Things of Interest Recommendation with Multidimensional Context Embedding in the Internet of Things

Shuhua Li[1], Jingmin An[2]
School of Computer, Guangdong University of Science and Technology, Guangdong, China[1]
School of Information Science and Technology, Dalian Maritime University, Dalian, China[2]

***Abstract*—The emerging Internet of Things (IoT) makes users and things closely related together, and the interactions between users and things generate massive context data, where the preference information in time, space, and textual content is embedded. Traditional recommendation methods (e.g., movie, music, and location recommendations) are based on static intrinsic context information, which lacks consideration regarding real-time content and spatiotemporal features, failing to adapt to the personalized recommendation in IoT. Therefore, to meet users' interests and needs in IoT, a novel effective and efficient recommendation method is urgently needed. The paper focuses on mining users' things of interest in IoT via leveraging multidimensional context embedding. Specifically, to address the challenge from massive context data embedding different user preference information, the paper employs Convolutional Neural Networks (CNN) to mine the intrinsic content information of things and learn their represent. To solve the real-time recommendation problem, the paper proposes a real-time multimodal model embedded into location, time, and some instant content information to track the features of users and things. Furthermore, the paper proposes a matrix factorization-based framework using the regularization method to fuse real-time context embedding and intrinsic information embedding. The experimental results demonstrate the proposed method tailored to IoT is adaptable and flexible, and able to capture user personalized preference effectively.**

Keywords—Internet of things; things of interest; multidimensional context embedding; intrinsic information; instant information; matrix factorization

I. Introduction

The emerging Internet of Things (IoT) is promoting the growth of connected things (e.g., sensors, actuators, and mobile devices), which makes a large amount of data available from interactions between users and things. The data includes time, locations, textual contents, and interaction records. In addition, in IoT, some context data is changing over time to time, such as users' locations, and things' function availability (things in use or not in use), and in term of that one, the data could be divided into intrinsic context and instant context. Therefore, the data is characterized by massive, multidimensional, and variant. To accelerate proactively searching and promote convenient life from the massive and overloaded data, an intelligent and automated method capable of deeper understanding and mining the information is needed for personalized recommendations in IoT. The things recommendations tailored to IoT should put more emphasis on users' and things' states under different scenarios and time besides historical interactive preferences. Hence, the things recommendation is more complex than the conventional recommender systems like movie recommendations [1-4], music recommendations [5-7], and other location recommendations [8-10].

In IoT, each user has its own unique behavior pattern, and the things of interest and interactive behaviors usually vary with time in a day, and these behaviors are regular and cyclical. To clearly observe users' real behaviors, the paper conducts an example for the spatiotemporal feature analysis on the real datasets from CASAS [1], which is a database collected from a smart home environment. Due to the space limitation, only three users' behavior records are shown on the locations and time. In Fig. 1, the three users interact with similar things except for their own locations, such as latitudes and longitudes. And Fig. 2 depicts the three users' action frequencies in the different time period are unique: user 1 usually interacts with things in the morning and afternoon, user 2 at noon, and user 3 in the early morning and evening. Consequently, the recommendation in IoT is personalized independence, context-dependence, real-time, and complexity. The following are the main challenges of achievement for things recommendations in IoT.

- *Mining and indicating things intrinsic content information*. When users decide to use a thing, they always make a primary assessment that the function of the thing meets interest or not. The descriptions of functional features are derived from the textual contents of things. Failure to mine and indicate the intrinsic content of things may result in some inaccurate recommendation results.
- *Highly dynamic*. In IoT, the locations and interests of users and the availability of things are dynamic, calling for the model capable of adapting to the changes in real-time and presenting the most timely recommendation results.
- *Data sparsity*. Compared with massive things in IoT, the things generating interactions with each user are limited, namely, the density of user-thing rating matrix

[1]http://ailab.wsu.edu/casas/datasets/

is quite low. Therefore, under this circumstance, it is difficult to sharply explore what users may be interested in.

In light of the challenges above, the paper proposes a matrix factorization framework fusing multidimensional context embedding (McEMF), including time, intrinsic textual content, and instant location and status information to address the recommendations in IoT. Specifically, to represent and learn users' periodic behavior regularly, the paper develops a temporal-user-thing rating matrix to record interactions between users and things. Then the rating matrix is used to implement the users' preference model. To mine and indicate the things intrinsic content, the paper employs CNN to learn intrinsic content embedding, which could be used to measure the semantic relationships on the functions of things. And leveraging the semantic relationships, users' preferences could be further explored with CF. To embed instant information, the paper adopts a particle filtering-based tracking method to capture the latest states of users and things. The benefit of instant information embedding is that it could help the recommender system enhance the efficiency of real-time state awareness and solve the cold-start problem. Indeed, data sparsity is a critical problem for historical data based recommender systems, and the textual content, location information, and time information are fused into the model to effectively alleviate the problem of data sparsity.

To sum up, the main contributions of the proposed McEMF are as follows:

- McEMF is a personalized things recommendation method tailored to IoT. By taking multidimensional contexts into account, McEMF captures both intrinsic content and instant information, addressing time awareness.

- Intrinsic contents and instant information are fused with improved matrix factorization technique (MF). In particular, the paper develops a CNN-based method to concatenate textual content and real-time states of things, and the real-time locations of users can constantly updated in the model to estimate the geographical relationships between things and users.

- The paper implements experiments to validate and evaluate the performance of the proposed McEMF on a real-world IoT database. The experimental results demonstrate that McEMF outperforms state-of-the- art baseline methods in effectiveness and efficiency, and it achieves the capability of IoT recommendation in real-time.

The rest of the paper is organized as follows. The paper reviews the related researches on things recommendation and IoT-oriented things recommendation in Section II. The paper presents the proposed McEMF model and describes the technical details of each procedure in Section III. The paper gives the experimental settings and reports the evaluated results of performance in Section IV. The paper is concluded in Section V.

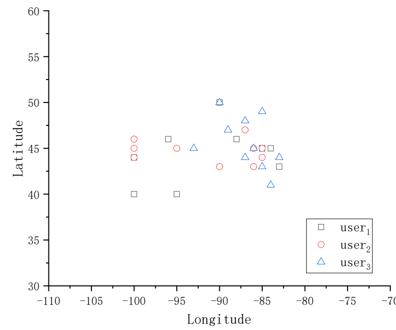

Fig. 1. Distributions of interactions between users and things.

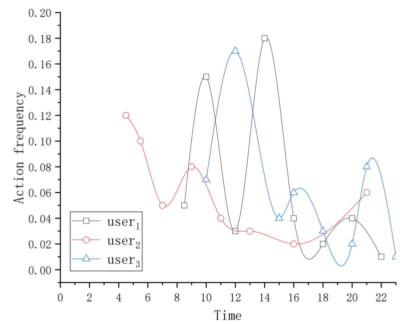

Fig. 2. Probabilities of interactions between users and things.

II. Related Works

A. Things Recommendation

At present, things recommendation is a hot academic issue, and it not only benefits the person but also the third-party business. The recommender could proactively recommend users some things that they may be interested in or need, and the third-party business could also obtain more potential preferences of users. Traditional recommendation systems employ two kinds of techniques in general. One is collaborative filtering (CF) technique, and the achievement of many popular recommendation methods is based on CF to learn user preference on things from user daily records. The CF techniques are divided into the memory-based CF and the model-CF. The works [11, 12] adopt the memory-based CF, namely, they use the data collected from user behavior records to compute the similarity of users or things, to recommend things of interest for users, called as user-based CF and item-based CF respectively. Besides, the studies [3, 4, 13-15] leverage the model-based CF (e.g., Matrix Factorization) in the recommendation system. They treat each thing as an item and conduct a user-item matrix to learn the user preference on things, and each user and thing in the matrix are indicated with a k-dimensional latent vector respectively. MF has become one of the most popular techniques in the personalized recommendation, due to its effectiveness and efficiency for large sparse user-item rating matrix. Focusing on solving the problem of data sparsity, [16, 17] employ singular value decomposition (SVD) for matrices, and [18-20] propose a non-negative MF to handle high-dimensional and sparse data collected from industrial applications. Meanwhile, probabilistic MF (PMF) [21] is proposed and shown good achievements. Different from previous works with the explicit rating as feedback, [22,23] propose a second-order decomposing method to treat feedback records as implicit information. Another technique is content-based filtering, which works with the profiles of users or things in historical behavior records and recommends things in terms of the similarity from profiles. Study [10] presents a CF fusing the contents of users and things, and it recommends preference things according to the satisfaction of similar users for different things. [24, 25] expand the contents of things, associating some geographical information and social information with the contents, for recommendations. These methods alleviate the data sparsity problem to some extent. Recently, a few works integrate more context information, and they achieve better performance. Among them, some works [26-30] combine contexts with CF. e.g., [30] proposes a social spatio-temporal PMF framework, which exploits things similarity and user similarity via modeling the social space, geographical space and things category space, to achieve recommendations. The others [31-33] employ neural networks to fuse contexts, such as, [33] proposes a multisource fusion recommendation model, which jointly considers user preference, geographical information, and social information modeled by performing network representation. However, the methods above only consider some intrinsic information or historical records and ignore the important real-time or instant information that could play an important role for recommendations in IoT. For example, a chef cooks in a restaurant during the day and in the kitchen at home at night, and in this situation, people have different spatiotemporal characteristics that change over time to time, and obviously, traditional recommendation systems fail to it.

B. Things Recommendation in IoT

Time is a critical factor in modeling recommenders, as data is changing from time to time, and some works [27,29,32,34,35] have shown the importance of temporal features for the improvement of the efficiency of the recommenders. In IoT, the growth of data is exponential, and they have obvious temporal features. Therefore, when modeling user preference, temporal information is essential. Recently, there are few works on things recommendation systems in IoT. Research [36] proposes a Trinity method, and the method constructs three categories of graphs related things from things usage records, namely, user-thing graph, time-thing graph, and location-thing graph, to mine possible user preference. The author in [37] presents a STUnion model, whose core work is two created graphs. One is the spatiotemporal graph that represents the relationships between users, things, time, and locations. Another is the social graph that indicates the social relationships of users. And the two graph relationships are used to model the user preferences on things with a linear combination. Recently, [38] proposes a time-aware smart thing recommendation model, which integrates user preferences and different social relationships between objects learned via graph embedding. And then, to capture more potential relationships between users and things, [39] models the influences of geographical, social, manufacturer, and economic factors on interactions and integrates them in the recommender system by deriving transition probabilities. These methods above take advantage of spatial information and temporal information in the recommenders. However, they are insufficient for real-time information. Compared with traditional web data, physical things and users are more dynamic in IoT, thus the recommendation model tailored to IoT should be able to adapt up-to-date information. Consequently, the paper focuses on achieving a real-time recommendation system in the paper, and proposes a things-recommendation method with multidimensional context embedding, which captures intrinsic information and instant information, to make more accurate recommendations.

III. MULTIDIMENSIONAL CONTEXT EMBEDDING FOR THINGS RECOMMENDATIONS

In this section, the paper develops a multidimensional context embedding framework fusing intrinsic information and instant information as Fig. 3 and gives the procedures in detail. The proposed framework employs the historical interactive records (data) between users and things to model user preference. Meanwhile, it captures the instant information on locations of users and states (availability) of things for fitting the historical user preference to make the most up-to-date recommendations. Specifically, the framework consists of four procedures: (1) problem definition and notations in the framework; (2) intrinsic information embedding model; (3) real-time information embedding model; (4) fused model.

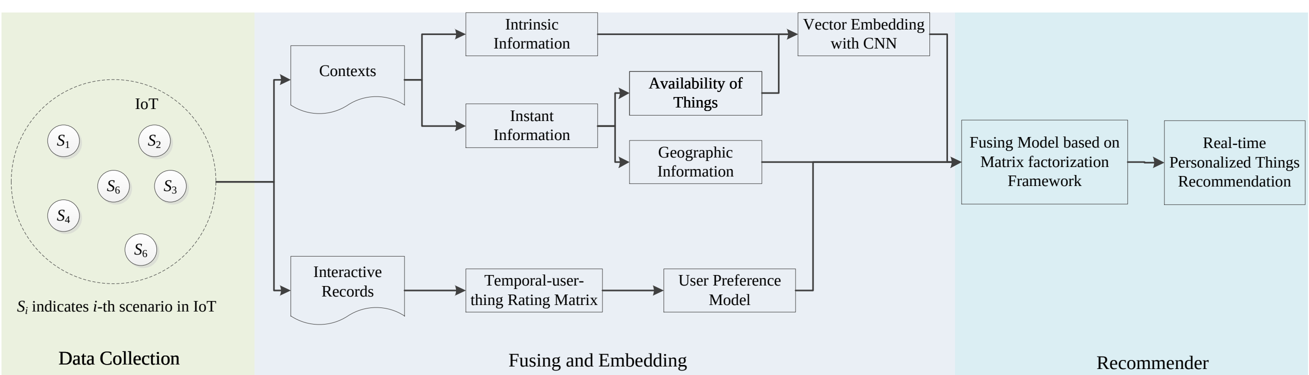

Fig. 3. Overview of the system framework.

A. Problem Definition and Notations

Formally, a thing's interactive record is created when a person interacts with a particular thing. Let U be a set of users, ui ∈ U, and T a set of things, tj ∈ T, and the record is represented as a pairwise ri,j=(ui, tj), which forms a matrix ri,j ∈R|U||T|. Besides, for each record, some context information is considered, such as, temporal information that indicates when the interaction happens, and spatial information that indicates where the interaction happens, and the profiles of things that indicate the functions of things. Each record is a quadruple of User, Thing with a certain function, Timestamp, Location as the following definitions.

Definition 1 (Things Interactive Record) Let U={u1, u2,…, um}, T={t1, t2,…, tn}, L={l1, l2,…, lp}, and I={ r1, r2,…, rq} represent the set of users, things, locations, and timestamps, respectively, where each tj has a bag of keywords Cd indicating its function review. An interactive record of a thing t is denoted by h∈H={h1, h2,…,hj}={<u, t, l, r>| u∈U∧t∈T∧l∈L∧r∈I}, indicating that user u uses thing t in the location l at timestamp r.

Definition 2 (Three-Level Time Granularity) According to our life experiences, users behave periodically in daily life. Specifically, people (users) may regularly stay at the workplace on weekdays and at the entertainment places or home on the weekends, namely, each user may have a different periodic behavior pattern on weekdays and weekends respectively. Let the temporal states Γ ={weekday, weekend}. Furthermore, one day is divided into 24 hours, thus 2∗24=48 temporal units Γ́ can be obtained, and r∈Γ́∈Γ. The paper exploits the three types of time to learn the temporal features of interactions.

Definition 3 (Temporal-User-Thing Rating Matrix) In a record h, the timestamp r is cyclical, therefore, matrix R|U||T| is extended to R|U||T, I|, ru,t,r∈R|U||T,I|. Note that due to location l with random and real-time, it is not suitable for merging into the matrix R|U||T,I| indicating the historical interaction records. Location l is a piece of instant information.

R|U||T, I| is a sparse matrix, and user historical preference is approximately expressed as $R_{|U||T,I|} \approx p_u^T q_{t,r}$, where p_u and $q_{t,r}$ are the user latent k-dimensional vector and the thing latent k-dimensional vector at timestamp r, respectively. The proposed framework is to predict the missing entities, and it will recommend the things with the bigger predicted ratings to given users. The major notations in the framework are summarized in Table I.

B. Intrinsic Information Embedding Model

When a user uses a specific thing, he/she mainly considers whether the functions of the thing can meet his/her needs or not. Therefore, the paper proposes an intrinsic information embedding model to represent the functional features of things as the low-dimensional vectors so that the model could learn the semantic relationships between things. The descriptions on things' functions are captured from the textual content, and CNN is employed for intrinsic information embedding. More specifically, as Fig. 4 illustrates. Given a text with C_d, each word c_z in C_d will be represented by an n-dimensional vector leveraging a non-static word embedding function. Supposing that there are N words in C_d, an $N \times n$ embedding matrix of C_d could be constructed, represented as:

$$\prod(C_d) = \Phi(c_1) \oplus \Phi(c_2) \oplus \dots \oplus \Phi(c_N), \quad (1)$$

where $\prod(C_d)$ indicates the $N \times n$ embedding matrix of C_d, and $\Phi(c_z)$ is a word embedding function to map the c_z into an n-dimensional vector, and $\oplus$ is the concatenation operator. The model inputs the word embedding into CNN, and uses convolution layers with filter windows of unigram, bigram and trigram, where the model applies a convolution operation to the inputs. Each convolution filter applies a filter f_j to a window of s words to generate a new feature z_j^k:

$$z_j^k = F\left(\prod(C_d) * f_j + b_j\right), \quad (2)$$

TABLE I. NOTATIONS IN THE FRAMEWORK

Notation	Description
U, T, L, I	user set, things set, location set, and timestamp set.
$r, Γ́, Γ$	the timestamp, the temporal unit, and the temporal state, r∈Γ́∈Γ.
$R_{\|U\|\|T,I\|}$	temporal-user-thing rating matrix, $r_{u,t,r} \in R_{\|U\|\|T,I\|}$.
p_u, $q_{t,r}$	user latent vector, thing latent vector at r.
k	the dimension of the latent vector.
$q^*_{t,r}$	the real-time thing latent vector.
$[y_{u,t,r}]$	the user preference matrix.
[g(u, t, r)]	the user-thing geographical relationship matrix.

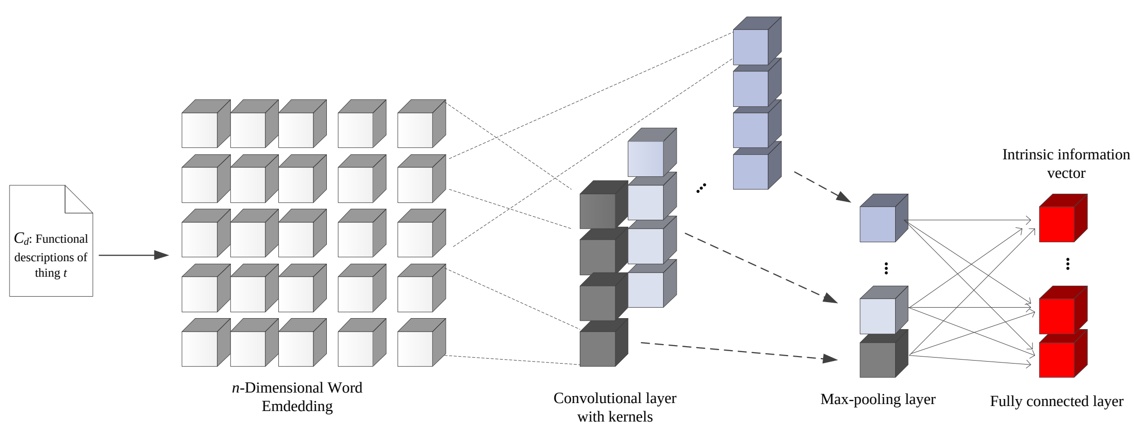

Fig. 4. The representation learning model of the intrinsic information.

where F is the activation function, and inspiring by [40], in Eq. (2), Rectified Linear Units (ReLUs) is used as the activation function. $*$ is the convolution operation, and b_j is a bias term. Furthermore, the model sorts the biggest value from each feature map in the max-pooling layer:

$$\mathcal{L}_\hbar = \{\ell_1, \ell_2, \dots, \ell_{n_f}\}, \hbar \in \{1,2,3\}, \quad (3)$$

where ℓ_j denotes the feature corresponding to filter f_j, and there are n_f filters. For each ℓ_j:

$$\ell_j = max\{z_j^1, z_j^2, \dots, z_j^{N-s+1}\}. \quad (4)$$

Afterward, the max-pooling layer $\mathcal{L}_1$, $\mathcal{L}_2$,and $\mathcal{L}_3$ are concatenated:

$$\mathcal{L} = \mathcal{L}_1 \oplus \mathcal{L}_2 \oplus \mathcal{L}_3. \quad (5)$$

The model feeds $\mathcal{L}$ obtained by max-pooling into the fully connected layer to integrate the intrinsic information embedding vector of the functional features of the thing t $q_t^{intrinsic}$:

$$q_t^{intrinsic} = F(W_{intrinsic} \times \mathcal{L} + B_{intrinsic}), \quad (6)$$

where $W_{intrinsic}$ is the weight corresponding to $\mathcal{L}$, and $B_{intrinsic}$ denotes a bias term. $W_{intrinsic}$, $B_{intrinsic}$, and other parameters in the model are trained together with McEMF via backpropagation. $q_t^{intrinsic}$ is a high-level feature vector combining non-linearly $n_{intrinsic}$ feature vectors of the thing's functions.

C. Real-time Information Embedding Model

$p_u^T q_{t,\text{r}}$ could capture user preference things only based on the past interactive records and could not make use of some instant information to meet user real-time needs. To tackle the issue, the paper proposes an instant information embedding model for recommender in IoT. There are two main kinds of instant information in IoT, namely, two subproblems that need to be solved. One is the thing's availability, which indicates a thing in use or not in use at timestamp *r*, and that is because a thing could not be used by multiple users at the same time in general (e.g., Automatic Teller Machine, ATM). To address the subproblem, the paper firstly employs a particle filtering [41] to track the latest things' availability and continuously refine the functions of things, obtaining the instant information embedding vector of the thing t $q_t^{instant}$ with $n_{instant}$-dimensional features. Then $q_t^{intrinsic}$ and $q_t^{instant}$ are fused into a new $(n_{intrinsic} + n_{instant})$-dimensional vector $V_{t,\text{r}}$:

$$V_{t,\text{r}} = q_t^{intrinsic} \oplus q_t^{instant}. \quad (7)$$

$V_{t,\text{r}}$ is input into the fully connected layer to synthesize a high-level feature vector $q_{t,\text{r}}^*$:

$$q_{t,\text{r}}^* = F(W \times V_{t,\text{r}} + B), \quad (8)$$

where W is the weight corresponding to $V_{t,\text{r}}$ inside layer, and B denotes a bias term and $q_{t,\text{r}}^*$ is a high-level vector including the intrinsic functional features and instant functional features, indicating the real-time features of thing t at timestamp *r*.

Another one is the users' locations, which are used to measure the geographical relationships between users and things, and according to Tobler's First Law of Geography [42]: "near things are more related than distant things". Thus, users intuitively tend to use nearby things. To address this subproblem, the users' locations l_p at timestamp r is tracked and the geographical relationships between users and things geo-tagged l_j' are computed as follows:

$$D_r(\text{u}, \text{t}, \text{r}) = \begin{cases} 1 & , l_p \textit{ and } l_j' \textit{ in same region at } \text{r} \\ 1 + \frac{d(l_p, l_j') + d(c(u), c(t))}{2 \times min} & , otherwise \end{cases} \quad (9)$$

where the paper partitions the regions based on the things' coordinates (longitudes and latitudes). Each region has a region center, and when both user *u* and thing *t* are in the same region, they have the same region center, assuming the geographical relationship as 1. Otherwise, the paper computes the geographical relationship by leveraging two types of distances. $d(l_p, l_j')$ is the distance between user *u* and thing *t*, and $d(c(u), c(t))$ corresponds to the distance between the region centers of user *u* and thing *t*, and min indicates the minimum distance between different region centers. Furthermore, the geographical influences $\text{g}(\text{u}, \text{t}, \text{r})$ is as $\text{g}(\text{u}, \text{t}, \text{r}) \propto \frac{1}{D_r(\text{u}, \text{t}, \text{r})}$. Therefore, the user real-time interactive preference $y_{u,t,\text{r}}$ is given as:

$$y_{u,t,\text{r}} = p_u^T q_{t,\text{r}} \cdot \text{g}(\text{u}, \text{t}, \text{r}). \quad (10)$$

D. Fused Model

In this section, the paper proposes a comprehensive framework based on MF with regularization (McEMF). The user real-time interactive preference $y_{u,t,\text{r}}$ is used to be close to the historical interaction rating $r_{u,t,r}$ for the user preference model. Moreover, the paper employs the Frobenius norm to fuse $q_{t,\text{r}}^*$ into the framework for the influence of the functional features of things, and adopts the regularizations to avoid over-fitting. Specifically, the objective function is as follows:

$$\mathcal{G} = \frac{1}{2} \sum_{\text{u} \in U, \text{t} \in T, \text{r} \in I} (r_{u,t,\text{r}} - y_{u,t,\text{r}})^2 + \frac{\lambda_q}{2} \sum_{t \in T} \sum_{\text{r} \in \Gamma} \sum_{\Gamma \in \Gamma} ||q_{t,\text{r}} - q_{t,\text{r}}^*||^2 + \frac{\lambda_p}{2} \sum_{u \in U} ||p_u||^2 + \frac{\lambda_v}{2} \sum_{t \in T} \sum_{\text{r} \in \Gamma} \sum_{\Gamma \in \Gamma} ||q_{t,\text{r}}||^2, \quad (11)$$

where λ_q is a trade-off parameter, which balances the importance between the user preference model and the real-time functional features of things. When λ_q is equal to zero, the objective function ignores the influence of the functional features, and a bigger λ_q means that $q_{t,\text{r}}^*$ is closer to $q_{t,\text{r}}$. λ_p and λ_v are the regularization coefficients. When λ_p and λ_v are limited to zero, the objective function tends to over-fit the training dataset, and when λ_p and λ_v approach to infinite, it tends to under-fitting. For the framework optimization, the paper uses gradient descent in *u* and *t* to find a minimum of the objective function as follows:

$$\mathrm{u} \leftarrow \mathrm{u} - \delta \cdot \frac{\partial \mathcal{G}}{\partial u} \quad (12)$$

and

$$\mathrm{t} \leftarrow \mathrm{t} - \delta \cdot \frac{\partial \mathcal{G}}{\partial t}, \quad (13)$$

where δ is the learning rate. After optimizing the framework, the user personalized preferences on things in real-time could be found by computing the rating given the interactive records, intrinsic information embedding, and instant information embedding. Then a recommendation list is constructed via sorting the rating for each user at a specific time period.

Time complexity analysis: The time complexity analysis of the framework is divided into three parts. The first one is the time for updating the user and thing latent vectors, which takes O(k2|R|+k3(|U|+|T|)). The second one is the time for updating a CNN embedding things' information, and the complexity is O(nf×N×n×|T|). The third one is the time complexity for updating a fully connected layer, which takes O(|W|×|T|). Therefore, the total time complexity is O(k2|R|+k3(|U|+|T|)+ nf×N×n×|T|+|W|×|T|) for per given data.

IV. Experiments

A. Experimental Setup

1) Dataset: The paper leverages CASAS datasets, which are augmented with the dataset as in [36, 37] collected from a smart home environment for a period of four months. The environment contains six diverse categories: entertainment, office, cooking, transportation, medical, and house appliances. And the users with less than 10-time records and things that are used less than five times are eliminated to reduce abnormal data. To the end, these datasets contain 1020 users, 855 things, and 108650 interactive records, illustrated in Table II. The paper separates the data into 8:2 ratio as training data and testing data, respectively. Then, some abnormal data filtered out is selected to validate the proposed model on the cold-start problem.

TABLE II. Statistics of the Dataset

Dataset	|Users|	|Things|	|Categories|	|Records|	Density
CASAS	228635	10256	6	1586507	6.8×10^{-4}

2) Evaluation metrics: As the proposed model generates a recommendation list for each user in real-time, the users will receive different recommendation results at different time period. The recommendation list is denoted as $\mathcal{Y}_i = \{y_i^1, y_i^2, \dots, y_i^K\}$, where y_i^j is the *j*-th recommended thing in terms of the rating and $i = (u, \mathrm{r})$. Therefore, the paper evaluates the proposed model via leveraging two ranking-based metrics: Recall@*K* and mean reciprocal rank (MRR).

Recall@*K* means the proportion of the things related to the ground-truth things in the top-*K* recommendation results. The Recall@*K* is defined as:

$$\mathrm{Re}call@K = \frac{1}{|S_{test}|}\sum_{i=1}^{|S_{test}|}\frac{|\mathcal{R}_i \cap \mathcal{Y}_i|}{|\mathcal{R}_i|}, \quad (14)$$

where $\mathcal{R}_i$ is the ground-truth things, and S_{test} represents the testing dataset, $\mathcal{R}_i \in S_{test}$.

MRR is a metric for ranking position. It refers to that in the recommendation list, the more things related to the ground-truth things are ranked in the front, the better the recommendation results are. MRR is defined as:

$$\mathrm{MRR} = \frac{1}{|S_{test}|}\sum_{i=1}^{|S_{test}|}\frac{1}{rank\ i}, \quad (15)$$

where $rank\ i$ means the ranking position of the relevant thing in $\mathcal{Y}_i$ found together from $\mathcal{R}_i$ and $\mathcal{Y}_i$ for the first time.

3) Baseline methods: To evaluate the performance of the proposed model, the paper selects the following classical and state-of-the-art methods as the baseline methods:

MF: A classical and popular collaborative filtering method for things of interest recommendation.

STUnion [37]: The method uses the context information to create a spatiotemporal graph and a social graph, which are linearly combined to model the user preference things.

KGE[39]:The method proposes to fuse various things social relationships via graph embedding for enhancing user preference predictions.

SORec[40]: The method proposes to represent richer relationships between users and things by contexts and integrate them in the recommender system.

McEMF: The proposed method.

4) Parameter settings: In McEMF, there are four set parameters, including λ_q, λ_p, λ_v, and δ. The paper uses the grid search method to adjust the parameters and finally sets $\lambda_q = 1$, regularization coefficients $\lambda_p = \lambda_v = 0.1$, and $\delta = 0.01$. In other compared baseline methods, the paper tries the best to ensure that their parameters are consistent with the original papers.

B. Discussion of Baselines

To further elaborate on the contributions and innovations of McEMF, the paper shows the difference between baselines as Table III illustrated. MF is a classical and popular collaborative filtering method without fusing any contexts, and it could only depend on the historical interactions to learn user preferences, which fails to overcome the data sparsity and meet users' real-time demand. The others, such as STUnion, KGE, and SORec, consider intrinsic context factors like spatiotemporal and social information etc, however, they ignore the instant information that can indicate users' real-time demands and fail to achieve the best things recommendation under different scenarios and time. The specific performance evaluation is given in the next section.

C. Experimental Results

Firstly, the paper evaluates the performance of all methods with top-5 recommendation things in different dimensions, *k*={5, 10, 20, 40}. The results are shown in Table IV. McEMF outperforms all other baselines on both Recall@5 and MRR. Specifically, the popular STUnion, KGE, SORec, and the proposed McEMF are superior to the classical MF. Besides, McEMF achieves 51.12% improvement over MF. When k is smaller, the performance of STUnion, KGE, SORec, and McEMF is closer, however, when k is bigger and k=20, the other baselines are inferior to McEMF, obviously. The reason is that McEMF naturally integrates more real-time information into the recommender system, such as the real-time geographical relationships and the real-time availability of things, which play an important role in things of interest recommendation in IoT. There is little difference between KGE and SORec, because both of them fuse multiple intrinsic context information. It can observe that McEMF achieves the best performance when k=20, hence let k=20 in the next experiments.

Then, the paper evaluates the performance of all methods with top-K recommendation things in the dimension k=20. The results are illustrated in Table V. From the results, it can observe that the longer recommendation lists recommenders give, the higher recall recommenders can achieve. McEMF achieves better performance compared with the other baselines on MRR and Recall with different K. Moreover, when K=1, McEMF achieves 29.20% improvements over SORec respectively, which implies McEMF could give the best recommendation result in real-time. With the increase of K, the changing rates of Recall and MRR are slowing down. The reason may be that there is little difference in user preferences for things at the end of the lists when the recommended lists reach a certain length.

Furthermore, the paper evaluates the performance of the proposed McEMF for the cold-start problem. The paper adopts the eliminated data, where the number of the interactive records of a user is n={1, 3, 5}. The results are shown in following Table VI. Obviously, for the cold-start problem, McEMF has outstanding advantages compared with other baselines, which benefits from real-time information embedding. In particular, when n=1, McEMF achieves 32.79% improvement over SORec, respectively. MF that ignores the context information is the worst among all methods.

TABLE III. THE DIFFERENCE BETWEEN BASELINES

Methods	Intrinsic context			Instant context	
	Social	Spatiotemporal	Content	Users' states	Things' states
MF					
STUnion	√	√			
KGE	√	√			
SORec	√	√	√		
McEMF	√	√	√	√	√

TABLE IV. TOP-5 RESULTS OF RECOMMENDATIONS IN DIFFERENT DIMENSIONS

k	metrics	MF	STUnion	KGE	SORec	McEMF
5	Recall@5	0.2524	0.3505	0.3603	0.3617	**0.3603**
	MRR	0.1552	0.2310	0.2412	0.2489	**0.2416**
10	Recall@5	0.2546	0.3505	0.3603	0.3617	**0.3635**
	MRR	0.1587	0.2310	0.2412	0.2489	**0.2485**
20	Recall@5	0.2581	0.3505	0.3603	0.3617	**0.3705**
	MRR	0.1601	0.2310	0.2412	0.2489	**0.2596**
40	Recall@5	0.2577	0.3505	0.3603	0.3617	**0.3690**
	MRR	0.1593	0.2310	0.2412	0.2489	**0.2572**

TABLE V. TOP-K RESULTS OF RECOMMENDATIONS IN THE DIMENSION K=20

metrics	MF	STUnion	KGE	SORec	McEMF
Recall@1	0.0324	0.1206	0.1420	0.1429	**0.1876**
MRR	0.0175	0.0797	0.1248	0.1250	**0.1589**
Recall@5	0.2581	0.3505	0.3603	0.3617	**0.3705**
MRR	0.1601	0.2310	0.2412	0.2489	**0.2596**
Recall@10	0.3150	0.4442	0.4748	0.4756	**0.4903**
MRR	0.2364	0.3526	0.3665	0.3666	**0.3812**
Recall@20	0.3345	0.4608	0.4893	0.4897	**0.5147**
MRR	0.2555	0.3687	0.3815	0.3820	**0.3995**

TABLE VI. GIVEN-N RESULTS OF RECOMMENDATIONS IN THE DIMENSION K=20

n	metrics	MF	STUnion	FST	KGE	SORec	McEMF
1	Recall@5	0.0096	0.0305	0.0770	0.0766	0.0767	**0.1033**
	MRR	0.0042	0.0210	0.0489	0.0482	0.0479	**0.0627**
3	Recall@5	0.0131	0.0354	0.0805	0.0800	0.0801	**0.1094**
	MRR	0.0079	0.0253	0.0525	0.0511	0.0510	**0.0660**
5	Recall@5	0.0168	0.0465	0.1062	0.1053	0.1056	**0.1122**
	MRR	0.0102	0.0313	0.0867	0.0856	0.0854	**0.0977**

Next, the paper examines the sensitivity of McEMF to parameters λ_q, λ_p, and λ_v. The paper conducts three sets of experiments. Fig. 5 shows that the performance is changing with λ_q when fixing $\lambda_p = \lambda_v = 0.1$. It can observe that the performance is sensitive to λ_q. At first, Recall@5 and MRR are increasing with λ_q and then decreasing after $\lambda_q = 1$. Therefore, it is believed that $\lambda_q = 1$ is the optimal setting. When λ_q is smaller, McEMF will degenerate into the classical MF as expected. Fig. 6 shows that the performance is changing with λ_p when fixing $\lambda_q = 1$ and $\lambda_v = 0.1$. It can observe that both Reacll@5 and MRR achieve the best performance somewhere near $\lambda_p = 0.1$. Fig. 7 illustrates that the performance is changing with λ_v when fixing $\lambda_q = 1$ and $\lambda_p = 0.1$. It can observe that both Reacll@5 and MRR achieve the best performance somewhere near $\lambda_v = 0.1$.

To evaluate the impact of the instant contextual embedding and the intrinsic information embedding to the proposed McEMF, the paper excludes the intrinsic information embedding from the model, denoted as McEMF/intrinsic, the instant contextual embedding from the model, denoted as McEMF/instant, and both the instant contextual embedding and the intrinsic information embedding from the model, denoted as McEMF/instant_intrinsic, respectively. The comparison results are shown in Fig. 8. McEMF outperforms McEMF/intrinsic, McEMF/instant, and McEMF/instant_intrinsic, which indicates both the instant contextual embedding and the intrinsic information embedding effectively promote the recommendation effects. Furthermore, McEMF/intrinsic is superior to McEMF/instant, which implies that the real-time information play a more important role than the intrinsic information in IoT scenarios.

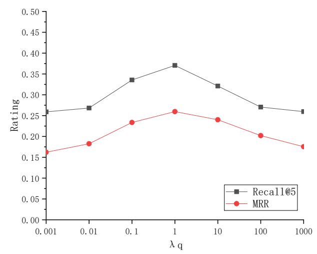

Fig. 5. Recall@5 and MRR for different λ_q.

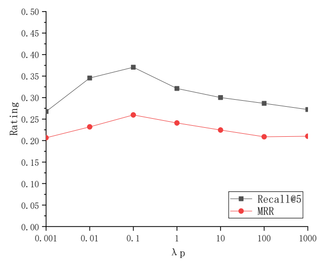

Fig. 6. Recall@5 and MRR for different λ_p.

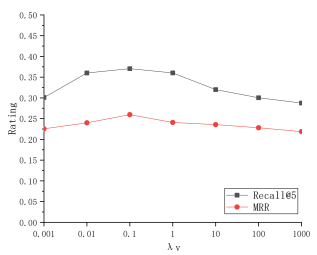

Fig. 7. Recall@5 and MRR for different λ_v.

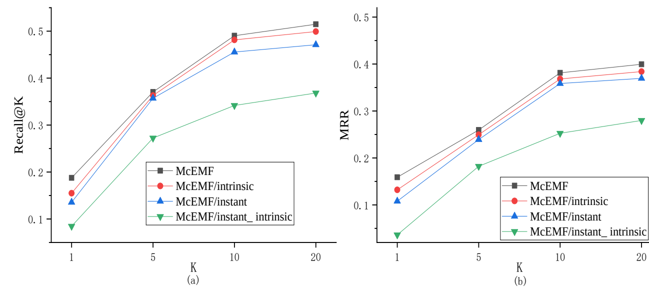

Fig. 8. The comparison of the impacts of the instant contextual embedding and the intrinsic information embedding to McEMF.

The paper uses the traditional Control Variate Technique to adjust the hyper-parameters of CNN in McEMF, as shown in Table VII. As it can see, the optimal combination of the parameters is Cross features + Hidden layers (512-256-128-64-1, ReLUs, Normal initializer, Dropout). From the results, the following conclusions can be inferred. (1) The number of neural units and hidden layers are not the most crucial factors for recommendation performance, and less hidden layers could avoid over-fitting. (2) Both activation function and initializer play an important role, because a proper activation or initializer can adjust the data distribution fed in each layer well. (3) Dropout and L2 regular are widely used to avoid over-fitting, and in the experiments, Dropout has more advantages.

TABLE VII. THE HYPER-PARAMETERS ADJUSTMENT OF CNN IN MCEMF

Hyper-parameters of hidden layers		Reacll@1	MRR	Recll@5	MRR	Recall@10	MRR	Recall@20	MRR
Activation Functions	Tanh	0.1335	0.1127	0.3116	0.2003	0.4289	0.3210	0.4478	0.3286
	ReLUs	0.1876	0.1589	0.3705	0.2596	0.4903	0.3812	0.5147	0.3995
Initializers	Uniform	0.1606	0.1325	0.3412	0.2294	0.4622	0.3565	0.4840	0.3757
	Normal	0.1876	0.1589	0.3705	0.2596	0.4903	0.3812	0.5147	0.3995
Number of layers and neural units	2048-1024-512-256-1	0.1822	0.1489	0.3677	0.2459	0.4882	0.3779	0.5002	0.3898
	1024-512-256-128-1	0.1851	0.1516	0.3658	0.2450	0.4896	0.3785	0.5010	0.3923
	512-256-128-64-1	0.1876	0.1589	0.3705	0.2596	0.4903	0.3812	0.5147	0.3995
Regular Terms	L2reg	0.1821	0.1514	0.3894	0.2555	0.4891	0.3801	0.5112	0.3973
	Dropout	0.1876	0.1589	0.3705	0.2596	0.4903	0.3812	0.5147	0.3995

V. CONCLUSION

The user-thing interactions in real IoT scenarios are dynamic and sparse, which is a challenge for things of interest recommendations. To solve the challenge, the paper proposes a recommendation model with multi-dimensional context embedding, which learns the user preferences in real-time by fusing the instant information and the intrinsic information into the matrix factorization framework. In the proposed model, the paper employs CNN to model the functional features of things. As conventional rating matrices fail to represent the temporal features, the paper proposes a temporal-user-thing rating matrix, which is used to model user preference via integrating the instant geographical information. The paper evaluates the performance of the proposed model through experiments on real-world IoT datasets. The paper compares the model with other state-of-the-art methods on three sets of experiments, as Table III, Table IV, and Table V. The results demonstrate that the effectiveness and efficiency of the proposed model.

Though the proposed model achieves an improvement for things recommendations in IoT, its performance is still limited by the interaction data sparsity. Therefore, the future work will focus on investigating a novel method against data sparsity.

ACKNOWLEDGEMENT

Key Project of Guangdong University of Science and Technology IoT Engineering Collaborative Innovation Center under grant GKY-2019CQYJ-9.

REFERENCES

[1] S. Aramuthakannan, M. R. Devi, S. Lokesh, R. Manimegalai, "Movie recommendation system via fuzzy decision making based dual deep neural networks," *J. Intell. Fuzzy Syst.*, vol.44, no.3, pp. 5481-5494 2023.

[2] Y. Liu, J. Miyazaki,"Knowledge-aware attentional neural network for review-based movie recommendation with explanations," *Neural Comput. Appl.*, vol.35, no.3, pp.2717-2735, 2023.

[3] Z.Z.Darban, M.H.Valipour, "GHRS: Graph-based hybrid recommendation system with application to movie recommendation," *Expert Syst. Appl.*, vol.200, pp. 116850, 2022.

[4] W. Sun, J.H.Jiang, Y.B. Huang, J.L. Li, Mengmeng Zhang, "An Integrated PCA-DAEGCN Model for Movie Recommendation in the Social Internet of Things," *IEEE Internet Things J.*, vol.9, no.12, pp. 9410-9418, 2022.

[5] Z.Y. Liu, W. Xu, W.P. Zhang, Q.Q. Jiang, "An emotion-based personalized music recommendation framework for emotion improvement," *Inf. Process. Manag.*, vol.60, no.3, pp. 103256, 2023.

[6] J. Yi, Y.C. Zhu, J.Y. Xie, Zhenzhong Chen, "Cross-Modal Variational Auto-Encoder for Content-Based Micro-Video Background Music Recommendation," *IEEE Trans. Multim.*, vol.25, pp.515-528, 2023.

[7] H. Weng, J.J. Chen, D.J. Wang, X. Zhang, D.J. Yu, "Graph-Based Attentive Sequential Model With Metadata for Music Recommendation," IEEE Access, vol.10, pp.108226-108240, 2022.

[8] W.J. Chang, D. Sun, Q.D. Du, "Intelligent Sensors for POI Recommendation Model Using Deep Learning in Location-Based Social Network Big Data," *Sensors*, vol.23, no.2, pp. 850, 2023.

[9] Z.Dong, X.W. Meng, Y.J. Zhang, "Exploiting Category-Level Multiple Characteristics for POI Recommendation," IEEE Trans. Knowl. Data Eng., vol.35, no.2, pp. 1488-1501, 2023.

[10] J.F. Fang, X.F. Meng, X.Y. Qi, "A top-k POI recommendation approach based on LBSN and multi-graph fusion," *Neurocomputing*, vol.518, pp. 219-230, 2023.

[11] M. Ye, P.F. Yin, W.C. Lee, and D.L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," *in Proc. 34th Int. Conf., ACM SIGIR.*, 2011, pp. 325-334.

[12] Q. Yuan, G. Cong, Z.Y. Ma, A.X. Sun, and N.M. Thalmann, "Time-aware point-of-interest recommendation," *in Proc. 36th Int. Conf., ACM SIGIR.*, 2013, pp. 363-372.

[13] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation," *in Proc. 20th Int. Conf., ACM SIGKDD.*, 2014, pp. 831-840.

[14] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," *in Proc. 19th Int. Conf., ACM SIGKDD.*, 2013, pp. 1043-1051.

[15] Y. Liu, W. Wei, A. Sun, and C. Miao, "Exploiting geographical neighborhood characteristics for location recommendation," *in Proc. 23rd Int. Conf., ACM CIKM.*, 2014, pp. 739-748.

[16] S. Zhang, W.H. Wang, J. Ford, F. Makedon, and J.D. Pearlman, "Using singular value decomposition approximation for collaborative filtering," *in Proc. 7th Int. Conf., IEEE Computer Society* (*CEC*), 2005, pp. 257-264.

[17] Y. Koren, "Collaborative filtering with temporal dynamics," *in Proc. 15th Int. Conf., ACM SIGKDD.*, 2009, pp. 447-456.

[18] X. Luo, M.C. Zhou, S. Li, and M.S. Shang, "An inherently non-negative latent factor model for high-dimensional and sparse matrices from industrial applications," *IEEE Trans. Ind. Inf.*, vol. 14, no. 5, pp. 2011–2022, 2018.

[19] X. Luo, M.C. Zhou, S. Li, Z.-H. You, Y. Xia, and Q.S. Zhu, "A non-negative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method," *IEEE Trans Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 524–537, 2016.

[20] X. Luo, M.C. Zhou, Y. Xia, and Q.S. Zhu, "An efficient non-negative matrix factorization-based approach to collaborative-filtering for recommender systems," *IEEE Trans. Ind. Inf.*, vol. 10, no. 2, pp. 1273–1284, 2014.

[21] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," *in Proc. 25th Int. Conf., ACM ICML.*, 2008, pp. 880-887.

[22] X. Luo, M.C. Zhou, S. Li, Y. Xia, Z.-H. You, and Q.S. Zhu, "Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing QoS data," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1216–1228, 2018.

[23] X. Luo, M.C. Zhou, S. Li, Z.-H. You, Y. Xia, Q.S. Zhu, H. Leung, "An efficient second-order approach to factorizing sparse matrices in recommender systems," *IEEE Trans. Ind. Inf.*, vol. 11, no. 4, pp. 946–956, 2015.

[24] S.N. Xing, F.A. Liu, Q.Q. Wang, X.H. Zhao, and T.L. Li, "Content-aware point-of-interest recommendation based on convolutional neural network," *Appl. Intell.*, vol. 49, no. 3, pp. 858-871, 2019.

[25] D.Q. Yang, D.Q. Zhang, Z.Y. Yu, and Z. Wang, "A sentiment-enhanced personalized location recommendation system," i*n Proc. 24th Int. Conf., ACM HT.*, 2013, pp. 119-128.

[26] J.D. Zhang and C.Y. Chow, "GeoSoCa: Exploiting Geographical, Social and Categorical Correlations for Point-of-Interest Recommendations," *in Proc. 38th Int. Conf., ACM SIGIR.*, 2015, pp. 443-452.

[27] G.H. Li, Q. Chen, B.L. Zheng, N.Q.V. Hung, P. Zhou, and G.F. Liu, "Time-aspect-sentiment Recommendation Models Based on Novel Similarity Measure Methods," *ACM Trans. Web*, vol. 14, no. 2, pp. 5:1-5:26, 2020.

[28] Z.Y. Zhang, Y. Liu, Z.J. Zhang, and B. Shen, "Fused matrix factorization with multi-tag, social and geographical influences for POI recommendation," *World Wide Web*, vol. 22, no. 3, pp. 1135-1150, 2019.

[29] X. Xiong, S.J. Qiao, Y.Y.Li, N. Han, G. Yuan, and Y.Q. Zhang, "A point-of-interest suggestion algorithm in Multi-source geo-social networks," *Eng. Appl. Artif. Intell.*, vol. 88, pp. 1-11, 2020.

[30] M. Davtalab and A. A. Alesheikh, "A POI recommendation approach integrating social spatio-temporal information into probabilistic matrix factorization," *Knowl. Inf. Syst*., vol.63, no.1, pp. 65-85, 2021

[31] Sh.L. Li, J.B. Zhou, T. Xu, H. Liu, X.J. Lu, and H. Xiong, "Competitive Analysis for Points of Interes," *in Proc. 26th Int. Conf., ACM SIGKDD.*, 2020, pp. 1265-1274.

[32] L. Gao, Y.H. Li, R.X. Li, Z.L. Zhu, X.W. Gu, and O. Habimana, "ST-RNet: A Time-aware Point-of-interest Recommendation Method based on Neural Network," *in Proc. 2019 Int. Conf., IEEE IJCNN*., 2019, pp. 1-8.

[33] .H.X. Hu, Zh.W. Jiang, Y. Zhang, H. Wang, and W. Wang, "Network Representation Learning-Enhanced Multisource Information Fusion Model for POI Recommendation in Smart City," *IEEE Internet Things J.*, vol.8, no.12, pp.9539-9548, 2021.

[34] Y.F. Zhang, M. Zhang, Y. Zhang, G.K Lai, Y.Q. Liu, H.H. Zhang, and S.P Ma, "Daily-Aware Personalized Recommendation based on Feature-Level Time Series Analysis," *in Proc. 24th Int. Conf., World Wide Web*, 2015, pp. 1373-1383.

[35] X.R. Wang, A. McCallum, "Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends," *in Proc. 12th Int. Conf., ACM SIGKDD.*, 2006, pp. 424-433.

[36] L. Yao and Q.Z. Sheng, "Exploiting latent relevance for relational learning of ubiquitous things," *in Proc. 21st Int. Conf., ACM CIKM.*, 2012, pp. 1547-1551.

[37] L. Yao, Q.Z. Sheng, B. J. Gao, A. H. H. Ngu, and X. Li, “A Model for Discovering Correlations of Ubiquitous Things,” *in Proc. 13th Int. Conf., IEEE Computer Society* (*CIDM*), 2013, pp. 1253-1258.

[38] Y.Y. Chen, M.X. Zhou, Z.W. Zheng, D. Chen, “Time-Aware Smart Object Recommendation in Social Internet of Things,” *IEEE Internet Things J.*, vol. 7, no.3, pp. 2014-2027, 2020.

[39] P. Mahajan, P.D. Kaur, “Smart object recommendation (SORec) architecture using representation learning in Smart objects-Based Social Network (SBSN),” *J. Supercomput.* vol.77, no.12, pp.14180-14206, 2021.

[40] V. Nair and G.E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” *in Proc. 27th Int. Conf., ACM ICML.*, 2010, pp. 807-814.

[41] L. Yao, Q.Z. Sheng, “Particle filtering based availability prediction for web services,” *in Proc. 9th Int. Conf., Springer ICSOC.*, 2011, pp. 566-573.

[42] W. R. Tobler, “A Computer Movie Simulating Urban Growth in the Detroit Region,” *Economic Geography*, vol. 46, pp. 234-240, 1970.