TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 15/22 pp185–196
DOI: 10.26599/TST.2023.9010025
Volume 29, Number 1, February 2024

Time-Aware LSTM Neural Networks for Dynamic Personalized Recommendation on Business Intelligence

Xuan Yang and James A. Esquivel*

Abstract: Personalized recommendation plays a critical role in providing decision-making support for product and service analysis in the field of business intelligence. Recently, deep neural network-based sequential recommendation models gained considerable attention. However, existing approaches pay little attention to users' dynamically evolving interests, which are influenced by product attributes, especially product category. To overcome these challenges, we propose a dynamic personalized recommendation model: DynaPR. Specifically, we first embed product information and attribute information into a unified data space. Then, we exploit long short-term memory (LSTM) networks to characterize sequential behavior over multiple time periods and seize evolving interests by hierarchical LSTM networks. Finally, similarity values between users are measured through pairwise interest features, and personalized recommendation lists are generated. A series of experiments reveal the superiority of the proposed method compared with other advanced methods.

Key words: personalized recommendations; evolving interests; embedding; LSTM networks

1 Introduction

Business intelligence refers to technologies, systems, practices, and applications that are used to analyze critical business data, which provide business managers and analysts with the ability to manipulate and transform business data, such as developing strategies to optimize business data[1–3]. Nowadays, with the rapid development of Web 3.0, people are creating a substantial amount of content, such as weblogs and product reviews, in a variety of practical business applications[4].

The data generated by users in such applications contain numerous valuable and useful information that can reveal the intentions, needs, and interests of users[5–7]. Therefore, it has become necessary to model user data from these commercial applications by using mathematical statistical models and machine learning methods[8–10]. Recently, personalized recommendation methods based on machine learning have gained significant attention. These methods model the user's behavioral data and intelligently recommend services to reduce the decision burden of users in various commercial activities[11]. For example, the Amazon platform uses personalized intelligent recommendation algorithms to recommend the right products to consumers, which increases the economic revenue of the platform.

To date, two main popular paradigms are used for personalized recommendation methods based on business intelligence: matrix factorization-based recommendation methods[12–15] and sequential recommendation methods[16–19]. The matrix factorization-based recommendation approach mainly decomposes the user-product matrix into a user matrix and a product matrix, which together assist in predicting unpurchased products. However, given the uniformity

• Xuan Yang is with Graduate School, Angeles University Foundation, Angeles City 2009, Philippines, and also with Shandong Provincial University Laboratory for Protected Horticulture, Weifang University of Science and Technology, Weifang 262700, China. E-mail: yang. xuan@auf.edu.ph.
• James A. Esquivel is with Graduate School, Angeles University Foundation, Angeles City 2009, Philippines. E-mail: esquivel.james@auf.edu.ph.
∗ To whom correspondence should be addressed.
Manuscript received: 2023-02-22; revised: 2023-03-20; accepted: 2023-03-30

of matrix decomposition, these approaches fail to integrate multiple product attributes, such as product category information, which reduces the sensitivity of capturing user interest. Meanwhile, the sequential recommendation is devoted to modeling the sequence of user-product interactions over time sequences and recommending to users the next product moment[20, 21]. However, such approaches are only able to explore users' short-term interests, but not sufficiently explore the evolution of users' purchasing interests in long-term behaviors.

The purchasing interests of users are dynamic and evolving. As shown in Fig. 1, each user purchases a wide variety of products. The target user u_{tar} purchases *Great Expectations*, computers, guitars, etc., at time T. We cannot fully explore the user's interests due to the sparsity of the user's purchase history in a short period of time. At time $T+1, u_{tar}$ purchased *Anna Karenina*, food, flowers, and other products. Meanwhile, the categories of products *Great Expectations* and *Anna Karenina* are books, which may be a long-term reading behavior of the user. From the purchase order of multiple time periods, the user's long-term interest may be in books. Thus, the attribute information of the product, such as category, can not only reflect the real demand of the user, but also indicate the purchase interest of the user. Moreover, the purchase sequence over multiple time periods not only contributes to uncovering users' long- and short-term interests, but also helps to supplement the sparse purchase records.

Driven by the above observations, we propose a time-aware long short-term memory (LSTM)-based dynamic personalized recommendations for business intelligence, named DynaPR. For better modeling of users' purchase records, the DynaPR framework contains two key components that correlate to the fusing of attribute information and the fusion of time-aware information to better model the user's purchase history. First, to cope with the uniformity of product information and attribute information, we employ an embedding mechanism to fuse product features and attribute features that belong to different data types and then transform them into the same data space. Through linear transformation, the embedding mechanism possesses a powerful learning ability to map sparse matrices into continuous dense matrices. Secondly, to fuse time-aware sequence information and extract continuous relevance, we adopt LSTM[22] to mine users' purchase records in multiple time periods, where users' long-term interests are extracted. Finally, we employ scalar product to measure the similarity of any two users for predicting the sequence of products that users have not purchased.

The main contributions of the paper are as follows.

(1) For unification of product and attribute information, we propose the use of an embedding mechanism to convert user information of multiple data types into the same data space and obtain user representations.

(2) To extract the long- and short-term interests of users, we propose the employment of the LSTM mechanism to mine the sequences of products purchased by users across multiple time periods and seize evolving interests by hierarchical LSTM networks.

(3) A series of experiments are performed on a real dataset, and the experimental results show that DynaPR outperforms other state-of-the-art methods.

The rest of the paper is structured as follows: Section 2 is an introduction and summary of the approaches regarding personalized recommendation; Section 3 describes the problem definition and research goals of the proposed approach; Section 4 presents the proposed model DynaPR in detail; Section 5 validates the effectiveness of the DynaPR approach with a large number of experiments. Section 6 concludes the work.

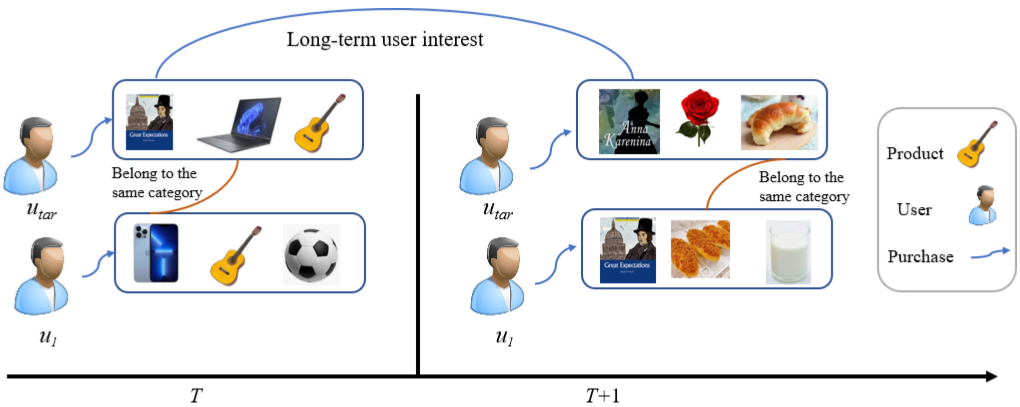

Fig. 1 Research motivation case.

2 Related Work

Personalized recommendations for business intelligence have attracted extensive attention. Here, we focus on matrix factorization-based and sequential recommendation methods.

2.1 Matrix factorization-based recommendation methods

Given their high scalability, matrix-factorization-based recommendation methods have gained widespread attention from scholars. The traditional matrix factorization-based approach first constructs the user-product rating matrix and decomposes it into user matrix and product matrix, which learn the user latent vector and product latent vector according to the objective function[12]. However, the decomposition of rating data cannot explicitly portray users' preferences, list-wise probabilistic matrix factorization (ListPMF) model was proposed by Liu et al.[13] ListPMF maximizes the predicted preference order with the actual preference order by log posterior, which yields an accurate list of recommendations. Alternatively, Salakhutdinov et al.[14] proposed the probabilistic matrix factorization (PMF) model that extends a finite set of unbalanced ratings linearly to obtain user preference features.

The objective function, as an essential part of matrix decomposition, plays a key role in the accuracy of recommendation results. Cui et al.[15] proposed a multi-objective optimal recommendation model to optimize multiple objectives, such as novelty and diversity. By restricting the regularization parameters, multi-objective evolutionary algorithm is empolyed to ensure the validity of the recommendation results. Further, considering the uncertainty and multilevel of recommendation information, Ye and Liu[23] developed a dynamic three-way recommendation model that uses a matrix decomposition framework, such as singular value decomposition (SVD), to construct the granularity structure of recommendation information based on user preferences and behavior.

Considering the data sparsity of user-product rating matrices, social relationships between users (e.g., trust relationships) are introduced into the matrix factorization-based approach. Xu et al.[24] decomposed user-user trust matrices by using matrix decomposition and combine user preferences with the trust strength between users to predict ratings. Although the matrix decomposition-based approaches are capable obtaining user preferences, the above approach ignores the influence of product attribute information on user long- and short-term interest.

2.2 Sequential recommendation methods

Sequential recommendation is a crucial task in recommender systems that analyzes the development of user behavior and discovers changes in user interests to foretell the user's subsequent behavior.

To predict users' behavior in the future, it is essential to consider both their long-term behavioral preferences and short-term, dynamic preferences in their behavior sequences. To address this challenge, He and MeAuley[17] proposed a mixed matrix decomposition and Markov chain model that can capture both of these preferences. Meanwhile, Xie et al.[18] utilized a personalized transition graph model, based on underlying Markov chains, to estimate the transition probabilities of users purchasing the same products at different moments. However, the matrix decomposition approach may struggle to fully incorporate long user behavior sequences, leading to potentially suboptimal performance. As a solution, Hidasi et al.[19] proposed an recurrent neural network (RNN)-based personalized sequence recommendation approach. They suggested modifying the RNN structure and utilizing a ranking loss function to enhance the model's ability to cope with specific tasks.

Based on the fact that product description information can also influence users' purchase interest, Tuan and Phuong[25] proposed a 3D convolutional neural network (CNN) based model that enhances the accuracy of recommendation results by combining product content features. With 3D architectures, spatio-temporal patterns of user behavior are captured. However, since CNN architectures allow only modelling to obtain local information about the whole sequence, while the correlation between sequences is ignored. You et al.[26] combined temporal convolutional network (TCN) with gate recurrent unit (GRU) to build recommendation models. As a result, the local information of the sequences can be captured by the CNN, while the overall properties of the sequence are maintained by the RNN.

Most existing sequence recommendation methods focus on feature engineering based on the whole sequence, but tend to overlook the potential interactions between attributes of the products within sequences. Therefore, modeling the interaction information between user behavior sequences remains a challenging task.

3 Problem Definition

As shown in business websites, users create multiple forms of purchase records, such as products ids, ratings of products, category of the product and so on. Here, we will model the behavioral records left by users.

- $U = \{u_1, u_2, \ldots, u_M\}$ denotes the set of users on the business website, where u_{tar} is one of the elements in U.
- $P = \{p_1, p_2, \ldots, p_N\}$ denotes the set of n products.

For each user in the business website, the purchase behavior of each user can be modeled. Particularly, the purchase history of u_{tar} across multiple time periods can be represented as follows:

- $P(t) = \{p^{tar}_{1,t}, p^{tar}_{2,t}, \ldots, p^{tar}_{Q,t}\}$ denotes the set of products purchased by u_{tar} during the t $(1 \leqslant t \leqslant T)$ time period.
- $S(t) = \{R_{tar}(t), C_{tar}(t)\}$ denotes product attribute information.
- $R_{tar}(t) = \{r^{tar}_{1,t}, r^{tar}_{2,t}, \ldots, r^{tar}_{Q,t}\}$ denotes the rating set of products purchased by u_{tar} during the t $(1 \leqslant t \leqslant T)$ time period.
- $C_{tar}(t) = \{c^{tar}_{1,t}, c^{tar}_{2,t}, \ldots, c^{tar}_{Q,t}\}$ denotes the category set of products purchased by u_{tar} during the t $(1 \leqslant t \leqslant T)$ time period.

Due to the presence of thousands of products on business websites, the purchase history of each user is dynamic and sparse. Hence, it is necessary to incorporate product attribute information to compensate for the effect of sparse data, while also exploring the multiple interests and preferences of users. However, it is a challenge to fuse multiple data types to form a user representation. We adopt two main mechanisms to address these challenges.

(1) Embedding mechanism[27]: Embedding mechanism can convert a sparse matrix into a dense and continuous matrix by linearity. Moreover, multiple data types can be transformed into a uniform data space.

(2) LSTM mechanism: LSTM mechanism with its inherent "gate" design can effectively extract the features and correlations of multiple time sequences. Such network structure helps to explore the long- and short-term interests of users.

With these two mechanisms, the challenges in user modeling are effectively solved. Here, the goal of the DynaPR model is to sketch the behavior sequence of the target user and to recommend products that have not been purchased.

4 DynaPR Model

For each user who leaves multiple time periods of purchase records on business websites, we utilize a deep learning approach to maximize the extraction of each user's feature representation. Figure 2 illustrates our proposed network model. Firstly, the embedding mechanism is utilized to preserve the original user purchase records of the products, while obtaining product embeddings and attributing embeddings. Secondly, the LSTM mechanism is utilized to extract

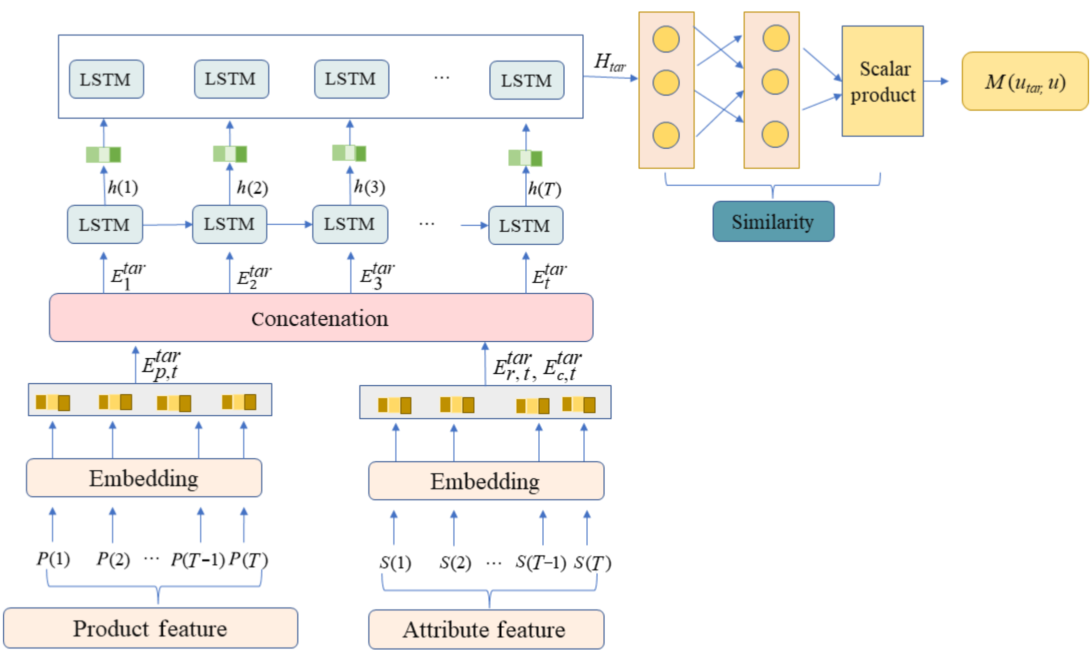

Fig. 2 DynaPR model.

the purchase records of users over multiple time periods, and allows the long-term purchase interest to be analyzed. Finally, similarity measures are used to generate recommendation results for the target users, where collaborative information of similar friends is considered.

4.1 Embedding representation

In business websites, there are a great variety of product attribute information, such as categories. With the process of user interaction with the platform, users will browse/purchase a large number of products, which generate a large number of behavioral records, such as ratings[28, 29]. On the one hand, the attribute information of the products can effectively reflect the usage of the products, which in turn reflects the potential interest of users. On the other hand, ratings reflect the quality evaluation of the product by users. The combination of these two types of information helps to seek the characteristics of users' interest in the purchased products. However, product and attribute information are different. It is necessary to find an integrated way to blend these two kinds of information. Notably, the embedding mechanism can not only effectively preserve the user's initial purchase behavior information, but also transform the information of different data types into the same data space. In particular, the embedding mechanism is applied in the following equations:

$$E_{p,\,t}^{tar} \leftarrow f(P_{tar}(t)) \tag{1}$$

$$E_{r,\,t}^{tar} \leftarrow f(R_{tar}(t)) \tag{2}$$

$$E_{c,\,t}^{tar} \leftarrow f(C_{tar}(t)) \tag{3}$$

In the above equations, $f(\cdot)$ denotes the embedding transformation function. $E_{p,\,t}^{tar} \in \mathbb{R}^{Q\times e}$, $E_{r,\,t}^{tar} \in \mathbb{R}^{Q\times e}$, $E_{c,\,t}^{tar} \in \mathbb{R}^{Q\times e}$ denote the product embedding matrix, the product rating embedding matrix, and the product category embedding matrix, respectively, which are the purchase behavior feature of u_{tar} at t time period.

$$E_t^{tar} \leftarrow E_{p,\,t}^{tar} \parallel E_{r,\,t}^{tar} \parallel E_{c,\,t}^{tar} \tag{4}$$

where $\parallel$ denotes concatenation operation. E_t^{tar} denotes the behavior embedding of u_{tar} at time period t. Thus, the product embedding is combined with attribute features to represent the linear transformation of the user's purchase behavior records.

Here, the embedding mechanism possesses the following two main capabilities. (1) Multiple types of product information can be converted into the same data space. Thus, a unified feature representation of the user can be achieved and fused. (2) The fused product features and attribute features provide a technical basis for analyzing the long-term interests of users.

4.2 Long short-term interest exaction

LSTM mechanism is the most widely applied framework for time series analysis, which is a special kind of RNN to learn long-term dependencies[30, 31]. It is mainly designed to solve the problem of gradient disappearance and gradient explosion during the training of long sequences. In brief, it means that LSTM can have better performance in long sequences compared to normal RNNs.Therefore, the LSTM mechanism is suitable for the research goal, which is to extract the long and short-term purchase interests of users by their purchase sequences over multiple time periods. For each LSTM block, the specific structure is shown as the following equations:

$$\tilde{\boldsymbol{c}}_t \leftarrow \varphi(E_t^{tar}\boldsymbol{W}_c + h_{t-1}\boldsymbol{U}_c + \boldsymbol{b}_c) \tag{5}$$

$$\boldsymbol{i}_t \leftarrow \sigma(E_t^{tar}\boldsymbol{W}_i + h_{t-1}\boldsymbol{U}_i + \boldsymbol{b}_i) \tag{6}$$

$$\boldsymbol{f}_t \leftarrow \sigma(E_t^{tar}\boldsymbol{W}_f + h_{t-1}\boldsymbol{U}_f + \boldsymbol{b}_f) \tag{7}$$

$$\boldsymbol{o}_t \leftarrow \sigma(E_t^{tar}\boldsymbol{W}_o + h_{t-1}\boldsymbol{U}_o + \boldsymbol{b}_o) \tag{8}$$

$$\boldsymbol{c}_t \leftarrow \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t \tag{9}$$

$$\boldsymbol{h}_t \leftarrow \boldsymbol{o}_t \odot \varphi(\boldsymbol{c}_t) \tag{10}$$

The parameters in Formulas (5)–(10) represent the flow of data in LSTM memory block, where $\boldsymbol{W}_c$, $\boldsymbol{W}_i$, $\boldsymbol{W}_f$, and $\boldsymbol{W}_o$ denote the weight transformation matrices in the LSTM memory block, while $\boldsymbol{b}_c$, $\boldsymbol{b}_i$, $\boldsymbol{b}_f$, and $\boldsymbol{b}_o$ denote the bias matrices, respectively. σ denotes the sigmoid transformation function. $\varphi(\cdot)$ denotes the tanh function. $\odot$ denotes the multiplication of elements between two matrices. h_t denotes the output of the last hidden layer in the whole LSTM mechanism. Purchase behavior records of users over multiple time periods are processed and mined through the LSTM mechanism, which can effectively capture the long-term purchase interest characteristics of users.

In the LSTM mechanism, three key gate control modes are used to handle data flow. Through the input gate, the input information from the input layer at each moment will pass through first, and the switch of the input gate will decide whether the information will be inputted to the memory block at this moment. The output gate determines whether information is outputted from the memory block at each moment. The forgetting gate validates whether critical information has been forgotten. Through data processing and control by these three gates, the feature values of the time sequence will be within a certain limit, which will help in the optimization of the whole architecture.

Further, by adding the depth of the network, stacked LSTMs are helpful in extracting multi-level user features. Therefore, we increase the depth of the LSTM hidden layer to extend the feature dimension of the extracted time sequence. As a result, the combined long- and short-term interest features of users are obtained. The specific formulation is shown below.

$$H_{tar} = \sum_{t=1}^{T} h_t^d / T \tag{11}$$

where H_{tar} denotes the long- and short-term purchase interest features of u_{tar}. h_t^d denotes the output of the hidden layer of LSTMs when the depth is d and time period is t. The outputs of the hidden layer for each time period are considered and averaged to represent the user's interest features. After the stacked LSTM structure is trained, multidimensional interest features of users are obtained.

4.3 Similarity measurement

Based on the long- and short-term interest features of each user obtained from the LSTM mechanism, we feed such features into a feedforward neural network (FNN) to capture the final user features[32–34]. In the FNN structure, the input matrix is a, the weight transformation matrix from input to output is W, b is the bias moment corresponding to the weight matrix, and the final output matrix is Z. The multilayer FNN framework is structured as follows:

$$Z_1 = a^1 W_1 + b^1 \tag{12}$$

$$Z_s = a^{s-1} W^{(s-1)} + b^{(s-1)}, s = 2, 3, \ldots, S \tag{13}$$

$$Z = a^S W^{(S)} + b^{(S)}, s = 2, 3, \ldots, S \tag{14}$$

Consequently, the multilayer FNN features are extracted. The final behavioral feature representation corresponding to each user can be represented as follows:

$$I_{tar} = (W_{tar}^3 (W_{tar}^2 (H_{tar} W_{tar}^1 + b_{tar}^1) + b_{tar}^2) + b_{tar}^3) \tag{15}$$

$$I_j = (W_j^3 (W_j^2 (H_j W_j^1 + b_j^1) + b_j^2) + b_j^3) \tag{16}$$

where I_{tar} and I_j represent the final user behavior representation of u_{tar} and u_j. In order to recommend fresh products to the target users, we adopt similarity measures to obtain similar friends of the target users. The recommendation list is enriched with the help of friends' purchase behavior records. Specifically, we use scalar product to measure the similarity between users.

$$M(u_{tar}, u_j) = |I_{tar}||I_j| \cos\theta \tag{17}$$

where $M(u_{tar}, u_j)$ denotes the similarity value between u_{tar} and u_j that measures the similarity of their behavioral records. Low similarity value shows the dissimilarity of users, so we set the similarity threshold to discover similar friends. In Eq. (17), we use the scalar product formula to calculate the similarity between users, which can effectively capture the interdependence between user-item ratings and facilitate the prediction of unobserved ratings. U_{sim} denotes the set of similar friends with similarity value greater than 0.5 to the target user. The ratings prediction of the unpurchased products by target user is as follows:

$$r_{tar} = \frac{1}{|U_{sim}|} \times \sum_{u_j \in U_{sim}} r_{n,u_j} \tag{18}$$

where r_{n,u_j} denotes the rating of the product p_n purchased by u_j. By combining the purchase records of numerous similar friends, we can explore the products that the user may be interested in and predict the corresponding ratings to generate the optimal product recommendation list for the target user. Here, we put users whose similarity value to the target user is greater than 0.6 into set U_{sim}. Products that have been purchased by similar users are leveraged to generate fresh products for the target user.

Furthermore, the training of model parameters should be improved. We utilize the mean squared error (MSE) loss function in order to optimize the parameter regression for each layer of the neural network[35, 36]. The concrete formulation is shown below.

$$NLL(x, y) = \frac{1}{2} \times \sum_{i=1}^{N} (r_{tar} - \hat{r}_{tar})^2 \tag{19}$$

where r_{tar} represents the real rating of the product purchased by the user and $\hat{r}_{tar}$ represents the predicted rating after the model has been trained. The MSE loss function is a popular regression loss function for neural network models. Furthermore, the specific process of DynaPR model is shown in Algorithm 1.

5 Experiment

Datasets. We validated the performance of the DynaPR model with the Epinions dataset, which is a typical dataset in the real-world recommendation field. It includes 7450 users and 6100 products. Users who purchased less than 5 products and products with less than 8 ratings were filtered out to obtain a high quality dataset.

Competitive approaches. The model has been trained for numerous rounds, it is necessary to check

Algorithm 1 DynaPR model

Input: User set: $U = \{u_1, u_2, \ldots, u_M\}$; Product set: $P = \{p_1, p_2, \ldots, p_N\}$; Attribute information: Rating set $R(t)$ and Category set $C(t)(1 \leqslant t \leqslant T)$;
Output: Set of recommended products $P_{tar} = \{P_{tar}^1, P_{tar}^2, \ldots, P_{tar}^j\}$;
for user $u_{tar} \in U$ **do**
  for product $p_i \in P$ **do**
    Embedding Projection $E_{p,t}^{tar}, E_{r,t}^{tar}, E_{c,t}^{tar} \leftarrow$ use
    Eqs. (1)–(3);
    Embedding concatenation $E_t^{tar} \leftarrow$ use Eq. (4) ;
  End for;
End for;
for $Epo \in Epochs$ **do**
  for u_j, u_{tar} in U **do**
    obtain interest feature $H_{tar} \leftarrow$ use Eqs. (5)–(11)
    obtain interest feature $H_j \leftarrow$ use Eqs. (5)–(11)
    obtain behavior feature $H_{tar}, H_j \leftarrow$ use Eqs. (12)–(16)
    Parameters Regression $\leftarrow$ use Eq. (19);
  End for;
End for;
Calculate the similarity value $M(u_{tar}, u_j) \leftarrow$ use Eq. (17);
Obtain the set of similar friends U_{sim};
Return the set of recommended products $P_{tar} = \{P_{tar}^1, P_{tar}^2, \ldots, P_{tar}^j\}$.

the validity of the proposed model. We compare the proposed model with other state-of-the-art competitive methods. The competitive methods are described as follows.

• **LSTM**[37]. This method is capable of converting a user's purchase records over multiple time periods into multiple time series that are embedded in an LSTM network. The similarity values between the target user and similar friends are then calculated.

• **JC**[38]. Jaccard coefficient is a frequently used similarity measure in recommender systems. Here we consider the intersection of the purchase records of the target user and similar friends. The similarity value is then calculated.

• **Random**[39]. We randomly draw the user's purchase history for the product, without considering the user's dynamic long-term interest over multiple time periods. In this case, scalar product is applied as a similarity measure formula to discover similar friends of the target user.

Metrics. Metrics can optimize the goals of a recommendation system, and check the effectiveness of the model architecture, which in turn measures the strengths and weaknesses of the recommendation algorithm. As common metrics in recommender systems, precision[40–43] and recall[44–46] are used as metrics to test the effectiveness of DynaPR models.

5.1 Experiments results

We first consider the personalized recommendation effect of the proposed DynaPR model. Figure 3 shows the accuracy of rating predictions, i.e., precision and recall[47–50]. We have the following exploratory observations.

Effectiveness of the DynaPR model. In this section, the DynaPR model is compared with three other state-of-the-art methods.

• As shown in Fig. 3a, the random method exhibits the poorest precision compared with the other three methods, whose average precision remains at 50% across five dataset densities. This phenomenon is due to the fact that the randomly selected purchase sequences are unordered and unregularised, while the other three methods are able to regularise users' historical product purchase records[51, 52].

• The JC method is able to extract "common ratings" and "common categories" of users' purchases over multiple time periods, thus effectively extracting changes in users' interests over time. The JC method performs better than the Random method in comparing the differences and similarities between two users' individual purchases.

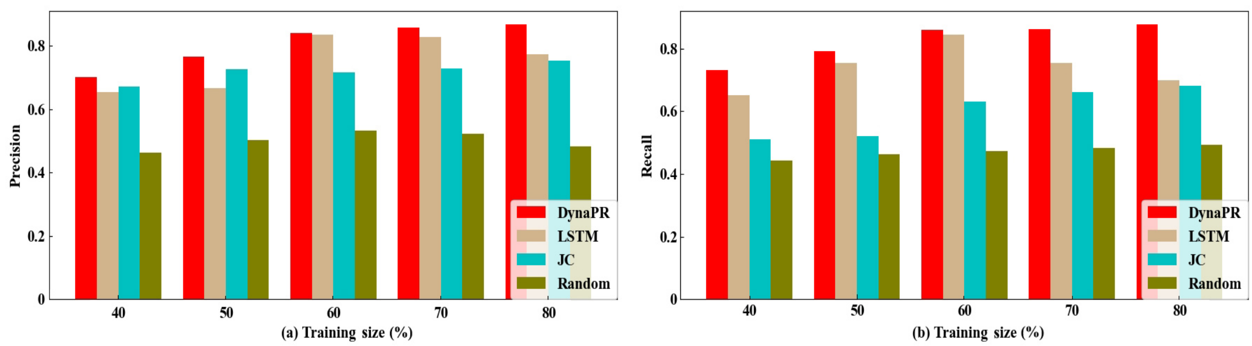

Fig. 3 Performance comparison of DynaPR model with competitive approaches.

• The LSTM mechanism[53–55] is suitable for training the correlation of purchase sequences over multiple time periods and extracting the corresponding purchase sequence features. After multiple rounds of recurrence and looping of the neural network, the long and short-term interests of users can be obtained[56].

• As shown in Figs. 3a and 3b, the DynaPR model provides the best performance compared to other baseline methods from the perspective of precision and recall. Firstly, the DynaPR model exploits the product and attribute profiles of each user's purchase sequence, which compensates for the effects of data sparsity as well as obtain a rich representation of the user[57, 58]. Secondly, the advanced framework components in the DynaPR model such as embedding and LSTM are able to capture the long-term and short-term interests of users as they evolve over multiple time periods. Finally, the semantic representation of richness can be measured by similarity measures.

In summary, the experimental results indicate that: (1) Attribute information can complement product information and contribute to a rich user feature representation; (2) Optimisation of neural network components can enhance the effectiveness of personalized recommendations; (3) DynaPR has the best results compared to other superior baseline methods.

5.2 Ablation experiments

Ablation experiments with different traing sizes. To further verify which part of the user data plays a pivotal role in the DynaPR model, we investigate ablation experiments to confirm the validity of the data. As shown in Table 1, the DynaPR–PF method indicates that the input data in the DynaPR model are only product features while the DynaPR–AF method indicates that the input data in the DynaPR model are only attribute features (as shown in Fig. 1). The performance of the three methods is presented under different metrics from 40% to 80% training size. On the one hand, the accuracy of the three models gradually increases as the training size increases, and DynaPR achieves the best results at 80% of the training set. On the other hand, the recall of the three models performs relatively smoothly under different training sizes. Collectively, the DynaPR model performs the best, which combines product features and attribute features. Single product features or attribute features fail to represent user features effectively.

Table 1 Effectiveness of ablation experiments.

Metric	Training ratio	DynaPR–PF	DynaPR–AF	DynaPR
Precision	40%	0.6787	0.6890	0.7065
	50%	0.8371	0.7868	0.7668
	60%	0.8435	**0.8526**	0.8410
	70%	0.8538	0.8261	0.8581
	80%	0.8616	0.8462	**0.8668**
Recall	40%	0.7205	0.6499	0.7313
	50%	0.7048	0.7541	0.7920
	60%	0.7916	0.7555	0.8591
	70%	0.8562	0.8102	0.8637
	80%	0.8563	0.8455	**0.8763**

Convergence of ablation experiments. As shown in Fig. 4. We track the performance of the three models during the network training. All models perform best when the three models were trained up to 30 epochs. When the models start training, the performance of the three models is fluctuating before training up to 15 epochs, which is due to the difficulty in finding effective feature information. The three models converge to smoothness at 25 to 30 epochs, and there is no significant change in the accuracy performance. Meanwhile, this phenomenon indicates that the network structure of the models is capable of capturing user features correctly[59–61].

5.3 Parameter effect of the DynaPR model

In this section we focus on parameters effect of the model's components on rating prediction.

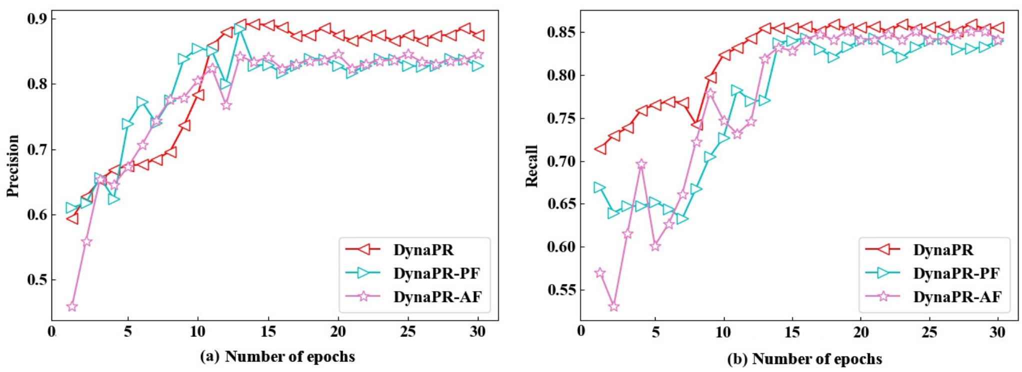

Fig. 4 Convergence comparison of ablation experiments.

Effect of embedding dimension. As a technique for processing the input information to the proposed model, embedding techniques can transform and fuse multiple types of data, while embedding dimensions can scale the original data. Different embedding dimensions can represent different levels of dense vectors.

To explore the effect of embedding dimension on the model, we increase the embedding dimension from 32 to 96 with an increment of 16. As shown in Fig. 5, the precision of our model increases progressively as the embedding dimension increases from 32 to 48. As the embedding dimension increases from 64 to 96, the precision tends to decrease. The recall metric are also taken into account to validate the effect of embedding dimensionality in combination[62, 63]. When the embedding dimensions are 48 and 64, the model has a similar recall performance. Among a couple of experiments, the proposed model performs best when the embedding dimension is 48. In general, it is difficult to achieve the best performance with both overly large and undersized embedding dimensions[64], and it is necessary to test multiple levels of embedding dimensions to achieve the optimal model performance.

Effect of LSTM depth. The LSTM mechanism, as an important component of the DynaPR model, allows for feature extraction of users' purchase sequence over multiple time periods. With the flexibility and controllability of the LSTM network, we explore the effect of the LSTM depths on the whole framework. the depth of the LSTM is set to {2, 3, 4, 5, 6}. As shown in Fig. 6, when the depth of the LSTM is set to 2, 3, and 4, the precision of our model gradually increases to 0.8734, and then the precision of the model gradually decreases at depths from 5 to 6. From a recall perspective, our model performs relatively consistently and maintains a high stability. This phenomenon shows that our model is always able to find the most like-minded friends and make the best recommendations regardless of the LSTM depth[65]. From the above experimental results, our proposed model has the most optimal performance when the LSTM depth is 4.

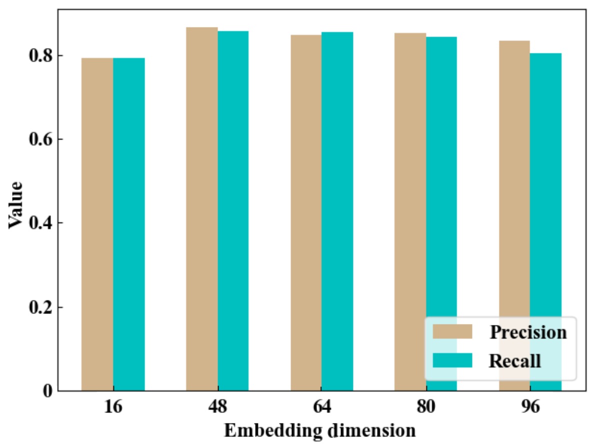

Fig. 5 Effect of embedding dimension.

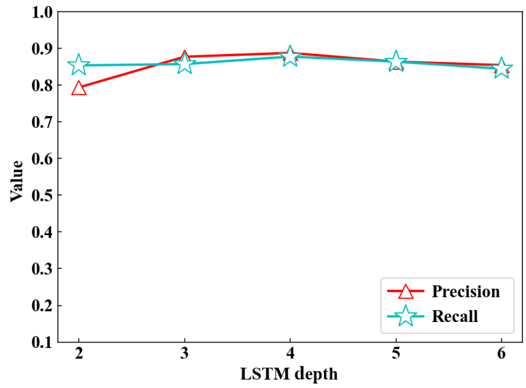

Fig. 6 Effect of LSTM depth.

In summary, the network parameters have a significant impact on the model. It is quite essential to examine the effect of parameter changes on the prediction results for the task.

6 Conclusion

We propose a time-aware LSTM neural network personalized recommendation model for business intelligence. Specifically, we design a comprehensive framework that combines both product features and attribute features using embedding and LSTM mechanisms. Then, users' long-term and short-term interests are extracted. Finally, the similarity values between users are measured by pairwise FNN. The scalable experiments show that our proposed model can discover the most similar friends and give the optimal recommendation results.

Moreover, the DynaPR model only considers the users' behavioral interaction sequences over multiple time periods while ignoring the social relationships between users (e.g., friends, family). Research shows that users are more likely to trust the recommendations of trusted friends[29, 44]. In future work, we will explore the impact of friendships on personalized recommendations for business intelligence to make more accurate rating predictions.

References

[1] V. V. Mišić and G. Perakis, Data analytics in operations

management: A review, *Manuf. Serv. Oper. Manag.*, vol. 22, no. 1, pp. 158–169, 2020.

[2] X. Zhou, W. Liang, K. Yan, W. Li, K. I. K. Wang, J. Ma, and Q. Jin, Edge-enabled two-stage scheduling based on deep reinforcement learning for Internet of everything, *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3295–3304, 2022.

[3] X. Zhou, Y. Hu, J. Wu, W. Liang, J. Ma, and Q. Jin, Distribution bias aware collaborative generative adversarial network for imbalanced deep learning in industrial IoT, *IEEE Trans. Ind. Inform.*, vol. 19, no. 1, pp. 570–580, 2022.

[4] M. R. Llave, Business intelligence and analytics in small and medium-sized enterprises: A systematic literature review, *Procedia Comput. Sci.*, vol. 121, pp. 194–205, 2017.

[5] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I. K. Wang, Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system, *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9310–9319, 2021.

[6] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT, *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12588–12596, 2021.

[7] X. Zhou, W. Liang, K. I. K. Wang, and L. T. Yang, Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations, *IEEE Trans. Comput. Soc. Syst.*, vol. 8, no. 1, pp. 171–178, 2021.

[8] M. I. Jordan and T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[9] L. Qi, Y. Yang, X. Zhou, W. Rafique, and J. Ma, Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0, *IEEE Trans. Ind. Inform.*, vol. 18, no. 9, pp. 6503–6511, 2022.

[10] H. J. Ye, X. Yang, X. Wang, and T. C. Stratopoulos, Monetization of digital content: Drivers of revenue on Q&A platforms, *J. Manag. Inf. Syst.*, vol. 38, no. 2, pp. 457–483, 2021.

[11] F. Wang, H. Zhu, G. Srivastava, S. Li, M. R. Khosravi, and L. Qi, Robust collaborative filtering recommendation with user-item-trust records, *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 4, pp. 986–996, 2022.

[12] F. Wang, L. Wang, G. Li, Y. Wang, C. Lv, and L. Qi, Edge-cloud-enabled matrix factorization for diversified APIs recommendation in mashup creation, *World Wide Web*, vol. 25, no. 5, pp. 1809–1829, 2022.

[13] J. Liu, C. Wu, Y. Xiong, and W. Liu, List-wise probabilistic matrix factorization for recommendation, *Inf. Sci.*, vol. 278, pp. 434–447, 2014.

[14] R. Salakhutdinov and A. Mnih, Probabilistic Matrix Factorization, in *Proc. 20th Int Conf. on Neural Information Processing Systems*, Red Hook, NY, USA, 2007, pp. 1257–1264.

[15] Z. Cui, P. Zhao, Z. Hu, X. Cai, W. Zhang, and J. Chen, An improved matrix factorization based model for many-objective optimization recommendation, *Inf. Sci.*, vol. 579, pp. 1–14, 2021.

[16] W. C. Kang and J. McAuley, Self-attentive sequential recommendation, in *Proc. 2018 IEEE Int. Conf. Data Mining*, Singapore, Singapore, 2018, pp. 197–206.

[17] R. He and J. McAuley, Fusing similarity models with Markov chains for sparse sequential recommendation, in *Proc. 2016 IEEE 16th Int. Conf. Data Mining*, Barcelona, Spain, 2017, pp. 191–200.

[18] F. Xie, Z. Chen, H. Xu, X. Feng, and Q. Hou, TST: Threshold based similarity transitivity method in collaborative filtering with cloud computing, *Tsinghua Science and Technology*, vol. 18, no. 3, pp. 318–327, 2013.

[19] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, Session-based recommendations with recurrent neural networks, arXiv preprint arXiv: 1511.06939, 2015.

[20] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, Sequential recommendation with user memory networks, in *Proc. Eleventh ACM Int. Conf. Web Search and Data Mining*, Marina Del Rey, CA, USA, 2018, pp. 108–116.

[21] L. Kong, G. Li, W. Rafique, S. Shen, Q. He, M. R. Khosravi, R. Wang, and L. Qi, Time-aware missing healthcare data prediction based on ARIMA model, *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 2022.

[22] J. Zhang, Y. Wang, Z. Yuan, and Q. Jin, Personalized real-time movie recommendation system: Practical prototype and evaluation, *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 180–191, 2019.

[23] X. Ye and D. Liu, A cost-sensitive temporal-spatial three-way recommendation with multi-granularity decision, *Inf. Sci.*, vol. 589, pp. 670–689, 2022.

[24] S. Xu, H. Zhuang, F. Sun, S. Wang, T. Wu, and J. Dong, Recommendation algorithm of probabilistic matrix factorization based on directed trust, *Comput. Electr. Eng.*, vol. 93, p. 107206, 2021.

[25] T. X. Tuan and T. M. Phuong, 3D convolutional networks for session-based recommendation with content features, in *Proc. Eleventh ACM Conf. Recommender Systems*, Como, Italy, 2017, pp. 138–146.

[26] J. You, Y. Wang, A. Pal, P. Eksombatchai, C. Rosenburg, and J. Leskovec, Hierarchical temporal convolutional networks for dynamic recommender systems, in *Proc. World Wide Web Conf.*, New York, NY, USA, 2019, pp. 2236–2246.

[27] Z. Yin and Y. Shen, On the dimensionality of word embedding, in *Proc. 32th Int. Conf. on Neural Information Processing Systems*, Red Hook, NY, USA, 2018, pp. 895–906. .

[28] S. Wu, S. Shen, X. Xu, Y. Chen, X. Zhou, D. Liu, X. Xue, and L. Qi, Popularity-aware and diverse web APIs recommendation based on correlation graph, *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 2, pp. 771–782, 2023.

[29] Y. Xu, Z. Feng, X. Xue, S. Chen, H. Wu, X. Zhou, M. Xing, and H. Chen, MemTrust: find deep trust in your mind, in *Proc. 2021 IEEE Int. Conf. Web Services*, Chicago, IL, USA, 2021, pp. 598–607.

[30] J. C. W. Lin, Y. Shao, Y. Djenouri, and U. Yun, ASRNN: A recurrent neural network with an attention model for sequence labeling, *Knowl. Based Syst.*, vol. 212, p. 106548, 2021.

[31] H. Dai, J. Yu, M. Li, W. Wang, A. X. Liu, J. Ma, L. Qi, and G. Chen, Bloom filter with noisy coding framework for multi-set membership testing, *IEEE Trans. Knowl. Data Eng.*, 2022.

[32] S. Cheng, Y. Wu, Y. Li, F. Yao, and F. Min, TWD-SFNN: Three-way decisions with a single hidden layer feedforward neural network, *Inf. Sci.*, vol. 579, pp. 15–32, 2021.

[33] Y. Jin, W. Guo, and Y. Zhang, A time-aware dynamic service quality prediction approach for services, *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 227–238, 2019.

[34] Y. Yang, X. Yang, M. Heidari, M. A. Khan, G. Srivastava, M. Khosravi, and L. Qi, ASTREAM: data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment, *IEEE Trans. Netw. Sci. Eng.*, 2022.

[35] C. Huang, Q. Zhang, D. Guo, X. Zhao, and X. Wang, Discovering association rules with graph patterns in temporal networks, *Tsinghua Science and Technology*, vol. 28, no. 2, pp. 344–359, 2022.

[36] X. Xu, T. Gao, Y. Wang, and X. Xuan, Event temporal relation extraction with attention mechanism and graph neural network, *Tsinghua Science and Technology*, vol. 27, no. 1, pp. 79–90, 2021.

[37] Y. Zhang, C. Yin, Z. Lu, D. Yan, M. Qiu, and Q. Tang, Recurrent Tensor Factorization for time-aware service recommendation, *Appl. Soft Comput.*, vol. 85, p. 105762, 2019.

[38] M. Ayub, M. Ali Ghazanfar, M. Maqsood, and A. Saleem, A Jaccard base similarity measure to improve performance of CF based recommender systems, in *Proc. 2018 Int. Conf. Information Networking*, Chiang Mai, Thailand, 2018, pp. 1–6.

[39] E. B. Barker and J. Kelsey, Recommendation for random bit generator (RBG) constructions, Gaithersburg, MD, USA: NIST, 2012.

[40] D. Ma, Y. Wang, J. Ma, and Q. Jin, SGNR: A social graph neural network based interactive recommendation scheme for E-commerce, *Tsinghua Science and Technology*, vol. 28, no. 4, pp. 786–798, 2023.

[41] L. Qi, W. Lin, X. Zhang, W. Dou, X. Xu, and J. Chen, A correlation graph based approach for personalized and compatible web APIs recommendation in mobile APP development, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5444–5457, 2023.

[42] L. Zhang, C. Xu, Y. Gao, Y. Han, X. Du, and Z. Tian, Improved Dota2 lineup recommendation model based on a bidirectional LSTM, *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 712–720, 2020.

[43] M. Jamali and M. Ester, *TrustWalker*: a random walk model for combining trust-based and item-based recommendation, in *Proc. 15th ACM SIGKDD Int. Conf. Knowledge discovery and data mining*, Paris, France, 2009, pp. 397–406.

[44] C. Yan, Y. Zhang, W. Zhong, C. Zhang, and B. Xin, A truncated SVD-based ARIMA model for multiple QoS prediction in mobile edge computing, *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 315–324, 2021.

[45] H. Dai, Y. Xu, G. Chen, W. Dou, C. Tian, X. Wu, and T. He, ROSE: robustly safe charging for wireless power transfer, *IEEE Trans. Mob. Comput.*, vol. 21, no. 6, pp. 2180–2197, 2022.

[46] R. Gu, Y. Chen, S. Liu, H. Dai, G. Chen, K. Zhang, Y. Che, and Y. Huang, Liquid: Intelligent resource estimation and network-efficient scheduling for deep learning jobs on distributed GPU clusters, *IEEE Trans. Parallel Distributed Syst.*, vol. 33, no. 11, pp. 2808–2820, 2022.

[47] H. Dai, X. Wang, X. Lin, R. Gu, S. Shi, Y. Liu, W. Dou, and G. Chen, Placing wireless chargers with limited mobility, *IEEE Trans. Mob. Comput.*, vol. 22, no. 6, pp. 3589-3603, 2023.

[48] R. Gu, K. Zhang, Z. Xu, Y. Che, B. Fan, H. Hou, H. Dai, L. Yi, Y. Ding, G. Chen et al., Fluid: Dataset abstraction and elastic acceleration for cloud-native deep learning training jobs, in *Proc. 2022 IEEE 38th Int. Conf. Data Engineering*, Kuala Lumpur, Malaysia, 2022, pp. 2182–2195.

[49] Y. Li, J. Liu, B. Cao, and C. Wang, Joint optimization of radio and virtual machine resources with uncertain user demands in mobile cloud computing, *IEEE Trans. Multimed.*, vol. 20, no. 9, pp. 2427–2438, 2018.

[50] Y. Li, C. Liao, Y. Wang, and C. Wang, Energy-efficient optimal relay selection in cooperative cellular networks based on double auction, *IEEE Trans. Wirel. Commun.*, vol. 14, no. 8, pp. 4093–4104, 2015.

[51] Y. Li, S. Xia, M. Zheng, B. Cao, and Q. Liu, Lyapunov optimization-based trade-off policy for mobile cloud offloading in heterogeneous wireless networks, *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 491–505, 2022.

[52] D. Zhou, X. Xue, and Z. Zhou, SLE2: the improved social learning evolution model of cloud manufacturing service ecosystem, *IEEE Trans. Ind. Inform.*, vol. 18, no. 12, pp. 9017–9026, 2022.

[53] X. Xue, S. Wang, L. Zhang, Z. Feng, and Y. Guo, Social learning evolution (SLE): Computational experiment-based modeling framework of social manufacturing, *IEEE Trans. Ind. Inform.*, vol. 15, no. 6, pp. 3343–3355, 2019.

[54] X. Xue, G. D. Li, D. Y. Zhou, Y. P. Zhang, L. Zhang, Y. Zhao, Z. Y. Feng, L. Z. Cui, Z. B. Zhou, X. Sun, X. D. Lu, and S. Z. Chen, The Research Roadmap of Service Ecosystem: A Crowd Intelligence Perspective, *International Journal of Crowd Science*, vol. 6, no. 4, pp. 195–222, 2022.

[55] Z. Xie, Y. Huang, D. Yu, R. M. Parizi, Y. Zheng, and J. Pang, FedEE: A federated graph learning solution for extended enterprise collaboration, *IEEE Trans. Ind. Inform.*, 2022.

[56] D. Yu, Z. Zou, S. Chen, Y. Tao, B. Tian, W. Lv, and X. Cheng, Decentralized parallel SGD with privacy preservation in vehicular networks, *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5211–5220, 2021.

[57] L. Yang, Y. Zou, M. Xu, Y. Xu, D. Yu, and X. Cheng, Distributed consensus for blockchains in internet-of-things networks, *Tsinghua Science and Technology*, vol. 27, no. 5, pp. 817–831, 2022.

[58] D. Yu, D. Wang, Q. Luo, Y. Zheng, G. Wang, and Z. Cai, Stable structural clustering in uncertain graphs, *Inf. Sci.*, vol. 586, pp. 596–610, 2022.

[59] Q. Luo, D. X. Yu, Z. P. Cai, X. M. Lin, G. H.Wang, X. Z. Cheng, Towards Maintenance of k-Hypercores in Large-Scale Dynamic Hypergraphs, *VLDB*, vol. 32, no. 3, pp. 647–664. 2023.
[60] W. Wang, Y. Wang, P. Duan, T. Liu, X. Tong, and Z. Cai, A Triple Real-time Trajectory Privacy Protection mechanism Based on Edge computing and Blockchain in Mobile Crowdsourcing, *IEEE Trans. Mob. Comput.*, 2022.
[61] Q. Zhang, Y. Wang, G. Yin, X. Tong, A. M. V. V. Sai, and Z. Cai, Two-stage bilateral online priority assignment in spatio-temporal crowdsourcing, *IEEE Trans. Serv. Comput.*, 2022.
[62] Y. Xu, Z. Liu, C. Zhang, J. Ren, Y. Zhang, and X. Shen, Blockchain-based Trustworthy Energy Dispatching Approach for High Renewable Energy Penetrated Power Systems, *IEEE Internet Things J.*, vol. 9, no. 12, pp. 10036–10047, 2022.
[63] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, A blockchain-enabled deduplicatable data auditing mechanism for network storage services, *IEEE Trans. Emerg. Top. Comput.*, vol. 9, no. 3, pp. 1421–1432, 2021.
[64] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, A blockchain-based multi-cloud storage data auditing scheme to locate faults, *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2252–2263, 2022.
[65] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, Blockchain empowered arbitrable data auditing scheme for network storage as a service, *IEEE Trans. Serv. Comput.*, vol. 13, no. 2, pp. 289–300, 2020.

James A. Esquivel received the Bachelor's degree in mathematics and computer science from University of Santo Tomas, Philippines in 1989, the Master's degree in computer science from De La Salle University, Philippines in 1998, and the PhD degree in Information Technology from Angeles University Foundation, Philippines in 2014. He is currently the professor of computer science at Angeles University Foundation. His main research interests include machine learning and predictive analytics.

Xuan Yang received the MS degree in chemical engineering from Qilu University of Technology, China, in 2014. He is currently pursuing the PhD degree in information technology at Angeles University Foundation in Philippines. His main research interests include business intelligence and recommendation system.