Precision Construction of Salary Prediction System Based on Deep Neural Network

Yuping Wang*, MingYan Bai, Changjiang Liao
Southern Power Grid Digital Enterprise Technology (Guangdong) Co., Ltd, GuangZhou, 510000 China

***Abstract*—Currently, most recruitment websites use keyword search or job nature classification to filter the salary information that job seekers are most concerned about. Job seekers need to spend much time and effort to understand the salary range of their desired position. In order to help job seekers quickly and accurately understand the salary of their desired position and market value, Word2vec model and latent Dirichlet allocation model are used to obtain topic features, which are used as the basis for the salary prediction model. The study uses deep neural networks and adaptive moment estimation algorithms to construct the salary prediction model. Based on the constructed salary prediction model, the final salary prediction system is constructed based on a browser/server model. The results showed that on the training set, the maximum accuracy of the salary prediction model was 96.71%, the minimum was 93.75%, and the average was 95.07%. The mean absolute percentage error and mean square error of this model were 5.661% and 0.3462, respectively. The maximum average response time of the salary prediction system was 134.2s, the minimum was 2.02s, and the maximum throughput was 1500000byte/s. The salary prediction model has good performance, which can provide technical support for salary prediction.**

Keywords—Deep neural network; Adam; salary; prediction; system

I. INTRODUCTION

Employment is the foundation of people's lives, social stability, and economic development. People can obtain economic income, improve their living standards, and realize their self-worth through employment [1-2]. However, due to insufficient industry knowledge among many job seekers and an excessive number of unemployed individuals, the current employment situation in China is not ideal. Salary is a key consideration for job seekers. With the development of Internet technology, there are many recruitment websites around the world. These recruitment websites are not limited by time and space, with low recruitment costs, extensive information, etc. [3-4]. In fact, excessive, repetitive, and false recruitment information requires job seekers to spend more time and effort understanding the salary range of their desired positions. Therefore, designing a salary prediction model and system is of great significance. The commonly used methods for constructing salary prediction models include Doc2vec algorithm, Support Vector Machine (SVM), random forest, Ridge regression, k-means clustering algorithm, etc. [5].

To predict the salary level of graduates, Kuo J Y et al. constructed a salary prediction model on the basis of deep learning. The stacked denoising auto-encoder was applied to train the model. The method could accurately predict the salary level of graduates [6]. Grmez Y et al. designed a prediction system based on deep learning and a corresponding performance rating scale to predict wage growth. The system had significant advantages in prediction accuracy and time consumption [7]. James O et al. built a neural network method to predict worker wages. The model was trained on data from 35 common job skills. The prediction accuracy exceeded 70%, with good performance [8]. Lombu A S et al. designed a classification model based on SVM. The Python programming language was used to predict individual wages. The accuracy was 87%, which exceeded the K-nearest neighbor model [9].

However, these methods also have certain shortcomings, such as the sensitivity of SVM to parameter selection, high computational complexity, and the difficulty in selecting regularization coefficients for Ridge regression. In order to accurately predict the salary of desired positions for users, the Deep Neural Network (DNN) is used in the study, and the Adaptive Moment Estimation (Adam) is also introduced to improve the training effect of DNN. The final salary prediction model is constructed. In addition, the study also utilizes technologies such as browser/server mode and Linux server to construct a salary prediction system. The research aims to predict job compensation, help job seekers understand their market value, and enhance industry awareness. The innovation of the research is reflected in the combination of DNN and Adam, as well as the Word2vec model and the Latent Dirichlet Allocation (LDA) model. The contribution of the research is to predict the salary of ideal positions for users, promoting job seekers' understanding of the salary for desired positions, and facilitating salary negotiations for seekers during the job search process.

The research is divided into five sections. Section II constructs a DNN-based salary prediction system, involving the design of functional modules for the salary prediction model and the overall design of the salary prediction system. Section III is the performance validation of the salary prediction model and salary prediction system, including prediction accuracy, convergence speed, concurrent test results, and throughput comparison. Section IV is the discussion, which includes personal insights and opinions. Section V is the conclusion, which includes the important results, shortcomings, and prospects of this research.

II. CONSTRUCTION OF DNN-BASED SALARY PREDICTION SYSTEM

In order to build a salary prediction system and help users understand the salary of the desired position, DNN algorithm is adopted in the study, and Adam algorithm is used to improve the training effect of DNN. The study also uses Word2vec and LDA models to obtain topic features, laying the foundation for the salary prediction model. In addition, technologies such as browser/server mode and Linux server are utilized to construct the final salary prediction system.

A. Design of DNN Salary Prediction Model Based on Adam Optimization

To predict the salary of desired positions, the Word2vec model and LDA model are used to achieve text clustering and topic feature construction in the study. Afterwards, the study uses DNN and Adam algorithms to construct the final salary prediction model (Adam-DNN). The web crawler technology is used to capture job information from third-party recruitment software, which is divided into structured and unstructured types. Among them, structured job information includes salary ranges, work locations, and educational requirements. To ensure the salary prediction accuracy, the raw data is preprocessed, including removing duplicate values, and handling missing values and outliers. To perform topic clustering on job description texts, the LDA is adopted in the study. The LDA topic model explores the topic structure of text through the common features of word items in text information, which has unsupervised learning, strong flexibility, and interpretability [10-11]. However, the LDA ignores the syntax and order of the document, and the weight scores between the generated topic keywords are relatively close, making it difficult to distinguish. In response to this issue, the Word2vec model is used to optimize the LDA. The specific optimization model is shown in Eq. (1).

$$new_{original_{score_j}} = \frac{original_{score_j}}{\sum_{j=1}^{M} original_{score_j}} \times \frac{\sum_{i=1}^{M-1} sim(\theta_i, \theta_j)}{M-1} \tag{1}$$

In Eq. (1), $original_{score_j}$ signifies the weight of the j-th topic word calculated by the original LDA model. $new_{original_{score_j}}$ signifies the weight of the newly defined j-th topic word. M is the number of topic words. θ_i and θ_j represent the i-th and j-th topic words, respectively. $sim(\theta_i, \theta_j)$ signifies the similarity in the i-th and j-th topic words. Cosine similarity is used to calculate word similarity, as displayed in Eq. (2) [12].

$$sim(X,Y) = \frac{X \cdot Y}{\|X\|\|Y\|} = \frac{\sum_{i=1}^{M} X_i \cdot Y_i}{\sqrt{\sum_{i=1}^{M}(X_i)^2}\sqrt{\sum_{i=1}^{M}(Y_i)^2}} \tag{2}$$

In Eq. (2), X and Y represent different vectors, respectively. X_i and Y_i represent the sub-vectors of X and Y, respectively. Word2vec+LDA topic clustering is used to optimize the model, which can extract topic features from job texts and lay the foundation for the salary prediction model. To construct a salary prediction model, DNN is used in the study, which is optimized by Adam. DNN has stronger non-linear fitting ability, which can demonstrate deep correlation between data. It has been widely applied in the processing and prediction of relevant data in different fields [13-14].

The DNN prediction model mainly includes Input Layer (IL), Hidden Layer (HL), and Output Layer (OL). The HL is at least two. In addition, each link between IL and OL network units is a fully connected chain that can be learned and trained. DNN uses forward and back propagation algorithms during training, which also needs to set hyper-parameters to determine the number of HLs and activation function before training. The forward propagation process involves the input and output of different nodes in HL and OL. The back propagation process involves the total error function, weight correction of HL and OL, and bias correction between HL and OL. In order to reduce the loss function value of DNN, the weights and biases of HL in DNN are updated. Therefore, the study adopts the Adam algorithm to improve the training performance of DNN. The advantage of Adam algorithm is its ability to automatically adjust learning rate, high computational efficiency, and fast convergence speed [15-16]. The steps for optimizing the Adam algorithm are shown in Fig. 1.

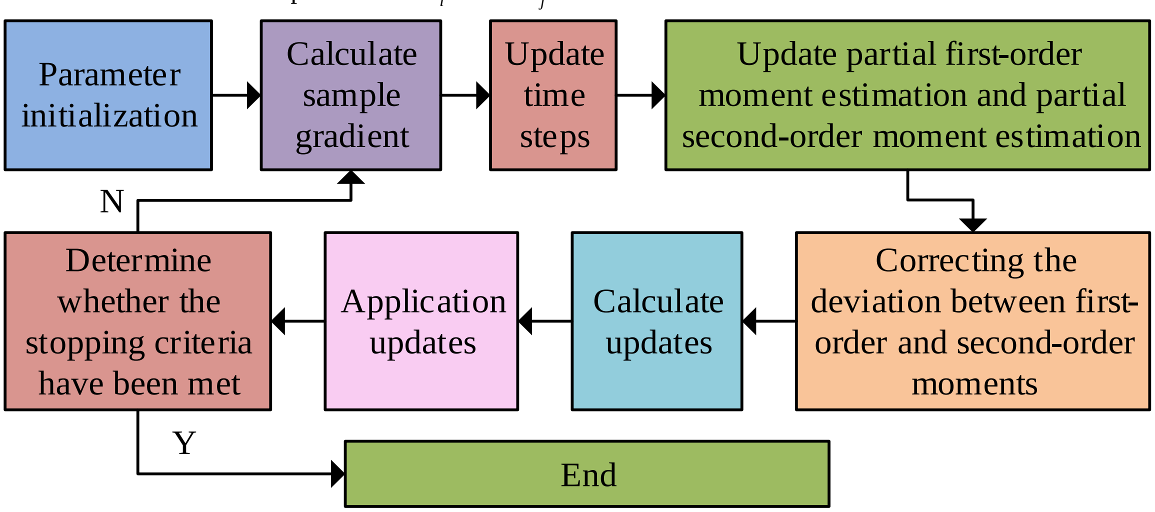


Fig. 1. Steps for optimizing Adam algorithm.

From Fig. 1, the first step of the Adam algorithm is parameter initialization. The second is to calculate the sample gradient. The third is to update the time step. The fourth is to update the partial first-order moment estimation and partial second-order moment estimation. The fifth step is to correct the deviation between the first-order and second-order moments. The sixth step is to compute the updated value. The seventh is to apply the updated value. The eighth is to determine whether the stop criterion is met. If the stop criterion is reached, the process ends. Otherwise, it returns to the second step. The activation function used in the prediction model is the Rectified Linear Unit (ReLU) function, and the loss function is the mean squared error function. The prediction process of the Adam-DNN salary prediction model is shown in Fig. 2.

From Fig. 2, the first step of the Adam-DNN salary prediction model is to input the learning sample dataset. The second uses the Adam algorithm to process the DNN model. The third is to output the predicted value, and the fourth is to determine whether the predicted output value is consistent with the sample output value. If it is consistent, the model training ends. Otherwise, it returns to the second. The fifth is to save the model. The sixth is to input the basic parameters of the prediction model. The seventh step is to make a salary prediction.

B. Construction of Salary Prediction System

In order to accurately construct the salary prediction system, a DNN-based model is first designed in the study. The DNN-based salary prediction model is the most core functional module in the salary prediction system. To build a salary prediction system, the study adopts browser/server mode, Python Web development technology, My Structured Query Language (MySQL) database, and Linux server. Python Web development technology has strong simplicity, readability, and scalability, which has been widely applied in system scheduling and Web development [17]. The advantages of MySQL database are fast, convenient, and simple [18]. The salary prediction system is shown in Fig. 3.

From Fig. 3, the overall structure of the salary prediction system includes the client, server, Flask application, Redis database, MySQL database, MongoDB database, Nginx proxy server, u Web Server Gateway Interface (uWSGI) process, and response. Before using this prediction model, it is necessary to perform a crawler task and save the obtained data to the MySQL database. In addition, the Crontab command in Linux servers is used to monitor Web crawler operations to ensure that the data obtained by the Web crawler is relatively new. The Crontab command drives the Spider to request job information from the target host and store it in the MongoDB database. The functional structure of the salary prediction system is displayed in Fig. 4.

From Fig. 4, the designed system mainly includes four functions, namely crawler management, model update management, user core function, and user basic function. Crawler management mainly consists of timed crawler tasks, crawler network management, and manual execution tasks. Model update management includes model updates and model version rollback. The primary task of a timed crawler is to select a timed crawler website, enter the scheduled crawler time, and choose to repeat the crawler task or execute it only once. After the primary task, scheduled crawlers need to determine whether the crawler website is available. If available, the crawler task data table is updated, the server Crontab script is refreshed, and the process ends. Otherwise, the process is terminated directly. The crawler process of the salary prediction system mainly has two steps. The first determines whether there are asynchronous requests in the recruitment network. If it is determined to be yes, the request module is used to request the Web interface. Otherwise, the request module is used to request the Web page source code. Then the Beautiful Soup is used to parse the Web page source code. The second step is to obtain information and store it in the database, and then terminate the process. Job search and salary prediction constitute the core functions of users, while the use basic function consists of login registration, personal information management, and password modification. The functional modules of the system are specifically designed. Among them, salary prediction belongs to the user function request, and its corresponding time sequence diagram is shown in Fig. 5.

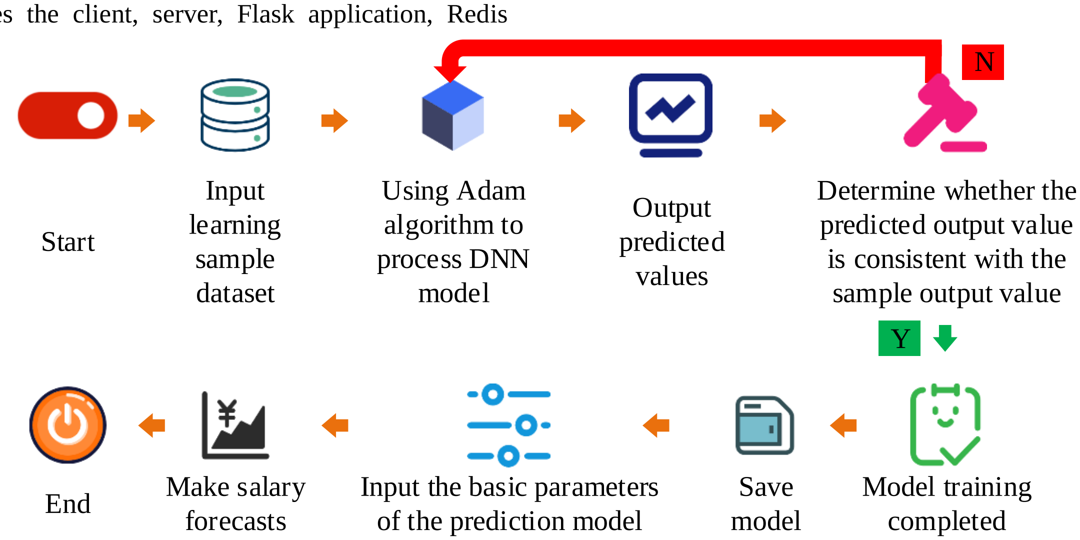


Fig. 2. The prediction process of Adam-DNN salary prediction model.

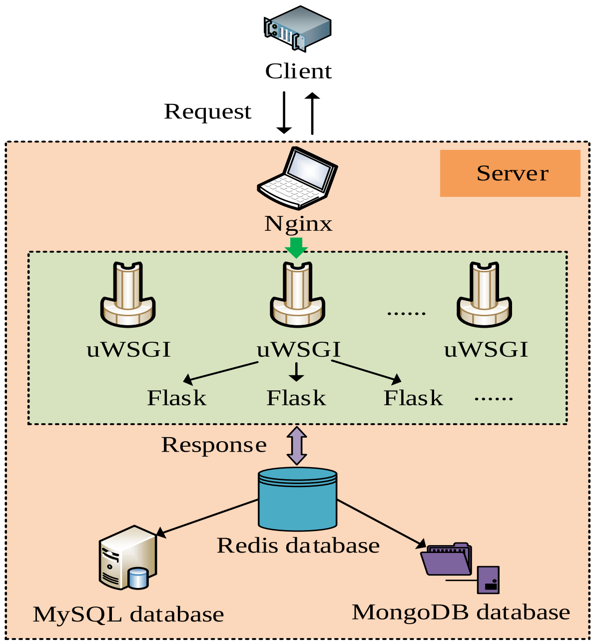


Fig. 3. The overall architecture of the salary prediction system.

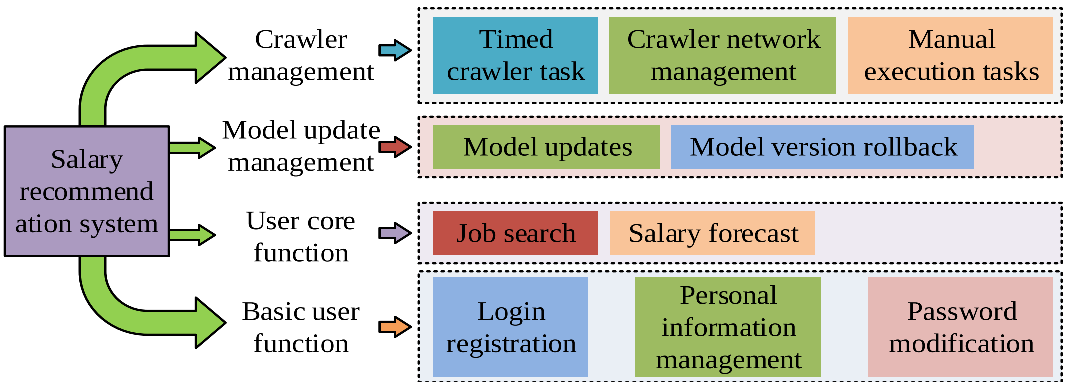


Fig. 4. The functional structure of salary prediction system.

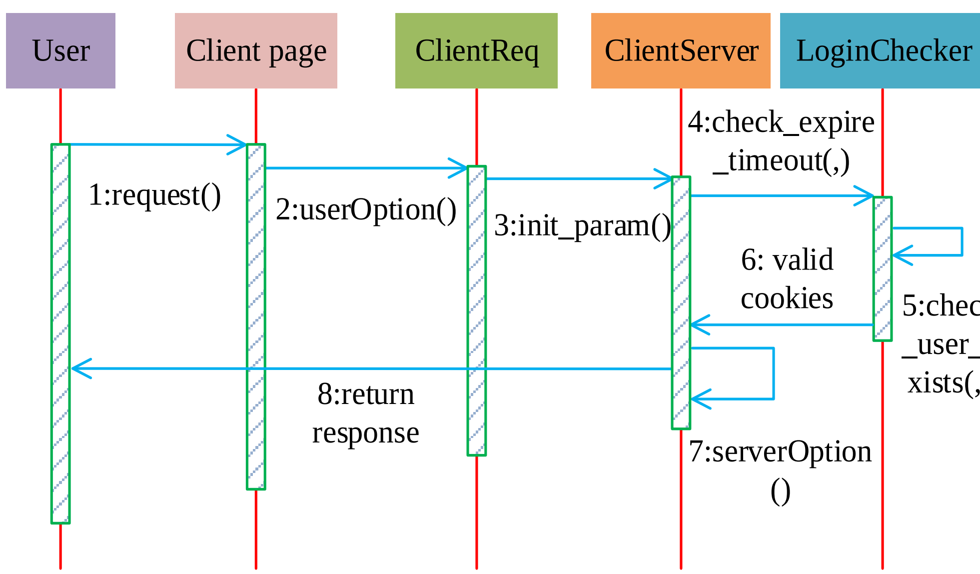


Fig. 5. Sequence diagram of user function request class.

From Fig. 5, the time sequence diagram of the user function request class involves user, Client page, ClientReq objects, ClientServer objects, and LoginChecker objects. The sequence diagram of the user function request class has eight steps. The first step is for the user to submit the parameters required to complete the operation through the request method. The second step provides the submitted parameters through the userOption method. The third step calls the init_param method to initialize the member variables of the ClientServer objects. The fourth step uses the check_expire_timeout method to verify whether the time is within the validity period. The fifth step uses the check_user_exists method to verify whether the user corresponding to the Cookies exists. The sixth step returns valid Cookies information. The seventh step performs server operations corresponding to the serverOption method on the ClientServer objects. The eighth step returns the information that the user operation is successful.

In database design, the records of users browsing positions are stored through the position browsing record table t_log_browse. This record table includes record numbers, position information record numbers, user record numbers, access time, and stay time. Among them, the position information record number is the foreign key of the position information table t_job, while the position information table includes salary, recruiters, company type, company size, and company location.

III. Results

To verify the performance of the designed salary prediction model and system, an experimental environment is set up and comparative methods are selected. In addition, the study also explains and divides the dataset required for the experiment. The performance verification of the salary prediction model includes accuracy, comparison between predicted values and true values, etc. The performance verification of the system involves response time and throughput, etc. Through comparative verification, it can better reflect the performance advantages of the salary prediction model and system designed in the paper, as well as the areas where the salary prediction model and system can be further improved.

A. *Performance Verification of Adam-Dnn Salary Prediction Model*

To validate the performance of the Adam-DNN, Extreme Gradient Boosting (XGBoost), logistic regression algorithm, SVM, and Back Propagation neural network (BP) are selected for comparison. The browser used in the experiment is Google Chrome 122.0.6261.6, with an Intel Core i5-13600KF processor, a maximum Intel Turbo Boost Technology of 5.1GHz, a basic power consumption of 125W, and a maximum memory of 192GB. The operating system is a dual system, which includes Windows 10 (64 bit) and Ubuntu version 20.04. In addition, the experiment uses Alibaba Cloud cloud servers, with a bandwidth of 1Mbps. The HL in DNN is 6. The sample data obtained by the crawler is divided into training and testing sets in a 7:3 ratio, with 300 samples in the testing set and 700 samples in the training set. The comparison of salary prediction accuracy for different models is shown in Fig. 6.

From Fig. 6(a), the maximum accuracy of the Adam-DNN model was 96.71%, the minimum was 93.75%, and the average was 95.07%. The maximum values of XGBoost, logistic regression algorithm, SVM, and BP were 92.13%, 93.42%, 90.68%, and 89.35%, respectively, while the minimum values were 89.47%, 90.31%, 86.98%, and 85.75%, respectively. According to Fig. 6(b), the maximum values of the five models on the testing set were 98.54%, 92.98%, 94.37%, 91.56%, and 90.27%, respectively. The maximum accuracy of the Adam-DNN model was 5.56%, 4.17%, 6.98%, and 8.27% higher than the maximum values of XGBoost, logistic regression algorithm, SVM, and BP, respectively. In summary, the Adam-DNN model has better accuracy and performance in salary prediction. The comparison results between the predicted and true salary values of different models are shown in Fig. 7. Mean Square Error (MSE) and Mean Absolute Percentage Error (MAPE) are taken as evaluation indicators.

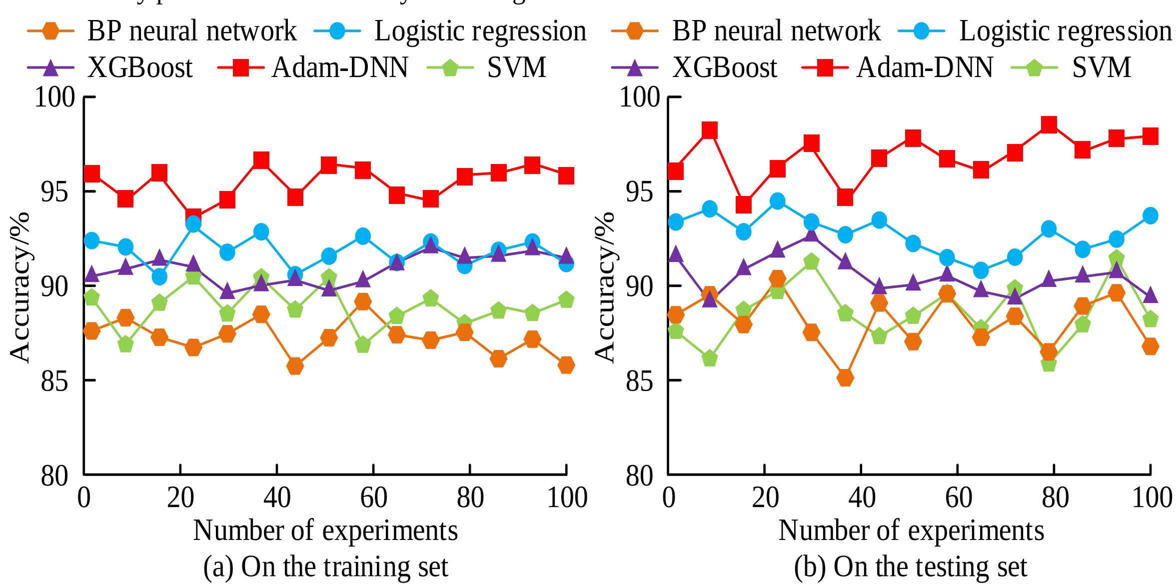
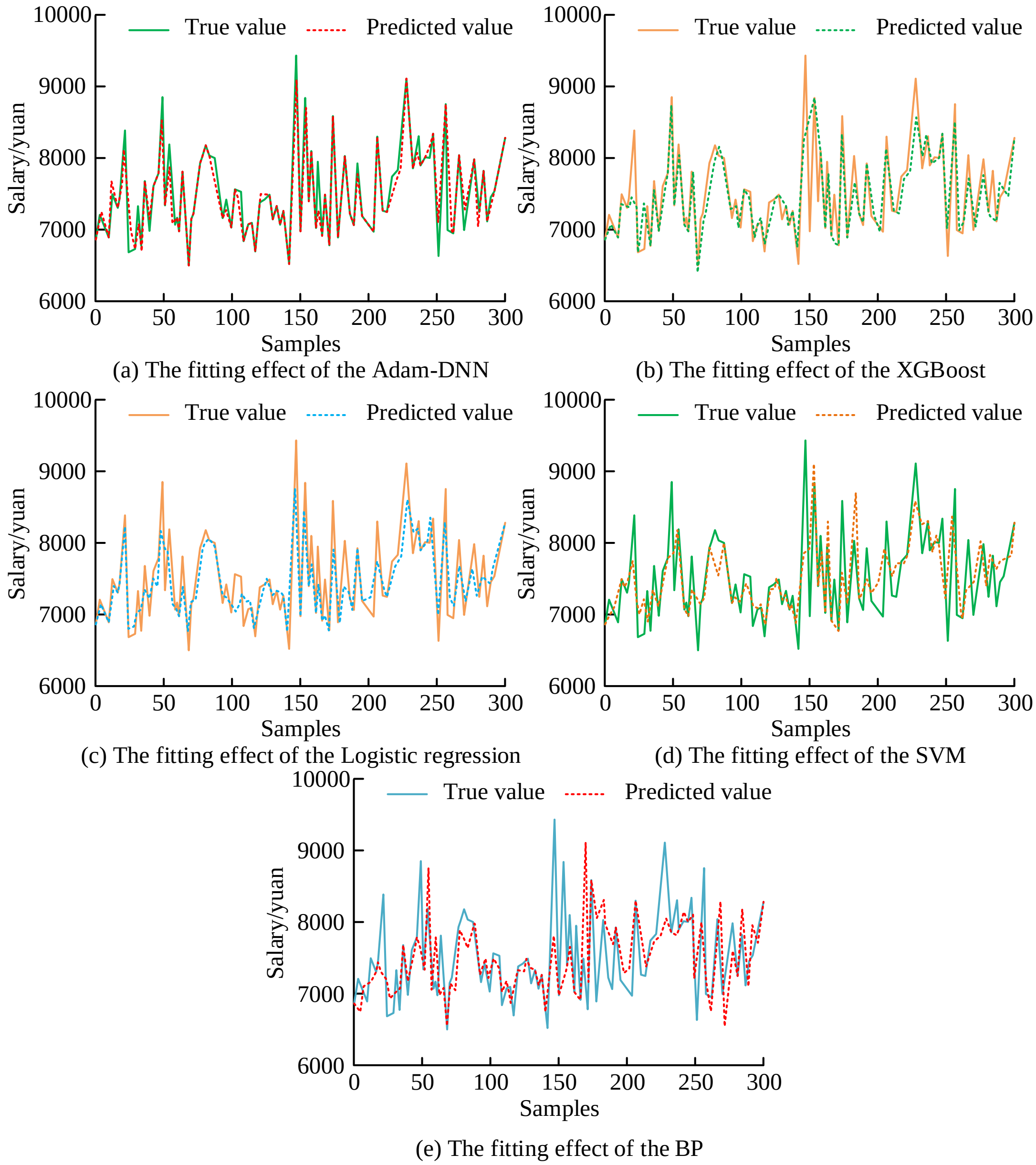


Fig. 6. Comparison of salary prediction accuracy for different models.



(a) The fitting effect of the Adam-DNN

(b) The fitting effect of the XGBoost

(c) The fitting effect of the Logistic regression

(d) The fitting effect of the SVM

(e) The fitting effect of the BP

Fig. 7. Comparison results between predicted and true salary values of different models.

Fig. 7 (a) to Fig. 7 (e) show the error curves in the predicted and true salary values of the Adam-DNN model, XGBoost, logistic regression algorithm, SVM, and BP, respectively. According to Fig. 7 (a), in most sample data, the predicted values of the Adam-DNN model were consistent with the true values. The MAPE of the Adam-DNN model was 5.661%, and the MSE was 0.3462. From Fig. 7 (b), the MAPE and MSE of XGBoost were 6.283% and 0.4237%, respectively. According to Fig. 7(c), 7 (d), and 7 (e), the MAPE of the logistic regression algorithm, SVM, and BP were 6.139%, 6.482%, and 6.667%, respectively, and the MSE was 0.4067, 0.4442, and 0.4586, respectively. Overall, the Adam-DNN model has better fitting effects, which can predict salary more accurately. The performance comparison of different optimization methods for DNN is shown in Fig. 8.

From Fig. 8 (a), on the training set, the Adam algorithm tended to flatten out after nearly 182 iterations, with a minimum loss value of 0.0052. The Momentum algorithm, Nestrov Accelerated Gradient algorithm, Adagrad algorithm, and Root Mean Square prop algorithm only reached a plateau after nearly 1620, 1540, 1020, and 225 iterations, respectively. According to Fig. 8 (b), on the testing set, the five algorithms iterated nearly 160, 1600, 1490, 1002, and 213 iterations respectively before stabilizing. Therefore, the Adam algorithm has better optimization performance and faster convergence speed. To better validate the performance of the Adam-DNN salary prediction model designed in the paper, other related models are selected for comparison, including natural neighbor classification algorithm, stacking fusion algorithm, and random forest algorithm. The comparison of Area Under the Curve (AUC) values and F1 values for different models is shown in Table I.

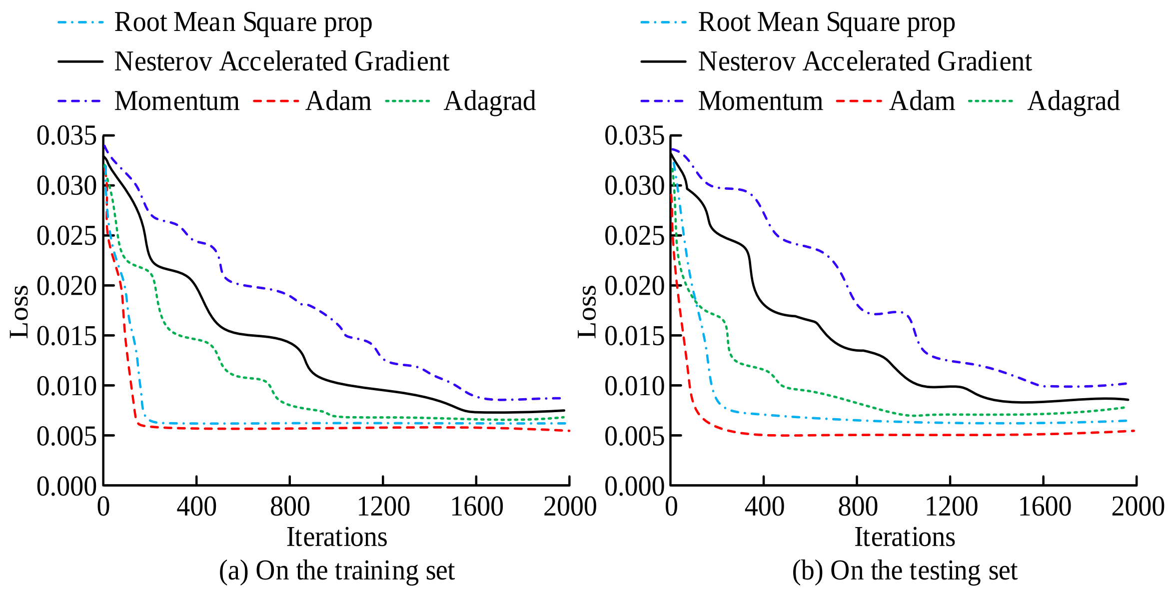


Fig. 8. Performance comparison of different optimization methods for DNN.

TABLE I. COMPARISON OF AUC AND F1 VALUES FOR DIFFERENT MODELS

Model	**AUC**					**F1**				
	Number of experiments					**Number of experiments**				
	1	***2***	***3***	***4***	***5***	***1***	***2***	***3***	***4***	***5***
BP	0.904	0.916	0.918	0.904	0.909	0.918	0.922	0.912	0.913	0.917
SVM	0.911	0.927	0.921	0.915	0.924	0.930	0.928	0.927	0.926	0.931
Random forest	0.927	0.938	0.928	0.931	0.938	0.938	0.932	0.929	0.931	0.933
Natural neighbor classification	0.933	0.941	0.939	0.937	0.945	0.942	0.939	0.941	0.935	0.937
XGBoost	0.948	0.950	0.943	0.949	0.953	0.950	0.947	0.943	0.952	0.949
Logistic regression	0.957	0.952	0.948	0.956	0.961	0.952	0.959	0.962	0.957	0.953
Stacking	0.968	0.977	0.976	0.978	0.972	0.975	0.965	0.977	0.964	0.961
Adam-DNN	0.987	0.994	0.997	0.982	0.988	0.987	0.992	0.993	0.989	0.995

From Table I, the average AUC of the Adam-DNN model was 0.9896, which was 0.0794, 0.07, 0.0572, 0.0506, 0.041, 0.0348, and 0.0154 higher than the average values of the other six models, respectively. Furthermore, in terms of F1 value, the Adam-DNN model also scored significantly higher than other comparison models. The average F1 value of the Adam-DNN model was 0.9912, while the average values of the other six models were 0.9164, 0.9284, 0.9326, 0.9388, 0.9482, 0.9566, and 0.9684, respectively. Overall, the Adam-DNN model performs better.

B. Performance Verification of Salary Prediction System Based on DNN

To verify the performance of the designed salary prediction system, the study selects similar systems designed by other researchers for comparison. The comparison systems include the human resources recruitment system with salary prediction function designed by Tian X et al., the human resources information system designed by Anupa M, and the human resources system based on firefly optimization algorithm designed by Li L et al [19-21]. The experimental settings used in the system are consistent with the performance verification of the deigned model. The concurrency test results and throughput comparison of different systems are shown in Fig. 9.

From Fig. 9(a), overall, as the number of concurrency increased, the average response time used by different systems also increased synchronously. After the concurrency exceeded 8000 times, the average response time of the constructed salary prediction system showed a rapid increase. The systems designed by Tian X et al., Anupa M, and Li L et al. showed a rapid increase after exceeding 3300, 4500, and 4300 times, respectively. The maximum average response time of the four systems was 134.2s, 162.7s, 159.5s, and 167.8s, while the minimum values were 2.02s, 16.48s, 14.59s, and 17.86s, respectively. The designed salary prediction system has better performance, and can withstand more concurrency. As shown in Fig. 9(b), as the user load increased, the throughput of all systems first increased and then decreased. The maximum throughput values of the four systems were 1500000byte/s, 1230000byte/s, 1320000byte/s, and 1190000byte/s, respectively, and the corresponding user loads for each system were 8120, 3210, 4380, and 4210. The designed salary prediction system performs better. The comparison of Central Processing Unit (CPU) utilization and memory usage is displayed in Table II.

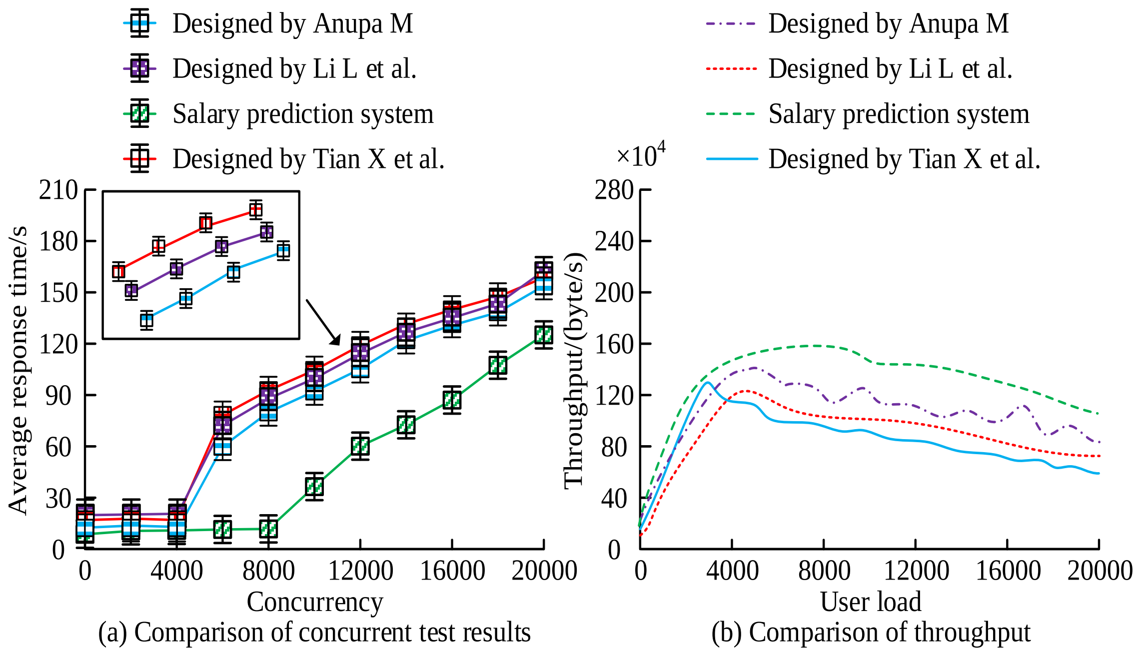


Fig. 9. Comparison of concurrency test results and throughput of different systems.

TABLE II. COMPARISON OF CPU UTILIZATION AND MEMORY USAGE OF VARIOUS SYSTEMS

System	**CPU utilization/%**					**Memory usage/%**				
	Experiment times					**Experiment times**				
	1	***2***	***3***	***4***	***5***	***1***	***2***	***3***	***4***	***5***
Designed by Tian X et al.	32.71	33.06	33.81	31.54	34.97	35.81	36.42	35.19	35.47	34.76
Designed by Anupa M	25.43	24.19	25.97	26.49	25.14	29.66	30.31	31.72	30.04	29.15
Designed by Li L et al.	28.65	28.08	29.34	29.18	28.46	31.22	31.75	32.29	31.55	33.26
Manuscript	13.27	12.18	12.94	13.53	11.03	15.37	14.08	15.65	15.91	13.77

From Table II, the maximum CPU utilization values for the designed salary prediction system, Tian X et al., Anupa M, and Li L et al. were 13.53%, 34.97%, 26.49%, and 29.34%, respectively, with average values of 12.590%, 33.218%, 25.444%, and 28.742%. In addition, the average memory usage rates of the four systems were 14.956%, 35.530%, 30.176%, and 32.014%, respectively. The average memory usage of the designed salary prediction system was 20.574%, 15.22%, and 17.058% lower than the systems designed by Tian X et al., Anupa M, and Li L et al., respectively. That is, the designed salary prediction system has better performance.

IV. Discussion

To predict the salary range of job seekers' desired positions, an Adam-DNN salary prediction model and system were designed. The results showed that the Adam-DNN salary prediction model designed in the paper had good prediction accuracy, small prediction error, and excellent performance. This is because DNN has stronger non-linear fitting ability, and the Adam algorithm converges faster and has high computational efficiency. Tavares I et al. designed a method based on multi-layer feedforward artificial neural network and a combination of convolutional neural network layer combined with DNN to predict photovoltaic power generation with smaller errors [22]. The salary prediction system designed in the paper has faster response time and maximum throughput, with lower CPU utilization and memory usage. In order to further improve the performance of the salary prediction system, it is recommended that future research adopt architecture technologies with better performance to optimize the overall architecture of the salary prediction system.

V. Conclusion

A DNN-based salary prediction model and system were designed for job seekers who want to quickly and accurately understand the salary range of their desired positions. The results showed that on the testing set, the maximum accuracy values of the Adam-DNN model, XGBoost, logistic regression algorithm, SVM, and BP were 98.54%, 92.98%, 94.37%, 91.56%, and 90.27%, respectively. Moreover, the maximum accuracy values of the Adam-DNN model were 5.56%, 4.17%, 6.98%, and 8.27% higher than those of comparison models, respectively. In addition, the MAPE of the five models was 5.661%, 6.283%, 6.139%, 6.482%, and 6.667%, respectively, and the MSE was 0.3462%, 0.4237%, 0.4067%, 0.4442%, and 0.4586%, respectively. The Adam-DNN model had better fitting effects. On the testing set, the Adam algorithm, Momentum algorithm, Nestrov Accelerated Gradient algorithm, Adagrad algorithm, and Root Mean Square prop algorithm iterated nearly 160, 1600, 1490, 1002, and 213 times respectively before becoming smoother, indicating that the optimization effect of Adam algorithm was better. The maximum average response time of the salary prediction system designed in this study, as well as the systems designed by Tian X et al., Anupa M, and Li L et al., were 134.2s, 162.7s, 159.5s, and 167.8s, respectively. The maximum throughput values were 1500000byte/s, 1230000byte/s, 1320000byte/s, and 1190000byte/s, respectively. In addition, the average CPU utilization rates of the four systems were 12.590%, 33.218%, 25.444%, and 28.742%, respectively, and the average memory usage rates were 14.956%, 35.530%, 30.176%, and 32.014%. The performance of the salary prediction model and system is good. However, this study still needs improvement. Firstly, the salary prediction system may also experience long response time when the concurrent quantity is not high. Future research can utilize deep learning algorithms to construct models with higher accuracy. Secondly, the research data is mainly obtained through Web crawling. This method has high complexity, and the amount of data obtained is small, making it difficult to verify the data authenticity, which affects the predictive performance of the model. Future research can analyze data from different recruitment networks, select platforms with relatively good data quality, or use data migration methods.

REFERENCES

[1] Ashaye O R, Mahmoud A B, Munna A S, Ali N. The role of social media engagement and emotional intelligence in successful employment. Higher Education, Skills and Work-Based Learning, 2023, 13(2):315-337.

[2] Spencer P, Haneghan J P V, Baxter A. Exploring social networks, employment and self-determination outcomes of graduates from a postsecondary program for young adults with an intellectual disability. Journal of Vocational Rehabilitation, 2021, 55(3):251-270.

[3] Huang D, He H, Liu T. City size and employment dynamics in China: evidence from recruitment website data. Journal of Geographical Sciences, 2021, 31(12):1737-1756.

[4] Huang J C. Effects of person-organization fit objective feedback and subjective perception on organizational attractiveness in online recruitment. Personnel Review, 2022, 51(4):1262-1276.

[5] Ye H, Cao B, Geng J, Wen Y. Web services classification via combining Doc2Vec and LINE model. International Journal of Computational Science and Engineering, 2020, 23(3):250-261.

[6] Kuo J Y, Liu C H, Lin H C. Building Graduate Salary Grading Prediction Model Based on Deep Learning. Intelligent Automation and Soft Computing, 2021, 27(1):53-68.

[7] Grmez Y, Arslan H, Sari S, Dani M. SALDA-ML: Machine Learning Based System Design to Predict Salary In-crease. Advances in Artificial Intelligence Research, 2022, 2(1):15-19.

[8] James O, Han C, Tomasi S. Using neural networks to predict wages based on worker skills. Studies in Business and Economics, 2021, 16(1):95-108.

[9] Lombu A S, Hidayat S, Hidayatullah A F. Pemodelan Klasifikasi Gaji Menggunakan Support Vector Machine. Journal of Computer System and Informatics (JoSYC), 2022, 3(4):363-370.

[10] Nanda G, Douglas K A, Waller D R, Merzdorf H E, Goldwasser D. Analyzing large collections of open-ended feedback from MOOC learners using LDA topic modeling and qualitative analysis. IEEE Transactions on Learning Technologies, 2021, 14(2):146-160.

[11] Fei X. An LDA based model for semantic annotation of Web English educational resources. Journal of Intelligent & Fuzzy Systems, 2021, 40(2):3445-3454.

[12] Rathnasabapathy P, Palanisami D. A theoretical development of improved cosine similarity measure for interval valued intuitionistic fuzzy sets and its applications. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(12):16575-16587.

[13] Liu Z, Dang Z, Liu Z, Li Y, He X, Dai Y, Fang Z. Self-design of arbitrary polarization-control waveplates via deep neural networks. Photonics Research, 2023, 11(5):695-711.

[14] Gholaminejad A, Jorkesh S, Poshtan J. A comparative case study between shallow and deep neural networks in induction motor's fault diagnosis. IET Science, Measurement & Technology, 2023, 17(5):195-207.

[15] Kiran K K, Farsangi E N. Blast Demand Estimation of RC-moment-resisting Frames using a Proposed Multi-modal Adaptive Pushover Analysis Procedure. International Journal of Engineering, Transactions B: Applications, 2021, 34(1):46-55.

[16] Xiao P, Yin Y, Liu B, Jiang B, Malaiya Y K. Adaptive Testing Based on Moment Estimation. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(3):911-922.

[17] Mahesha C. Telugu Optical Character Recognition Using Cloud Computing and Python. International journal of computer science engineering and information technology research: IJCSEITR, 2022, 12(1):25-34.

[18] Bouamrane K, Matallah H, Belalem G. Comparative Study Between the MySQL Relational Database and the MongoDB NoSQL Database. International journal of software science and computational intelligence, 2021, 13(3):38-63.

[19] Tian X, Pavur R, Han H, Zhang L. A machine learning-based human resources recruitment system for business process management: using LSA, BERT and SVM. Business Process Management Journal, 2023, 29(1):202-222.

[20] Anupa M. Role of Human Resources Information System (Hris) in Accelerating Organizational Effectiveness-It Companies Perspective. International Journal of Management and Humanities, 2021, 5(6):22-25.

[21] Li L, Pahlevanzadeh B. Evaluation of the trust values among human resources in the enterprise cloud using an optimization algorithm and fuzzy logic. Kybernetes: The International Journal of Systems & Cybernetics, 2022, 51(6):2008-2029.

[22] Tavares I, Manfredini R, Almeida J, Soares J, Ramos S, Foroozandeh Z, Vale Z. Comparison of PV power generation forecasting in a residential building using ANN and DNN. IFAC-PapersOnLine, 2022, 55(9):291-296.