

## Research Article

# Intelligent Recommendation Algorithm Combining RNN and Knowledge Graph

Fengsheng Zeng  and Qin Wang 

*Academy of Engineering and Technology, Yang-En University, Quanzhou 362014, China*

Correspondence should be addressed to Fengsheng Zeng; zengfengsheng@yeu.edu.cn

Received 26 August 2022; Revised 22 November 2022; Accepted 3 December 2022; Published 21 December 2022

Academic Editor: Man Leung Wong

Copyright © 2022 Fengsheng Zeng and Qin Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the continuous application and development of big data and algorithm technology, intelligent recommendation algorithms are gradually affecting all aspects of people's daily life. The impact of smart recommendation algorithm has both advantages and disadvantages; it can facilitate people's life, but also exists at the same time the invasion of privacy, information cocoon, and other problems. How to optimize intelligent recommendation algorithms to serve the society more safely and efficiently becomes a problem that needs to be solved nowadays. We propose an intelligent recommendation algorithm combining recurrent neural network (RNN) and knowledge graph (KG) and analyze and demonstrate its performance by building models and experiments. The results show that among the five different recommendation models, the intelligent recommendation algorithm model combining RNN and knowledge graph has the highest AUC and ACC values in the Book-Crossing and MovieLens-1M. At the same time, the algorithm's rating prediction error values are small (less than 2%) in extracting different users' ratings for different books. In addition, the intelligent recommendation algorithm combining RNN and knowledge graph has the lowest RMSE and MAE values in the comparison of three different recommendation algorithms, indicating that it has better performance and stability, which is important for the improvement of user recommendation effect.

## 1. Introduction

The continuous development of intelligent technology makes recommendation algorithms have an increasingly important impact on people's lives. For example, in daily life, shopping, travel, and watching videos are closely related to intelligent recommendation algorithms. Intelligent recommendation algorithms mainly rely on the application of big data, through the extraction and analysis of a large amount of data to know the behavior preferences of different users, so as to make intelligent recommendations to meet the actual needs of different users [1]. For example, in the field of short video, relevant short video platforms extract and analyze users' past viewing habits to generate user profiles and push their favorite contents or products to them based on this. With its excellent performance, smart recommendation algorithms can accurately push different information to each user, thus reducing the cost and difficulty of users' access to information. There are different types and forms

of intelligent recommendation algorithms, such as content-based, model-based, social relationship-based recommendations, and collaborative filtering. In addition, at the algorithm level, there are clustering algorithms, classification algorithms, association algorithms, etc. [2]. For different recommendation algorithms, the resulting recommendation effect varies. In this study, we conduct experiments and analysis on intelligent recommendation algorithms combining recurrent neural network (RNN) and knowledge graph (KG) to make a study on how to improve the performance and efficiency of recommendation algorithms.

There are two main innovation points in the research. The first point is to combine knowledge map with recommendation algorithm to solve the sparse problem and cold start problem of intelligent recommendation algorithm. The second point is to use deep learning recurrent neural network to conduct deep mining of user preferences and propose a recommendation algorithm based on recurrent neural network and weight knowledge map, thus solving

the problem that the recommendation results are easily affected by low correlation. The proposed method solves the problem that the relationship between entities has phase weight when knowledge atlas is used as a heterogeneous information network for recommendation algorithm. The combination of knowledge map and deep learning improves the recommendation effect and solves the problem of data sparsity and cold start of traditional recommendation algorithm to a certain extent.

## 2. Related Works

RNN and knowledge graphs have been widely studied and applied in different fields. For example, to address the problem of the proliferation of fake news, Pimpalkar et al. applied techniques such as recurrent neural networks to analyze and detect fake news, as well as to classify them, and the results showed that the method was effective in intervening fake news [3]. Marquez et al. applied a recurrent neural network approach to predict ethanol consumption in Brazil and showed that it had good performance and prediction accuracy [4]. Ananthanatarajan et al. applied recurrent neural networks to the prediction of wind energy change and showed that the RNN model has better performance compared to other traditional methods [5]. For problems such as stock market value prediction, Moghar and Hamiche applied recurrent neural network methods for model construction and analysis, which helped to improve the prediction accuracy [6]. Zheng et al. applied recurrent neural networks to electricity consumption prediction and showed that the performance of household electricity consumption prediction can be significantly improved if prediction is performed at the granularity level of appliances [7]. Laib et al. applied recurrent neural networks to predict the consumption of natural gas and found that the method used in the study possessed higher accuracy by building a model and comparing it with the models of other methods [8]. Wang et al. applied recurrent neural network to the prediction of oil field production and verified that the method has good prediction effect and higher prediction accuracy by building a model [9]. To address the problem of frequent earthquakes in Taiwan, China, Chin et al. applied recurrent neural networks to earthquake early warning in the region and tested and analyzed them with previously recorded seismograms, showing that the method can reduce processing time and has high accuracy [10]. The study validated and analyzed the method through different scenarios and showed that it has better performance than traditional methods [11].

Shi applied the deep learning-based knowledge graph to the intelligent construction engineering Q&A system, and the study proved its good horizontal scalability [12]. Do et al. combined the knowledge graph with deep learning and applied it to the development of Vietnam tourism Q&A system, and the experiment showed its high accuracy and effectiveness [13]. For the problem of matching radiological imaging with medical reports, Hou et al. applied knowledge graph to improve the original method and proved that the improvement possessed better perfor-

mance [14]. Huang C L. and Huang C C. applied knowledge graph to customize the flight training of student pilots, and the results showed that it could effectively demonstrate the problems in course learning as well as provide a basis for subsequent [15]. Fan et al. applied knowledge mapping to the analysis of sepsis-related knowledge literature to sort out its historical evolution, research content, etc., so as to achieve the prediction of its development trend [16]. Zeng et al. applied knowledge mapping to drug discovery by this method to promote that the method was used to facilitate the recycling of drugs and the prediction of adverse drug reactions [17]. In response to the problem of financial fraud in listed companies, Wu et al. constructed a knowledge graph framework to detect fraudulent companies and showed that it has high recognition ability, which helps to effectively regulate companies [18]. The knowledge graph was applied to the construction of a medical intelligent question and answer system for hepatitis B. Through this method, it provides reference for doctors' diagnosis and treatment, etc. [19].

By combing through related studies, it is found that most researchers have applied RNN and knowledge graph to prediction analysis of different problems and achieved good research results by combining other methods, but there are fewer studies combining RNN with knowledge graph. To address this, the study proposes an intelligent recommendation algorithm combining RNN and knowledge graph and conducts experimental analysis on some worthy optimization problems in user recommendation to improve the effect and performance of intelligent recommendation.

## 3. Intelligent Recommendation Algorithm Model Construction Combining RNN and Knowledge Graph

*3.1. RNN Algorithm Model Construction.* RNN is a neural network that inputs sequential data and performs data processing in a recursive manner. It uses sequential relationships for input and output during operation, such that the output of the previous neuron can be used as the input of the next neuron, which makes the preceding and following data have a certain causal relationship [20]. The structure of RNN is shown in Figure 1.

As can be seen from the RNN structure diagram, when each input layer has data input, it performs the same computation internally, so the parameters in this structure are in a shared state, and the training time it needs to spend is greatly reduced. When a sequence of input layers  $x = \{x_1, x_2, \dots, x_t\}$  is given, the expression of the implicit layer  $s_t$  is shown in

$$s_t = f(Ux_t + Ws_{t-1}). \quad (1)$$

In Equation (1),  $f$  is a nonlinear function,  $U$  and  $W$  are the weight matrices, and  $s_0$  is made zero when the hidden state values are calculated for the first layer. In different task-specific environments, the output at the moment of  $t$  is not necessary; in some tasks, only the last output is of

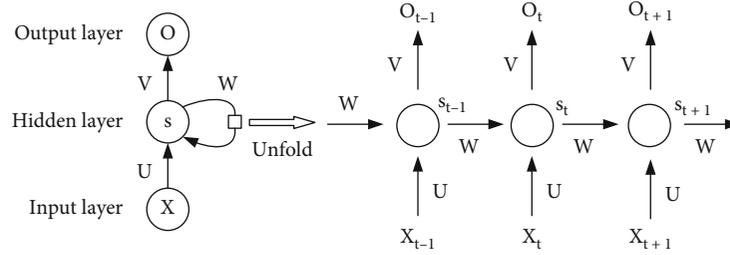


FIGURE 1: RNN structure diagram.

interest; in others, the output at each moment is required. The expression for the output value  $o_t$  is shown in

$$o_t = g(Vs_t). \quad (2)$$

Equation (2) indicates that the hidden layer  $s_t$  captures the input information for all previous moments, where  $g$  is the activation function. Due to the problems of gradient disappearance and explosion in RNN, it is difficult to preserve long-term information effectively. Therefore, RNN have evolved into long short-term memory (LSTM) networks and gated recurrent units (GRU), which are the two main streams. There are three types of gating: input gates, forgetting gates, and output gates. During the operation of LSTM recurrent neural network, data sampling and preprocessing are required first, and the received transmission sequence can only be transmitted to the next layer after data normalization. Then the normalized data will be transferred layer by layer between the multilayer LSTM cells. During this period, the LSTM cells will compensate and reconstruct the distorted signals, and finally determine the estimated signal sequence as the output. The door structure of LSTM unit is shown in Figure 2.

The gradient disappearance problem is solved by changing the gradient of numeric decimals from concatenated multiplication to concatenated addition. The forgetting gate of LSTM  $f_t$  is used to determine whether the memory of the previous unit should be forgotten or retained, and its expression is shown in

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \quad (3)$$

In Equation (3),  $W$  denotes the weight,  $[h_{t-1}, x_t]$  denotes the input dimension summed in series with the dimension of the previous hidden layer, and  $b$  denotes the bias. The input cell is extracted from the input information of this cell, and its expression is shown in

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \end{cases} \quad (4)$$

The memory unit  $C_t$  is formed by combining the current memory content with the new memory content, and its calculation formula is shown in

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t. \quad (5)$$

Equation (5) is the update of the cell state  $C$ , which carries the memory signal  $C_t$  which effectively solves the problem of network gradient disappearance. The expressions of the output cell  $o_t$  and the hidden cell  $h_t$  are shown in

$$\begin{cases} o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t = o_t * \tanh(C_t). \end{cases} \quad (6)$$

The gated loop unit removes the memory unit  $C$ , which transmits information by hiding it, and forms an update gate that combines forgetting and input, thus simplifying the internal structure of the unit for efficiency. The update gate  $z_t$  plays a role in determining the degree to which the current and previous information is retained, and its expression is shown in

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z). \quad (7)$$

The reset gate  $r_t$  then plays a role in determining the extent to which previous information is discarded, and its expression is shown in

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r). \quad (8)$$

The implied state of the gated loop unit  $h_t$  is a linear interpolation between the previous implied state  $h_{t-1}$  and the candidate implied state  $\tilde{h}_t$ , whose expression is shown in

$$\begin{cases} \tilde{h}_t = \tanh(W_h[r_t * h_{t-1}, x_t]), \\ h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t. \end{cases} \quad (9)$$

**3.2. Knowledge Graph Model Construction.** A knowledge graph is essentially a semantic network that describes various things and their relationships. In the semantic network of knowledge graph, nodes represent different things or concepts, and edges are constituted by semantic relationships. Its construction of knowledge graph is mainly achieved by extracting a large amount of data and then knowledge fusion, storage, editing, and annotation [21]. The knowledge graph structure diagram is shown in Figure 3.

As can be seen from the structure diagram, knowledge mapping is generally divided into three parts: information extraction, knowledge fusion, and knowledge processing.

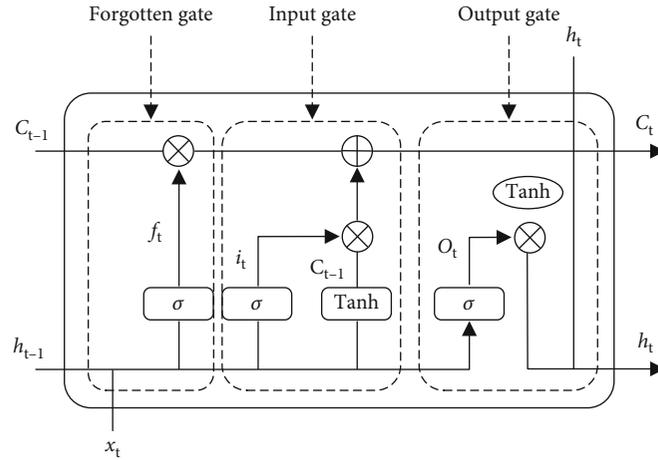


FIGURE 2: Door structure of LSTM unit.

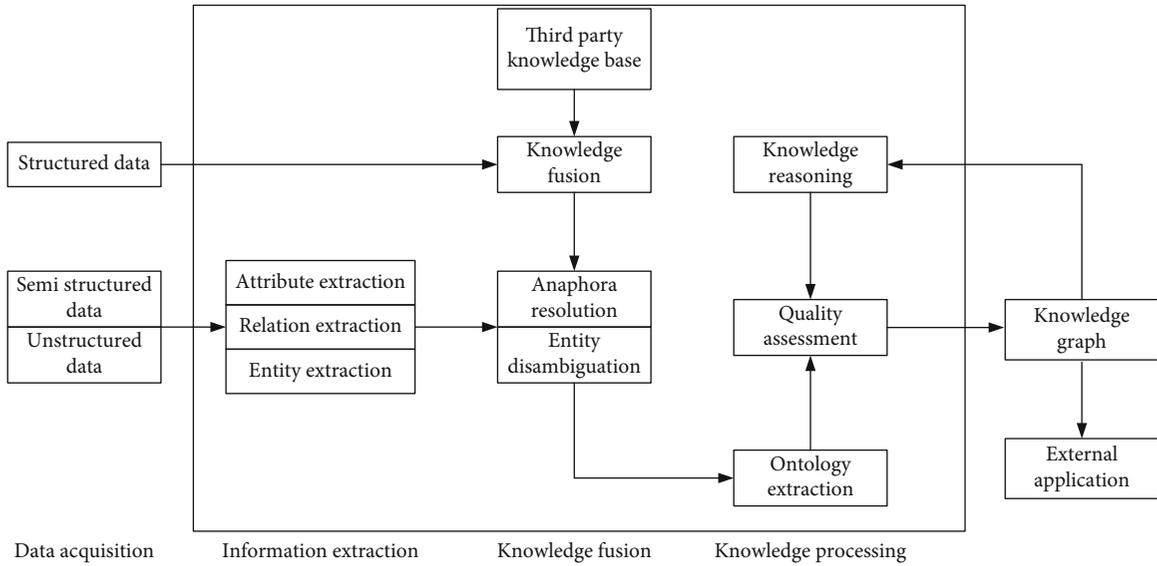


FIGURE 3: Knowledge graph structure.

Information extraction means extracting things and relationships between things from various kinds of data, and the forms of data are mainly divided into unstructured, semistructured, and structured. The extracted information is often not complete and effective, but lacks hierarchy and logic, and even contains many duplicated or wrong information fragments, so knowledge fusion should be performed on the extracted information. In order to make knowledge mapping more accurate and of higher quality, after information extraction and knowledge fusion, knowledge processing is also needed to lay the foundation for subsequent applications. The application of knowledge mapping mainly includes two aspects, namely, general knowledge mapping and industry knowledge mapping. The general knowledge graph is a horizontal application with more extensive expansion, such as widely used in knowledge Q&A and Internet

search. On the other hand, industry knowledge graph emphasizes vertical expansion, which is mainly applied in a certain industry or field and has in-depth mining and application. For example, intelligent recommendation is one of the application scenarios of knowledge graph, which constructs knowledge graphs for a large number of different user information to form user portraits and lay the foundation for recommending preference information to users. When applying knowledge graph for intelligent recommendation, we can incorporate collaborative filtering information technology and build a knowledge graph with weights to reduce the influence of interference factors and improve the accuracy of recommendation. There are three steps in building the knowledge graph with weights: constructing the similarity set of things, generating the explanation path, and generating the knowledge graph with weights. First, let

the number of users be  $m$ , and its set be  $\{Q_1, Q_2, \dots, Q_m\}$ ; the number of items be  $n$ , and its set be  $\{G_1, G_2, \dots, G_m\}$ , then the scoring matrix between them is shown in

$$R_{m \times n} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1j} & \cdots & R_{1n} \\ R_{21} & R_{22} & \cdots & R_{2j} & \cdots & R_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{i1} & R_{i2} & \cdots & R_{ij} & \cdots & R_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{m1} & R_{m2} & \cdots & R_{mj} & \cdots & R_{mn} \end{bmatrix}. \quad (10)$$

In Equation (10),  $R_{ij}$  is the rating of the user  $Q_i$  on the thing  $G_j$ , which indicates the preference of the user  $i$  on the thing  $j$ . From the matrix, for thing  $a$  and thing  $b$ , their corresponding rating vectors are  $I_a = \{R_{1a}, R_{2a}, \dots, R_{ma}\}$  and  $I_b = \{R_{1b}, R_{2b}, \dots, R_{mb}\}$ , then the expression of cosine similarity between thing  $a$  and thing  $b$  is shown in

$$\text{sim}(a, b) = \cos(I_a, I_b) = \frac{\sum_{k=1}^m R_{ka} \cdot R_{kb}}{\sqrt{\sum_{k=1}^m R_{ka}^2} \cdot \sqrt{\sum_{k=1}^m R_{kb}^2}}. \quad (11)$$

The rating matrix tends to have problems such as uneven distribution, i.e., when the number of participants in rating two things is low, higher similarity tends to occur. To address this problem, the item similarity calculation is improved, as shown in

$$\text{sim}'(a, b) = \frac{|Q_{ab}|}{|Q_{ab}| + \beta} \cdot \text{sim}(a, b). \quad (12)$$

In Equation (12),  $\text{sim}'(a, b)$  is the improved similarity, and  $|Q_{ab}|$  indicates that there are both  $|Q_{ab}|$  individuals rating things  $a$  and  $b$ .  $\beta$  is a shrinkage parameter. When  $|Q_{ab}|$  is very small,  $\beta$  can shrink the similarity. At the same time, when  $|Q_{ab}| \gg \beta$  is small, then it has less effect on the original similarity. According to Equation (10), it can be transformed into a similarity matrix between things and things, as shown in

$$S_{n \times n} = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1b} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2b} & \cdots & S_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ S_{a1} & S_{a2} & \cdots & S_{ab} & \cdots & S_{an} \\ \vdots & \vdots & & \vdots & & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nb} & \cdots & S_{nn} \end{bmatrix}. \quad (13)$$

In Equation (13),  $S_{ab}$  is the improved similarity between items  $a$  and  $b$ . First, the threshold  $k$  is set initially, and then,

the similarity matrix is filtered to exclude the values smaller than  $k$  to get the set of similarity of things as  $M = \{(a, b) | a, b \in G, S_{ab} > k, a \neq b\}$ . The knowledge graph can describe the relatedness between similar items by constructing paths, and the paths are called interpretable paths. In the set of all explainable paths  $T$ , the weight of the relationship between two things  $c$  and  $d$  is related to the number of paths that pass between them, and the weight expression is shown in

$$w_c^{(r,d)} = \frac{|\{T_{c \sim (r,d)}\}|}{\sum_{(r,d)' \in S_k} |\{T_{c \sim (r,d)'}\}|}. \quad (14)$$

In Equation (14),  $r$  is the relationship between things  $c$  and  $d$ , and  $|\{T_{c \sim (r,d)}\}|$  indicates the number of paths between thing  $c$  to thing  $d$  in all explainable paths  $T$ .  $\sum_{(r,d)' \in S_k} |\{T_{c \sim (r,d)'}\}|$  denotes the number of paths that passes through the thing  $d$ . So the final form of the weighted knowledge graph is  $WKG = (E, R, S, W)$ ,  $W = \{w_c^{(r,d)} | c, d \in E \text{ and } r \in R\}$ .

**3.3. An Intelligent Recommendation Algorithm Model Combining RNN and Knowledge Graph.** Firstly, the user's preference set is obtained through knowledge graph, and then, the algorithm is applied to learn its representation in order to get the vector representation of different things in the user's preference set. The user's preference set is input according to different hierarchical order, and the attention mechanism is used in the process of obtaining different preference degrees, so as to obtain the user's preference vector; finally, the vector between the user and the candidate things is combined, so as to output user's click probability on things to generate recommendation results. The flow chart is shown in Figure 4.

Analytical propagation of users' historical interests is done through weighted knowledge graphs, while the direction of propagation is controlled based on the weights. Let the set of users  $U$  and the set of things  $V$ , and the interaction matrix of users and things  $Y = \{y_{uv} | u \in U, v \in V\}$ , and then, the implicit feedback of users is

$$y_{uv} = \begin{cases} 1 & u \cap v \\ 0 & \text{others} \end{cases}, \quad (15)$$

where  $y_{uv} = 1$  indicates that there is an implicit feedback between users  $u$  and things  $v$ . The set of  $k$  hopping preferences of user  $u$  in the weighted knowledge graph WKG is shown in

$$E_u^k = \left\{ t \mid (c, r, d) \in \text{WKG} \text{ and } h \in E_u^{k-1} \text{ and } w_c^{(r,d)} > \tau \right\}. \quad (16)$$

In Equation (16),  $k = 1, 2, \dots, H$ , and  $E_u^0 = \{v | y_{uv} = 1\}$  denotes the set of users' clicks on historical things, which

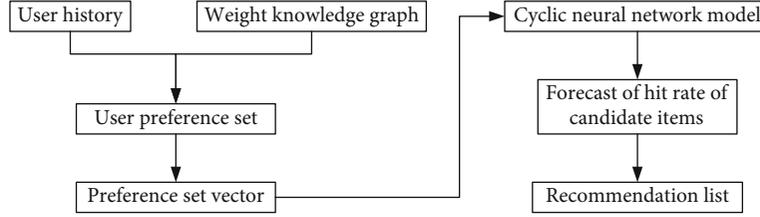


FIGURE 4: RNWKG algorithm flow chart.

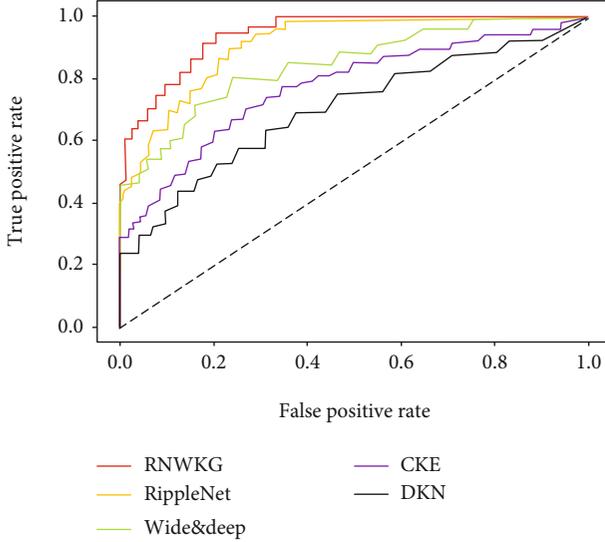


FIGURE 5: Comparison of ROC curves of five models in Book-Crossing.

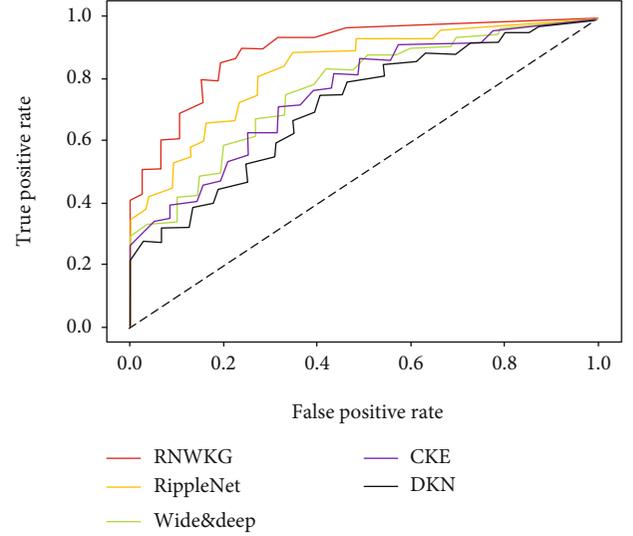


FIGURE 6: Comparison of ROC curves of five models in MovieLens-1M.

can be considered as the initialized set of users  $u$  in the knowledge graph.  $w_c^{(r,d)}$  denotes the relationship weight of things  $c$  and things  $d$ , and  $\tau$  is the weight threshold, which mainly excludes the propagation direction with smaller weight. According to the datasets of related things, the set of  $k$  hopping triples of user  $u$  is shown in

$$S_u^k = \left\{ (c, r, d) \mid (c, r, d) \in \text{WKG and } h \in E_u^{k-1} \right\}. \quad (17)$$

Propagating user preferences through weighted knowledge graphs can achieve better results. After obtaining the user's preference set by weight knowledge mapping, it can be learned by using recurrent neural network, which can represent the user's deeper preferences and thus achieve better user usage prediction. For each user's preference set  $\{E_u^k\}_{k=1}^H$ , the algorithm is used to embed it into the vector space with different hierarchical order for input, with the first one being the inner layer data and the second one being the outer layer data, so as to retain the inner layer information. The hidden layer base unit then uses a long short-term memory (LSTM) network for the purpose of solving the gra-

dient problem. Since users have various preferences for different things, the attention mechanism is used to measure the weights of each thing, and the different outputs are linearly combined to obtain the final preference characteristics, as shown in

$$u = \sum_{j=1}^n \alpha_{nj} c_j. \quad (18)$$

In Equation (18),  $\alpha_{nj}$  denotes the match between the  $j$  and the  $n$  output and the relative importance of the  $j$  thing.  $n$  is the number of entities in the preference set  $E_u$  of the user  $u$ . The expression of  $\alpha_{nj}$  is shown in

$$\alpha_{nj} = s(c_n, c_j) = \sigma(c_n^T c_j). \quad (19)$$

Finally, the user's preferred features and the combination of features of things are used to predict the user's click probability on things, as shown in

$$P_u^i = \sigma(u^T v_i). \quad (20)$$

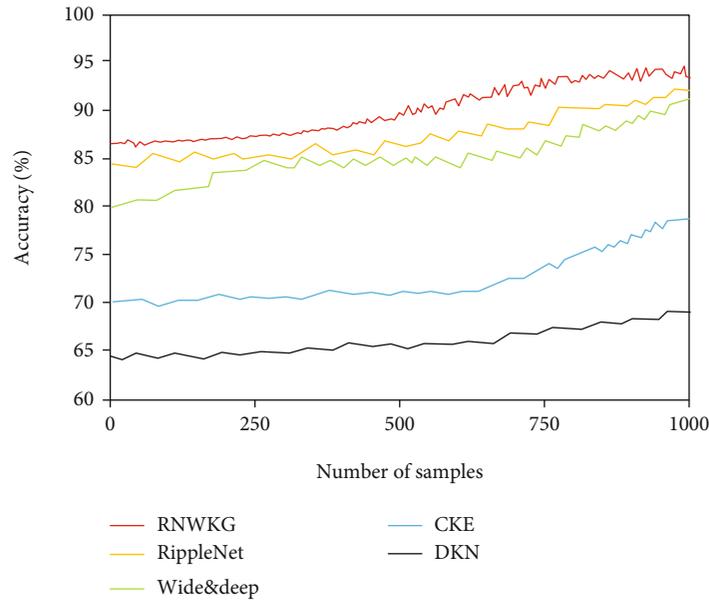


FIGURE 7: Comparison of accuracy of five models in Book-Crossing.

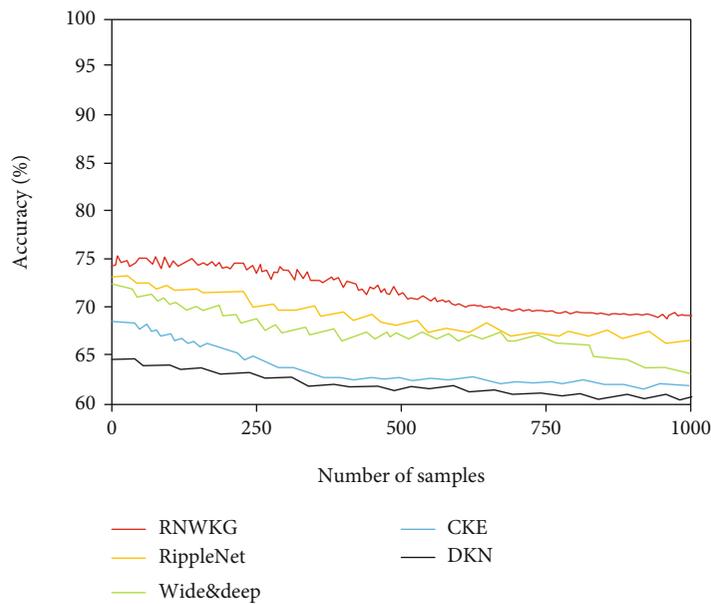


FIGURE 8: Comparison of accuracy of five models in MovieLens-1M.

In Equation (20),  $\sigma$  is the Sigmoid activation function. The expression of the loss function of the algorithm is shown in

$$L = L_{KG} + L_{RS} = \sum_{(c,r,d) \in S} \sum_{(c',r',d') \in S'} \left[ \tilde{d}(c+r,d) - \tilde{d}(c'+r',d') + \gamma \right]_+ + \sum_{u \in U, v \in V} \partial(\hat{y}_{uv}, y_{uv}). \quad (21)$$

In Equation (21),  $L_{KG}$  is the loss function of knowledge graph entity embedding,  $L_{RS}$  is the loss value of click-

through rate prediction, and  $\partial$  is the cross-entropy loss function of actual probability and predicted probability.

#### 4. Performance Analysis of Intelligent Recommendation Algorithm Combining RNN and Knowledge Graph

4.1. Model Comparison of Intelligent Recommendation Algorithms Combining RNN and Knowledge Graph. The study takes two scenarios of book and movie recommendations as examples, and the datasets used are Book-Crossing and MovieLens-1M datasets. In the experimental model

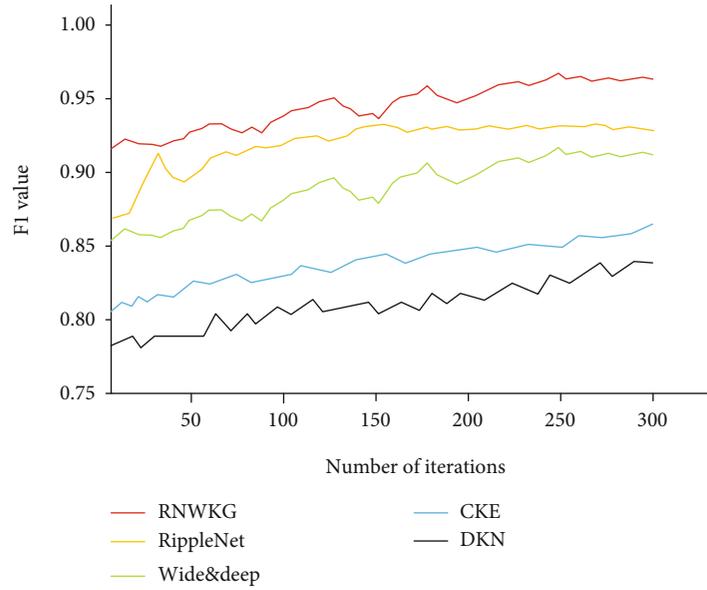


FIGURE 9: Comparison of  $F1$  values of five models in Book-Crossing.

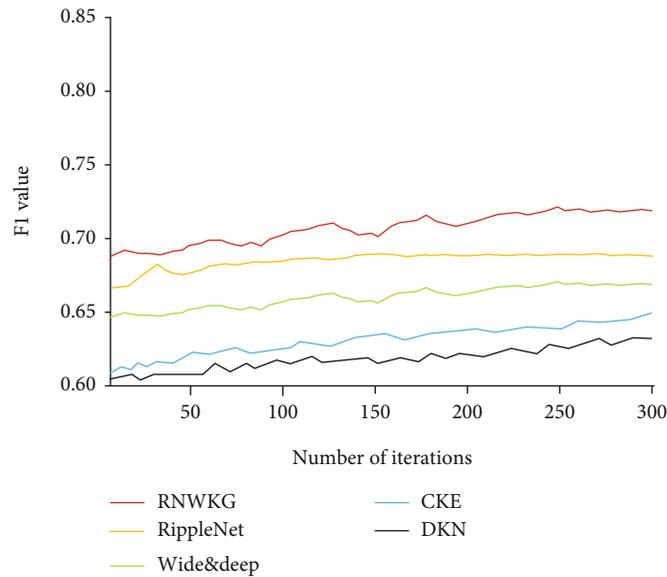


FIGURE 10: Comparison of  $F1$  values of five models in MovieLens-1M.

setup, the models used in the study are compared and analyzed with the following models: the CKE (collaborative knowledge base embedding) model [22], the DKN (deep knowledge-aware network for news recommendation) model [23], the wide and deep model [24], and RippleNet model [25]. Firstly, we combine the knowledge graph and collaborative filtering information to construct the weight knowledge graph. Then the threshold value is set. In the book and movie dataset, the data with a rating of not less than 8 is set to 1, indicating that the user likes the book or movie more. In the Book-Crossing [26] and MovieLens-1M datasets [27], the data that the user has not seen or has

not participated in the rating is set to 0, which means the user does not like the book or movie. The longest interpretable path  $L$  is set to 4 and 3, the weight threshold  $\tau$  is set to 0.06 and 0.1, and the number of hops  $H$  is set to 2. For the parameter settings of the other four models, see References [22–25]. The results of the five models are compared in the book and movie datasets. Among them, the ROC comparison of different models in the Book-Crossing datasets is shown in Figure 5.

As can be seen from Figure 4, the ROC curve combining RNN and knowledge graph works best in the Book-Crossing datasets, and its corresponding lower area AUC value is the

TABLE 1: Scoring error of ten users.

User number	Book number	User rating	Forecast score	Error
18	6	8	8.103	1.29%
65	33	9	8.856	1.6%
79	58	9	9.124	1.38%
101	74	7	7.108	1.54%
134	91	6	5.923	1.28%
289	112	8	7.841	1.99%
289	153	7	7.095	1.36%
412	178	7	6.904	1.37%
503	178	9	8.927	0.81%
791	239	6	5.941	0.98%
925	301	8	8.092	1.15%

highest at 0.931. Among them, the DKN model has the lowest AUC value, which indicates that the method is less effective in extracting information in the Book-Crossing datasets. The ROC curves of the five models in the MovieLens-1M datasets were also compared, as shown in Figure 6.

Figure 6 shows the comparison of the ROC curves of five different models in the MovieLens-1M datasets, and it can be seen from the figure that the ROC curve combining RNN and knowledge graph has the best effect, with the highest AUC value of 0.784 and the lowest AUC value of 0.617 for the DKN model. However, the performance is better in the Book-Crossing datasets. The reason is that there are more knowledge graph relations in the Book-Crossing datasets, which can produce better results when controlling them by using weights. The accuracy of the five models in the two datasets is also compared, and the comparison in the Book-Crossing datasets is shown in Figure 7.

As can be seen from Figure 7, the accuracy rates of the five models gradually increase and level off as the number of samples increases. Among them, the accuracy rates of CKE model and DKN model are lower compared with the other three algorithmic models, 76.45% and 66.72%, respectively, indicating that the extraction effect of CKE model and DKN model is poor. In addition, the algorithmic model combining RNN and knowledge graph achieves the highest accuracy, and its rising trend is smoother as the number of samples increases. When the number of samples is 1000, its accuracy is 93.37%, which is 2.45% higher than the RippleNet model, 6.87% higher than the wide and deep model, 16.92% higher than the CKE model, and 26.65% higher than the DKN model by 26.65%. Also, the accuracy of the five models in the MovieLens-1M datasets is compared, as shown in Figure 8.

As can be seen from Figure 8, the extraction accuracy of all five algorithmic models in the MovieLens-1M datasets is reduced compared to the Book-Crossing datasets. It is the same reason as the five algorithmic models achieved lower AUC values in the MovieLens-1M datasets. With the increase of sample size, the accuracy rate also shows a slow increasing trend. When the number of samples reaches 1000, the algorithm model combining RNN and knowledge

graph achieves the highest accuracy of 74.15%, which is 2.97% higher than the RippleNet model, 3.31% higher than the wide and deep model, 8.32% higher than the CKE model, and 11.43% higher than the DKN model. The  $F1$  values of the five models in the two datasets were also compared, and the comparison in the Book-Crossing datasets is shown in Figure 9.

As shown in Figure 9, the  $F1$  values of the five models show a slowly increasing trend with the increase of the number of iterations in the Book-Crossing datasets, which is consistent with the trend of the accuracy rate. The  $F1$  values of the CKE and DKN models are also lower than the other three models. Meanwhile, the  $F1$  values of the five models in the MovieLens-1M datasets are compared, as shown in Figure 10.

As can be seen from Figure 10, the  $F1$  values of all five models in the MovieLens-1M datasets decreased compared to the  $F1$  values of the five models in the Book-Crossing datasets. With the increase of the number of iterations, the  $F1$  values of the five models show a slowly increasing trend and tend to be stable. Among them, the algorithm model combining RNN and knowledge graph achieves the highest  $F1$  value. The change of the AUC and ACC values together shows that the recommendation algorithm combining RNN and knowledge graph has the best performance.

*4.2. Performance Analysis of Intelligent Recommendation Algorithms Combining RNN and Knowledge Graph.* Ten users in the Book-Crossing datasets were randomly selected for error analysis on the ratings of different books, and the results are shown in Table 1.

As can be seen from Table 1, the predicted scores of ten users randomly selected from the Book-Crossing datasets scores, extracted by combining RNN and knowledge graph, do not differ much from the true scores, and the error values generated are low (less than 2%). The algorithm predicts the scores of 7.841 and 7.095 for the user with number 289, whose scores for different books (numbers 112 and 153) are 8 and 7, respectively, and the corresponding errors are 1.99% and 1.36%, indicating that the errors generated by the algorithm for the same user on the scores of different books are not very different. Meanwhile, for the book numbered 178, two different users (numbered 412 and 503) rated it 7 and 9, and the algorithm predicted 6.904 and 8.927, with an error of 1.37% and 0.81%, respectively, indicating that the error of the algorithm for different users on the same book is not significant, which proves the high performance of the algorithm used in the study. The recommendation algorithm used in the study has a high performance. In order to further verify the effectiveness of combining RNN and knowledge graph recommendation algorithms, two other different recommendation algorithms are introduced: LDA-ALS algorithm and SVD algorithm, and the RMSE and MAE values of the three algorithms are compared in actual operation, as shown in Figure 11.

As can be seen from Figure 11, the recommendation algorithm combining RNN and knowledge graph has the lowest RMSE value and MAE value among the three recommendation algorithms. Its RMSE value is 0.467, which is

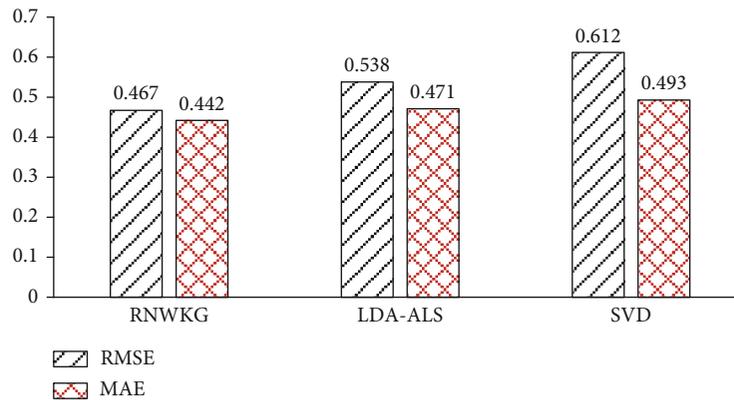


FIGURE 11: Performance comparison of three recommended algorithms.

13.2% lower than that of the LDA-ALS algorithm and 23.7% lower than that of the SVD algorithm. Its MAE value is 0.442, which is 6.2% lower than the LDA-ALS algorithm and 10.3% lower than the SVD algorithm. It can be seen that the RNWKG algorithm performs better and is more stable than the LDA-ALS algorithm and the SVD algorithm, and it produces better recommendation results.

## 5. Conclusion

Smart recommendation algorithms use deep learning techniques to extract effective features from a large amount of user data, thus forming different user profiles and recommending users' favorite information based on historical preferences. The study combines recurrent neural network algorithms with knowledge graphs and uses the advantages of both to study and improve the intelligent recommendation algorithms. The results show that the AUC and ACC values of the five recommendation models are higher in the Book-Crossing datasets than the MovieLens-1M datasets, indicating that the five models have the best extraction effect in the Book-Crossing datasets. And among the five recommendation models, the recommendation algorithm model combining RNN and knowledge graph has the highest AUC value and ACC value. At the same time, the ratings of several users were randomly selected in the Book-Crossing datasets, and it was found that the predicted ratings of the recommendation algorithm combining RNN and knowledge graph were closer to the real ratings, and the error values were smaller (less than 2%), indicating that it has better extraction and prediction performance. Comparing the RMSE and MAE values of the results of the three different recommendation algorithms, it is found that the recommendation algorithm combining RNN and knowledge graph has the lowest RMSE and MAE values. Its RMSE value is 0.467, which is 13.2% lower than that of the LDA-ALS algorithm and 23.7% lower than that of the SVD algorithm. Its MAE value is 0.442, which is 6.2% lower than the LDA-ALS algorithm and 10.3% lower than the SVD algorithm, indicating its stable performance and good recommendation effect. In summary, the recommendation algorithm combining RNN and knowledge graph has high performance and

can play a greater value in user recommendation. However, there are still some areas for improvement, such as only the book and movie datasets, which can be combined with more application scenarios for analysis and validation. Also, the examination of users' short-term preferences can be added to achieve a more comprehensive effect.

## Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Conflicts of Interest

It is declared by the authors that this article is free of conflict of interest.

## References

- [1] Y. Cui, "Intelligent recommendation system based on mathematical modeling in personalized data mining," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6672036, 11 pages, 2021.
- [2] W. Fu, J. Liu, Y. Lai, and Key Laboratory of Industrial Internet of Things & Networked Control, Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing 400065, China, "Collaborative filtering recommendation algorithm towards intelligent community," *Discrete and Continuous Dynamical Systems*, vol. 12, no. 4-5, pp. 811-822, 2019.
- [3] A. Pimpalkar, M. Singh, S. Sheikh, K. Gedam, and A. Khadgi, "Fake news classification using bi-directional LSTM-recurrent neural network," *Journal of Huazhong University of Science and Technology*, vol. 50, no. 6, pp. 1-9, 2021.
- [4] J. P. Marquez, C. D. O. Ribeiro, E. R. Santoyo, and V. F. Fernandez, "Ethanol fuel demand forecasting in Brazil using a LSTM recurrent neural network approach," *IEEE Latin America Transactions*, vol. 19, no. 4, pp. 551-558, 2021.
- [5] V. Ananthanatarajan, M. S. Kumar, and V. Tamizhazhagan, "Forecasting of wind power using LSTM recurrent neural network," *Journal of Green Engineering*, vol. 10, no. 11, pp. 11105-11115, 2021.

- [6] A. Moghar and M. Hamiche, "Stock market prediction using LSTM recurrent neural network," *Procedia Computer Science*, vol. 170, pp. 1168–1173, 2020.
- [7] Z. Zheng, H. Chen, and X. Luo, "Spatial granularity analysis on electricity consumption prediction using LSTM recurrent neural network," *Energy Procedia*, vol. 158, pp. 2713–2718, 2019.
- [8] O. Laib, M. T. Khadir, and L. Mihaylova, "Toward efficient energy systems based on natural gas consumption prediction with LSTM recurrent neural networks," *Energy*, vol. 177, pp. 530–542, 2019.
- [9] H. Wang, M. U. Longxin, F. Shi, and H. Dou, "Production prediction at ultra-high water cut stage via recurrent neural network," *Petroleum Exploration and Development*, vol. 47, no. 5, pp. 1084–1090, 2020.
- [10] T. L. Chin, K. Y. Chen, D. Y. Chen, and D. E. Lin, "Intelligent real-time earthquake detection by recurrent neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5440–5449, 2020.
- [11] R.-Q. Li, X.-H. Zhou, G.-B. Bian, X.-L. Xie, and Z.-G. Hou, "Recognition of endovascular manipulations using recurrent neural networks," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 7010–7013, Berlin, Germany, 2019.
- [12] M. Shi, "Knowledge graph question and answer system for mechanical intelligent manufacturing based on deep learning," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6627114, 8 pages, 2021.
- [13] P. Do, T. Phan, and B. B. Gupta, "Developing a Vietnamese tourism question answering system using knowledge graph and deep learning," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, pp. 1–18, 2021.
- [14] D. Hou, Z. Zhao, and S. Hu, "Multi-label learning with visual-semantic embedded knowledge graph for diagnosis of radiology imaging," *IEEE Access*, vol. 9, pp. 15720–15730, 2021.
- [15] C. L. Huang and C. C. Huang, "Study on customized knowledge graph of student pilot learning in FITS training," *Journal of Intelligent Fuzzy Systems*, vol. 40, no. 4, pp. 7969–7979, 2021.
- [16] T. Fan, S. Song, H. Li, Y. Bai, Y. Chen, and B. Cheng, "Knowledge graph characteristics of sepsis research based on scientometric study," *Zhonghua Wei Zhong Bing Ji Jiu Yi Xue*, vol. 33, no. 4, pp. 433–437, 2021.
- [17] X. Zeng, X. Tu, Y. Liu, X. Fu, and Y. Su, "Toward better drug discovery with knowledge graph," *Current Opinion in Structural Biology*, vol. 72, pp. 114–126, 2022.
- [18] H. Wu, Y. Chang, J. Li, and X. Zhu, "Financial fraud risk analysis based on audit information knowledge graph," *Procedia Computer Science*, vol. 199, pp. 780–787, 2022.
- [19] Y. Yin, L. Zhang, Y. Wang, M. Wang, Q. Zhang, and G. Z. Li, "Question answering system based on knowledge graph in traditional Chinese medicine diagnosis and treatment of viral hepatitis B," *BioMed Research International*, vol. 2022, Article ID 7139904, 8 pages, 2022.
- [20] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.
- [21] B. Oh, S. Seo, J. Hwang, D. Lee, and K. H. Lee, "Open-world knowledge graph completion for unseen entities and relations via attentive feature aggregation," *Information Sciences*, vol. 586, pp. 468–484, 2022.
- [22] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 353–362, San Francisco, California, USA, 2016.
- [23] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: deep knowledge-aware network for news recommendation," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, pp. 1835–1844, Lyon, France, 2018.
- [24] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, "Adam optimization algorithm for wide and deep neural network," *Knowledge Engineering and Data Science*, vol. 2, no. 1, pp. 41–46, 2019.
- [25] L. U. O. Yutai, X. U. Tao, and X. U. Zhangbo, "Recommended method study based on incorporating complex network ripple net," *Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University*, vol. 39, no. 5, pp. 1070–1076, 2021.
- [26] Z. Huang, X. Xu, J. Ni, H. Zhu, and C. Wang, "Multimodal representation learning for recommendation in Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10675–10685, 2019.
- [27] R. Li and L. F. Capretz, "Assessing the performance of recommender systems with movietweetings and movielens datasets," *International Journal of Innovation, Management, and Technology*, vol. 10, no. 6, pp. 229–234, 2019.