IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 17, 2020, accepted January 5, 2021, date of publication January 18, 2021, date of current version January 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052054

Bond Default Prediction Based on Deep Learning and Knowledge Graph Technology

MA CHI[1], SUN HONGYAN[2], WANG SHAOFAN[1,2], LU SHENGLIANG[1,2], AND LI JINGYAN[1]

[1]School of Computer Science and Engineering, Huizhou University, Huizhou 516007, China
[2]School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China

Corresponding author: Sun Hongyan (ustl_linux@126.com)

This work was supported in the part by the Foundation of Guangdong Educational Committee under Grant 2018KTSCX218, and in the part by the Professorial and Doctoral Scientific Research Foundation of Huizhou University under Grant 2018JB020.

ABSTRACT The traditional financial models used in bond default mainly focus on the analysis and prediction of bonds issued by listed companies, and they lack early warning abilities for a large number of bonds of nonlisted companies. At the same time, there is a great deal of relational data and category data in bond data. It is of great significance for bond default prediction to use these data reasonably, which may bring considerable revenue to companies in the near future. Therefore, this paper uses multisource information from bonds and issuers as well as macroeconomic data to predict bond defaults based on a knowledge graph and deep learning technology. On the basis of constructing a bond knowledge graph, knowledge representation learning technology is used to vectorize the knowledge in the graph, and the extracted vectors are inputted into the deep learning model as features to forecast bond default. The applied model is the deep factorization machine model, and good prediction results are obtained.

INDEX TERMS Default prediction, deep learning, DeepFM, knowledge graph, knowledge representation learning.

I. INTRODUCTION

With the recent epidemic of credit risk in the bond market, bond defaults have occurred frequently in China, especially in 2018. Thus, it is of great significance for bond investors and practitioners to use computer technology to predict bond default based on objective data.

Scientists have done more research on default prediction and credit risk measurement. Altman [1] uses multiple discriminant method and proposes a Z-score credit scoring model to analyze the probability of bankruptcy or bank default. Ohlson [2] applies a logistic function to the calculation of default probability. KMV puts forward the KMV model, which uses stock price and the public financial data of listed companies to measure the expected default rate of loans and bonds [3]. Morgan [4] introduced the Credit Metrics model to quantify credit risk.

At present, the traditional financial model KMV model is mainly used to predict bond defaults in China [5], [6]. Wei [7] combines the KMV model and the Logit regression model to study 12 listed companies and 115 control companies that have bond defaults. The results shows the validity of the classical model and predicts their default risk without relying on the actual sample default data. Hu [8] uses stochastic forest to analyze the default characteristics of bonds and concludes that the types of major shareholders and proportion of shareholders play important roles in default prediction. KMV is more effective for bonds issued by listed companies, but it lacks early warning ability for a large number of bonds issued by nonlisted companies.

With the development of machine learning, neural networks, support vector machines and other models have gradually begun to be used in credit risk prediction. Dutta and Shekhar [9] used a neural network to forecast bond credit rating, which proved the effectiveness of the neural network. Lee [10] uses a support vector machine to predict enterprise credit rating.

Bond default prediction is a data mining problem. In the field of data mining, deep learning has been successfully applied in recommendation systems, including in Click Through Rate (CTR) and Click Value Rate (CVR) prediction. The Wide&Deep model [11] was proposed by

The associate editor coordinating the review of this manuscript and approving it for publication was Xin Luo.

Google in 2016. The model integrates linear model Logistic Regression (LR) and Deep Neural Networks (DNN), which gives the model memory and generalization ability. On this basis, scholars have proposed a series of CTR prediction models combined with in-depth learning, such as Factorization-machine supported Neural Network (FNN) [12], Product-based Neural Network (PNN) [13], and Deep Factorization Machines (DeepFM) [14] and have achieved good results for CTR problems.

With the rapid development of the knowledge atlas, researchers have realized that knowledge graphs can be used as a feature supplement and input into in-depth learning to improve the effect of the model. Zhang *et al.* [15] proposed the Collaborative Knowledge Base Embedding (CKE) model, which embeds the structured knowledge map into the network through the TransR model based on Bayesian improvement. The film vectorization representation is obtained, which is fused with the text knowledge features and image knowledge features. The representation is inputted into the collaborative integrated learning framework, and a personalized recommendation is made by the fusion of this knowledge. Experiments show that film vectorization representation can effectively improve the performance of the model. Wang *et al.* [16] constructed four medical knowledge maps with medical data, used TransR and the LINE model [17] to express the knowledge graph, generated expression vectors, and finally recommended drugs for patients through joint learning. Wang *et al.* [18] applied a knowledge map to news recommendations. A content-based deep knowledge perception network (DKN) was proposed, and the given knowledge map was embedded to improve the performance of news recommendations.

Although the application of knowledge representation learning in deep learning models is still in the exploratory stage, existing models prove that a knowledge graph, as a constraint of prior knowledge, can improve the performance of the model to a certain extent.

Because there are many types of characteristics of bond information, including data, text, and some implied holding relationships, the use of a deep learning model alone cannot well reflect the complex relationship in the bond market, while the importance of the time series data characteristics cannot be reflected directly and simply through the establishment of a knowledge graph for classification prediction. In the bond prediction experiment, the relationship between the hidden layer and the historical time series data is equally important.

Therefore, in order to improve the accuracy and rationality of prediction, we build a hybrid model based on a knowledge graph and deep learning. According to the characteristics of bond information, including multiple relationships, we build a bond knowledge graph. We use the knowledge to represent the learning model to learn the semantic and structural information of the knowledge graph, as prior knowledge of bond default and supplementary input to the deep learning model to improve the effect of the hybrid prediction model.

The rest of this paper is organized as follows: In Section 2, data acquisition and preprocessing are introduced in detail. Section 3 and 4 present the knowledge vector representation of the bond knowledge graph and bond default prediction model based on optimized DeepFM. The detailed experiments and results analysis are given in Section 5. Finally, the conclusion and future studies are given in Section 6.

II. DATA ACQUISITIONAND PREPROCESSING

This paper mainly uses the credit bonds of the interbank and exchange market as the research object to forecast bond default.

A. DATA ACQUISITION

We classify bond related data into four acquisition categories.

1) BOND BASIC DATA

Bond data mainly includes the following three types: bond information, bond issuer information and credit analysis indicators. We obtain bond data from the Wind information platform. The specific indicators of each part are shown in Table 1.

TABLE 1. Bond basic data from Wind.

Type	Specific indicators
Bond information	Bond code, bond abbreviation, listing place, total amount of issuance, listing date, starting date, maturity date, bond maturity, coupon interest rate, tax rate, interest-bearing method, number of interest payments per year, urban investment bonds or not
Bond issuer information	Debt subject, issuer's Chinese name, date of establishment, registered capital, legal representative, chairman, general manager, main product and business, main product type, total number of employees, provinces, cities, listed companies, company attributes, ten shareholder names, shareholding ratio, shareholder types, actual controller names
Credit analysis indicators	Bond Rating and Subject Rating

We use Comma-Separated Values (CSV) format to store the information. Since the first bond default occurred in 2014, we choose bonds that are issued after January 1, 2010, and maturity between January 1, 2014, and September 1, 2018.

The financial information of a bond issuer can well reflect a company's operating situation. The company's earning ability and debt situation will affect whether the company has the ability to pay bonds. This paper obtains financial data such as net asset yield, net sales interest rate, liability growth rate, and current liabilities/total liabilities of bond issuers through a Python quantization interface provided by the Wind platform.

2) MACROECONOMIC DATA

Macroeconomic factors also have a great impact on bond default. Currently, relevant studies have shown that bond

default has a greater relationship with the growth of Gross Domestic Product (GDP) and Clock cycle Per Instruction (CPI) in the macroeconomy [8]. This paper uses the Python quantitative interface provided by Wind to obtain GDP growth rate, regional GDP growth rate, CPI growth rate and industry index from January 2010 to August 2018. Among these, the GDP growth rate is the cumulative year-on-year constant price of GDP, and the data frequency is quarterly. CPI growth rate is cumulative year-on-year, the data frequency is monthly; and the industry index frequency is daily. Specific indicator information is shown in Table 2.

TABLE 2. Macroeconomic indicators.

Name	Meaning	Frequency	Unit
GDP growth rate	Cumulative year-on-year GDP invariance	Quarterly	%
Regional GDP Growth Rate	GDP Growth Rate of 31 Provincial Administrative Regions	Quarterly	%
CPI growth rate	CPI cumulative year-on-year	Monthly	%
Sector index	Sector index of Wind	Day	Point

3) BOND ANNOUNCEMENT

Issuer announcement information is also helpful to the prediction of bond default. The obtained bond announcement information is mainly used for data verification, because the basic bond information and issuing company information are from the interface provided by the Wind terminal, which is needed to extract the important features of bond announcements to verify the accuracy of the data obtained. In addition, by consulting the announcement, we can further understand the major bond issues and the latest operating financial situation of the issuer, which is helpful information for verifying and analyzing our experimental prediction results. We obtain the bond announcement information through a web crawler that stores the information in a MySQL database. For exchange bonds, there are also various bond market data. We obtain these market data through the financial data interface and data providers.

B. DATA PREPROCESSING

We regard the prediction of bond default as a two-class problem. For the bonds we analyze, we mark the bonds that have substantive default as 1 and the bonds that have not defaulted as 0, with a total of 118 defaulted bonds. For defaulted bonds, there are two main types. One is bonds that cannot pay interest or principal and interest at maturity and have a default time as the day of maturity or the next 1-2 days; the other is substantive default caused by failure to pay interest on the annual or designated interest date during the bond life, which is prior to the maturity date of the bond. For the obtained macro data, we take the corresponding index value of the half year before the maturity or default of the bond as the characteristic value. Taking the growth rate of regional GDP as an example, for each bond, first the region is matched, and then the corresponding index value is obtained before the bond maturity date. As the GDP growth rate is released once a quarter, it can be obtained by pushing the GDP growth rate from half a year ago forward by two quarters.

To avoid the influence of data format, missing data and value range on subsequent experiments, we cleaned the bond data. We normalize the numerical data as shown in Formula 1.

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

In the formula, x_{norm} represents the normalized result of the data, x represents the value before normalization, $x_{\min}$ and $x_{\max}$ represent the minimum and maximum values of samples in this feature, respectively.

After pretreatment, we find that bond-related data contains more relational data, for example, shareholder relationship, actual controller relationship, industry relationship and so on. To make full use of these relationship data, we use a knowledge graph to mine the implicit relationship between bonds. At the same time, there are many kinds of bond data characteristics, and the feature correlation on the surface is low. Therefore, we use deep learning to discover low-order and high-order characteristics and predict bond defaults.

III. KNOWLEDGE VECTOR REPRESENTATION OF THE BOND KNOWLEDGE GRAPH

The whole process of knowledge vector representation based on a bond knowledge map is shown in Fig. 1, which is divided into the following steps.

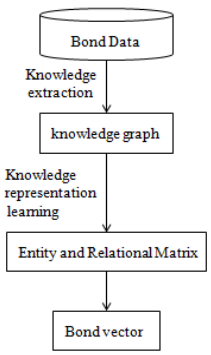


FIGURE 1. Knowledge vector representation based on a bond knowledge graph.

- Construct a bond knowledge graph based on existing data.
- For the constructed knowledge map, the knowledge representation learning model is used to learn, and the entity matrix and the relation matrix are obtained.
- Correspond the entity matrix with the entity to obtain the required bond knowledge representation.

We use the structured data in Wind database as the data source to construct the knowledge graph, and use the top-down method to build the graph. We extract entities, entity attributes and the relationship between entities, generate the data format required by the knowledge representation

FIGURE 2. Bond knowledge graph.

model, and train the entity vector through the knowledge representation model. The bond vector representation is obtained from the trained entity matrix.

A. CONSTRUCTION OF THE BOND KNOWLEDGE GRAPH

There are two main ways to construct a knowledge graph: top-down and bottom-up. Since the ultimate goal of constructing a bond knowledge map is to provide knowledge for bond default prediction, we adopt a top-down approach to build the graph. We use structured data obtained from Wind as the source of the knowledge graph to construct the bond knowledge graph.

In the graph, entities are represented by nodes, and the edges connecting nodes represent the relationships between entities. The relationship is directed, and the final result is a directed graph. We take bonds, companies, provinces, industries, people, and bond types as entities, including the relationship between the issuance of bonds between companies, legal persons, chairmen, general managers and other positions, shareholders, actual controllers and so on. At the same time, the bond code, bond issuance time, maturity time, company registered capital and so on are taken as the attributes of the corresponding entities. Specific entities and relationship information are shown in Table 3.

After this process, we extract the entity, entity attributes and the relationship between entities and complete construction of the bond knowledge map. The Neo4j graph database is selected as the storage database to store the constructed bond knowledge map. Fig. 2 shows the knowledge map we finally constructed, which contains 25242 nodes and 11 relationships.

TABLE 3. Entity, relation and attribute statistics.

Type	Name	Quantity
Entities	Bond, location (bond listing place, issuer's province/city), company, person, industry, bond type, shareholder type	25242
Relationship	Issue relationship, listing location relationship, industry relationship, provincial-urban relationship, bond type relationship, legal person relationship, chairman relationship, general manager relationship, shareholder relationship, major shareholder type relationship, actual controller relationship	11
Attributes	Bond attributes: securities code, coupon rate, total issuance, issuance time, maturity time, bond rating and other attributes Corporate attributes: date of establishment, registered capital, whether listed companies, total number of employees, major products and business attributes	/

Taking the bond "15 Le Shi 01" as an example, Fig. 3 shows the query results of the bond. Different types of entities are represented by nodes of different colors.

The corresponding node and the specific attributes of the node are shown below, including the corresponding coupon interest rate, total issuance, bond rating and other attributes.

B. KNOWLEDGE VECTOR REPRESENTATION OF THE BOND KNOWLEDGE GRAPH

The knowledge map represented by symbols is difficult to be directly used by computers. However, knowledge representation learning can embed entities and relationships

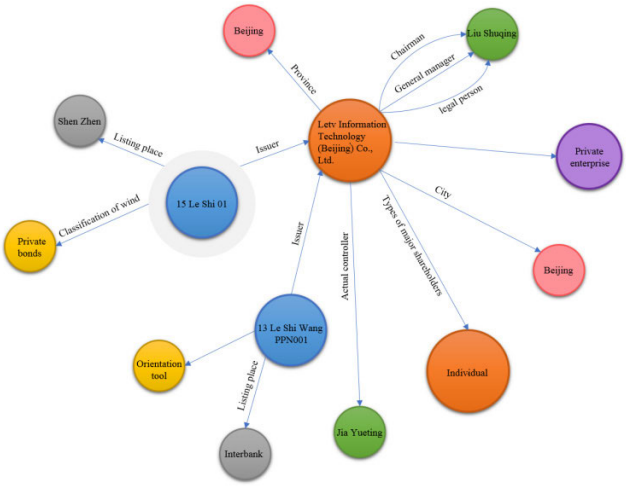


FIGURE 3. Bond knowledge graph query.

in a knowledge graph into vector space, express them in the form of vectors, and input them into machine learning and deep learning models as features. The training vectors retain the semantic information and structure of the original graph.

We preprocess the bond knowledge map to generate the data format needed for the knowledge representation model. First, we numbered all entities; each entity was given a unique id, and each relationship number was given a unique ID. Then, according to the labeled entity ID and relationship id, each pair of triples (h, t, r) is mapped with ID to get triples in the form of head entity ID, tail entity ID and relationship ID, where h is the head vector, t is the tail vector, and r is the relationship vector. The knowledge representation of learning document generated from the knowledge graph after the above processing is shown in Table 4.

TABLE 4. Knowledge representation learning documents.

file name	Content	Quantity
entity2id.txt	entity-id pair	25242
relation2id.txt	relation-id pair	11
triple2id.txt	Triple represented by id	72905

Entity2id file: There are a total of 25242 entity-id pairs, including 17624 bond entities.

Relationon2id file: There are 11 relationship-id pairs, including the issuance relationship between company and bond, the position relationship between legal person, chairman and general manager, the relationship between shareholders and actual controllers.

Triple2id file: There are a total of 72905 pairs of triples, and each pair of triples are represented by the corresponding ID.

The core idea of the knowledge representation model is to embed the entities and relationships in the knowledge graph into the m-dimensional space and learn a low-dimensional dense vector for each entity. The vector contains the similarity between entities and the network structure information of the graph. Knowledge representation learning reduces the high dimension and heterogeneity of the knowledge graph and reduces the extra computational burden caused by the introduction of the knowledge graph. At the same time, the continuous low-dimensional vector can also be easily inputted into machine learning, deep learning and other models, so that the model can make better use of the symbolic knowledge in the knowledge graph and further improve the performance of the model. The commonly used knowledge representation models include the TransE model, which was proposed by Borders in 2013 [19]. It is a knowledge representation learning model based on translation. Wang *et al.* proposed the TransH model in 2014 [20]. TransH maps the relationship to a hyperplane, which balances the complexity of the model and the ability to express it. Lin *et al.* proposed the TransR model [21]. The TransE and TransH models embed entities and relationships in the same vector space, without considering that they are essentially different objects that may not be well represented in the same vector space. At the same time, an entity may have multiple semantic attributes, which may correspond to different relationships. Although TransH maps relationships to hyperplanes, it still cannot break the constraints on entities and relationships in the same space.

Therefore, for the preprocessed knowledge graph, we use TransR as the training entity vector of the knowledge representation model [21]. This model embeds entities and relationships into two different spaces, and the entities in the entity space are projected into the relational space through the entity-relational projection matrix M_r. For triples (h, t, r), head vectors h and tail vectors t are projected by projection matrix M_r to obtain the projected head vectors h_r and tail vectors t_r.

$$\begin{aligned} h_r &= hM_r \\ t_r &= tM_r \end{aligned} \tag{2}$$

h_r and t_r are connected by relational vectors r. Those entities that were previously close to each other in physical space would be far away from each other in some specific relational space, as shown in Fig. 4.

The corresponding scoring function is defined as follows.

$$f_r(h, t) = \|h_r + r - t_r\|_2^2 \tag{3}$$

The loss function is defined as follows.

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'_{(h,r,t)}} [\gamma + f_r(h, t) - f_r(h', t')]_+ \tag{4}$$

Here, γ is a marginal parameter, $[x]_+$ is a hinge loss function, and $S'_{(h,r,t)}$ is a constructed error tuple.

When generating negative samples in the training process, we choose the Bern negative sampling method in the TransH model to generate negative samples, because this sampling method is more reasonable than others. The sampling method proposed in TransE is called the Unif sampling method, and the sampling method proposed in TransH is called the Bern sampling method. There is a many-to-one relationship and a one-to-many relationship in the knowledge graph we build. Taking many-to-one as an example, for bond-to-bond

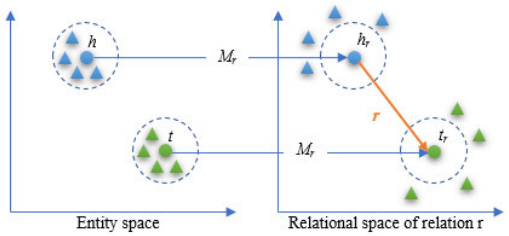


FIGURE 4. TransR model.

types, multiple bonds may correspond to the same bond type. "15 Le Shi 01," "13 Xin Tian Yang," and "13 Bai Chuan" bonds are all private placement bonds. In the case of many-to-one, if the Unif sampling method is used to replace the head and tail entities with the same probability, when entities are randomly selected from the entity set to replace the tail entities, the negative examples that are easy to generate are not actually negative examples but are still correct triples. Specifically, the known triples (15 Le Shi 01, bond type, private placement bond) and (13 Xin Tian Yang, bond type, private placement bond) are both correct triples. When only Le Shi is present in the training set and generates a negative example for the triple, it is possible to replace the header entity "15 Le Shi 01" with "13 Xin Tian Yang" to generate a triple (13 Xin Tian Yang, bond type, private placement bond) that can be considered a negative example.

The Bern sampling method uses different probabilities to replace the head and tail entities for many-to-one, one-to-many, and many-to-many relation triplets. For many-to-one relations, a larger probability replaces the tail nodes, and for one-to-many relations, a larger probability replaces the head nodes. This sampling method is more reasonable, so we use this method to generate negative samples.

For relation r, *tph* represents the average number of head entities corresponding to tail entities, and *hpt* represents the average number of head entities corresponding to tail entities. In the construction of negative cases, the head node is replaced by the probability $\frac{tph}{tph+hpt}$, and the end node is replaced by the probability $\frac{hpt}{tph+hpt}$. That is, for many-to-one relationships, a larger probability replaces the tail node, and for one-to-many relationship, a larger probability replaces the head node.

After model training, the following three matrices are generated: the entity matrix, the relation matrix and the projection matrix. The bond vector from the entity matrix is shown in Fig. 5.

```
5980,14 Zheng Zhou Jiao Tou PPN001,-0.007828794,0.0021384577,-0.0029689022,0.0019
2667,14 Zhong Ye SCP004,-0.038404092,0.01705473,-0.015120124,-0.0013837904,-0.039
9385,15 Ping AN CD185,0.017166318,-0.037501357,0.015627874,-0.042882107,0.077600
5463,14 Gan Gong Tou CP001,0.012970981,-0.030862646,0.029915318,-0.02410593,0.087
7018,15 Zhong Xin CP002,0.019255757,-0.031875037,0.0012651284,-0.032110825,0.0559
13728,16 Pang Da Qi Mao CP001,0.0010844995,-0.0053431815,-0.0039749043,-0.014842
3912,14 Tian Fu CP001,-0.0024695555,0.023539018,0.00796744,0.007286392,-0.0146530
```

FIGURE 5. Bond vector representation.

As shown in the figure above, the first column is the ID of the bond, the second column is the name of the bond, and the third column is the vector representation of the bond.

IV. BOND DEFAULT PREDICTION MODEL BASED ON OPTIMIZED DeepFM

The Factorization Machines (FM) was proposed by Steffen Rendle in 2010 [22]. It considers the correlation between features and can learn from a sparse matrix very well. It is a general model that can be used in any case where the feature is a real value. In the general linear model, the features are considered separately, without considering their relationship. However, in fact, many of the features are related.

The general linear model is as follows:

$$y = w_0 + \sum_{i=1}^{n} w_i x_i \tag{5}$$

It does not consider the association between features. The FM model is proposed to solve the problem of how to combine features. For simplicity, the second-order polynomial model is generally discussed.

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_i x_j \tag{6}$$

Among them, n represents the number of features after one hot, x_i represents features i, and w_o, w_i, and w_{ij} are model parameters. It can be seen that this model has more polynomial parts than the general linear model, and $x_i x_j$ represents a combination of features x_i and x_j. However, due to the sparse sample data, the nonzero term of $x_i x_j$ will be few, and the lack of training samples will lead to w_{ij} inaccuracy.

To find w_{ij}, a hidden vector $v_i = (v_{i1}, v_{i2}, \cdots, v_{ik})$ is introduced for each feature x_i. Parameter w_{ij} constitutes a symmetric matrix W, which can be decomposed into $W = V^T V$, that is to say, each parameter $w_{ij} = \langle v_i, v_j \rangle$, so we can get the following:

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j \tag{7}$$

Among them,

$$\langle v_i, v_j \rangle = \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \tag{8}$$

Finally, the second term of FM is simplified:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j = \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2 x_i^2 \right) \tag{9}$$

After simplification, the complexity of FM is optimized to $O(kn)$, that is, its time complexity is linear.

The FM algorithm extracts feature combination by the implicit variable inner product of each one-dimensional feature. Although in theory, FM can model high-order feature

combination, in fact, only second-order feature combination is used because of the complexity of calculation.

Then, for the high-order feature combination, it can be solved by the neural network of multilayer structure, namely, DNN.

The concept of DNN comes from a typical multilayer structure Multilayer Perception (MLP) in traditional neural network, which consists of input layer, output layer and hidden layer. DNN can be understood as a neural network with many hidden layers, and all layers are connected.

Although DNN can implicitly reflect the combination of low-order and high-order features in the hidden layer, at this time, the combination of low-order features cannot be modeled separately. To solve this problem, Guo *et al.* [14] and others at the Harbin University of Technology have integrated DNN and FM model and proposed DeepFM model.

DeepFM [23], as a deep learning model in the field of CTR prediction, can well learn low-order and high-order combinatorial features without manual feature extraction and has a strong ability to learn from sparse data. In the sample of bond default prediction in this paper, there are many types of features, which become sparse after one-hot encoding. At the same time, these types of features have low correlation on the surface. We need to use deep neural network to further mine the association features. Therefore, the DeepFM model is used to better learn the correlation characteristics between bonds. Based on the DeepFM model, this paper introduces the knowledge graph as feature embedding of the model and proposes an optimized DeepFM model that integrates the knowledge graph information.

A. CONSTRUCTION OF THE OPTIMIZED DeepFM MODEL

1) CONSTRUCTION OF DeepFM

For default forecasting, we hope to learn the characteristic combination behind bond default. Low-order combination features or high-order combination features may have an impact on the final prediction results. DeepFM is composed of factor decomposer FM and neural network DNN. Its model structure is shown in Fig. 6.

As shown in the figure above, it is a parallel structure with the same input shared at the bottom. DeepFM model solves the problem of learning low-order and high-order features at the same time. Its main models can be expressed as follows:

$$\hat{y} = \text{sigmoid}(y_{\text{FM}} + y_{\text{DNN}}) \tag{10}$$

In Formula 10, y_{FM} is the output of FM, and y_{DNN} is the output of DNN. FM is responsible for extracting low-order features, and DNN is responsible for extracting high-order features. Finally, FM and DNN results are combined to activate the output.The model structure of the FM part is shown in Fig. 7.

The Deep part is a fully connected DNN, which is used to learn higher-order feature combinations. The network structure of the Deep section is shown in Fig. 8.

Unlike the input of image or voice classes, some of the features of bond classes are very sparse after one-hot encoding. Therefore, DeepFM adds an embedded layer before the first hidden layer to convert the input vector into a dense low-dimensional vector. The structure of the embedded layer is shown in Fig. 9.

First, the feature is divided into different fields, and the same field represents the same feature. For each input record, only one neuron in a field has a value of 1, and the others are all 0. That is, for the embedding process, only one neuron in each field works. Suppose that $k = 5$, and the weights V_{i1}, V_{i2}, V_{i3}, V_{i4} and V_{i5} of the five lines from the input layer to the embedding layer connected with the neuron are the hidden vectors V_i introduced in FM. DNN takes the hidden

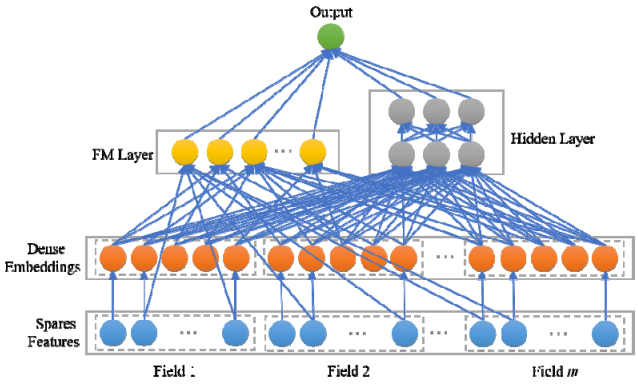


FIGURE 6. Structure of the DeepFM model.

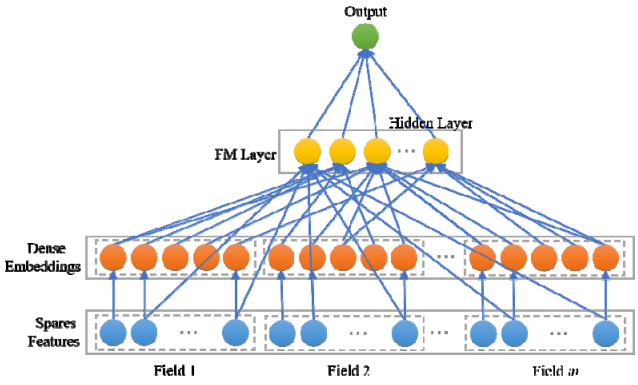


FIGURE 7. The model structure of the FM part.

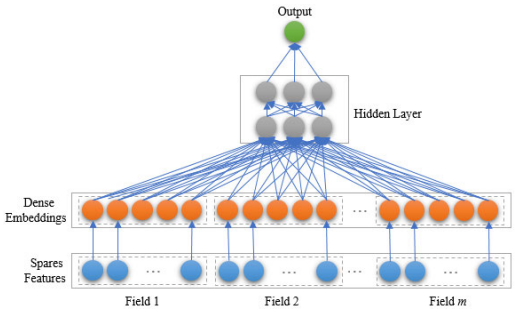


FIGURE 8. Deep partial network structure.

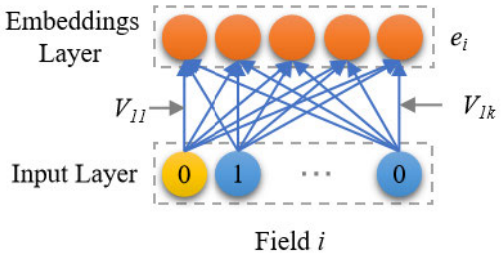


FIGURE 9. Structure of embedded layer.

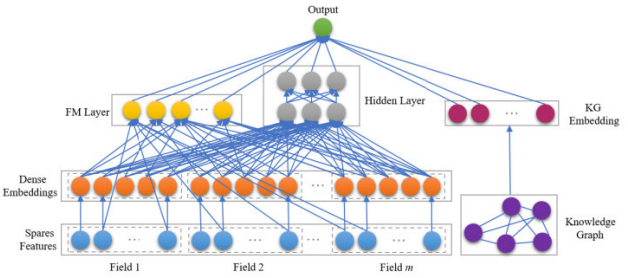


FIGURE 10. Structure of the DeepFM-KG model.

vector V_i from FM as the weight of the embedded layer network, which shares the weight.

The output of the embedded layer is:

$$a^{(0)} = [e_1, e_2, \cdots, e_m] \tag{11}$$

In the formula, e_i represents the embedding of the i_{th} field, and m is the number of fields. Then, $a^{(0)}$ is passed into DNN, and the output of the next layer is:

$$a^{(l+1)} = \sigma\left(W^{(l)}a^l + b^l\right) \tag{12}$$

In the formula, $a^{(l)}$ represents the output, l represents the number of layers, σ is the activation function, $W^{(l)}$ is the weight of the model, and $b^{(l)}$ is the bias of the l layer. Finally, DNN outputs a dense real-valued vector and combines a SIGMOD function with FM output to get the final prediction result. The DNN output is:

$$y_{\text{DNN}} = \sigma\left(W^{|H|+1} \cdot a^H + b^{|H|+1}\right) \tag{13}$$

In this formula, $|H|$ is the number of hidden layers.

FM and Deep share the same feature embedding, which enables the model to learn low-order and high-order feature interactions from the original features.

2) OPTIMIZED DeepFM

Based on DeepFM, this paper proposes a Deep Factorization Machines-Knowledge Graph (DeepFM-KG) model that integrates the semantic information of the knowledge graph. The knowledge graph is embedded in the n-dimensional space by representation learning, and the bond vector representation is obtained. The model is combined with the results of FM and DNN and output through the sigmoid layer. In the actual modeling process, we use DeepFM's idea of concurrent integration of FM and DNN to further design the network structure for bond default prediction. At the same time, bond vectors obtained from knowledge representation learning training are added to the model training process, and the final training output is obtained. The structure of the DeepFM model with knowledge representation is shown in Fig. 10.

B. MODEL TRAINING AND OPTIMIZATION

1) TRAINING INPUT AND OUTPUT

The DeepFM model is a supervised learning model, so it is necessary to design the corresponding training set. We input embedding encoded discrete features such as provinces and industries into FM as features. The DNN part of the neural network accepts continuous data input and normalizes the continuous features, such as the coupon rate at the time of bond issuance and the proportion of large shareholders' shareholding, and then inputs them into the DNN structure. Finally, the output of FM, DNN and the corresponding bond vectors from knowledge representation learning are put into the sigmoid activation function for training, and the final output results are obtained. The input of the model is shown in Formula 14.

$$y = \text{sigmoid}(y_{\text{FM}} + y_{\text{DNN}} + y_{\text{KG}}) \tag{14}$$

In the formula, y_{FM} represents FM output, y_{DNN} represents DNN output, and y_{KG} represents vectors trained from the bond knowledge graph.

2) NETWORK DESIGN

In construction of the DNN network, batch-normalization is applied to the input of each layer, so the values of each layer are passed down in an effective range, thus improving the learning efficiency. The Rectified Linear Unit (ReLU) function is selected as the activation function. The function schematic is shown in Fig. 11. In the positive interval, its derivative is constant, so the problem of gradient disappearance is avoided, and the convergence speed of the model is faster.

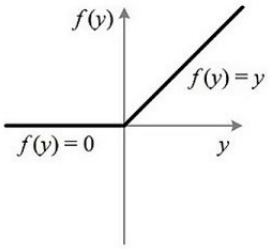


FIGURE 11. Diagram of the ReLU function.

3) OPTIMIZATION OF TRAINING METHOD

When training the model, it is necessary to select the best optimizer so the model can converge quickly and learn correctly, while minimizing the loss function to the greatest extent. Adaptive moment estimation (Adam) works well in practical

application, so this paper uses Adam as an optimizer to train the model in the construction of the DeepFM model.

The essence of the Adam algorithm is that the current gradient updating utilizes the exponential decay mean $\widehat{m_t}$ of the gradient m_t at the previous moment and the exponential decay mean $\widehat{v_t}$ of the square gradient v_t at the previous moment. g_t represents the first derivative of the objective function to the parameters at time t. m_t and v_t can be obtained by the following formulas.

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1)\, g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2)\, g_t^2 \end{aligned} \tag{15}$$

According to the above formula, we can calculate:

$$\begin{aligned} \widehat{m_t} &= \frac{m_t}{1-\beta_1^t} \\ \widehat{v_t} &= \frac{v_t}{1-\beta_2^t} \end{aligned} \tag{16}$$

Finally, the gradient updating method is:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\widehat{m_t}}{\sqrt{\widehat{n_t}} + \epsilon} \tag{17}$$

In the formula, m_t represents the gradient, $\widehat{m_t}$ represents the corresponding exponential decay mean, v_t represents the square gradient, $\widehat{v_t}$ represents the corresponding exponential decay mean, β_1 represents the exponential decay rate of m_t, β_2 represents the exponential decay rate of v_t, and η represents the learning step.

V. EXPERIMENTS AND RESULTS ANALYSIS

A. EXPERIMENTAL DATA

This paper chooses bonds in the interbank and exchange markets as the experimental data. Taking January 1, 2010, as the start date and September 1, 2018, as the end date, the government bond data were excluded, and a total of 17624 matured bonds were finally obtained as the experimental subjects; the defaulted bonds were labeled. The defaulted sample is marked as positive sample 1, and the remaining bonds are marked as 0. A total of 118 defaulted bonds are marked.

The bond default forecasting samples are out of balance, so we randomly sampled negative samples and retained all positive samples before training. When dividing the training set and the test set, 5-fold cross validation is used.

In each training, for samples in the training set, a positive sample is duplicated by the upsampling method, so the ratio of positive and negative samples in the final training set is approximately 1:15. Precision Recall Curve (PRC) is used as the evaluation index to evaluate the predicted results of the classifier.

B. EXPERIMENTAL SETUP

By constructing a bond knowledge map and training bond knowledge representation, an optimized deep learning model based on the knowledge graph is designed. The knowledge representation of bonds is taken as part of the model input, and the default prediction of bonds is realized by model training. The model is implemented with the Keras framework. After several groups of comparative experiments, the optimal parameters of the model are obtained. The specific parameters are shown in Table 5.

TABLE 5. Optimized DeepFM model parameters.

Parameter	Value
DNN Hidden Size	3
DNN Activation	ReLU
Learning rate	0.005
Batch size	64
Epoch	1
Optimizer	Adam
Loss	binary_crossentropy

When the distribution of positive and negative samples is very uneven, that is, when the number of negative samples is much larger than that of positive samples, the PRC can be used to measure the classifier more effectively [24]. The samples are not in balance in this paper, so we finally choose PRC as the evaluation criterion to evaluate the model by comparing the area under the PRC curve. PRC is sensitive to unbalanced data and can evaluate whether the classifier is good or bad for overall classification. The abscissa of PRC is recall rate, and the ordinate is precision.

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$Precision = \frac{TP}{TP + FP} \tag{19}$$

TP represents the number of samples predicted to be 1 and actually 1. *FP* represents the number of samples predicted to be 1 and actually 0. *FN* represents the number of samples predicted to be 0 and actually 1.

C. EXPERIMENTAL RESULTS

In this paper, five groups of experiments are compared and analyzed from different angles on the issue of bond default.

1) CONTRAST EXPERIMENT OF DeepFM AND THE TRADITIONAL MACHINE LEARNING MODEL

This experiment explores the effects of DeepFM and traditional machine learning models on bond default prediction. The LR, Support Vector Machine (SVM), Support Vector Regression (SVR), eXtreme Gradient Boosting (Xgboost) and DeepFM models are compared under the same input characteristics. The experimental results are shown in Table 6.

TABLE 6. Prediction results of DeepFM and the traditional machine learning model.

Model	PRC
LR	0.322
SVM	0.504
SVR	0.569
Xgboost[47]	0.601
DeepFM	0.778

From Table 6, we can see that DeepFM is superior to other traditional machine learning models in model performance. This shows that the deep learning model has a strong ability to learn higher-order cross-features. It also proves the importance of learning high-order cross-features in bond default prediction.

2) CONTRAST EXPERIMENT OF DNN WITH DIFFERENT LINEAR LAYERS

This experiment explores the effects of DNN with different linear layers on bond default prediction results. In the case of the same input characteristics, DNN, LR + DNN, DeepFM and the results of knowledge graph embedding are added to each model. The model name has + KG, which means that the knowledge map embedding information is added. The knowledge representation learning model of knowledge graph adopts TransR model. The results are shown in the following table.

It is obvious from Table 7 that adding knowledge graph semantic information can improve the performance of the model. By learning the representation of the knowledge graph, the semantic and structural information contained in knowledge graph can be used to the greatest extent. At the same time, the experimental results also prove the influence of the first-order features and the second-order features on the model.

TABLE 7. Prediction results of DNN with different linear layers and KG.

Model	PRC
DNN	0.760
DNN+KG	0.775
LR+DNN	0.762
LR+DNN+KG	0.781
DeepFM	0.778
DeepFM+KG	0.802

3) CONTRAST EXPERIMENT OF KNOWLEDGE REPRESENTATION MODEL

This experiment explores the influence of different knowledge representation models on the experimental results. For the constructed bond knowledge graph, the TransE, TransH, and TransR models are trained, from which the bond representation is obtained and input into the subsequent model as a feature. The Bern sampling method was used in training, and the training dimension was 80 dimensions. The experimental results are shown in Table 8.

From the above table, it can be seen that the TransR model, as the embedding of knowledge representation learning model, improves the model results most obviously. The results may occur because the entities in the knowledge graph have a variety of semantic attributes, corresponding to different relationships. The TransR model considers that an entity has multiple semantic attributes and embeds entities and relationships into different spaces. It also proves that considering heterogeneous information can improve the effect of knowledge embedding.

TABLE 8. Results of DeepFM with different knowledge representation models.

Model	PRC
DeepFM+TransE	0.782
DeepFM+TransH	0.788
DeepFM+TransR	0.801

4) CONTRAST EXPERIMENT OF DIFFERENT EMBEDDING DIMENSIONS IN TransR MODEL

This experiment explores the influence of different embedding dimensions of the knowledge representation model on the final results. Entity and relationship embedding dimensions D are selected in {20, 50, 80, 100, 200}, respectively. The TransR model is selected in the model. The experimental results are shown in Table 9.

TABLE 9. Prediction results of different embedding dimensions in TransR model.

Embedding dimensions	PRC
20	0.770
50	0.784
80	0.802
100	0.796
200	0.788

Fig. 12 shows the results of bond default prediction models with different embedding dimensions. As seen from the figure, when the dimension is low, the value of PRC increases with the increase in the dimension. When the dimension is 80, PRC reaches the highest value. After that, with the increase of dimension, PRC value has a downward trend. The reason for this phenomenon may be that the model cannot learn the representation of knowledge well when the dimension is low. However, when the dimension is too high, a certain amount of noise is introduced.

5) BOND DEFAULT PREDICTION BASED ON AN OPTIMIZED DeepFM MODEL

The optimized DeepFM model proposed in this paper is used for training and testing, and the final scores are sorted from high to low. The top 100 bonds were selected each time, the experiment was conducted five times, and Table 10 shows one of the test results, i.e., the ranking of actual default bonds in the top 100 bonds that are likely to default. As seen from the table, of the top 100 bonds that are expected to default, 8 bonds have substantial defaults.

Table 11 shows the number of default bonds in the top 100 possible default bonds in the five experiments.

By averaging the results of the five experiments, the number of bonds that actually default among the top 100 bonds that are likely to default is 7.8.

TABLE 10. Ranking of bonds that actually default in the top 100 possible default bonds.

Bond name	Rank
15 Le Shi 01	1
13 Bo Yuan MTN001	6
15 Chuan Mei Tan PPN001	16
13 Dan Dong Gang MTN1	17
15 Dan Dong Gang MTN001	35
17 Hu Hua Xin SCP002	62
15 Dan Dong Gang PPN001	76
15Dan Dong Gang PPN002	94

TABLE 11. Numbers of actual default bonds in the top 100 bonds.

Bond name	Occurrence number
15 Le Shi 01	5
15Chuan Mei TAN PPN001	5
13 Bo Yuan MTN001	5
13 Dan Dong Gang MTN1	5
17 Hu Hua Xin SCP005	2
17 Hu Hua Xin SCP004	2
15 Dan Dong Gang MTN001	5
15 Dan Dong Gang PPN001	3
15 Dan Dong Gang PPN002	3
17 Hu Hua Xin SCP003	2
17 Hu Hua Xin SCP002	2

To verify the effectiveness of the proposed method, the prediction results of optimized DeepFM are compared with those of the LR and Xgboost algorithms commonly used in default prediction. The top 100 bonds, the top 150 bonds and the top 200 bonds that are likely to default are selected to compare the predicted results. The experimental results are shown in Table 12 and the comparison figures are shown in Fig. 13.

TABLE 12. Comparisons between optimized DeepFM model and traditional methods.

Model	100	150	200
LR	4	6	11
Xgboost	7	9	11
DeepFM-KG	7.8	10	12

As seen from the figure, among the Top100, Top150 and Top200 bonds that may default, the optimized DeepFM model has the highest hit rate. Generally, the predicted results of the optimized DeepFM model are similar to those of Xgboost. The reason is that the number of samples is too small for a deep learning model. However, as an integrated learning model, Xgboost can learn features better when the number of samples is small. It is believed that the actual prediction performance of optimized DeepFM will be significantly reflected when there are relatively many samples. LR has the worst predictive performance in these three models, and it is relatively dependent on feature engineering.

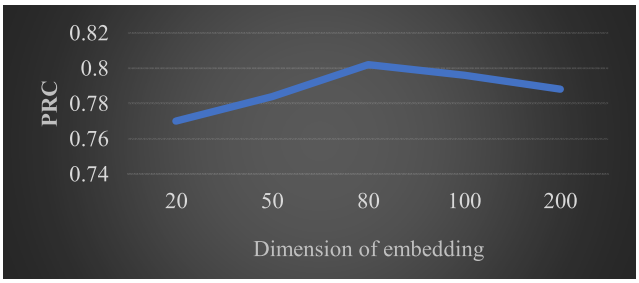


FIGURE 12. Prediction results of different embedding dimensions in the TransR model.

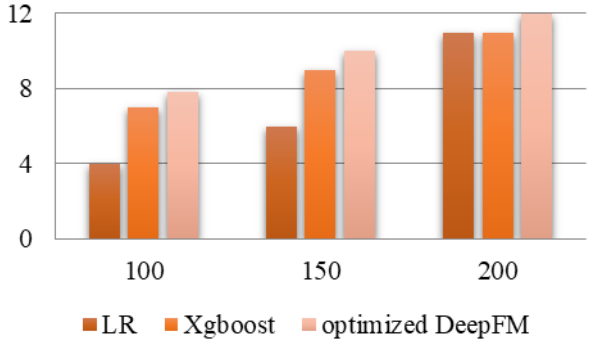


FIGURE 13. Actual number of defaulted bonds in the top 100, 150 and 200 bonds.

VI. CONCLUSION AND FUTURE STUDIES

According to the characteristics of publicly available bond data, this paper proposes a deep learning model that integrates the semantic information of a knowledge graph and applies it to bond default prediction. Using the knowledge representation learning model, we embed the knowledge atlas of discrete symbolized representation into the vector space to obtain the knowledge representation of bonds. The bond knowledge representation used to construct the knowledge graph and train the knowledge representation model effectively utilizes the bond relationship data and excavates the implicit relationship between bonds. A deep learning model is used to automatically learn higher-order features, and the bond knowledge graph is used as prior knowledge to expand higher-order cross-features.

The experimental results show that the deep learning model improves the prediction accuracy greatly compared with the traditional machine learning model. At the same time, the optimized DeepFM model that fuses the semantic information of the knowledge graphs outperforms the original DeepFM model that does not include knowledge information in the prediction task, which proves the feasibility and validity of the deep learning of knowledge maps fusion.

In future research, more knowledge information should be added to the bond knowledge graphs. At present, the data obtained are not sufficient. Adding more knowledge can help researchers obtain more comprehensive bond semantic information, which is conducive to knowledge representation

learning. In addition, knowledge representation learning is developing rapidly, and new models are proposed every year. In the future, other knowledge representation learning models can be tested to find the most appropriate method.

REFERENCES

[1] Altman El Financial Ratios, "Discriminant analysis and the prediction of corporate bankruptcy," *J. Finance*, vol. 23, no. 4, pp. 589–609, 1968.

[2] J. A. Ohlson, "Financial ratios and the probabilistic prediction of bankruptcy," *J. Accounting Res.*, vol. 18, no. 1, pp. 109–131, Apr. 1980.

[3] S. T. Bharath and T. Shumway, "Forecasting default with the KMV-Merton model," *Social Sci. Electron. Publishing*, 2004.

[4] J. Morgan, "Creditmetrics-technical document," JP Morgan, New York, NY, USA, Tech. Rep., 1997, no. 1, pp. 102–127.

[5] Y. Shiwei and L. Jincheng, "Credit risk measurement, bond default prediction and structured model extension," *Securities Market Herald*, no. 10, pp. 41–48, 2015.

[6] H. Xiaorong, "Analysis and measurement of bond default risk of listed companies in China," Northeast Finance Econ. Univ., Dalian, China, Tech. Rep., 2016.

[7] W. Guojian, "Measurement and empirical study on default risk of credit bonds based on KMV-LOGIT hybrid model," China Univ. Sci. Technol., Taipei, Taiwan, Tech. Rep., 2018.

[8] H. Die, "Bond default analysis based on stochastic forests," *Contemp. Economy*, no. 3, pp. 28–30, 2018.

[9] S. Dutta and S. Shekhar, "Bond rating: A nonconservative application of neural networks," in *Proc. IEEE Int. Conf. Neural Netw.*, Dec. 1988, pp. 443–450.

[10] Y.-C. Lee, "Application of support vector machines to corporate credit rating prediction," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 67–74, Jul. 2007.

[11] H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, and R. Anil, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[12] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data," in *Proc. Eur. Conf. Inf. Retr.*, 2016, pp. 45–57.

[13] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1149–1154.

[14] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," 2017, *arXiv:1703.04247*. [Online]. Available: http://arxiv.org/abs/1703.04247

[15] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 353–362.

[16] F. Gong, M. Wang, H. Wang, S. Wang, and M. Liu, "SMR: Medical knowledge graph embedding for safe medicine recommendation," 2017, *arXiv:1710.05980*. [Online]. Available: https://arxiv.org/abs/1710.05980

[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.

[18] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," 2018, *arXiv:1801.08284*. [Online]. Available: http://arxiv.org/abs/1801.08284

[19] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[20] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1112–1119.

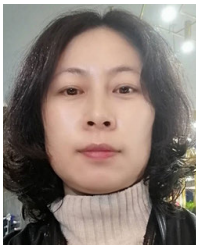
[21] C. Moon, P. Jones, and N. F. Samatova, "Learning entity type embeddings for knowledge graph completion," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 2181–2187.

[22] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 995–1000.

[23] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 985–991.

[24] T. Saito and M. Rehmsmeier, "Precrec: Fast and accurate precision–recall and ROC curve calculations in R," *Bioinformatics*, vol. 33, no. 1, pp. 145–147, Jan. 2017.

MA CHI received the Ph.D. degree from the Dongling School of Economics and Management, University of Science and Technology, Beijing. Since 2010, he has been working as an Associate Professor with the School of Software Engineering, University of Science and Technology Liaoning. Since 2019, he has been working with the School of Computer Science and Engineering, Huizhou University. His research interests include pattern recognition and data mining.



SUN HONGYAN received the M.S. degree from the University of Science and Technology Liaoning. She worked as a Lecturer with the School of Computer Science and Software Engineering, University of Science and Technology Liaoning. Her research interests include pattern recognition and data mining.

WANG SHAOFAN received the degree from Zaozhuang University. He is currently pursuing the M.S. degree in computer science and technology with the School of Computer Science and Software Engineering, University of Science and Technology Liaoning. His research interests include deep learning and data mining.

LU SHENGLIANG received the degree from the Dalian Neusoft University of Information. He is currently pursuing the M.S. degree in computer science and technology with the School of Computer Science and Software Engineering, University of Science and Technology Liaoning. He worked with Dalian Modern High-Tech Company Ltd. His research interests include deep learning and data mining.

LI JINGYAN is currently working with the College of Computer Science and Technology, Huizhou University. She has in-depth research in computer science and has achieved certain results. Her research interests include pattern recognition and data mining.