

## Battlefield Situation Information Recommendation Based on Recall-Ranking

Chunhua Zhou\*, Jianjing Shen, Yuncheng Wang and Xiaofeng Guo

PLA Strategic Support Force Information Engineering University, Zhengzhou, 450000, China

\*Corresponding Author: Chunhua Zhou. Email: zchgjb@126.com

Received: 28 May 2020; Accepted: 02 July 2020

**Abstract:** With the rapid development of information technology, battlefield situation data presents the characteristics of “4V” such as Volume, Variety, Value and Velocity. While enhancing situational awareness, it also brings many challenges to battlefield situation information recommendation (BSIR), such as big data volume, high timeliness, implicit feedback and no negative feedback. Focusing on the challenges faced by BSIR, we propose a two-stage BSIR model based on deep neural network (DNN). The model utilizes DNN to extract the nonlinear relationship between the data features effectively, mine the potential content features, and then improves the accuracy of recommendation. These two stages are the recall stage and the ranking stage. In the recall stage, the candidate set of situation information is generated, and the massive situation information is reduced to a small candidate subset that is highly relevant to the commanders. In the ranking stage, the situation information in the candidate set is accurately scored and then sorted, and the situation information with a high score is eventually recommended to the commanders. Finally, we use the historical data from the situation management of real combat training information systems as the dataset and verify the effective-ness of the algorithm through experiments, and analyze the effects of the depth and width of the neural network on the performance of the algorithm.

**Keywords:** Battlefield situation; battlefield situation information recommendation; recommendation system; deep learning; DNN

### 1 Challenges Faced by BSIR

With the rapid development of information technology and its extensive application in the military field, battlefield situation data has presented the “4V” characteristics of big data, namely “Volume”, “Variety”, “Value” and “Velocity”. Moreover, due to the accuracy and timeliness sensitivity of military applications, the recommendation of battlefield situation information has also encountered great challenges, mainly in the following aspects:

1. Big data volume. Currently, most of the recommendation algorithms can determine the recommendation issue of small data volume, and they have good performance in both time complexity and space complexity, but they can do nothing for massive situation data, such as “dimension disaster” problem



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

- [1]. Therefore, highly specialized distributed learning algorithm and efficient service system are very necessary for processing the massive situation information data in BSIR.
2. **Timeliness.** Battlefield situation database is changing rapidly and dynamically, with many data uploads per second. The recommendation system must have adequate responsiveness to model lately uploaded data, so that it can recommend the information to the commanders who need it within the time-limit of situation information and maximize the value of the information. In addition, the recommendation system should be able to respond to the commanders' latest operations, recommend information to the commander when the commander needs it most, and support the commanders in decision-making.
  3. **Implicit feedback.** Unlike movie or book recommendations, items can be scored directly by users, or given explicit feedback. The commanders' feedback to the situation information can only indirectly reflect the commanders' opinion by observing the commander's operation behaviors, that are, implicit feedback. Commander's attention to different situation information can be reflected by commander's clicks on situation information, browsing time, whether to collect, print, label, subscribe and other information. For example, if a commander subscribes to the situation information of a particular target entity, this target entity may be the focus of the commander's focus.
  4. **No negative feedback.** Through observing the user's behavior, it can be inferred which items they may like, but it is difficult to infer which items the user does not like probably. Therefore, the commander's implicit feedback can only focus on collecting positive feedback data, which will greatly distort the users' complete behavioral preferences. Consequently, it is important to resolve "missing data" in addition to positive feedback, where most negative feedback will be produced. In BSIR, because of the data sparseness, there is a huge amount of "missing data" will be handled, which will severely raise the amount of data and make difficulty in the recommendation.

In order to solve the above problems, we propose a two-stage BSIR model based on DNN. The model uses DNN to learn the implicit features of commanders and situation information more precisely, and captures the non-linear correlation between commanders and situation information. Besides, via two DNNs to generate the candidate sets and rank the situation information in candidate set. During the sorting phase, the situation information in the candidate set is scored and correctly sorted to compose the best recommendation list. In the recall stage, the candidate set of situation information is generated, and the massive situation information is reduced to a small candidate subset that is highly relevant to the commanders. In the ranking stage, the situation information in the candidate set is accurately scored and then sorted, and the best recommendation list is generated.

## 2 Intelligent Recommendation Based on Deep Learning

The big data environment is encountered with more complex data features and various data information that is hard to recognize. Hence, the entire application of the DNN capability to learn the data features to build up a recommendation mechanism between the command and decision requirements and situation information re-sources. It can better guarantee decision-makers have access to the most matching and high valuable information based on their individual demands.

The development of deep learning and intelligent recommendation technology provides a good technical way to resolve the problem of information overload. Particularly in recent years, with the deep learning as the core of cognitive intelligence technology rapid rise and the study of deep learning explosive growth, deep learning has caused breakthroughs and contributed rich outcomes in speech recognition, image processing, natural language processing, and other departments [2–5]. International academia and industry have paid more and more attention to the use of deep learning in recommendation assignments. Since 2016, the international conference on recommendation systems (ACM RecSys) has regularly held

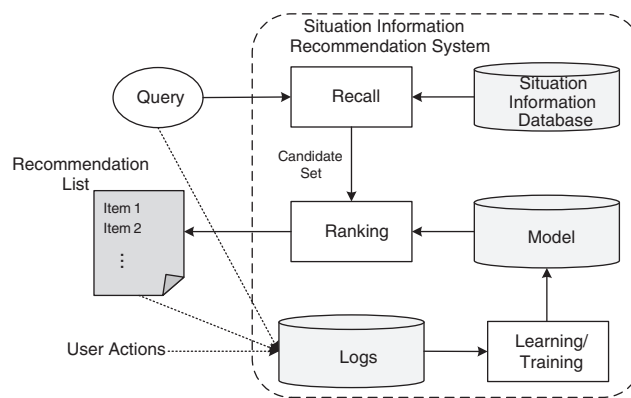
seminars on recommendation systems based on deep learning. Many universities and research institutions at home and abroad have also carried out extensive studies on recommendation systems based on deep learning [6–10].

Then, inserting commander data, situation data, and commander's historical records into the neural network, such as the commander's age, education background, level, arms, the label of situation information, clicks, and other data are inserted into the neural network. And deeply mine the complicated potential association between users and items and learn the behavior preferences of commanders. At the same time, suggest high-value situation information to the commander to efficiently support the commander in operational decision-making. Advance the effectiveness of command and control, to transform the information benefit into the decision-making benefit.

Deep learning has brought revolutionary changes to the architecture of recommendation system and brought more opportunities to improve the performance of recommendation system. Deep learning can play an important role in BSIR by combining low-level features with multi-layer network structure and nonlinear transformation to form low-dimensional, abstract and easily differentiated high-level representations to discover distributed feature representations of data. Applying deep learning to the intelligent analysis and processing of battlefield situation information data can effectively learn and represent the features of situation data through neural network. Input the commander data, situation information as well as the historical records into the neural network, such as the age, educational background, rank, data labels, click amount et al, to deeply mine the complicated potential association between users and items and learn the behavior preferences of commanders. Recommend the high-value information to commanders to efficiently support the commander in operational decision-making. Advance the effectiveness of command and control, to transform the information benefit into the decision-making benefit.

### 3 The Architecture of BSIR Based on Recall-Ranking

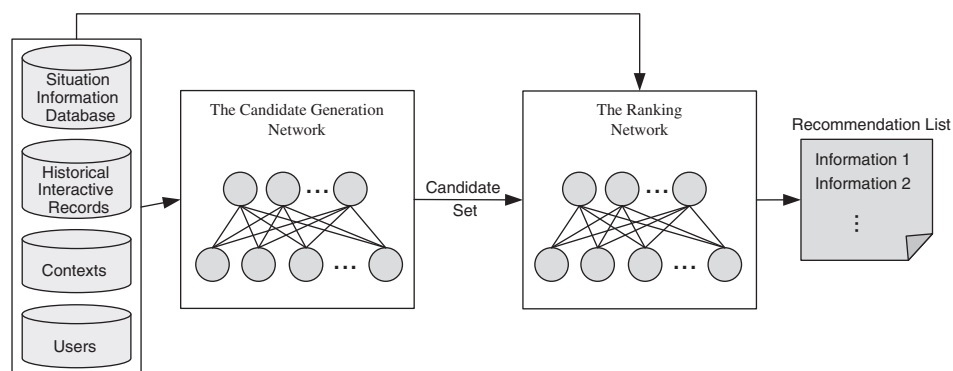
BSIR based on recall-ranking mainly adopts the recommendation idea of YouTube videos [11], which has been introduced into recommendation systems by BaiDu, Alibaba, Tencent and other major Internet companies. The whole process is divided into two stages: The recall stage and the ranking stage, as shown in Fig. 1 [12]. When a user accesses the BSIR system, a query is generated that can contain a variety of user and context information. The recommendation system returns a list of situation information (recommendation candidate set of situation information), and users can perform some operations on the information in the candidate set, such as favorite, click, and subscription. These user actions, along with queries and candidate sets, will be logged as training data for learning.



**Figure 1:** The BSIR system based on recall-ranking

Due to the increasing data scale of battlefield situation information, the data in the database has reached millions or even tens of millions. In addition, the combat re-quires high timeliness of information, so it is difficult to make a precise ranting for each piece of information before recommending it. Therefore, when a user accesses the recommendation system, the system will first query the matching situation information based on the combination of machine learning models and human-defined constraint association rules, and generate a candidate set of situation information (hundreds). Then, the ranking system will rank the situation information in the candidate set and recommend the top-k item with the highest score to the user.

In this paper, two DNNs are used to realize the recall and ranking of situation information. DNN in the recall stage is used to generate the candidate set of situation information, and DNN in the ranking stage is used to rank the situation information in the candidate set. The overall structure of the recommendation model is shown in Fig. 2. The candidate set generation network takes the historical interaction records of user-situation information as input and filters out the candidate subset with the highest matching degree from the original situation information database. Here, collaborative filtering algorithm is used to calculate the similarity. For example, the similarity between users can be matched by demographic information, such as age, rank, service and arms, and other demographic information. The best list of situation information to be recommended is generated by ranking the situation information in the candidate set according to its relevance to the current combat mission or to the commanders' interesting targets. Those with the highest relevance are deemed to be the most valuable, and the score will be the highest, which will be recommended priority to the commanders.



**Figure 2:** The overall architecture of recommendation model with two DNNs

### 3.1 The Recall Stage

During the recall phase, the massive situation information dataset is filtered into a subset of situation information that is strongly related to the commanders. In this paper, we model the BSIR problem as a “large-scale multiclass classification” problem. The large-scale classification problem is handled by *softmax* neural network, and calculated by the nonlinear factorization [13]. Stochastic Gradient Descent (SGD) and Two-pass Approximate Adaptive Sampling (TAPAS) method [14] are used to complete parameters learning of DNN and the approximate ranking of situation information.

This is inspired by Continues Bag of Word (CBOW) in Word2Vec [15]. Multi-classification problem is common in machine learning. In computer vision, natural language processing and recommendation system, many tasks are essentially multi-classification problem. In CBOW, a word is predicted based on the context of the word in the sentence. Similarly, multiple classifications can be used to model battle-field situation information recommendation problems, where context represents user features, such as demographic information and user historical behavior activities, and each label represents a situation information that a user might be interested in.

### 3.1.1 Problem Modelling

The multiclass classification problem can be defined as: Label the context features and discrete categories based on the empirical observation, for any  $x$ , learn to predict the corresponding category label  $y$ . For multiclass classification assignments, an especially effective solution is to model conditional probability using the *softmax* model. In the model,  $p(y|x)$  is proportional to  $\exp(\Phi(x) \cdot \psi(y))$ , that is,  $p(y|x) \propto \exp(\Phi(x) \cdot \psi(y))$ , where  $\Phi$  and  $\psi$  are the parameterized functions that map each context and label to a high-dimensional space, respectively called context embedding and label embedding. The parameters of  $\Phi$  and  $\psi$  are normally learned by minimize the cross entropy loss function in gradient descent method. Cross entropy loss function is defined as follows:

$$Loss = - \sum_{y=1}^M q_y \log(p_y) \quad (1)$$

where,  $M$  is the number of categories;  $q_y$  is the judgment variable (0 or 1). If the category is equal to the sample category, it is 1; otherwise, it is 0.  $p_y$  is the prediction probability that the observed sample belongs to category  $y$ .

Similarly, the BSIR problem can be modeled as a “large-scale multiclass classification” problem. That is, in the context  $C$ , the probability that any one of the situation information  $i$  in the situation information database  $D$  will be concerned by the commander  $u$  is predicted, which is expressed in the following formula:

$$P(s = i|U, C) = \frac{e^{v_i u}}{\sum_{j \in D} e^{v_j u}} \quad (2)$$

Obviously, the above formula is a *softmax* multi-classifier. Where,  $u \in R^N$  represents the high-dimensional embedded vector of user;  $v_j \in R^N$  represents an embedded vector of candidate situation information. An embedded vector is a simple mapping that maps sparse high-dimensional data (situation information, users, etc.) to a dense vector in  $R^N$ . Therefore, the task of the deep neural network is to learn the embedded vector  $u$  of the commander with the input of commander and context, and then use a *softmax* activation function as a classifier to classify the situation information  $i$ . To express DNN by formula is to fit the following function:

$$u = f\_DNN(user\_infor, context\_infor) \quad (3)$$

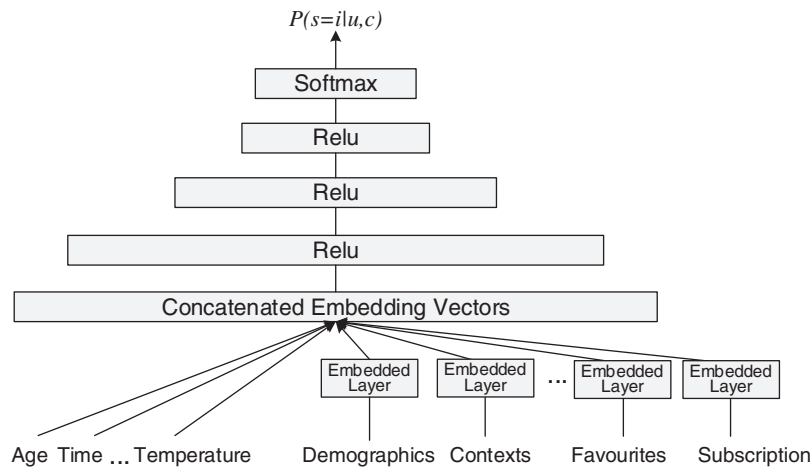
However, there is a difficulty in applying *softmax* multi-classifier. While we employ it for bigger and bigger assignments, the label vocabulary  $V$  may be extremely big. For example, the amount of situation information in the battlefield situation information database is around  $10^5$  to  $10^7$ , this is the so-called “extreme multiclass classification”. In the *softmax* model based on gradient training, every step of the training need to compute partition function, when  $V$  is very large, the cost of calculating the  $Z(x)$  can be very expensive. For extreme multiclass multi-classification problems, researchers have proposed a number of solutions, which can be roughly divided into two categories: The samples-based methods and the tree-based methods [16].

### 3.1.2 The Structure of Candidate Generation Network

The structure of candidate generation network is shown in Fig. 3.

The bottom of the network is the widest, and the number of cells in every successive hidden layer is halved. Embedding vectors are learned for continuous features and categorical features, such as the historical records (browse, retrieval, favorite, subscription), demographics, and other situation information. We concatenate all the embeddings together with the dense features. The concatenated

vector is then input into the network. The output layer of offline training is a *softmax* layer, and the output is the possibility shown by Eq. (2). Situation information, user information, scenario information, and historical interaction records are entered into the embedding layer based on the data type and are mapped into vectors of fixed dimensions. These embedding vectors are combined and linked as the first layer of DNN. Additionally, the embedding vector is reached via the normal gradient descent and back-propagation update and is acquired together with all other parameters of the model. In the situation information management system, users seldom have accurate explicit feedback, so we use implicit feedback information for deep neural network training. Situation information, user, context, and historical interaction records will be input into the embedded layer according to the data type, and are mapped to fixed dimensional vectors, and these embedded vectors will be concatenated as a wide embedding vector. And the embedded vector is updated through normal gradient descent back propagation, which is learned together with all other parameters of the model.



**Figure 3:** The structure of candidate generation network

In the training stage, in order to effectively train such a model containing millions, even tens of millions of classifications, we adopt the idea of CBOW and approximate processing technology to solve large-scale problems. Hierarchical *softmax* and negative sampling techniques are two very popular and effective approximation techniques in CBOW. However, Covington Weston et al. [11] argue that hierarchical *softmax* uses a binary tree hierarchy, and traversing each node in the tree involves differentiating between normally unrelated sets of classes, which makes the classification problem more difficult and reduces performance. Therefore, we use TAPAS, a two-pass approximate adaptive sampling method, to effectively train the *softmax* model with a large vocabulary. TAPAS adds an adaptive sampling step on the basis of negative sampling in CBOW to improve efficiency.

### 3.1.3 TAPAS

TAPAS is based on the idea of *softmax* loss function and gradient approximation calculation based on sampling. It divides the sampling into two steps: The first step is non-adaptive sampling, and the second step is adaptive sampling. In the first step, as in the negative sampling of COBW, a subset of the sample is sampled based on a predetermined distribution. In the second step, a smaller set  $S$  that “approaches” the context is re-sampled from the subset, that is, has a higher probability prediction in a given context.  $S$  is then used to calculate the gradient to update the model parameters. Compared with existing methods, TAPAS selects samples based on context and current model parameters. Also, resampling reduces the

sample size, so the gradient is measured more efficiently. The sampling algorithm is combined with the mini-batch SGD method to sample each batch.

#### *3.1.4 Feature Selection*

Using DNN for matrix factorization can easily add any continuous and categorical features to the model. For example, the commanders' behavior records of searching situation information can be processed similar to viewing behavior, and the historical query records can be converted into embedded vector and weighted average, which can reflect the commanders' overall search history state. Additionally, demographic features are very important for providing prior information, which can make the recommendations more reasonable for new users. Many recommendation systems use demographic information to solve the cold start problem of users. Gender, login status, age, and other simple categorical or continuous features are normalized to 0–1 and directly fed to the network.

We use behavioral information and demographic information to represent commanders. Behavioral information includes browsing, searching, collecting, subscribing, etc. Behavior features can be represented by the average after the word is embedded. Demographic information principally includes name, gender, age, ethnicity, education, rank, service type, service variety, unit and interests, etc. Besides, to perform the recommended situation information more targeted, combat tasks are input into the network as context, to advise more personalized situation information recommendation for commanders for specific combat tasks. Situation information is a description of entity attributes, states, and relationships. Different entities have different feature selections. For example, the attributes of mobile entities are number, name, category, speed, position, action, discovery time, etc. The attributes of static entities are number, name, category, volume, location, threat level, discovery time, etc. The specific feature selections can be based on the actuality and data sampling, and then conduct the feature screening.

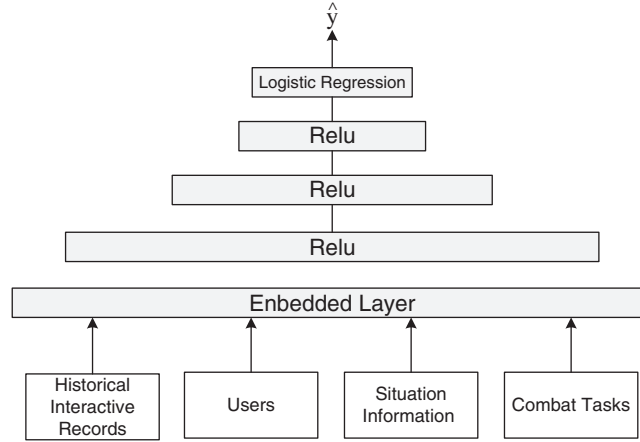
### *3.2 The Ranking Stage*

Equations in display format are separated from the paragraphs of the text. The main task of the ranking stage is to accurately estimate the commanders' preference for situational information. Unlike the recall stage, which faces the massive situation information set, the ranking stage only faces a very limited number of situation information subsets, so we can use more and more elaborate features to describe the situation information and the relationships between the commander and the situation information. For example, a commander may be very interested in a particular situation information, but may not be affected by the display format, so in the ranking stage, you need to use more feature information to describe the similarity between the situation information and the situation information concerned by the commander. In addition, ranking the situation information in the candidate set is also a reevaluation of the situation information, because in the candidate generation stage, there are often many sources of data that cannot be directly compared with each other, and in the ranking stage, data from different sources need to be effectively compared.

#### *3.2.1 The Structure of Ranking Network*

The structure of ranking network is similar to the structure of candidate generation network. The difference is that the last layer of the model uses logistic regression to assign an independent score to every situation information item, and then sorts the list of situation information items according to their scores. The structure of ranking network is shown in [Fig. 4](#). Commanders, situation information, contexts, etc. are fed into the network, and logistic regression models are employed to train the weights and generate the ranking scores. The ranking model still uses diverse features as input, which involves the commander's demographic information, resume and situation information, contexts, etc. This stage still requires feature selection, and the categorical extraction of potential features of multiple types of

input information, mapping to low-dimensional dense embedded vectors, input to the hidden layer of the neural network.



**Figure 4:** The structure of ranking network

In the process of model training, firstly, the embedded vector is randomly initialized, and then the value of the embedded vector is trained to minimize the final loss function. Each hidden layer is calculated as follows:

$$\alpha^{(l+1)} = f\left(W^{(l)}\alpha^{(l)} + b^{(l)}\right) \quad (4)$$

where,  $l$  is the number of layers, and  $f$  is the activation function. The ReLU activation function is adopted here.  $\alpha^{(l)}$ ,  $W^{(l)}$  and  $b^{(l)}$  are the input, model parameters, and offset of layer  $l$  respectively.

### 3.2.2 Feature Representation

In order to solve the problems of too many training parameters and too low computational efficiency caused by data sparseness and high feature dimension, DNN based recommendation systems at present usually used field-based feature representation, i.e., clustering data features into several fields, and then compressing each field into a low-dimensional, dense real-value embedding vector through a feature embedding layer [17,18]. In this way, even fields with arbitrary length and sparse discretization will become a dense and fixed-length continuous numerical vector after passing through the embedding layer. This paper uses field-based feature representation to extract potential content features of users and battlefield situation information. As shown in the model, this paper directly uses the embedding layer to transform the embedded features of the users' information, situation information, contexts and combat tasks etc.

### 3.2.3 Logistic Regression

Our purpose is to predict the possibility of the commander's attention situation information  $i$  according to provided training samples. Logistic regression was used for probability prediction. The logistic regression models are based on linear regression and use sigmoid function for nonlinear mapping. The prediction function of logistic regression is as follows:

$$\hat{y} = P(s = i|u, c) = \sigma(W^T x + b) \quad (5)$$



$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (6)$$

Then,

$$\hat{y} = \frac{1}{1 + e^{-(W^T x + b)}} \quad (7)$$

The model performs logistic regression training with a minimum cross entropy loss. The cross entropy loss function is:

$$L(W, b) = -\frac{1}{N} \sum_{i=1}^N \left( y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log (1 - \hat{y}^{(i)}) \right) \quad (8)$$

#### 4 Experiments

Accuracy is a very important index for evaluating the recommendation system. The purpose of BSIR is also to recommend the situation information that the commander needs, so the accuracy of the situation information recommendation is very important. We selected *Mean Absolute Error (MAE)* and *Root Mean Squared Error (RMSE)* as the evaluation indexes of our algorithm to evaluate the recommendations effectiveness.

The dataset is obtained from the situation management of real combat training information systems, 80% of which is used as training set, and 20% of which is used as the test set. The calculation formulas of *MAE* and *RMSE* are as follows:

$$MAE = \frac{1}{|R|} \sum_{r_{ui} \in R} |r_{ui} - \hat{r}_{ui}| \quad (9)$$

$$RMSE = \sqrt{\frac{1}{|R|} \sum_{r_{ui} \in R} (r_{ui} - \hat{r}_{ui})^2} \quad (10)$$

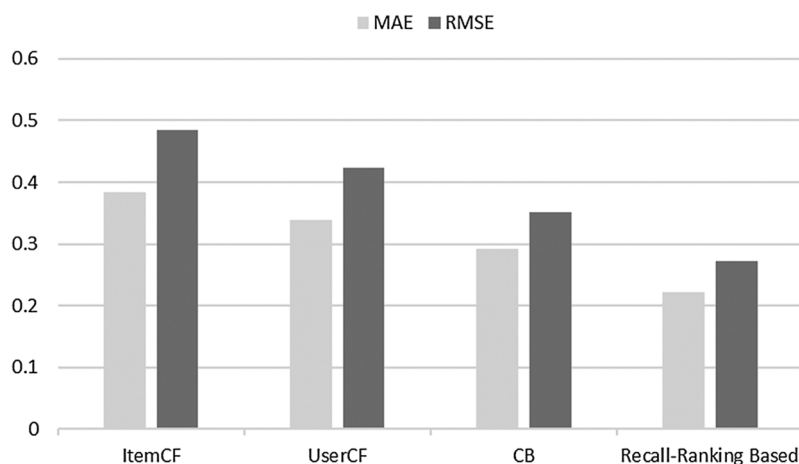
where,  $|R|$  is the number of situation information in the battlefield situation database.

Compare our model with the following three algorithms [19]: Content-based filtering algorithm (CB), user-based collaborative filtering algorithm (UserCF), item-based collaboration filtering algorithms (ItemCF). Experimental results of *MAE* and *RMSE* are shown in [Tab. 1](#) and [Fig. 5](#).

**Table 1:** Experimental results of *MAE* and *RMSE*

Methods	<i>MAE</i>	<i>RMSE</i>
ItemCF	0.3131	0.3652
UserCF	0.2980	0.3437
CB	0.2712	0.3010
Recall-Ranking Based	0.2011	0.2515

As can be seen from [Tab. 1](#) and [Fig. 5](#), our algorithm has smaller error and higher accuracy compared with the other three algorithms.



**Figure 5:** Comparison of *MAE* and *RMSE*

We also conducted experiments on the effect of DNN depth and width on algorithm performance, and the results are shown in [Tab. 2](#).

As can be seen from [Tab. 2](#), the experimental result of 1024 ReLU→512 ReLU→256 ReLU is the best, which indicates that increasing the depth and width of the hid-den layer can improve the accuracy of prediction. However, this comes at the cost of increasing the computation time of CPU. In practice, we can make a comprehensive balance of time and accuracy to meet our needs.

**Table 2:** The effect of depth and width of DNN on algorithm performance

Hidden Layers	<i>RMSE</i>
256ReLU	0.3421
512ReLU	0.3247
1024ReLU	0.3100
512 ReLU→256 ReLU	0.2915
1024 ReLU→512 ReLU	0.2601
1024 ReLU→512 ReLU→256 ReLU	0.2515

## 5 Conclusion

In order to solve the challenges faced by BSIR, this paper proposes a battlefield situation information recommendation model based on recalling-ranking, which is divided into two stages: recall phase and ranking phase. During the recall phase, the candidate set of situation information is generated, and the massive situation information dataset will be reduced to a subset of situation information that is strongly related to the commanders. The ranking phase performs the scoring and sorting of situation information in the candidate set, and the top-k items with the highest scores are finally recommended to users. The two stages are implemented with DNNs, and the network architecture is described in detail. DNN can effectively extract the non-linear relationship between data features, mine potential content features, and improve the accuracy of prediction. We transform the categorical and the continuous features respectively, and map the high-dimensional and sparse data feature to the low-dimensional and dense embedded vector space, so as to improve the calculation speed and avoid the dimension disaster. Finally, the effect of the

depth and width of the neural network on the performance of the algorithm is analyzed through experiments. The experiment finds that increasing the depth and width of the network can improve the accuracy of prediction, but at the cost of computation time.

**Acknowledgement:** We gratefully acknowledge the financial supports by the National Natural Science Fund of China under Grant number 61773399, as well as the National Social Science Fund of China under Grant numbers 14gj003-073.

**Funding Statement:** The National Natural Science Fund of China (No. 61773399) and the National Social Science Fund of China (No. 14gj003-073) supported our work with RMB 200,000 and RMB 100,000 respectively.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] V. Spruyt, "The curse of dimensionality in classification," *Computer Vision for Dummies*, vol. 21, no. 3, pp. 35–40, 2014.
- [2] J. W. Liu, Y. Liu and X. L. Luo, "Research and development on deep learning," *Application Research of Computers*, vol. 31, no. 7, pp. 1921–1942, 2014.
- [3] L. W. Huang, B. T. Jiang, S. Y. Lu, Y. B. Liu and D. Y. Li, "Survey on deep learning based recommender systems," *Chinese Journal of Computers*, vol. 41, no. 7, pp. 1619–1647, 2018.
- [4] H. Wu, Q. Liu and X. D. Liu, "A review on deep learning approaches to image classification and object segmentation," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 575–597, 2019.
- [5] J. Y. Yeh, "Rank-order-correlation-based feature vector context transformation for learning to rank for information retrieval," *Computer Systems Science and Engineering*, vol. 33, no. 1, pp. 41–52, 2018.
- [6] J. Yang and J. Li, "Application of deep convolution neural network," *2017 14th Int. Computer Conf. on Wavelet Active Media Technology and Information Processing*, Chengdu, pp. 229–232, 2017.
- [7] Y. G. Wang and G. Shang, "Deep collaborative filtering recommendation with attention mechanism," *Computer Engineering and Applications*, vol. 55, no. 13, pp. 8–14, 2019.
- [8] Z. W. Li, L. Chen and Z. X. Cao, "Research on collaborative filtering algorithm in intelligent recommendation system under big data environment," *Computer Programming Skills and Maintenance*, vol. 21, pp. 78–79+85, 2016.
- [9] K. K. Li, X. Z. Zhou, F. Lin, W. H. Zeng and G. Alterovitz, "Deep probabilistic matrix factorization framework for online collaborative filtering," *IEEE Access*, vol. 7, pp. 56117–56128, 2019.
- [10] Z. Batmaz, A. Yurekli, A. Bilge and C. Kaleli, "A review on deep learning for recommender systems: Challenges and remedies," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 1–37, 2019.
- [11] P. Covington, J. Adams and E. Sargin, "Deep neural networks for youtube recommendations," in *Proc. of the 10th ACM Conf. on Recommender Systems*, ACM, Boston Massachusetts USA, pp. 191–198, 2016.
- [12] H. T. Cheng, L. Koc and J. Harmsen, "Wide & deep learning for recommender systems," in *Proc. of the 1st Workshop on Deep Learning for Recommender Systems*, ACM, Boston MA USA, pp. 7–10, 2016.
- [13] J. Weston, S. Bengio and N. Usunier, "Wsabie: Scaling up to large vocabulary image annotation," in *Proc. of the Int. Joint Conf. on Artificial Intelligence*, IJCAI, Barcelona, Spain, 2011.
- [14] Y. Bai, S. Goldman and L. Zhang, "TAPAS: Two-pass approximate adaptive sampling for softmax," pp. 53–55, 2017, CoRR, abs/1707.03073.
- [15] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. of ICLR Workshop*, Scottsdale, Arizona, USA, 2013, arXiv: 1301.3781[cs.CL].
- [16] W. Chen, G. David and A. Michael, "Strategies for training large vocabulary neural language models," in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL, Berlin, Germany, Long Papers, 2016.

- [17] H. Guo, R. Tang, Y. Ye, Z. Li and X. He, “DeepFM: A factorization-machine based neural network for CTR prediction,” *Proc. of the 26th Int. Joint Conf. on Artificial Intelligence*, AAAI Press, Melbourne, Australia, pp. 1725–1731, 2017.
- [18] W. Zhang, T. Du and J. Wang, “Deep learning over multi-field categorical data,” in *European Conf. on Information Retrieval*. Springer, Padua, Italy, pp. 45–57, 2016.
- [19] L. Xiang, *Recommendation system practice*. Beijing, China: People’s Posts and Telecommunications Press, Barcelona, pp. 44–59, 2019.