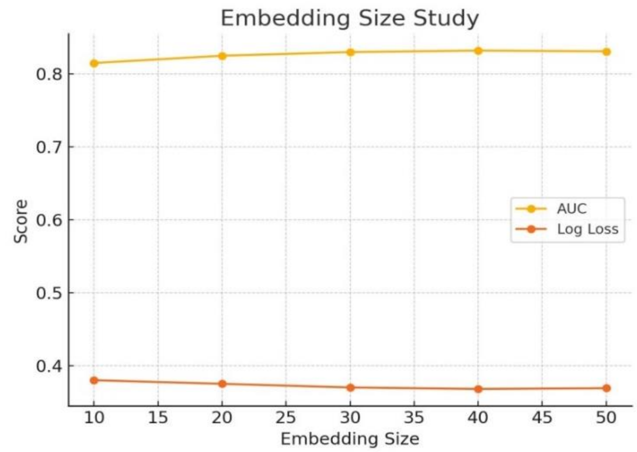

Fig. 1. Embedding size study.

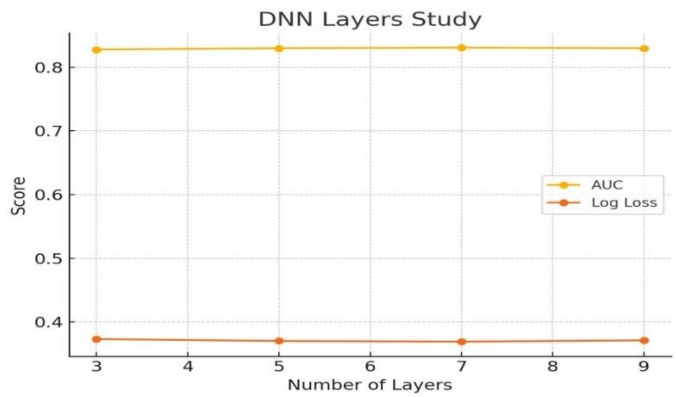

Fig. 2. DNN layers study.

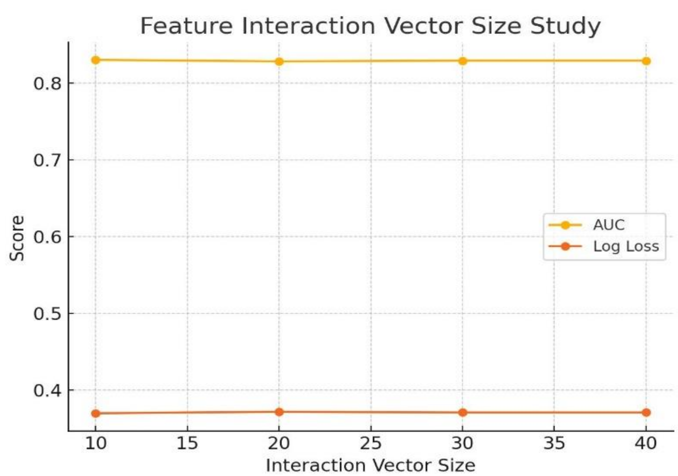

Fig. 3. Feature interaction vector size study.

TABLE VIII. VARIABLE IMPORTANCE OF STUDY

Variable	Importance Score
Monthly Salary	0.25
Job Satisfaction	0.22
Company Size	0.18
Weekly Working Hours	0.15

V. DISCUSSION

The prediction of job turnover among early career university graduates is a crucial task for both employees and employers. Accurate predictions can help employees navigate their career paths more effectively and assist employers in developing strategies to enhance employee retention, thereby reducing the costs associated with recruitment and training. This study proposes a Feature Interaction based Neural Network (FINN) model designed to address the complexities inherent in employment data and improve the accuracy of job turnover predictions.

In this paper, the results demonstrated that the FINN model outperforms all baseline models in terms of both AUC and Log Loss. Specifically, the FINN model achieved an AUC of 0.830 and a Log Loss of 0.370, indicating its superior ability to distinguish between employees who are likely to leave their jobs and those who are not. This performance can be attributed to the model's ability to effectively capture complex feature interactions through its feature interaction layer, which traditional models and even some advanced neural network models struggle to do [20, 21].

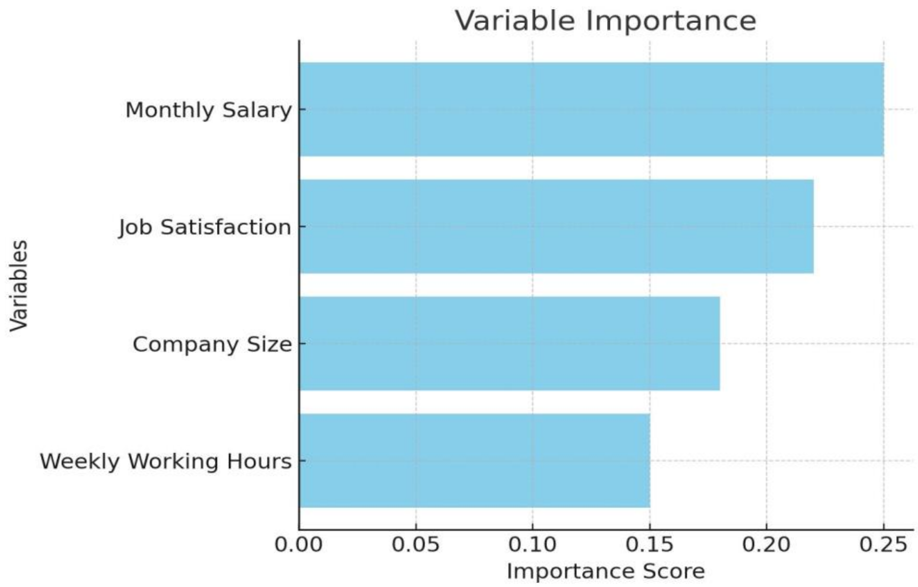

Fig. 4. Variable importance of study.

The study conducted an extensive parameter study to identify the optimal settings for the FINN model. This included varying the embedding sizes, the depth of the DNN, and the size of the feature interaction vector. The results provide valuable insights into the impact of these hyper-parameters on model performance [35,36]. Additionally, the analysis identified key variables that significantly influence job turnover, such as monthly salary, job satisfaction, company size, and weekly working hours. These findings can inform both policy and practice by highlighting areas where interventions might be most effective in reducing turnover rates. This study makes several significant contributions to the field of employment data analysis and predictive modeling. The primary contribution is the development and validation of the FINN model. This model enhances the predictive accuracy of job turnover by effectively modeling complex interactions between features. By leveraging the rich GOMS dataset, the study provides a detailed analysis of various factors influencing job turnover among early career graduates. This comprehensive dataset allows for a nuanced understanding of the predictors of job turnover. These findings have several practical implications. Employers can use the insights from the FINN model to develop targeted retention strategies. For instance, improving job satisfaction and offering competitive salaries could be effective measures to reduce turnover rates among early career employees. Career counselors and advisors can use the model's predictions to provide personalized guidance to graduates, helping them make informed decisions about their career paths and job choices. Policymakers can leverage the findings to design programs and policies aimed at improving job stability among young graduates. This could include initiatives to enhance job satisfaction and provide better working conditions.

While the FINN model has demonstrated significant improvements in predictive accuracy, there are several avenues for future research. First, future research could explore the integration of additional advanced techniques, such as attention mechanisms and sequence modeling, to further enhance the model's ability to capture complex feature interactions. Second, testing the FINN model on different datasets from various regions and industries could validate its generalizability and robustness across different contexts. Third, conducting longitudinal studies to track job turnover over a more extended period could provide deeper insights into the long-term predictors of job stability and career success. Fourth, experimenting with different intervention strategies based on the model's predictions could help in identifying the most effective measures for reducing job turnover.

VI. Conclusion

In this paper, we have demonstrated that the FINN model represents a significant advancement in the field of predictive modeling for job turnover among early career graduates. By effectively capturing complex feature interactions and leveraging a rich dataset, the FINN model provides superior predictive performance compared to both traditional and contemporary models. The insights gained from this study have the potential to inform strategies and policies aimed at improving job retention and career outcomes for young professionals. As employment data analysis continues to advance, the FINN model is expected to provide a strong foundation for both future research and practical applications, enabling more precise and actionable predictions in the labor market.

Acknowledgment

This research Supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF- RS-2023-00237287, NRF-2021S1A5A8062526) and local government-university cooperation-based regional innovation projects (2021RIS-003).

References

[1] J. M. Abowd and F. Kramarz, "The analysis of labor markets using matched employer-employee data," Handbook of Labor Economics, vol. 3, pp. 2629-2710, 1999.

[2] R. W. Crandall, W. Lehr, and R. E. Litan, "The effects of broadband deployment on output and employment: A cross-sectional analysis of US data," Washington, DC: Brookings Institution, vol. 6, 2007.

[3] M. Doede, "Race as a predictor of job satisfaction and turnover in US nurses," Journal of Nursing Management, vol. 25, no. 3, pp. 207-214, 2017.

[4] Y. Zhao, M. K. Hryniewicki, F. Cheng, B. Fu, and X. Zhu, "Employee turnover prediction with machine learning: A reliable approach," in Intelligent Systems and Applications: Proceedings of the 2018 Intelligent Systems Conference (IntelliSys) Volume 2, Springer International Publishing, pp. 737-758, 2019.

[5] M. Lazzari, J. M. Alvarez, and S. Ruggieri, "Predicting and explaining employee turnover intention," International Journal of Data Science and Analytics, vol. 14, no. 3, pp. 279-292, 2022.

[6] A. R. Skelton, D. Nattress, and R. J. Dwyer, "Predicting manufacturing employee turnover intentions," Journal of Economics, Finance and Administrative Science, vol. 25, no. 49, pp. 101-117, 2020.

[7] S. Sajjadiani, A. J. Sojourner, J. D. Kammeyer-Mueller, and E. Mykerezi, "Using machine learning to translate applicant work history into predictors of performance and turnover," Journal of Applied Psychology, vol. 104, no. 10, pp. 1207-1225, 2019.

[8] E. Rombaut and M. A. Guerry, "Predicting voluntary turnover through human resources database analysis," Management Research Review, vol. 41, no. 1, pp. 96-112, 2018.

[9] J. R. Carlson, D. S. Carlson, S. Zivnuska, R. B. Harris, and K. J. Harris, "Applying the job demands resources model to understand technology as a predictor of turnover intentions," Computers in Human Behavior, vol. 77, pp. 317-325, 2017.

[10] H. Min, B. Yang, D. G. Allen, A. A. Grandey, and M. Liu, "Wisdom from the crowd: Can recommender systems predict employee turnover and its destinations?," Personnel Psychology, vol. 77, no. 2, pp. 475-496, 2024.

[11] X. Gao, J. Wen, and C. Zhang, "An improved random forest algorithm for predicting employee turnover," Mathematical Problems in Engineering, vol. 2019, article ID 4140707, 2019.

[12] A. Coetzer, C. Inma, P. Poisat, J. Redmond, and C. Standing, "Does job embeddedness predict turnover intentions in SMEs?," International Journal of Productivity and Performance Management, vol. 68, no. 2, pp. 340-361, 2019.

[13] M. Akeyo and P. Wezel, "Factors Influencing Staff Turnover in Logistics Management," American Journal of Economics, vol. 1, no. 1, pp. 79-94, 2016.

[14] A. F. Schlechter, C. Syce, and M. Bussin, "Predicting voluntary turnover in employees using demographic characteristics: A South African case study," Acta Commercii, vol. 16, no. 1, pp. 1-10, 2016.

[15] H. J. Lee and Y. C. Cho, "Relationship between job satisfaction and turnover intention among nurses in general hospitals," Journal of the Korea Academia-Industrial Cooperation Society, vol. 15, no. 7, pp. 4404-4415, 2014.

[16] Y. Sun, Z. Luo, and P. Fang, "Factors influencing the turnover intention of Chinese community health service workers based on the investigation results of five provinces," Journal of Community Health, vol. 38, pp. 1058-1066, 2013.

[17] F. Horn, R. Pack, and M. Rieger, "The autofeat python library for automated feature engineering and selection," in Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I, Springer International Publishing, pp. 111-120, 2020.

[18] S. M. Fati, "A Loan Default Prediction Model Using Machine Learning and Feature Engineering," ICIC Express Letters, vol. 18, no. 1, pp. 27-37, 2024.

[19] A. Abdulhafedh, "Comparison between common statistical modeling techniques used in research, including: Discriminant analysis vs logistic regression, ridge regression vs LASSO, and decision tree vs random forest," Open Access Library Journal, vol. 9, no. 2, pp. 1-19, 2022.

[20] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "Finn: A framework for fast, scalable binarized neural network inference," in Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays, pp. 65-74, 2017.

[21] F. Rezaei and A. Houmansadr, "FINN: Fingerprinting network flows using neural networks," in Proceedings of the 37th Annual Computer Security Applications Conference, pp. 1011-1024, 2021.

[22] M. M. Bejani and M. Ghatee, "A systematic review on overfitting control in shallow and deep neural networks," Artificial Intelligence Review, vol. 54, no. 8, pp. 6391-6438, 2021.

[23] S. Kost, O. Rheinbach, and H. Schaeben, "Using logistic regression model selection towards interpretable machine learning in mineral prospectivity modeling," Geochemistry, vol. 81, no. 4, p. 125826, 2021.

[24] R. Langone, A. Cuzzocrea, and N. Skantzos, "Interpretable Anomaly Prediction: Predicting anomalous behavior in industry 4.0 settings via regularized logistic regression tools," Data & Knowledge Engineering, vol. 130, p. 101850, 2020.

[25] J. Á. Martín-Baos, R. García-Ródenas, and L. Rodriguez-Benitez, "Revisiting kernel logistic regression under the random utility models perspective. An interpretable machine-learning approach," Transportation Letters, vol. 13, no. 3, pp. 151-162, 2021.

[26] Y. Izza, A. Ignatiev, and J. Marques-Silva, "On explaining decision trees," arXiv preprint arXiv:2010.11034, 2020.

[27] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," Journal of Applied Science and Technology Trends, vol. 2, no. 1, pp. 20-28, 2021.

[28] G. W. Cha, H. J. Moon, and Y. C. Kim, "Comparison of random forest and gradient boosting machine models for predicting demolition waste based on small datasets and categorical variables," International Journal of Environmental Research and Public Health, vol. 18, no. 16, p. 8530, 2021.

[29] H. Byeon, "Is the random forest algorithm suitable for predicting parkinson's disease with mild cognitive impairment out of parkinson's disease with normal cognition?," International Journal of Environmental Research and Public Health, vol. 17, no. 7, p. 2594, 2020.

[30] H. Byeon, "Can the random forests model improve the power to predict the intention of the elderly in a community to participate in a cognitive health promotion program?," Iranian Journal of Public Health, vol. 50, no. 2, pp. 315-324, 2021.

[31] H. Byeon, "Exploring the risk factors of impaired fasting glucose in middle-aged population living in South Korean communities by using categorical boosting machine," Frontiers in Endocrinology, vol. 13, p. 1013162, 2022.

[32] F. Lang, L. Liang, K. Huang, T. Chen, and S. Zhu, "Movie recommendation system for educational purposes based on field-aware factorization machine," Mobile Networks and Applications, vol. 26, no. 5, pp. 2199-2205, 2021.

[33] Y. Juan, Y. Zhuang, W. S. Chin, and C. J. Lin, "Field-aware factorization machines for CTR prediction," in Proceedings of the 10th ACM conference on recommender systems, pp. 43-50, 2016.

[34] Y. Juan, D. Lefortier, and O. Chapelle, "Field-aware factorization machines in a real-world online advertising system," in Proceedings of the 26th International Conference on World Wide Web Companion, pp. 680-688, 2017.

[35] A. Koutsoukas, K. J. Monaghan, X. Li, and J. Huan, "Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data," Journal of Cheminformatics, vol. 9, pp. 1-13, 2017.

[36] N. Tran, J. G. Schneider, I. Weber, and A. K. Qin, "Hyper-parameter optimization in classification: To-do or not-to-do," Pattern Recognition, vol. 103, p. 107245, 2020.