2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)

2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security
(IDS) | 978-1-6654-8069-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/BIGDATASECURITYHPSCIDS54978.2022.00031

TANN: Text-based Attention Neural Network Recommendation Model

Gang Qiu[1,2] , Yanli Guo[2] , Changjun Song[2] (✉)
[1] School of Software, Shandong University, Jinan, China
[2] Changji University, Changji, China
coral1001@163.com, gyl@cjc.edu.cn, 1484783865@qq.com

***Abstract*—The recommendation system can help users automatically select items of interest from a large amount of information. The use of artificial intelligence technology to find out the willingness to express emotions in the text has received widespread attention, but data sparseness and cold start problems have seriously affected the quality of recommendations. Based on this, we propose a *Text-based Attention Neural Network* (TANN) recommendation model, which is based on text perception. First, use word2vec to obtain the word vector representation of the user comment text. Then, the combination of LSTM and attention mechanism is used to capture the contextual information in the text, and finally the emotional factors are classified and predicted through the CNN layer. Compared with the advanced baseline method, the TANN recommendation model is better than other baselines and can effectively solve the user's data sparseness and cold start problem.**

***Index Terms*—Context; Deep learning; emotion analysis; Recommender System; Neural Networks**

I. Introduction

The recommendation system has become one of the important intelligent software to help users make decisions. The *Collaborative Filtering* (CF) recommendation algorithm uses the user's historical rating of items and the interaction or preference between the user and the item to generate a recommendation list [1]. However, when the user interacts very sparsely with the project, the CF method is often affected by limited performance, which is very common for large data collection scenarios [2] [3] [4] such as online shopping, news recommendation, and auction platforms. In addition, because the CF method does not have historical data when a new project is received, it will cause a cold start problem. [5] [6].

The content-based recommendation can solve the cold start problem in recommendation by analyzing the emotional expression of text information [7] [8], is not subject to user sparseness, and can find hidden information in the item, and has a good user experience. However, In the content based on recommendation system, some texts describe less content and insufficient information, which will bring difficulties to semantic analysis.

In response to the above problems, this paper proposes a *Text-based Attention Neural Network* (TANN) Recommendation Framework based on text perception. The model first obtains the word vector information from the user comment text through the Word2Vec model, the LSTM layer is used to capture the context information in the text word vector sequence, and finally uses the CNN layer to extract the feature information, and perform classification and prediction. The main contributions of our work are as follows:

- We propose a text-aware attention neural network recommendation method to solve the problem of data sparsity and user cold start in the recommendation system.
- We use the Word2Vec model to obtain the word vector representation in the text, and embed the word vector representation as an emotion vector in the LSTM layer, and obtain the hidden representation of the word vector through LSTM, so that more emotional information in the text can be obtained.
- We test on real data sets. Experiments show that the text-perception-based attention mechanism neural network recommendation method proposed in this paper can more fully capture the emotional characteristics of texts of different lengths. Our method is better than other baselines and can effectively solve the Sparse of data and cold start problems.

The rest of this paper is shown below. The related work is represented in Section 2. Section 3 introduces data collection and preprocess. Section 4 details Attention neural network recommendation model based on text-aware. Section 5 discusses the results of the experiment, and Section 6 briefly reviews the conclusions of this paper and future work.

II. Related Work

With the development of computer systems [9] [10] [11], and networks [12] [13] [14], recommendation system becomes an emerging area. The traditional recommendation system is mainly divided into collaborative filtering recommender system, content-based recommender system and base on neural network recommender system.

The collaborative filtering recommendation system completes the recommendation by calculating the similarity between users and finding the user most similar to the target user. In addition, the item-based collaborative filtering recommendation method is to complete the recommendation through the similarity between the items and the user's rating of the item [15]. Researchers also use real-time personalized recommendations based on implicit user feedback data streams to convert the relationship between users and items into user prediction problems and sort the recommendation list [16] [17]. Although collaborative filtering recommendation uses

978-1-6654-8069-7/22/$31.00 ©2022 IEEE
DOI 10.1109/BigDataSecurityHPSCIDS54978.2022.00031

historical rating data to achieve effective recommendation [18] [19] [20]. However, the collaborative filtering recommendation method cannot recommend new users or items because there is no purchase record or scoring data for new users or items.

Content-based recommendation is based on the user's description of product reviews or product content, mining products similar to the content as recommendations, which belong to the product-to-product association method. User comment information can be used as the basis of collaborative filtering recommendation, incorporating content information into the factor model of hybrid recommendation [21] [22]. In the review-based recommendation method, some researchers incorporate the valuable review information generated by users into the recommendation system to achieve in-depth integration of user review recommendation factors [23] [24]. Content-based recommendation ignores project information and can cause cold start problems for new users.

Deep learning technology has been widely used in machine translation, natural language processing, image and image processing, text sentiment analysis and other fields, thanks to rising of Big data [25] [26] [27] and AI technologies [28] [29] [30]. Gui et al. [31] transformed the text emotion recognition problem into a QA reading problem, and modeled the text context information. Yu et al. [32] proposed a framework structure based on a hierarchical network, which considers the characteristics of text information from the position of words and semantic levels, and finds the interaction between sentences. Cai et al. [33] make full use of text and image information and carry out the recognition and fusion of multi-modal information, and test it in the Twitter multi-modal irony detection data set. The experimental results show that the full use of multi-modal information Can effectively improve the accuracy of sentiment analysis.

In summary, although the above methods can effectively extract the emotional features of the text, they did not preprocess the natural language description in the review text, and did not fully consider the correlation between the upper and lower emotional words. This paper proposes an attention neural network recommendation model based on text perception, which can effectively extract the relevant features of the context information in the review text, vectorize the text information, use LSTM to obtain the hidden features of the text sequence, and use the attention for the text sequence Hidden features are used for weight distribution, and finally CNN neural network is used for classification and prediction.

III. Data acquisition and preprocess

A. Data acquisition

We collect (2020.1.12-2020.12.1) real data sets from MSN New logs [1]. The statistical information of this data set is shown in Table 1.

B. Data preprocessing

In the data preprocessing stage, the user comment text is vectorized. This article uses the Word2Vec model to vectorize

[1]https://www.msn.com/en-us/news

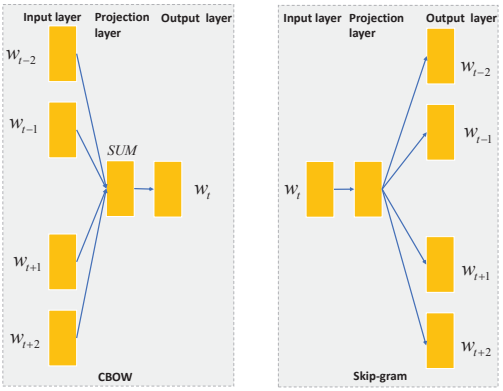

Fig. 1. Word2Vec two language models .

the text. Word2Vec is a word vector model proposed by Google in 2013. It can obtain the form of multi-dimensional vectors from a large number of unlabeled corpora, and calculate the semantic similarity of expressed text or words based on the similarity of words in the vector space. It has been widely used in language processing, such as sentiment analysis, topic extraction, and dictionary expansion. Word2Vec has two word vector training models, the distribution is Skip-gram and CBOW (Continuous bag of words), both of which are composed of input layer, projection layer and output layer. The difference is that Skip-gram predicts the probability of the context based on the current word vector, while CBOW uses the context to predict the probability of the current word. As shown in Figure 1.

In Skip-gram, surrounding words are generated based on the central word. Suppose there is a dictionary of index set $\{0, 1, \ldots, n\}$, a central word W_c and background word W_o, and one BB corresponds to the One-hot word vector as V_c, and W_o corresponds to the One-hot word vector as u_o. Therefore, given the central word in Skip-gram The probability of W_c generating background word W_o is:

$$P(W_o|W_c) = \frac{exp(u_o^T v_c)}{\sum_{i=0}^{n} exp(u_i^T v_c)} \tag{1}$$

where i,o,c denotes the position of the index in the dictionary.

Suppose, there exists a text sequence of length T. Then the probability of generating $W^{(t)}$ background words given the central word $W^{(t+j)}$ is:

$$\prod_{t=1}^{T} \prod_{-m \le j \le m, j \ne 0} P(W^{(i+j)|w^{(t)}}) \tag{2}$$

Where, t is the time step and m is the window size for generating background words. The minimization loss function can be obtained in equation (2) as:

$$\sum_{t=1}^{T} \sum_{-m \le j \le m, j \ne 0} \log P(W^{(i+j)}|w^{(t)}) \tag{3}$$

In the training process,, stochastic gradient descent is used to take a random subsequence to train the parameters of the updated model to obtain the gradient of the central word vector V_c:

$$\begin{aligned} &\log P(W_o|W_c) = u_o^T v_c - \log(\sum_{i=0}^{n} exp(u_i^T v_c)) \\ \Longrightarrow \quad &\frac{\partial \log P(W_o|W_c)}{\partial v_c} = u_o - \frac{\sum_{j=0}^{n} exp(u_j^T v_c) u_j}{\sum_{i=0}^{n} exp(u_i^T v_c)} \\ \Longrightarrow \quad &\frac{\partial \log P(W_o|W_c)}{\partial v_c} = u_o - \sum_{j=0}^{n} P(w_j|w_c) u_j \end{aligned} \tag{4}$$

When the training is finished Word2vec word vectors can be obtained. In addition, we can determine the size of the embedding dimension based on the word vectors in the review text.

In the CBOW model, assume that there exists a lexicon with an indexed set $\{0, 1, \ldots, n\}$ and a central word W_c, and a background word set $\{W_{O_1}, W_{O_2}, \ldots, W_{O_m}\}$. The One-hot word vector corresponding to W_c is u_c, and the set of background One-hot word vectors corresponding to $\{W_{O_1}, W_{O_2}, \ldots, W_{O_m}\}$ is $\{V_{O_1}, V_{O_2}, \ldots, V_{O_m}\}$. Therefore, in CBOW, the probability of generating a central word W_c given the set of background words $\{W_{O_1}, W_{O_2}, \ldots, W_{O_m}\}$ is:

$$P(W_c|W_{o_1}, W_{o_2}, \ldots, W_{o_m}) = \frac{exp(\frac{1}{m} u_i^T (v_{o_1}, v_{o_2}, \ldots, v_{o_m}))}{\sum_{i=0}^{n} exp(\frac{1}{m} u_i^T (v_{o_1}, v_{o_2}, \ldots, v_{o_m}))} \tag{5}$$

We will $W_o = \{W_{o_1}, W_{o_2}, \ldots, W_{o_m}\}$,$\overline{v}_o = \frac{(v_{o_1} + v_{o_2} + \ldots + v_{o_m})}{m}$. so formula 5 can be rewritten as:

$$P(W_c|W_o) = \frac{exp(u_c^T \overline{v}_o)}{\sum_{i=0}^{n} exp(u_i^T \overline{v}_o)} \tag{6}$$

Suppose there exists a text sequence of length T. Then given the background word $\{W^{(t-m)}, \ldots, W^{(t-2)}, W^{(t-1)}, W^{(t+1)}, W^{(t+2)}, \ldots, W^{(t+m)}\}$, the probability of generating the central word W^t is:

$$\prod_{t=1}^{T} P(W^{(t-m)}, \ldots, W^{(t-2)}, W^{(t-1)}, W^{(t+1)}, W^{(t+2)}, \ldots, W^{(t+m)}) \tag{7}$$

Where, t is the time step, and m is the window size for generating background words. The minimized loss function can be obtained in formula 7:

$$\begin{aligned} &\sum_{t=1}^{T} \log \Big(W^{(t)} | W^{(t-m)}, \ldots, W^{(t-2)}, \\ &W^{(t-1)}, W^{(t+1)}, W^{(t+2)}, \ldots, W^{(t+m)} \Big) \end{aligned} \tag{8}$$

In the training process, stochastic gradient descent is used to train the parameters of the updated model to obtain the gradient of any background word vector $V_{o_k} (k = 1, 2, \ldots, m)$:

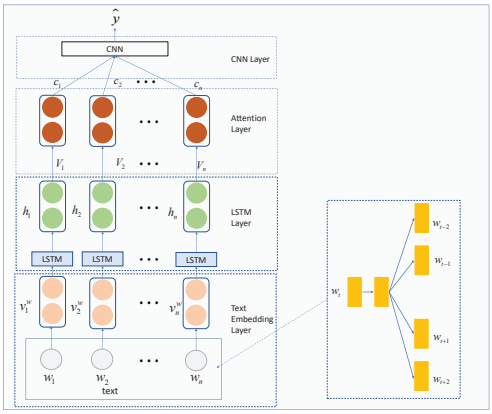

Fig. 2. Attention neural network recommendation model based on text-aware

$$\begin{aligned} \frac{\partial \log P(W_o|W_c)}{\partial v_{o_k}} &= \frac{1}{m} \left(u_c - \sum_{j=0}^{n} \frac{exp(u_j^T \overline{v}_o) u_j}{\sum_{i=0}^{n} exp(u_i^T \overline{v}_o)} \right) \\ &= \frac{1}{m} (u_c - \sum_{j=0}^{n} P(W_j|W_o u_j) \end{aligned} \tag{9}$$

When the training is finished Word2vec word vectors can be obtained. In addition, we can determine the size of the embedding dimension based on the word vectors in the review text.

In the actual model training process, Skip-gram takes more time than CBOW, but the accuracy of the result is better than the latter. Therefore, this article uses the Skip-gram model to train word vectors. The specific training parameters are shown in Table 1:

TABLE I
WORD2VEC TRAINING PARAMETER SETTINGS

Main parameters	Parameter value
Algorithm(sg)	Skip-gram
Word vector dimension(size)	500
Number of context windows(window)	6
Learning rate(alpha)	0.02
Number of iterations(iter)	6
Sampling threshold(sample)	0.001

After the text is vectorized by Word2vec, the word vector W_t is obtained, and the word vector W_t is input into the LSTM model.

sectionAttention neural network recommendation model based on text-aware

In this section, we propose a *Text-based Attention Neural Network* (TAAN) recommendation model. in Figure 2, the attention neural network recommendation model based on text perception includes text embedding layer, LSTM layer, attention layer and CNN layer [34]. First, we input the word vector representation into the text embedding layer for encoding. Then, the LSTM layer is used to capture the contextual

information of the text. Finally, the CNN layer classifies and predicts emotional factors based on the output mapping matrix.

C. Text Embedding Layer

The text embedding layer uses the Word2Vec model to obtain a weighted word vector representing w_i from the natural language description of user comments. The word vector with weight is expressed as $V^W = \{v_1^w, v_2^w, \ldots, v_n^w\}$. Then the word vector is represented as the embedding vector v_i^w.

D. LSTM layer

Due to the correlation between word vectors in the process of text information processing, and the RNN network is prone to problems such as gradient descent and gradient explosion when processing long sequences. Therefore, we introduced a long short-term memory network (LSTM). LSTM is a special kind of RNN, which consists of three types of gates, namely input gate, forget gate and output gate. Among them, the input gate conditionally determines the state value of the input terminal to update the memory, the forget gate conditionally discards some information of the unit, and the output gate conditionally outputs the vector. LSTM can not only effectively capture the time characteristics of training data, but also solve the problem of gradient disappearance and gradient explosion in the training process of long sequences. Based on this, we input the word vector matrix W_t of the text embedding layer into the LSTM:

$$h_t = LSTM(W_t, h_{t-1}) \tag{10}$$

Where h_t represents the output of the current cell, and h_{t-1} represents the output of the cell at the previous moment. Therefore, the hidden state H_i output by LSTM is$H_i = h_1, h_2, \ldots, h_{T-1}, h_T$. We convert the output hidden state H into a vector V_t:

$$V_t = W_1 * H + b \tag{11}$$

Where W_1 is the trainable parameter vector, and b is the deviation. We embed the word vector into the LSTM layer to obtain the hidden state representation of the text information. Then input the obtained vector V_t to the attention layer.

E. Attention layer

The attention mechanism is similar to human visual attention. We can effectively obtain the distribution weights of vector information through the attention mechanism, thereby improving the accuracy in sequence learning tasks. The word vector is processed by the LSTM layer to obtain the hidden representation sequence of the text information, and then the hidden representation sequence of the text information is input to the attention layer. In Figure 3, we learn the important vector representation h_i from the hidden representation sequence of the text information. And these feature information are given different weights, the size of the weight value determines the importance of the feature. Specifically, the vector V of normalized weight values is α_n:

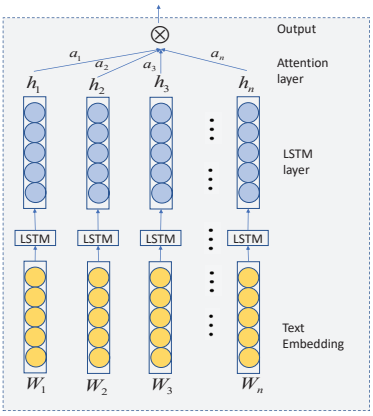

Fig. 3. LSTM with Attention

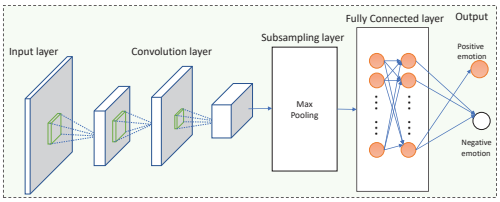

Fig. 4. CNN framework for sentiment analysis.

$$\alpha_n = softmax(W_n * (tanh(V))^T) \tag{12}$$

Where W_n is a trainable parameter, and α_n is a vector of length T. α_{nt} represents the weight of the hidden state output by the LSTM:

$$\alpha_{nt} = \frac{exp(W_n \cdot h_t)}{\sum_{i=1}^{N}(W_n \cdot h_t)}, t \in \{1, 2, \ldots, n\} \tag{13}$$

The sum of all α_{nt} weights is 1. Apply the weight value to the hidden state of the LSTM output, the output vector c_n is:

$$c_n = \sum_{t=1}^{T} \alpha_{nt} \cdot h_t \tag{14}$$

F. CNN layer

The CNN layer is used to extract text feature classification. It is composed of a convolutional layer, a maximum pooling layer, and a fully connected layer, as shown in Figure 4. We denote the CNN layer input as:

$$\hat{y}(X) = f(X, \upsilon) \tag{15}$$

Where X is the input vector of the neural network, $\hat{y}$ is the predicted output of the neural network, and υ represents the neural network parameters, that is, the set of weights and deviations of all neurons in the network.

IV. Experiments

In this section, we evaluate the performance of the proposed TANN model in sentiment analysis.

A. Data Set

We collect (2020.1.12-2020.12.1) real data sets from MSN New logs. The statistical information of this data set is shown in Table 1. In addition, we randomly selected 10 % as the validation set of the training data.

TABLE II
Data Sets Statistics

Users	12000	avg. words per title	12.23
news	43321	topic categories	14
impressions	451230	positive samples	496855
samples	714298	negative samples	6661254

In the experiment, our word embedding is 300-dimensional, CNN has 400 filters, and the window size is set to 3. Using Adam as the optimization algorithm, batch is 64. The indicators evaluated by our results include AUC, MRR, nDCG@10 and nDCG@20. The number of repetitions of the experiment is 10, and the results of the experiment are averaged.

B. Performance Evaluation

In the experiment, our TANN method is compared with multiple state-of-the-art baselines on the recommended performance. Including (1) DSSM [35], a recommendation model based on deep learning. DSSM mainly relies on a deep feed forward neural network to extract user attributes and potential vector features of items, and connect a one-hot vector and user attribute vector as the user The input feature vector. (2) DeepFM [36], a hit rate prediction model based on factorization machine (FM) and deep feed forward network. (3) Graph2R, an extended recommendation model of attribute-aware GraphRec [37]. GraphRec embeds the attributes of the user and the item into the input vector together, and is activated by the non-linear embedding layer and CRELU. (4) DKN [38] is a text-based recommendation model for deep knowledge perception. DKN combines news semantic level and knowledge level knowledge representation, and uses words and entities as multi-channel knowledge as input vectors. (5) wide & deep [39], a joint-trained generalized linear model and deep neural network recommendation model; (6) DFM [40] a deep matrix decomposition that uses multiple non-linear layers to process users and commodities Recommended model. (7) TANN-LSTM is TANN without attention perception; (8) TANN-LSTM&Attention is TANN with attention perception. The results are shown in Table 2.

The experimental results in Table 2 are shown. First, the text-based deep knowledge perception recommendation method is better than other methods. This is because DKN uses words and entities as multi-channel knowledge as input vectors, combined with text semantic-level knowledge representation, which is more conducive to acquiring rich semantic knowledge. Secondly, TANN-LSTM&Attention performs better in the method of attention network recommendation learning based on text perception. This is because we choose the more important vector information in the text message. The above experimental results show that our TANN is effective in the recommendation.

TABLE III
Performance comparison on the MSN New logs dataset

Methods	*AUC*	*MRR*	*NDCG@10*	*NDCG@20*
DSSM	0.5939	0.2677	0.2884	0.3790
wide & deep	0.5802	0.2543	0.2767	0.3664
DeepFM	0.5822	0.2573	0.2805	0.3697
DFM	0.5851	0.2605	0.2846	0.3732
DKN	0.6060	0.2803	0.3014	0.3823
Graph2R	0.5771	0.2471	0.2712	0.3550
TANN-LSTM	0.6042	0.2826	0.2856	0.3825
TANN-LSTM&Attention	0.6121	0.2940	0.2974	0.3913

C. Cold start performance analysis

In this section, we study the performance of TANN in cold start. In the recommendation system, some users may be new users or have never purchased items on the platform. Therefore, it is difficult for these users to effectively recommend items. Fortunately, we can comprehensively obtain user characterization information through various user behaviors such as user registration information, user search, or user browsing product information, which can effectively alleviate the cold start problem. As shown in Figure 5, we compared the performance of the TANN-Base, TANN-LSTM, TANN-LSTM&Attention methods with little or no user behavior. Among them, TANN-Base is the direct word vector obtained from the text embedding layer and enter it into the CNN network. TANN-LSTM is the word vector obtained from the text embedding layer and then the sequence representation of the word vector context is obtained through LSTM and then input into the CNN network. TANN-LSTM&Attention is the text-aware attention network method proposed in this paper. Experimental results show that our method also achieves good results under cold start conditions.

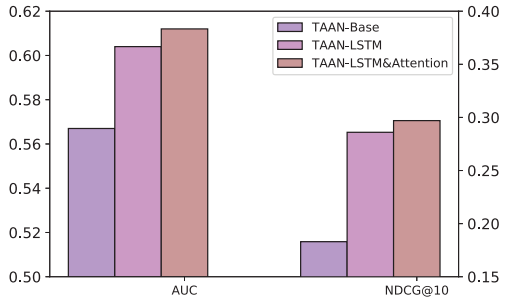

Fig. 5. Cold-start performance of our approach.

V. CONCLUSIONS

This paper proposed a neural network recommendation model based on text-aware attention mechanism. The model used preference analysis on the text information of user reviews, deeply mines the emotional words information in the text, and established a recommendation model based on the text-perceived attention mechanism. Specifically, TANN first obtained word vectors based on text context information through Word2Vec. Then through LSTM, the hidden features of the word vectors in the text were obtained, and the attention mechanism was used to capture the weight information in the text words. Finally, the representation of the word vectors was converted into advanced features and input into the CNN network, and the text information was classified through the CNN network predict.

ACKNOWLEDGEMENT

The work was supported by the 2021 Autonomous Region Innovation Environment (Talents, Bases) Construction Special-Natural Science Program (Natural Science Foundation) Joint Fund Project (2021D01C004), the 2019 Xinjiang Uygur Autonomous Region Higher Education Scientific Research Project (XJEDU2019Y057, XJEDU2019Y049).

REFERENCES

[1] J. B. Schafer, D. Frankowski, J. L. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web, Methods and Strategies of Web Personalization*, 2007, pp. 291–324.

[2] H. Qiu, Q. Zheng, and et al., "Topological graph convolutional network-based urban traffic flow and density prediction," *IEEE Trans. on ITS*, 2020.

[3] M. Qiu and et al., "Data allocation for hybrid memory with genetic algorithm," *IEEE Trans. on Emerging Topics in Computing*, 2015.

[4] M. Qiu, C. Xue, , and et al., "Efficient algorithm of energy minimization for heterogeneous wireless sensor network," in *IEEE EUC*, 2006, pp. 25–34.

[5] Z. H. Huang, J. W. Zhang, C. Q. Tian, S. L. Sun, and Y. Xiang, "Survey on learning-to-rank based recommendation algorithms," *Journal of Software*, vol. 27(03), pp. 691–713, 2016.

[6] A. Karatzoglou, L. Baltrunas, and Y. Shi, "Learning to rank for recommender systems," in *7th ACM Conf. RecSys*, 2013, pp. 493–494.

[7] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, 2013.

[8] W. Zhang and J. Wang, "A collective bayesian poisson factorization model for cold-start local event recommendation," in *21th Int'l Conf. SIGKDD*, 2015, pp. 1455–1464.

[9] M. Qiu, H. Li, and E. Sha, "Heterogeneous real-time embedded software optimization considering hardware platform," in *ACM sym. on Applied Comp.*, 2009, pp. 1637–1641.

[10] G. Wu, H. Zhang, M. Qiu, and et al., "A decentralized approach for mining event correlations in distributed system monitoring," *JPDC*, vol. 73, no. 3, pp. 330–340, 2013.

[11] M. Qiu, E. Sha, M. Liu, M. Lin, S. Hua, and L. Yang, "Energy minimization with loop fusion and multi-functional-unit scheduling for multidimensional DSP," *JPDC*, vol. 68, no. 4, pp. 443–455, 2008.

[12] M. Qiu, D. Cao, H. Su, and K. Gai, "Data transfer minimization for financial derivative pricing using monte carlo simulation with GPU in 5G," *Int'l J. of Comm. Sys.*, vol. 29, no. 16, pp. 2364–2374, 2016.

[13] Z. Lu, N. Wang, J. Wu, and M. Qiu, "IoTDeM: An IoT Big Data-oriented MapReduce performance prediction extended model in multiple edge clouds," *JPDC*, vol. 118, pp. 316–327, 2018.

[14] M. Qiu, J. Liu, J. Li, and et al., "A novel energy-aware fault tolerance mechanism for wireless sensor networks," in *IEEE Conf. GCC*, 2011.

[15] G. Linden, B. Smith, and J. York, "Industry report: Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Distributed Systems Online*, vol. 4, no. 1, 2003.

[16] K. Choi and Y. Suh, "A new similarity function for selecting neighbors for each target item in collaborative filtering," *Knowl.-Based Syst.*, vol. 37, pp. 146–153, 2013.

[17] X. Wu, B. Cheng, and J. Chen, "Collaborative filtering service recommendation based on a novel similarity computation method," *IEEE Trans. Serv. Com.*, vol. 10, no. 3, pp. 352–365, 2017.

[18] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *8th IEEE Int'l Conf. (ICDM*, 2008, pp. 263–272.

[19] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *14th Int'l Conf. SIGKDD*, 2008, pp. 426–434.

[20] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[21] E. Shmueli, A. Kagian, Y. Koren, and R. Lempel, "Care to comment?: recommendations for commenting on news stories," in *21st Conf. WWW*, 2012, pp. 429–438.

[22] Y. Zhang, Y. Tan, M. Zhang, Y. Liu, T. Chua, and S. Ma, "Catch the black sheep: Unified framework for shilling attack detection based on fraudulent action propagation," in *24th Int'l Conf. IJCAI*, 2015, pp. 2408–2414.

[23] H. Wang, N. Wang, and D. Yeung, "Collaborative deep learning for recommender systems," in *21th Int'l Conf. SIGKDD*, 2015, pp. 1235–1244.

[24] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *9th ACM Int'l Conf. WSDM*, 2016, pp. 153–162.

[25] L. Qiu, K. Gai, and M. Qiu, "Optimal big data sharing approach for tele-health in cloud computing," in *IEEE SmartCloud*, 2016, pp. 184–189.

[26] M. Qiu, K. Gai, and Z. Xiong, "Privacy-preserving wireless communications using bipartite matching in social big data," *FGCS*, vol. 87, pp. 772–781, 2018.

[27] J. Wang, M. Qiu, and B. Guo, "Enabling real-time information service on telehealth system over cloud-based big data platform," *Journal of Systems Architecture*, vol. 72, pp. 69–79, 2017.

[28] Y. Li, Y. Song, L. Jia, and et al., "Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning," *IEEE Trans. on Industrial Informatics*, vol. 17, no. 4, pp. 2833–2841, 2020.

[29] M. Liu, S. Zhang, and et al., "H infinite state estimation for discrete-time chaotic systems based on a unified model," *IEEE Trans. on Systems, Man, and Cybernetics (B)*, 2012.

[30] H. Qiu, M. Qiu, and Z. Lu, "Selective encryption on ecg data in body sensor network based on supervised machine learning," *Information Fusion*, vol. 55, pp. 59–67, 2020.

[31] L. Gui, J. Hu, Y. He, R. Xu, Q. Lu, and J. Du, "A question answering approach to emotion cause extraction," *CoRR*, vol. abs/1708.05482, 2017.

[32] X. Yu, W. Rong, Z. Zhang, Y. Ouyang, and Z. Xiong, "Multiple level hierarchical network-based clause selection for emotion cause extraction," *IEEE Access*, vol. 7, pp. 9071–9079, 2019.

[33] Y. Cai, H. Cai, and X. Wan, "Multi-modal sarcasm detection in twitter with hierarchical fusion model," in *57th Conf. ACL, Volume 1*, 2019, pp. 2506–2515.

[34] G. Qiu, X. Yu, L. Jiang, and B. Ma, "Text-aware recommendation model based on multi-attention neural networks," in *Knowledge Science, Engineering and Management - 14th Int'l Conf., Volume 12815 LNAI*, 2021, pp. 590–603.

[35] P. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. P. Heck, "Learning deep structured semantic models for web search using click through data," in *22nd Conf. CIKM*. ACM, 2013, pp. 2333–2338.

[36] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *26th Conf. WWW*, 2017, pp. 173–182.

[37] A. Rashed, J. Grabocka, and L. Schmidt-Thieme, "Attribute-aware non-linear co-embeddings of graph features," in *13th Conf. RecSys*.

[38] J. Gao, X. Xin, J. Liu, R. Wang, J. Lu, B. Li, X. Fan, and P. Guo, "Fine-grained deep knowledge-aware network for news recommendation with self-attention," in *IEEE/WIC/ACM Conf. WI*, 2018, pp. 81–88.

[39] H. Cheng, L. Koc, J. Harmsen, T. Shaked, and et al., "Wide & deep learning for recommender systems," in *1st Workshop on Deep Learning for Recomm. Systems, @RecSys*. ACM, 2016, pp. 7–10.

[40] J. Lian, F. Zhang, X. Xie, and G. Sun, "Towards better representation learning for personalized news recommendation: a multi-channel deep fusion approach," in *27th Int'l Conf. IJCAI*, 2018, pp. 3805–3811.