Future Generation Computer Systems 137 (2022) 173–185

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Review article

Modeling feature interactions for context-aware QoS prediction of IoT services

Yuanyi Chen [a,*], Peng Yu [a,b], Zengwei Zheng [a], Jiaxing Shen [c], Minyi Guo [d]

[a] School of Computer and Computing Science, Zhejiang University City College, China
[b] College of Computer Science and Technology, Zhejiang University, China
[c] Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[d] Department of Computer Science and Engineering, Shanghai Jiao Tong University, China

ARTICLE INFO

Article history:
Received 8 December 2020
Received in revised form 16 July 2022
Accepted 20 July 2022
Available online 25 July 2022

Keywords:
IoT services
Deep neural network
Feature interaction
Context-aware qoS prediction

ABSTRACT

Accurate Quality of Service (QoS) prediction of service is a key measure to accomplish successful applications such as QoS-aware service recommendation and composition in Internet of Things (IoT) environments. The key of this task is to consider contextual information like geographic location, network address and type of service, since they have subtle but powerful effects on QoS of IoT services. Recently proposed context-aware QoS prediction for IoT services follow two general paradigms: clustering contextual information for calculating similarity between users and services or integrating contextual information by extending latent factor models. However, the simple clustering contextual information or learning the latent feature of contextual information do not go much further to discover complex and intricate user–service interaction patterns. To this end, we propose a context-aware feature interaction modeling (CFM) approach for IoT services to perform QoS prediction, considering context as an additional feature similar to users and services and modeling their interactions.

The proposed method can capture both low-order and high-order feature interactions using contextual information and user's invoked records, which consists of three phases: (1) learn low-order feature interactions by decomposing the sparse user–service QoS matrix with factorization machine; (2) learn high-order feature interactions explicitly and implicitly with a multilayer perceptron and deep cross network; (3) aggregate the output of low-order and high-order feature interactions with a parametric-matrix-based fusion. Experimental results on a large-scale QoS dataset demonstrate that the proposed method consistently outperforms state-of-the-art baselines in terms of various evaluation metrics.

Contents

* Corresponding author.
E-mail addresses: chenyuanyi@zucc.edu.cn (Y. Chen), pyu@zju.edu.cn (P. Yu), zhengzw@zucc.edu.cn (Z. Zheng), jiaxshen@comp.polyu.edu.hk (J. Shen), guo-my@cs.sjtu.edu.cn (M. Guo).

https://doi.org/10.1016/j.future.2022.07.017
0167-739X/© 2022 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Motivation

As the fundamental building block of the Internet of Things (IoT), smart objects are defined as autonomous physical/digital objects augmented with sensing, processing and network capabilities [1]. The IoT paradigm aims to integrate physical world and the cyber world by interconnecting a tremendous number of smart objects [2] and facilitate numerous intelligent applications such as human activity and acoustic gesture recognition [3,4], context-aware workflow management [5], WiFi-based group detection [6,7], mobile crowdsensing [8] and blockchain-based mobile transaction databases [9,10]. How to effectively manage these massive and heterogeneous smart objects has become one of the fundamental tasks of the IoT, which has also become a hot research issue in both academia and industry [11]. One of the most promising technologies is service-oriented architecture (SOA) since it can achieve IoT system interoperability [12]. With the SOA-based service model, IoT applications can be implemented by combining various services instead of managing changes in sensors and controllers. Given the rapidly increasing number of IoT services, how to quickly and effectively find the appropriate service that meets the needs of developers and users from such a large collection of IoT services has become a challenging problem., To address this challenge, a few studies [13–15] propose service search or recommendation techniques to discover the most suitable service among candidates with similar or the same functions. Unfortunately, service selection or recommendation may fail to find services that meet user needs since the Quality of Service (QoS) (e.g., response time and throughput) change frequently in IoT. Therefore, QoS prediction of IoT service is the essential task for safely discovering services and objects based on users' preference [9]. However, QoS prediction in IoT environment is even more challenging and context dependent compared with traditional web QoS prediction due to some unique characteristics of IoT service.

- **Data sparsity due to few historical records of IoT service.** Users usually only invoke limited services in IoT environments, which results in high sparse user–service invocation matrix. This can be always observed in QoS prediction, for instance, the density for the well-known QoS dataset WS-DREAM [16]. In addition, QoS invoked records between users and services are difficult to collect due to privacy and security issue. Given such an extreme sparsity user–service invocation matrix, existing IoT QoS prediction methods are easily over-fitted and suffer from high computational complexity.
- **Contextual information plays an important role in QoS prediction of IoT service.** Several studies [17–19] demonstrate context information has a great impact on IoT QoS prediction. In general, users and services with similar latent factors tend to receive a similar QoS, so jointly considering contextual information of users and services is promising in improving QoS prediction performance in IoT environment.

Recently, many QoS prediction methods aim to leverage user-related and service-related contextual factors for QoS prediction of IoT services can be mainly divided into two groups: (1) Utilizing K-means or fuzzy clustering algorithm to cluster contextual information for calculating user's similarity to improve collaborative filtering based QoS prediction model [19,20]; (2) Integrating contextual information by extending latent factor models by regarding contexts as additional features of users and services (e.g., matrix factorization (MF) [18,21–23] and factorization machine (FM) [23].), which can capture the interaction pattern between contexts and users/services. However, these approaches suffer from a number of limitations, for example, existing methods based on matrix factorization or factorization machine perform low-order context feature interaction since they are linear and have shallow structures, thus cannot learn nonlinear and complex relationships within an extremely sparse QoS data set in IoT environment, thus resulting in poor QoS prediction performance. Another challenge of these methods is the dimension of the input context features will increase exponentially when enumerating all the possible combinatorial features, thus greatly increases model complexity and leads to overfitting. To address these challenges, several deep learning models [24,25] have been proposed for IoT QoS prediction inspired by the success of deep neural networks in computer vision [26] and speech recognition [27]. In these deep learning based models, raw datasets are directly fed to neural networks to learn combinatorial features. Such kinds of methods will suffer from two limitations: Firstly, most context factors in IoT QoS prediction are multi-field and discrete categorical features, thus leading combinatorial features space are more sparse than the raw features; Secondly, training deep neural networks on such a large feature space requires tuning a huge number of parameters, which is difficult to learn them effectively.

1.2. Problem statement

In this subsection, we focus on the problem formulation of context-aware QoS prediction of IoT service by firstly illustrating a set of formal definitions.

Definition 1 (***Service User***)**.** A service user can be described as a two-tuple $\langle u, c_u\rangle$, where u is the identity label of service user and c_u is the context factors set of user u (e.g., gender, country and network).

Definition 2 (***IoT Service***)**.** A IoT service can be described as $\langle s, c_s\rangle$, where s is the identifier of service and c_s is the context factors set of service s (e.g., location, service Provider and IP address).

Given a user set $U = \{u_1, u_2, \dots, u_M\}$ and a service set $S = \{s_1, s_2, \dots, s_N\}$ contains M users and N services respectively. Each user may make calls to some of these services, resulting in a record of real-world QoS evaluation results, such as response time, throughput, and failure rate. Let matrix $R^{M\times N}$ denote the QoS values generated by these M users call records for N services, where the element r_{ij} represents the QoS measurement when user u_i invoke service s_j. In a real scenario, a user will only access few services, which leads to numerous missing values in the QoS matrix R. The problem of QoS prediction in IoT aims to forecast these missing QoS values $R^{M\times N}$, to facilitate service selection and recommendation in IoT environment.

1.3. Our contributions

In light of the preceding challenges in existing IoT QoS prediction methods, we present a context-aware prediction model for IoT service by modeling both low-order context feature interactions and high-order context feature interactions simultaneously. Firstly, our model learns low-order feature interactions with FM, which can capture the linear and pairwise interactions between features with addition and inner product operations. Secondly, our model learns high-order feature interactions with a multilayer perceptron (MLP) [28] and deep cross network (DCN) [29]. The MLP and DCN can model implicit and explicit higher-order feature interactions respectively. For reducing the dimension of raw input features, we firstly learn a low-dimensional distributed representation of the input feature with an embedding layer. Then, we feed the embedding vector of sparse features into MLP and DCN to model high-order feature interactions. The principle underlying our approach is to capture both memorization and generalization by jointly considering low-order feature interactions and high-order feature interactions. Memorization means the capability of catching the direct features from historical data and paying attention to the historical behavior of the user, which is achieved by learning low-order feature interactions. Meanwhile, generalization means the capability of producing more general and abstract representations for users or services without historical records, which is achieved by learning high-order feature interactions. In a nutshell, our contributions are three-fold.

- We propose a context-aware QoS prediction model for IoT service with jointly modeling low-order and high-order context feature interactions. The invocation interactions between users and services are seamlessly integrated into factorization machine and deep neural network (i.e., DCN and MLP), which can be used to characterize complex relations between users and services within an extremely sparse interaction matrix.
- We co-train a factorization machine model and two deep neural networks: multilayer perceptron and deep cross network, to embed both implicit and explicit user/service features and relations towards a comprehensive representation of both low-order and high-order user–service interactions.
- We conduct a series of experiments using a large-scale QoS dataset to evaluate the proposed approach. Experimental results demonstrate that our approach can achieve better QoS prediction performance in IoT environment than several state-of-the-art methods.

The remainder of the paper is organized as follows: Section 2 surveys related work on QoS prediction of IoT services. Section 3 describes the proposed context-aware QoS prediction model for IoT service in detail. Section 4 reports and discusses the experimental results. Finally, we present our conclusion and future work in Section 5.

2. Related work

QoS prediction in IoT environment has attracted an enormous amount of research from both industry and academia, as its great importance to successful recommendation and composition of IoT services. The previous efforts to tackle IoT QoS prediction problem can be mainly divided into two groups: (1) Memory-based CF models and (2) Model-based CF models. This section reviews these mainstream methods.

Collaborative filtering (CF) is the most commonly used method in QoS prediction in IoT inspired from recommender systems [17,21,22,30–32]. The rationale is that users or services with similar context factors, such as geographic location and service type, tend to observe similar quality scores on the same service. Therefore, these CF-based QoS prediction methods can automatically predict personalized QoS values for current users with given active users and services. CF-based methods exploit crowd intelligence to facilitate QoS prediction, which can save time and effort for both service providers and users. Generally, existing CF-based methods consist of memory-based CF and model-based CF.

Memory-based CF methods use similar users or services to predict target QoS values. It can be classified into three types, user-based [30], service-based [17] and a combination of both [22], depending on different similarity calculation perspectives. For example, the work [17] presents a deviation-based neighborhood model for context-aware IoT QoS prediction. Their proposed method includes two phases: (1) Perform a baseline estimate for QoS prediction based on service's and user's deviations; (2) Fine-grained adjustments of the predictions with item-based CF. Specifically, the deviation-based model can efficiently optimize global model parameters. In [31], the QoS of IoT service is predicted by initially calculating user or edge server similarity and selecting top-K most-similar neighbors. Since it is difficult to find similar neighbors when user–service rating matrix is very sparse, the study in [31] firstly utilizes user-mean method to obtain essential prediction accuracy, then improves the QoS prediction with considering the historical response time variation of IoT services observed by different users. For finding the relevant QoS values from similar users/services in known QoS invoked records, the work [30] exploits Pearson Correlation Coefficient (PCC) to calculate the user/service's similarity. Then, using least mean square algorithm to analyze the hidden relationships between all the known QoS data and corresponding QoS data with the highest similarities, and finally predicting IoT QoS based on the derived coefficients. For predicting response time of IoT services, [33] proposed a new method to calculate service response time similarity and adjust initial similarity values for selecting similar neighbors based on a similarity threshold. Finally, the proposed method predicts QoS values based on neighborhood-based CF with a densified user–service rating matrix. Even though neighborhood-based CF are simplicity and efficiency, it will suffer from a few limitations, such as the large data set in the grading matrix and the data sparsity problem, since the IoT QoS records are usually sparse thus may lead to poor quality of similar neighbors identification.

Model-based CF methods can effectively address the challenge of sparse data that cannot be solved by neighbor-based CF for IoT QoS prediction [18,21–23,32], which employs matrix factorization to train a predefined model using historical user–service rating datasets. Specifically, matrix factorization decomposes the user–service rating matrix into the product of two lower dimensionality rectangular matrices, which can alleviate the problem of data sparsity. For instance, the work [21] exploits Pearson Correlation Coefficient to calculate the similarities of latent features for different users and services. The work [32] demonstrates model-based CF can be used in a self-managing, goal-driven service model for QoS prediction in the IoT. In [22], an ensemble model that combines the model-based CF and neighborhood-based CF are proposed, which consists of two stages: (1) utilizing an improved auto-encoder to alleviate the cold-start problem by pre-computing an estimate of the missing QoS values; (2) exploiting a novel computation method based on Euclidean distance to

Table 1
Notations used in the paper.

Symbol	Description
U, S, R	The set of users, services, QoS values
M, N	The number of users and services
$\boldsymbol{X_u}, \boldsymbol{X_s}$	The one-hot encoding vector of user u and service s
r_{ij}	The known quality-score observed by user u_i on service s_j
Θ	The model parameters of factorization machine
K	The size of embedding vector
P, Q	The embedding matrix for users and services
E_u, E_s	The embedding vector of user u and service s
$\boldsymbol{z_0}$	The output of embedding and stacking layer
L_1, L_2	The network depth of MLP and DCN network
$\boldsymbol{w_i^m}$, $\boldsymbol{b_i^m}$, $\kappa(\cdot)$	The weight matrix, bias, activation function for the ith layer of MLP
$\boldsymbol{w_i^d}$, $\boldsymbol{b_i^d}$	The weight matrix and bias for the ith layer of DCN
h_i, c_i	The output of the ith layer of MLP and DCN
$\boldsymbol{\pi^{FM}}$, $\boldsymbol{\pi^{MLP}}$, $\boldsymbol{\pi^{DCN}}$	The weight vector of the pre-trained FM, MLP and DCN model
$\boldsymbol{\Phi^{FM}}$, $\boldsymbol{\Phi^{MLP}}$, $\boldsymbol{\Phi^{DCN}}$	The output of FM, MLP and DCN model
I_{ij}	An indication function that is equal to 1 if user u_i invokes service s_j
α_1, α_3, α_3	The trade-off hyperparameters of FM, MLP and DCN model.
σ, π^T	The activation function and edge weights of the fusion layer.
y_{us}	The QoS prediction value when user u invokes service s

address the overestimation problem. The work [23] incorporates user's location to better fit the IoT mobile environments with factorization machines. To exploit context factors for further improving IoT QoS prediction, [18] firstly optimizes the SVM with an improved artificial bee colony algorithm, then utilizes the optimized SVM to predict the workload of IoT services. The above literature show model-based CF methods can achieve higher performance than memory-based CF methods in the case of high data sparsity.

Unfortunately, model-based CF methods can only generate low-order feature interaction between user's latent features and service's latent features, and the inner product cannot model high-order feature interaction. Therefore, a few studies based on deep neural networks have been proposed for QoS prediction in IoT by modeling high-order feature interaction [19,24,25], such as auto-decoder [24] and neural collaborative filtering [19]. To predict communication delay of IoT service, the work [25] adopts a deep neural network (DNN) to model the relationship between diverse communication parameters (e.g., queue size, application traffic rate and transmission power and delay). For reducing training and request time of QoS prediction, the study in [24] presents a stacked auto-encoder with dropout on a deep edge architecture. To leverage both local and global features for IoT QoS prediction, the work [19] develops a holistic framework based on neural collaborative filtering (NCF) and fuzzy clustering.

However, it is difficult to train large DNN models with numerous parameters due to the exponential growth in the dimensionality of the combined features caused by the extreme sparsity and high-dimensional features of the IoT QoS dataset. Inspired by these studies, we propose a hybrid model for QoS prediction in IoT by modeling both low-order and high-order feature interactions simultaneously, which aims to combine the memory ability of the linear model and the generalization ability of the DNN model. For reducing the dimension of the raw datasets and making different types of features to interact with each other, we learn low-dimensional distributed representation of the raw sparse datasets with an embedding layer.

Our proposed approach differs from the above-mentioned works in the following three aspects: (1) We investigate that the key challenge of addressing data sparsity of IoT QoS prediction is to effectively model feature interactions. Then we propose a hybrid model for QoS prediction in IoT by modeling both low-order and high-order feature interactions simultaneously, which aims to combine the memory ability of the linear model and the generalization ability of the DNN model; (2) For reducing the dimension of the raw datasets and making different types of features to interact with each other, we learn low-dimensional distributed representation of the raw sparse datasets with an embedding layer; (3) We generate a real-world benchmark dataset for IoT QoS prediction by adding heterogeneous traffic data to the public WSDream dataset with the HetHetNets traffic model.

3. The proposed QoS prediction model in IoT

In this section, we detail the proposed hybrid IoT QoS prediction model. Fig. 1 shows a detailed workflow of the IoT service QoS prediction process in our method, and the steps include: (1) Collecting observed QoS data from user's service invocation records. When users invoke working services, we can collect their QoS values by providing them from users, and keep this data in reserve in our prediction server; (2) Feature representation of target users and services. In most IoT QoS prediction tasks, the context factors of users and services (such as user's gender and service's provider) are collected in a multi-field categorical form, so that each data instance is normally transformed into a high-dimensional sparse (binary) vector via one-hot encoding, an example is shown in Fig. 2; (3) Generate QoS prediction value by the proposed hybrid model. As stated in Section 1, generating new features from raw features helps improve the performance of QoS prediction. To achieve this goal, the proposed hybrid model designs a proper neural network structure and FM to identify useful feature interactions then generate new features automatically.

Specifically, the proposed hybrid model performs QoS prediction by learning both low-order feature interactions and high-order interactions, which includes three components (as shown in Fig. 3): (1) the FM component learns a latent representation of users and services and models low-order interactions between them by decomposing the user–service QoS matrix; (2) the deep learning component models high-order interactions between users and services by firstly learning low-dimensional embedding representation of sparse user and service with an embedding layer, followed by stacking the embedding representation of users and services into one vector, then feeds the concatenated vector into a multilayer perception and deep cross network for learning high-order feature interactions; (3) to combine both low-order and high-order feature interactions, the outputs of FM component and deep learning component are aggregated with a parametric-matrix-based fusion to generate the ultimate QoS prediction. For ease of the following presentation, we define the key data structures and notations used in the proposed method. Table 1 lists the relevant notations used in this paper.

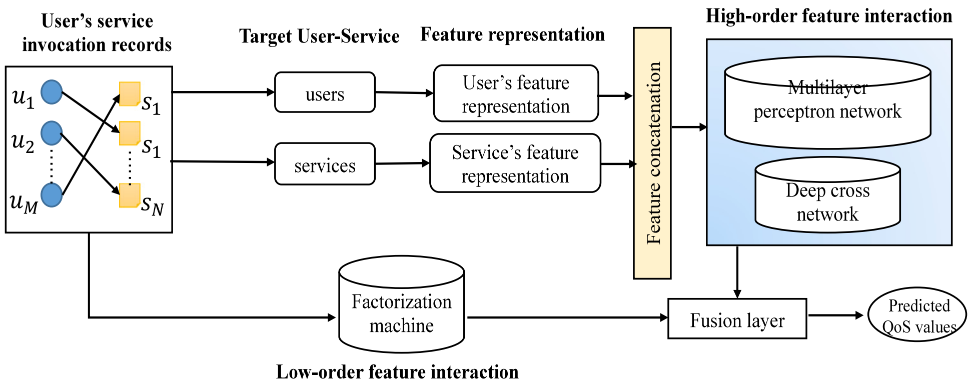
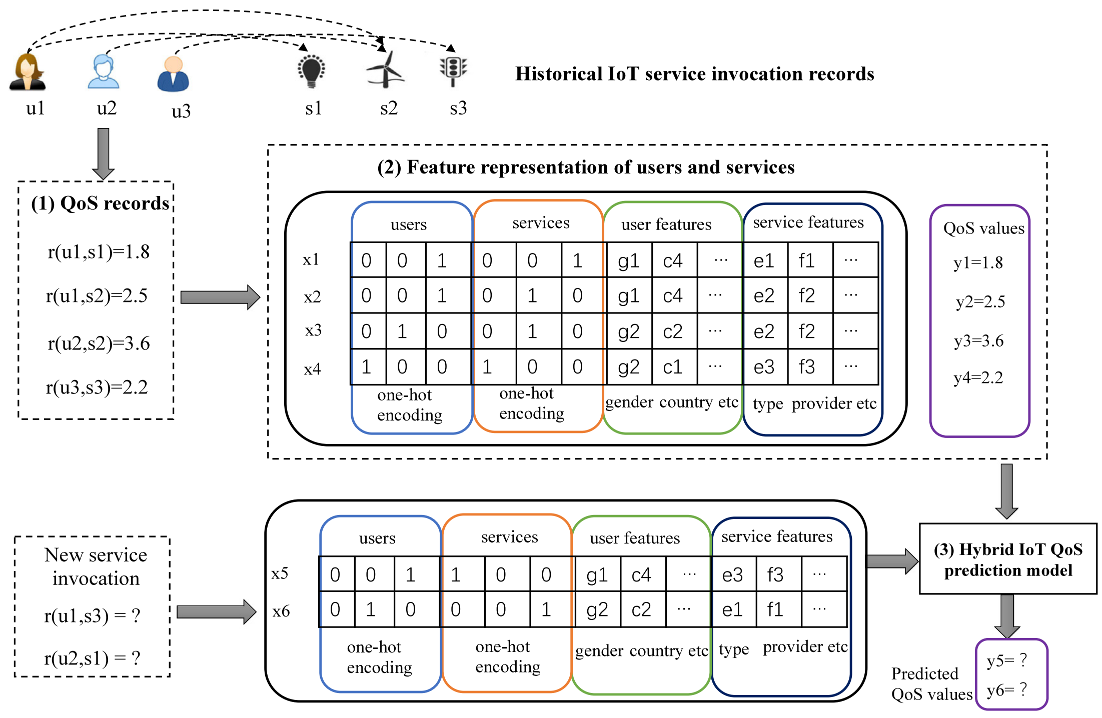

User's service invocation records
u_1
u_2
u_M
s_1
s_1
s_N
Target User-Service
users
services
Feature representation
User's feature representation
Service's feature representation
Feature concatenation
High-order feature interaction
Multilayer perceptron network
Deep cross network
Factorization machine
Low-order feature interaction
Fusion layer
Predicted QoS values

Fig. 1. The workflow of the proposed hybrid IoT QoS prediction model.



	users			services			user features			service features		
x1	0	0	1	0	0	1	g1	c4	···	e1	f1	···
x2	0	0	1	0	1	0	g1	c4	···	e2	f2	···
x3	0	1	0	0	1	0	g2	c2	···	e2	f2	···
x4	1	0	0	1	0	0	g2	c1	···	e3	f3	···

	users			services			user features			service features		
x5	0	0	1	1	0	0	g1	c4	···	e3	f3	···
x6	0	1	0	0	0	1	g2	c2	···	e1	f1	···

Fig. 2. An example of the proposed hybrid model for IoT QoS prediction.

3.1. Modeling low-order feature interaction with factorization machine

A key factor of IoT QoS prediction is service's QoS (e.g., response time and throughput) depends heavily on feature interactions. Clearly, QoS depends not only on whom the user is and what the service is, but also on the interaction between the user and the service. For example, a service that tends to have low latency may give a high response time to a particular user because the geographic distance between the user and the service provider is too far. For this reason, we focus on modeling pairwise feature interactions with FM in this work.

Formally, let U and S denote the user set and service set, the task of FM is to estimate a target function $f : U \times S \longmapsto R$. Each user–service interaction $(u, s) \in U \times S$ with a feature vector $\boldsymbol{X} \in R^G (i.e.,\ G = M + N)$ with one-hot encoding, which indicates which user invoked which service. For example, if user u has invoked service s, the feature vector $\mathbf{X}$ is denoted as:

$$\mathbf{X} = (\underbrace{0, \ldots, 1, 0, \ldots, 0}_{\text{user\_id}}, \underbrace{0, \ldots, 1, \ldots, 0, 0}_{\text{service\_id}})$$

where non-zero elements in $\mathbf{X}$ are corresponding to IDs of user u and service s.

Given an input tuple $(\boldsymbol{X}, y)$, $\boldsymbol{X} = (X_1, \ldots, X_G) \in R^G$ is a G-dimensional feature vector and y is its corresponding QoS value. FM learns a model based on the interaction between pairwise features, as shown in Eq. (1).

$$f(\boldsymbol{X}) = w_0 + \sum_{i=1}^{G} w_i X_i + \sum_{i=1}^{G} \sum_{j=i+1}^{G} w_{i,j} X_i X_j \tag{1}$$

where w_0 is the global bias, w_i and $w_{i,j}$ are 1-order interaction parameters and 2-order factorized interaction parameters respectively, with

$$w_{i,j} = \langle v_i \cdot v_j \rangle = \sum_{p=1}^{k} v_{i,p} v_{j,p}$$

$v_i = (v_{i,1}, \ldots, v_{i,k})$ is k-dimensional factorized vector for feature i.

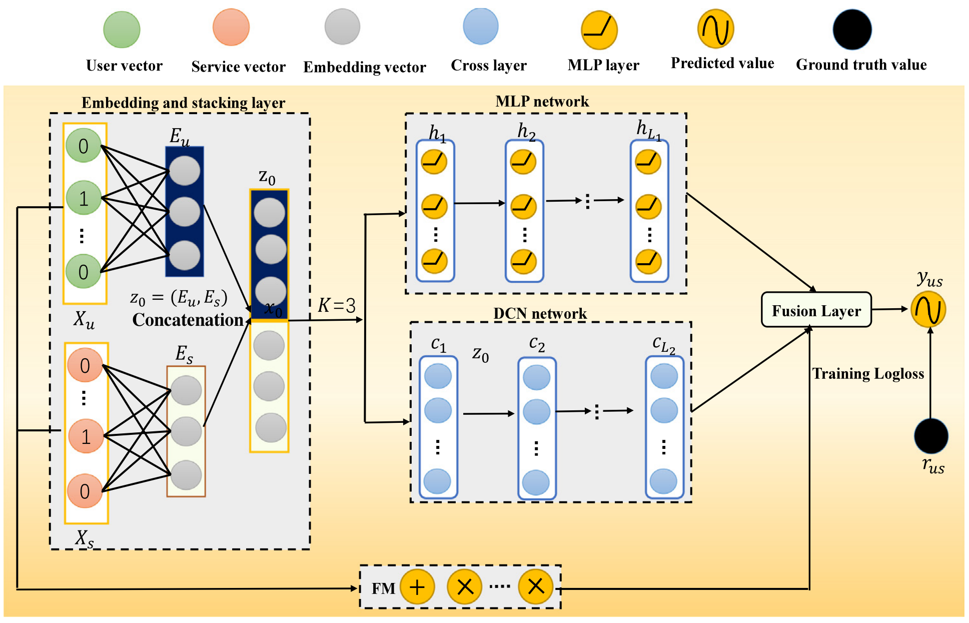


Fig. 3. The architecture of the proposed hybrid model for IoT QoS prediction.

The FM model can be computed in linear time by rewriting Eq. (1) as follows:

$$f(\boldsymbol{X}) = w_0 + \sum_{i=1}^{G} w_i X_i + \sum_{j=1}^{k}\left(\left(\sum_{i=1}^{G} v_{i,j} X_i\right)^2 - \sum_{i=1}^{G} v_{i,j}^2 X_i^2\right) \tag{2}$$

Given training dataset D, the model parameters of FM $\Theta = \{w_0, w_1, \ldots, w_G, v_{1,1}, \ldots, v_{G,k}\}$ can be learned by minimizing the sum of loss function, we further add a regularization term to avoid overfitting. The loss function Ψ is shown in Eq. (3):

$$\Psi(\Theta) = \sum_{(\boldsymbol{X},y)\in D} (f(\boldsymbol{X}|\Theta) - y)^2 + R(\Theta) \tag{3}$$

The regularization term $R(\Theta)$ is defined as follows:

$$R(\Theta) = \sum_{\theta\in\Theta} \lambda_\theta \theta^2 \tag{4}$$

For learning model parameter $\theta \in \Theta$, we calculate the derivatives of loss function, and the parameter optimization is achieved via the stochastic gradient descent (SGD) algorithm with the following update rule:

$$\theta \longleftarrow \theta - \eta\left(2\left(f(\boldsymbol{X}) - y\right)\frac{\partial f(\boldsymbol{X})}{\partial\theta} + 2\lambda_\theta\theta\right) \tag{5}$$

where η is the learning rate to control the step of each iteration. Once the model is learned when the whole SGD process achieves the convergence state, the predicted QoS of FM can be inferred using the model parameters.

3.2. Modeling high-order feature interaction with deep neural network

Traditional QoS prediction methods in IoT are usually neighborhood-based CF and model-based CF model, which ignore high-order feature interactions between users and services. Recently, enormous literature [34–36] demonstrate deep feature interactions are essential for good prediction performance on sparse features. However, existing deep learning based IoT QoS prediction methods usually treat one-hot encoding vector of user u and service s as input features to predict the QoS values, thus are easily over-fitted due to the extreme sparse and high-dimension characteristics of raw IoT QoS dataset. Another limitation of existing DNN-based QoS prediction in IoT is that the feature interactions learned by DNNs are implicit and highly nonlinear. To overcome these limitations, we firstly learn low-dimensional dense representation of raw QoS features in continuous space, then model deep feature interactions explicitly and implicitly between users and services with multilayer perceptron and deep cross network for IoT QoS prediction. The deep cross network consists of multiple layers and explicitly performs feature crossing automatically.

(1) Embedding and stacking layer

As shown in Fig. 3, the identifiers of users and services are firstly transformed into sparse binary vectors with one-hot encoding, denote as $\boldsymbol{X_u}$ and $\boldsymbol{X_s}$ with sizes are M and N respectively. Since the encoding vectors $\boldsymbol{X_u}$ and $\boldsymbol{X_s}$ are high dimensional and extreme sparse, we learn low-dimensional distributed representation of these encoding vectors with an embedding layer. Formally, let $P \in R^{M\times K}$ and $Q \in R^{N\times K}$ denote the embedding matrix of encoding vectors $\boldsymbol{X_u}$ and $\boldsymbol{X_s}$ respectively, and K denotes the embedding size. The outputs of embedding layer can be represented as $E_u = P \cdot \boldsymbol{X_u}$ and $E_s = Q \cdot \boldsymbol{X_s}$, respectively.

To effectively learn the dense embedding vector of encoding vectors, we initialize the embedding matrix with Probabilistic Matrix Factorization (PMF) [37]. Let P_i and Q_j denote the ith and jth embedding vector of user u_i and service s_j, PMF supposes the QoS value r_{ij} is a normal distribution with Gaussian noise:

$$p(R|P, Q, \sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[N(r_{ij}|P_i^T Q_j, \sigma^2)\right]^{I_{ij}} \tag{6}$$

where I_{ij} is an indication function, which means $I_{ij} = 1$ if user u_i invokes service s_j, otherwise it is 0.

Suppose the embedding matrix P and Q are normal distribution with Gaussian noise, as shown in Eqs. (7) and (8):

$$p(P|\sigma_P) = \prod_{i=1}^{N} N(0, \sigma_P^2\boldsymbol{I}) \tag{7}$$

$$p(Q|\sigma_Q) = \prod_{i=1}^{M} N(0, \sigma_Q^2\boldsymbol{I}) \tag{8}$$

The posterior probability of P and Q can be represented as:

$$p(P, Q|R, \sigma_P^2, \sigma_Q^2, \sigma^2) \propto p(R|P, Q, \sigma^2)p(P|\sigma_P^2)p(Q|\sigma_Q^2) \tag{9}$$

After adding a regularization term to avoid over-fitting, the loss function of posterior probability is shown in Eq. (10):

$$\begin{aligned} \Gamma = &-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{i,j}(R_{i,j} - P_i^T Q_j)^2 \\ &- \frac{\lambda_P}{2}\sum_{i=1}^{M} P_i^T P_i - \frac{\lambda_Q}{2}\sum_{j=1}^{N} Q_i^T Q_i \end{aligned} \tag{10}$$

where $\lambda_P = \sigma^2/\sigma_P^2$ and $\lambda_Q = \sigma^2/\sigma_Q^2$.

Based on loss function Γ, the parameter optimization is achieved via the SGD algorithm with the following update rule:

$$P_i = P_i + \eta\left[(R_{i,j} - P_i^T Q_j)Q_j - \lambda_P P_i\right] \tag{11}$$

$$Q_j = Q_j + \eta\left[(R_{i,j} - P_i^T Q_j)P_i - \lambda_Q Q_j\right] \tag{12}$$

where η is the learning rate.

The whole SGD process would achieve the convergence state after a considerable number of parameters updating iterations. Then, we can obtain the initial value of embedding matrix P and Q, and then jointly updated with other parts of the deep neural network.

The overall new feature is generated by stacking the embedding vectors of users and services, are formalized as:

$$\boldsymbol{z_0} = (E_u, E_s) = (P \cdot \boldsymbol{X_u}, Q \cdot \boldsymbol{X_s}) \tag{13}$$

The output $\boldsymbol{z_0}$ of embedding and stacking layer is fed into deep neural network for modeling high-order feature interaction, which will be elaborated in the next.

(2) Multilayer perceptron network

In this study, we utilize a multilayer perceptron (MLP) that consists of several full-connected hidden layers to implicitly model high-order feature interaction of QoS prediction in IoT, as shown in Fig. 3. The hidden layers can learn combinatorial features implicitly with some activation functions, such as exponential linear unit (ELU), rectified linear unit (ReLU), s-shaped rectified linear activation unit (SReLU) and logistic function. Note that the proposed method can model feature combinations in a linear or non-linear way depending on using linear or non-linear activation functions. Formally, let $\boldsymbol{z_0}$ as stated above denotes the output of embedding and stacking layer, which is fed into hidden layers of MLP and the forward process is as follows:

$$\begin{aligned} &\boldsymbol{h_1} = \kappa(\boldsymbol{w_1^m z_0} + \boldsymbol{b_1^m}) \\ &\boldsymbol{h_2} = \kappa(\boldsymbol{w_2^m h_1} + \boldsymbol{b_2^m}) \\ &\ldots\ldots \\ &\boldsymbol{h_{L_1}} = \kappa(\boldsymbol{w_{L_1}^m h_{L_1-1}} + \boldsymbol{b_{L_1}^m}) \end{aligned} \tag{14}$$

where L_1 is the depth of hidden layers, $\boldsymbol{w_i^m}$, $\boldsymbol{b_i^m}$, $\kappa(\cdot)$ and $\boldsymbol{h_i}$ are the model weight matrix, bias, activation function and output for the ith layer, respectively.

(3) Deep cross network

The MLP can automatically learn deep feature interaction implicitly based on the powerful learning capabilities of neural networks, but still bring two challenges: (1) the unexplainability brought by implicitly learning feature combinations and (2) the inefficient parameters learning since not all feature combinations are useful. To address these challenges, we exploit DCN for learning deep feature interactions explicitly, which is a state-of-art approach described in [29] due to its high performance. The DCN model performs feature interactions at each layer with crossover networks. Formally, for input $\boldsymbol{z_0}$, the calculation procedure of the lth in Fig. 4 is as follows:

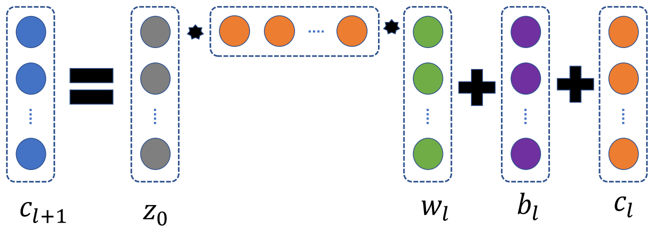


Fig. 4. The cross layer for QoS prediction in IoT.

$$\boldsymbol{c_{l+1}} = \boldsymbol{z_0 c_l^T w_l^d} + \boldsymbol{b_l^d} + \boldsymbol{c_l} = \phi(\boldsymbol{c_l}, \boldsymbol{w_l^d}, \boldsymbol{b_l^d}) + \boldsymbol{c_l} \tag{15}$$

And the overall calculation process is given in the following equation:

$$\begin{aligned} &\boldsymbol{c_1} = \boldsymbol{z_0 z_0^T w_0^d} + \boldsymbol{b_0^d} + \boldsymbol{z_0} \\ &\boldsymbol{c_2} = \boldsymbol{z_0 c_1^T w_1^d} + \boldsymbol{b_1^d} + \boldsymbol{c_1} \\ &\ldots\ldots \\ &\boldsymbol{c_{L_2}} = \boldsymbol{z_0 c_{L_2-1}^T w_{L_2-1}^d} + \boldsymbol{b_{L_2-1}^d} + \boldsymbol{c_{L_2-1}} \end{aligned} \tag{16}$$

where L_2 is the number of layers, $\boldsymbol{w_i^d}$ and $\boldsymbol{b_i^d}$ denote the weight parameters and bias parameters of the ith layer, respectively.

As seen in Eqs. (14) and (16), for an input $\boldsymbol{z_0}$ of dimension K, the number of parameters of the cross network is $K \times L_2 \times 2$; while for MLP, its number of parameters is $K \times m + m + (m^2 + m) \times (L_1 - 1)$, where m is the number of neurons per layer. The model parameters of DCN are almost an order of magnitude less than MLP. Additionally, the network structure of DCN is simple and efficient, since its complexity is determined by the depth of network layers.

3.3. QoS prediction with parametric-matrix-based fusion

Since our model adopts two pathways to model low-order and high-order feature interaction between users and services: (1) utilizing FM to model linear feature interactions of users and services; (2) adopting MLP and DCN to model non-linear feature interactions of both users and services. We generate the ultimate QoS prediction y_{us} by concatenating the output of FM, MLP and DCN and feed into a standard output layer.

$$y_{us} = \sigma(\boldsymbol{\pi^T}[\Phi^{FM}, \Phi^{MLP}, \Phi^{DCN}]^T) \tag{17}$$

where σ and $\boldsymbol{\pi^T}$ are the activation function and weights of the output layer. Φ^{FM}, Φ^{MLP} and Φ^{DCN} are the outputs of FM, MLP and DCN, respectively, calculated by Eq. (2), (14) and (16).

The proposed QoS prediction model in IoT captures user–service latent interactions by fusing the linearity of FM and non-linearity of MLP and DCN. Obviously, the model parameters can be inferred with back-propagation algorithm. However, gradient-based optimization methods only find locally-optimal parameters due to the non-convexity of the proposed model's objective function. The parameter initialization is essential for the convergence and performance of deep neutral network, we initialize the model parameters with pre-trained models of FM, MLP and DCN. We first train FM with via stochastic gradient descent algorithm as shown in Eq. (5), then train the model parameters of MLP and DCN with back-propagation algorithm until convergence. Then, the corresponding parts of the proposed model parameters are initialized with these pre-trained model parameters of FM, MLP and DCN. Finally, we concatenate weights of the three models as output with the following Equation:

$$\boldsymbol{\pi} \longleftarrow \begin{bmatrix} \alpha_1 \boldsymbol{\pi}^{FM} \\ \alpha_2 \boldsymbol{\pi}^{MLP} \\ \alpha_3 \boldsymbol{\pi}^{DCN} \end{bmatrix}, \alpha_1 + \alpha_2 + \alpha_3 = 1 \tag{18}$$

where $\boldsymbol{\pi}^{FM}$, $\boldsymbol{\pi}^{MLP}$ and $\boldsymbol{\pi}^{DCN}$ denote the weight vector of the pre-trained FM, MLP and DCN model, respectively; α_1, α_3 and α_3 are hyper-parameters to determine the trade-off between the three pre-trained models.

The whole procedure of our proposed method can be seen in Algorithm 1. First, as shown in Line 1, we encode user ID and service ID with one-hot encoding and stack them as intersection feature. Then, as seen in Line 2 to 5, we train FM model until converged. Next, as depicted in Line 6 to 10, we initial the parameters of embedding layer and update them until converged. Finally, we embed the sparse vector into dense vector, and optimize parameters of MLP and DCN until converged in Line 11 to 17.

Algorithm 1 The algorithm for training parameters of the hybrid IoT QoS prediction model

```
Input: User set $U$, service set $S$ and QoS matrix $R$
       Model parameters $\{\boldsymbol{P}, \boldsymbol{Q}, \pi^{FM}, \pi^{MLP}, \pi^{DCN}\}$
1: One hot code the user ID and service ID, stack them as intersection feature $\boldsymbol{X}$
2: while not converged do
3:     Predict QoS value according to Equation (1)
4:     Update $\pi^{FM}$ according Equation (5)
5: end while
6: Initial embedding matrix $\boldsymbol{P}$ and $\boldsymbol{Q}$ according to Equation (7) and (8)
7: while not converged do
8:     Predict QoS value according to Equation (10)
9:     Update $\boldsymbol{P}$ and $\boldsymbol{Q}$ according Equation (11) and (12)
10: end while
11: Embed sparse vector $\boldsymbol{X}$ to dense vector $\boldsymbol{z_0}$
12: while not converged do
13:     Calculate low-order feature interaction $\Phi^{FM}$ according to Equation (1)
14:     Calculate high-order feature interaction $\Phi^{MLP}$ and $\Phi^{DCN}$ according to Equation (14) and (16)
15:     Predict QoS value according to Equation (17)
16:     Update parameters $\pi^{MLP}$ and $\pi^{DCN}$
17: end while
18: return $\{\boldsymbol{P}, \boldsymbol{Q}, \pi^{FM}, \pi^{MLP}, \pi^{DCN}\}$
```

4. Experiment evaluation

In this section, we report the results of a series of experiments conducted to evaluate the performance of the proposed IoT QoS prediction model. To make a fair comparison between the proposed model and comparative methods, we first describe the settings of experiments including QoS datasets, comparative methods and evaluation metrics. Then, we report and discuss the experimental results.

4.1. Experimental settings

4.1.1. QoS Dataset in IoT

Currently, as there is no public dataset for IoT QoS prediction, almost all research literature on IoT QoS prediction published in reputable journals and conference (such as TSC [18,38,39], FGCS [17], IoT-J [19] and Percom [24]) use the dataset released by Zheng et al. [40] to perform experiment analysis, which consists of a matrix of the response time and throughput of 339 users for 5,825 web services. Therefore, we follow these work to measure the performance of the proposed method with this dataset, which can be freely downloaded from the website.[1] The reasons for the use of a public dataset instead of a testbed or simulation are: (1) the dataset contains more users and services to be evaluated than available testbeds; (2) a public dataset allows the proposed model can be compared to existing state-of-the-art. The WSDream dataset contains 339 users and 5,825 services, and the number of service invocation records is 1,974,675; (3) the method would be evaluated using different values generated by simulation experiments, since the QoS value may change due to congestion on the network, which make future algorithms difficult to compare with the existing baselines.

There are two types of QoS properties in this dataset, i.e., response time and throughput. Response time denotes the total time between a request for service and the fulfillment of that request, while throughput stands for the data transmission rate (e.g., kbps) when a user invoking a service. In this study, we evaluated the proposed QoS prediction model in both response time property and throughput property. In addition, this dataset is high-sparse as the sparsity ratio (i.e., the number of missing QoS values in user–service QoS matrix) is about 26%.

According to [41], the values of response time are between 0 and 20 s, and 91% of response time values are less than 2 s in this dataset. Obviously, such QoS indicators are unlikely to be provided by IoT devices with limited resource and energy. To make the dataset better meet the IoT scenario, we add heterogeneous traffic data to the existing dataset using the HetHetNets traffic model [42]. The model provides a realistic and manageable traffic model which can be applied to heterogeneous wireless cellular networks with heterogeneous traffic distributions, such as WiFi and wireless sensor network. And the parameter λ of this model is set to 2, because this is the standard practice for modeling network traffic in IoT scenarios [21]. The user–service QoS matrix derived from user's invocation records is usually highly sparse in the real-world IoT environment, since a user usually invokes a few services due to a finite amount of resources. Therefore, we randomly select a part of QoS records in the dataset as the training set, and utilizes the remaining QoS records as the test set.

4.1.2. Baseline methods

To verify the efficiency of the proposed QoS prediction technique in IoT, we compare the proposed method with the following state-of-the-art competitors, where the first two competitors are neighborhood-based CF methods for QoS prediction in IoT, the next four competitors are well-known model-based CF methods for QoS prediction in IoT, the last two competitors are deep neutral network based methods for QoS prediction in IoT. We detail these competitors as following.

- **User-based Collaborative Filtering for QoS Prediction (UPCC).** UPCC [43] is a collaborative filtering algorithm by using the Persons Correlation Coefficient to calculate user's similarity, and performs QoS prediction based on neighboring user's QoS value.
- **Hybrid Collaborative Filtering for QoS Prediction (UIPCC).** This approach [44] firstly predicts IoT QoS value with User-based and Item-based collaborative filtering (IPCC) respectively, then obtains the ultimate QoS prediction by fusing the QoS prediction values of UPCC and IPCC.
- **Collaborative QoS Prediction in IoT (IoTPredict).** Since UIPCC [44] utilizes Persons Correlation Coefficient to calculate the similarity between users and services, but a number of assumptions of UIPCC are not satisfied, such as having

1 https://github.com/wsdream/wsdream-dataset

no outliers and the variables being approximately normally distributed. To address these challenges, IoTPredict [21] exploits an alternative non-parametric similarity computation mechanism (i.e., Kendall's Tau) to calculate the similarity between users and services, then generates IoT QoS predictions for the missing values by using the top-K largest tau values of the users and services.

- **Non-negative Matrix Factorization based QoS Prediction (NMF).** This approach [45] is matrix factorization-based method for QoS prediction with the property that all factorized matrices have no negative elements.
- **Probabilistic Matrix Factorization based QoS Prediction (PMF).** In PMF [46], probability distribution is utilized to traditional matrix factorization method, then utilizes the Bayesian method to derive the posterior probability of implicit features of users and services.
- **Factorization machines based QoS Prediction (FM).** This study [23] utilizes IoT service's historical invoked records to make QoS predictions for service users. This approach integrates the QoS experiences of both similar users and nearby users with the Factorization Machines.
- **Autoencoder based QoS Prediction (Autoencoder).** This study [24] utilizes a stacked autoencoder with drop-out on a deep edge architecture to predict IoT QoS values, which aims to learn a representation for a set of data by training the network to ignore signal noise.
- **Deep neutral network based QoS Prediction (DNN).** In [25], a deep neural network based model is adopted to capture the relationship between diverse communication parameters (e.g., queue size, application traffic rate) and two IoT QoS metrics (i.e., transmission power and delay).

4.1.3. Evaluation metrics

To study the effectiveness of the proposed QoS prediction model in IoT, i.e., how well the method can predict the QoS values when a given user invokes a service, we use two widely used metrics, namely, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), following the work [23,24].

– **Mean Absolute Error (MAE)**, which calculates the average of absolute difference between predicted QoS value and ground truth as the following:

$$MAE = \frac{1}{|TE|}\sum_{(i,j)\in TE} |q(u_i, s_j) - r(u_i, s_j)| \tag{19}$$

where $q(u_i, s_j)$ is the ground truth QoS values when user u_i invokes service s_j, $r(u_i, s_j)$ is the predicted QoS value by prediction models, and $|TE|$ is the number of the test samples.

– **Root Mean Square Error (RMSE)**, which represents the standard deviation of the difference between the predicted values and observed values as the following:

$$RMSE(t) = \sqrt{\frac{1}{|TE|}\sum_{(i,j)\in TE} [q(u_i, s_j) - r(u_i, s_j)]^2} \tag{20}$$

4.2. Experimental results

In this subsection, we first study the impact of hyperparameters of the proposed QoS prediction model in IoT. Then we conduct extensive experiments and report their performance to validate the efficiency of the proposed method.

4.2.1. Impact of hyperparameters

Tuning hyperparameters, such as the number of neurons per layer, the number of hidden layers and activation functions, are critical to the performance of the proposed QoS prediction method. We study the impact of these hyperparameters on the response time property of the QoS dataset.

(1) Effect of Activation functions. We mainly compare the performance of five kinds of activation functions by applying these activation functions to all hidden layers, i.e., tanh, relu, elu, selu, softplus, the experiment results are shown in Fig. 5. From this figure, we observe selu achieves the best performance when the QoS dataset density from 10% increases to 30%. For instance, the MAE is 1.023 when applying selu as activation function and setting the dataset density as 10%, which is improved by 15.64% (relu), 4.39% (tanh), 2.93% (elu) and 26.32% (softplus) with the same dataset density. Similarly, the RMSE of selu is 1.597 when setting the dataset density as 30%, which is improved by 7.29% (relu), 5.12% (tanh), 1.7% (elu) and 18.81% (softplus) with the same dataset density.

(2) Effect of the number of neurons per layer. We set the number of neurons per layer to 64, 128, 256, 512 and 1024 when other hyperparameters remain the same. From Fig. 6, we can observe that the best performance is achieved when setting the number of neurons per layer as 256. We further observe the performance increases with the increasing of neurons per layer from 64 to 256, and then decrease when the number of neurons per layer is larger than 256. For example, the RMSE is 1.775 when setting the dataset density as 20% and the number of neurons per layer as 64, which decreases to 1.746 when the number of neurons per layer is 256, then increases to 1.758 when setting the number of neurons per layer as 1024. The result is not surprising since too many neurons per layer introduce model complexity thus may lead to model over-fitting, while fewer neurons per layer make the model suffer from under fitting. For the QoS dataset in our study, 256 neurons per layer are a good choice according to the experimental results.

(3) Effect of the number of hidden layers. One fundamental factor of the proposed QoS prediction model is the number of hidden layers for both DCN and MLP network. From Fig. 7, we observe that increasing the number of hidden layers from 1 to 4 can improve the performance of QoS prediction, however, the model performance will degrade when we continue to increase the number of hidden layers. Taking the MAE when dataset density is 10% as an example, the MAE is 0.99 when setting hidden layers number as 4, which improved 30.3% and 5.75% when setting hidden layers number as 1 and 8 respectively. According to these results, we can find that a single hidden layer with a finite number of neurons can approximate continuous functions with mild assumptions on the activation function, additional hidden layers can learn complex representations (i.e., automatic feature engineering) for complex high-sparsity datasets. The results suggest that one would like to avoid model over-fitting by setting the number of hidden layers to an appropriate small number.

4.2.2. Comparison of different QoS prediction models in IoT

We compare the performance of QoS prediction of different models on the two QoS properties, i.e., response time and throughout. Since there is no concept of hidden layer and activation functions for UPCC, UIPCC, PMF and NMF, it is impossible to compare the effect of these parameters in these methods on the prediction performance. To this end, we explore the impact of different activation functions and the number of hidden layers for Autoencoder [24] and DNN [25]. As mentioned earlier, we set the data density as 10% and repeat five times for each experiment, and report the average MAE and RMSE of the two methods in Fig. 8 and Table 2. We observe the best RMSE are

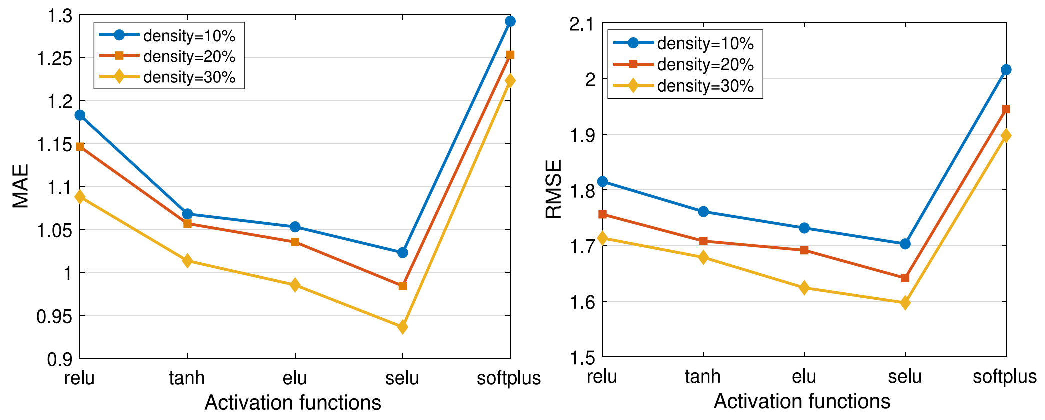


Fig. 5. Effect of different activation functions for the proposed QoS prediction model.

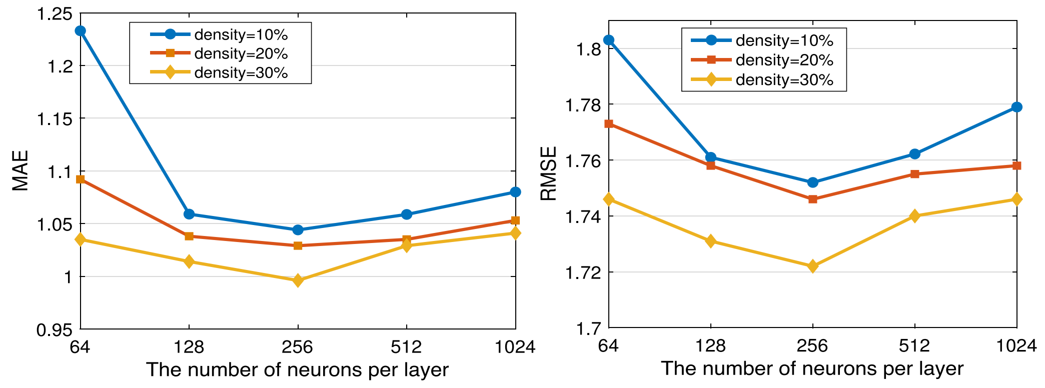


Fig. 6. Effect of the number of neurons per layer to the proposed QoS prediction model.

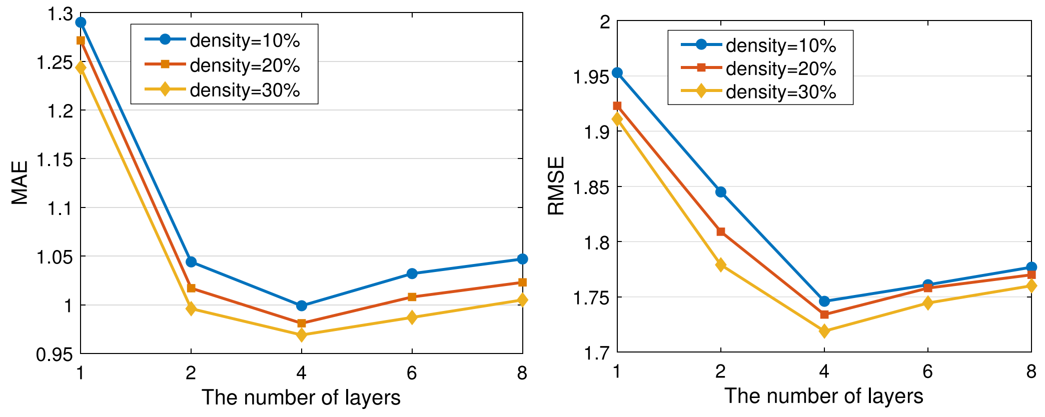


Fig. 7. Effect of the number of hidden layers to the proposed QoS prediction model.

Table 2
Effect of the number of hidden layers to Autoencoder and DNN.

Hidden layer		1	2	3	4	5	6	7	8	9	10
Autoencoder	MAE	1.594	**1.465**	1.486	1.712	1.687	1.703	1.686	1.677	1.695	1.665
	RMSE	2.263	**2.04**	2.167	2.276	2.321	2.358	2.324	2.389	2.353	2.368
DNN	MAE	1.318	**1.277**	1.279	1.292	1.301	1.325	1.353	1.348	1.353	1.347
	RMSE	1.982	**1.95**	1.965	1.971	1.965	1.963	1.978	1.9762	2.003	2.124

achieved when the activation function is set to selu, while the best MAE is achieved for Autoencoder and DNN with different activation functions (i.e., selu for DNN and relu for Autoencoder). For different hidden layers, the best MAE and RMSE are achieved when the layer is 2 The result is not surprising since deep learning based models would overfit to the training data with too many hidden layers.

In our experiment, there are six different data densities to be used, which are 5%, 10%, 15%, 20%, 25% and 30%. Finally, each case is repeated 10 times to produce the final results, and we report

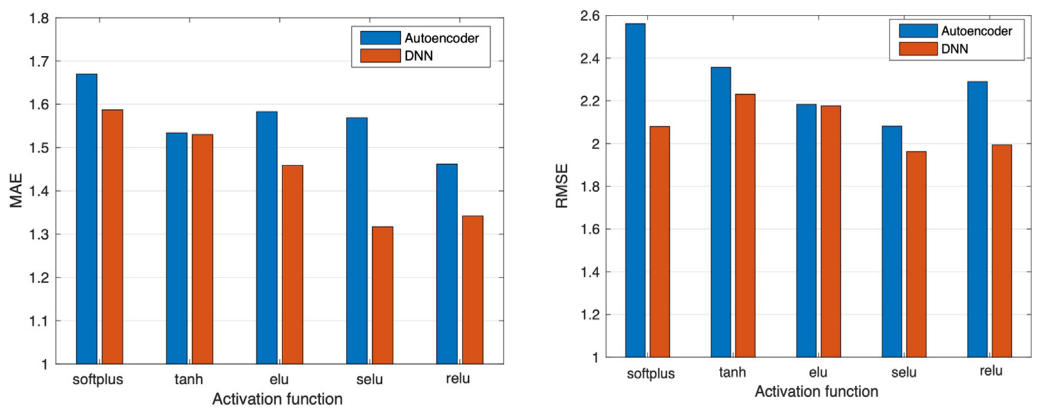


Fig. 8. Effect of different activation functions for Autoencoder and DNN.

the mean of the performance as the ultimate experiment results. The results are shown in Tables 3 and 4.

From Table 3, we have the following observations: (1) the proposed method that consider modeling both low-order and high-order feature interactions always outperforms traditional CF-based models (e.g., UPCC, UIPCC, MF, PMF, NMF and FM) and existing deep learning based models (e.g., Autoencoder and DNN). For example, the MAE of our method is 1.01 when the dataset density is 5%, while 1.54 for PMF (which is the best performance with CF-based models) and 1.345 for DNN(which is the best performance with deep learning based models). In fact, a small improvement in IoT QoS prediction is likely to lead to a significant increase with such a huge IoT service market. Therefore, the performance improvement of our method is significant (about 33.11% and 23% respectively), which clearly demonstrates the effectiveness of jointly considering low-order and high-order feature interactions for QoS prediction in IoT; (2) deep learning based models such as our method and DNN can achieve better performance than traditional CF-based models. For instance, compared to the best CF-based models PMF, our method and DNN achieve more than 29.57% and 6.24% in terms of MAE (12.36% and 3.94% in terms of RMSE) when the user–service QoS matrix density is 10%. The results suggest that existing CF-based QoS prediction methods perform low-order feature interaction since they are linear and have shallow structures, which cannot learn nonlinear and complex relationships with sparse QoS data set in IoT environment, thus will result in poor QoS prediction performance. Contrary to these CF-based QoS prediction methods, deep learning based methods can model non-linearity of sparse QoS dataset with nonlinear activations such as relu, elu, tanh and softplus. Therefore, deep learning based methods can capture complex and intricate user–service interaction patterns, thus can achieve better QoS prediction performance; (3) fusing both low-order and high-order feature interaction can significantly improve the performance of QoS prediction in IoT. This observation is from the fact that Antoencoder and DNN (which is similar to the proposed model without considering low-order feature interaction) performs much worse than the proposed model. As the best model, our method outperforms DNN by 24.13% in terms of MAE and 10.36% in terms of RMSE when the dataset density is 20%. This reason is that feature interactions behind service's invoked behaviors can be highly sophisticated, thus both low-order and high-order feature interactions should play important roles in IoT QoS prediction. Therefore, QoS prediction in IoT by considering low-order and high-order feature interactions simultaneously brings additional improvement over the cases of considering either alone.

Table 4 shows the prediction results of throughput property among these models with different dataset density, and both the MAE and RMSE of these models are shown in this Table. We observe the proposed method achieves the best performance, as the MAE and RMSE metric is always lower than other baseline models. For example, the MAE of our method is 10.07 when the dataset density is 15%, while the results are 16.55 for Autoencoder and 21.12 for IoTPredict. Similar results can be investigated in terms of RMSE with different dataset density. The result suggests again that jointly model low-order and high-order feature interactions can improve the prediction performance. On the contrary, existing CF-based models fail to predict service's throughout property as they are inability to uncover and generalize the underlying patterns from high-dimensional and high-sparse QoS dataset. Given such a high-sparse QoS dataset, the key challenge of QoS prediction is in effectively modeling feature interactions. Deep neural network is efficient in learning the underlying explanatory factors and useful representations from sparse QoS dataset. Another observation is that the prediction performance of all methods increase as the data density increases. Taking IoTpredict as example, the RMSE is 53.45 and 44.55 when the data density is 10% and 20% respectively. The reason is that increasing the data density makes the data denser, which leads more ground truth values in each test case thus leading to better prediction results.

5. Conclusion

This paper proposes a hybrid QoS prediction model (CFM) for IoT service by jointly modeling both low-order and high-order context feature interactions. CFM goes through three phases: learn low-order feature interaction with factorization machine, model high-order feature interaction with MLP and DCN, and aggregate the output of low-order and high-order feature interactions with a parametric-matrix-based fusion. Extensive experiments are conducted on a real-world benchmark dataset to evaluate CFM against the state-of-the-art models. The extensive results of the compared experiments verify the proposed method has significantly better performance than the state-of-the-art models in prediction accuracy, which suggest that jointly considering low-order and high-order feature interactions is significantly effective for IoT QoS prediction. Although CFM has shown promising prospects, one important issue remains unveiled: how to perform IoT QoS prediction in data privacy-sensitive IoT scenarios? Therefore, it is important to protect the private data of users while retaining the ability of generating accurate QoS prediction.

Currently, we are building an IoT testbed, which consists of several different places (e.g., bedroom, bathroom, garage, kitchen), where approximately 127 physical IoT objects (e.g., couch, laptop, microwave oven, fridge) are monitored by attaching WiFi tags

Table 3
Experiment results of different QoS prediction models in response time property (a smaller value indicates a higher accuracy).

Method	density=5%		density=10%		density=15%		density=20%		density=25%		density=30%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UPCC	2.57	3.356	2.535	3.32	2.511	3.276	2.457	3.108	2.398	2.967	2.375	2.856
UIPCC	2.235	3.123	2.21	3.04	2.146	2.914	2.042	2.85	1.912	2.768	1.876	2.75
IoTPredict	1.756	2.56	1.735	2.523	1.726	2.48	1.719	2.456	1.687	2.42	1.66	2.367
NMF	2.532	3.55	2.332	3.325	2.16	3.165	2.05	2.995	1.95	2.85	1.84	2.723
PMF	1.54	2.235	1.41	2.035	1.386	1.987	1.365	1.95	1.36	1.935	1.334	1.925
FM	2.056	3.04	2.01	3.023	1.95	3.01	1.932	2.96	1.92	2.964	1.895	2.921
Autoencoder	1.45	2.03	1.46	2.04	1.475	2.052	1.48	2.064	1.492	2.11	1.485	2.05
DNN	1.345	1.967	1.322	1.95	1.298	1.935	1.292	1.927	1.28	1.91	1.275	1.879
CFM	**1.01**	**1.756**	**0.993**	**1.751**	**0.985**	**1.745**	**0.975**	**1.732**	**0.972**	**1.729**	**0.968**	**1.726**

Table 4
Experiment results of different QoS prediction models in throughput property (a smaller value indicates a higher accuracy).

Method	density=5%		density=10%		density=15%		density=20%		density=25%		density=30%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UPCC	28.96	62.78	26.93	58.38	24.71	55.46	22.89	52.72	21.16	48.26	20.32	45.17
UIPCC	26.29	60.16	23.76	54.28	21.29	47.93	20.15	45.82	18.47	43.72	17.21	40.01
IoTPredict	25.35	59.56	23.35	53.45	21.12	48.92	19.24	44.556	18.18	42.52	17.36	40.167
NMF	27.35	60.4	25.32	57.56	23.83	54.35	21.56	50.56	20.25	46.17	19.95	43.15
PMF	19.74	40.62	18.46	37.86	17.57	34.78	16.18	32.18	15.22	29.84	14.36	27.79
FM	20.67	48.36	18.67	42.67	17.84	37.56	16.54	34.25	15.21	32.45	14.12	30.65
Autoencoder	18.55	40.15	17.15	36.23	16.55	33.2	15.78	31.05	14.54	28.56	13.82	26.46
DNN	16.48	36.45	14.61	33.91	13.97	30.27	13.57	27.68	13.02	25.87	12.59	23.72
CFM	**12.58**	**31.83**	**11.63**	**29.38**	**10.07**	**28.14**	**10.01**	**26.15**	**9.78**	**24.56**	**9.52**	**22.89**

and sensors. After collecting enough IoT QoS data in our testbed or other public datasets suitable for IoT QoS prediction emerge, we will assess the effectiveness of the proposed approach and compare with state-of-the-art techniques in the future.

CRediT authorship contribution statement

Yuanyi Chen: Writing – original draft, Design the algorithms. **Peng Yu:** Perform the experiments, Write Sections 4 of the manuscript. **Zengwei Zheng:** Design the algorithms, Research investigation. **Jiaxing Shen:** Write Sections 2 and 3 of the manuscript. **Minyi Guo:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data in this manuscript.

Acknowledgments

This work was supported in part by the Top Young Scholar of Zhejiang Province "Ten Thousands Talent Program", Top Young Scholar of Hangzhou "Ten Thousands Talent Program", National Natural Science Foundation of China under Grant 62072402, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LGN20F020003, and in part by Hangzhou Science and Technology Bureau under Grant 20191203B37 and the Intelligent Plant Factory of Zhejiang Province Engineering Lab.

References

[1] Maxim Chernyshev, Zubair Baig, Oladayo Bello, Sherali Zeadally, Internet of things (iot): Research, simulators, and testbeds, IEEE Internet Things J. 5 (3) (2017) 1637–1647.

[2] Ian F. Akyildiz, Ahan Kak, The internet of space things/CubeSats: A ubiquitous cyber-physical system for the connected world, Comput. Netw. 150 (2019) 134–149.

[3] He Li, Kaoru Ota, Mianxiong Dong, Minyi Guo, Learning human activities through Wi-Fi channel state information with multiple access points, IEEE Commun. Mag. 56 (5) (2018) 124–129.

[4] Yanwen Wang, Jiaxing Shen, Yuanqing Zheng, Push the limit of acoustic gesture recognition, IEEE Trans. Mob. Comput. (2020).

[5] Feilong Tang, Minyi Guo, Mianxiong Dong, Minglu Li, Hu Guan, Towards context-aware workflow management for ubiquitous computing, in: 2008 International Conference on Embedded Software and Systems, IEEE, 2008, pp. 221–228.

[6] J. Shen, J. Cao, X. Liu, BaG: Behavior-aware group detection in crowded Urban spaces using WiFi probes, IEEE Trans. Mob. Comput. (2020) 1.

[7] J. Shen, J. Cao, X. Liu, S. Tang, SNOW: Detecting shopping groups using WiFi, IEEE Internet Things J. 5 (5) (2018) 3908–3917.

[8] He Li, Kaoru Ota, Mianxiong Dong, Minyi Guo, Mobile crowdsensing in software defined opportunistic networks, IEEE Commun. Mag. 55 (6) (2017) 140–145.

[9] Jingyu Zhang, Siqi Zhong, Tian Wang, Han-Chieh Chao, Jin Wang, Blockchain-based systems and applications: A survey, J. Internet Technol. 21 (1) (2020) 1–14.

[10] Jingyu Zhang, Siqi Zhong, Jin Wang, Xiaofeng Yu, Alfarraj Osama, A storage optimization scheme for blockchain transaction databases, 2020.

[11] Yuanyi Chen, Jingyu Zhang, Liting Xu, Minyi Guo, Jiannong Cao, Modeling latent relation to boost things categorization service, IEEE Trans. Serv. Comput. (2017).

[12] Y. Zhang, J. Chen, B. Cheng, Integrating events into SOA for IoT services, IEEE Commun. Mag. 55 (9) (2017) 180–186.

[13] Yuanyi Chen, Jingyu Zhou, Minyi Guo, A context-aware search system for internet of things based on hierarchical context model, Telecommun. Syst. 62 (1) (2016) 77–91.

[14] Chang Xu, Jiachen Wang, Liehuang Zhu, Chuan Zhang, Kashif Sharif, PPMR: a privacy-preserving online medical service recommendation scheme in ehealthcare system, IEEE Internet Things J. 6 (3) (2019) 5665–5673.

[15] Buqing Cao, Jianxun Liu, Yiping Wen, Hongtao Li, Qiaoxiang Xiao, Jinjun Chen, Qos-aware service recommendation based on relational topic model and factorization machines for IoT mashup applications, J. Parallel Distrib. Comput. 132 (2019) 177–189.

[16] Z. Zheng, M.R. Lyu, Ws-dream-web service qos datasets, 2012, Retrieved from WS-Dream: http://www.wsdream.com/dataset.html.

[17] Hao Wu, Kun Yue, Ching-Hsien Hsu, Yiji Zhao, Binbin Zhang, Guoying Zhang, Deviation-based neighborhood model for context-aware QoS prediction of cloud and IoT services, Future Gener. Comput. Syst. 76 (2017) 550–560.

[18] Zhi-Zhong Liu, Quan Z Sheng, Xiaofei Xu, Dianhui Chu, Wei Emma Zhang, Context-aware and adaptive QoS prediction for mobile edge computing services, IEEE Trans. Serv. Comput. (2019).

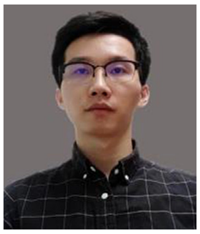
[19] Honghao Gao, Yueshen Xu, Yuyu Yin, Weipeng Zhang, Rui Li, Xinheng Wang, Context-aware QoS prediction with neural collaborative filtering for internet-of-things services, IEEE Internet Things J. 7 (5) (2019) 4532–4542.

[20] Shulong Chen, Yuxing Peng, Haibo Mi, Changjian Wang, Zhen Huang, A cluster feature based approach for QoS prediction in web service recommendation, in: 2018 IEEE Symposium on Service-Oriented System Engineering, SOSE, IEEE, 2018, pp. 246–251.
[21] Gary White, Andrei Palade, Christian Cabrera, Siobhán Clarke, IoTPredict: collaborative QoS prediction in IoT, in: 2018 IEEE International Conference on Pervasive Computing and Communications, PerCom, IEEE, 2018 pp. 1–10.
[22] Yuyu Yin, Weipeng Zhang, Yueshen Xu, He Zhang, Zhida Mai, Lifeng Yu, QoS prediction for mobile edge service recommendation with auto-encoder, IEEE Access 7 (2019) 62312–62324.
[23] Mingdong Tang, Wei Liang, Yatao Yang, Jianguo Xie, A factorization machine-based QoS prediction approach for mobile service selection, IEEE Access 7 (2019) 32961–32970.
[24] Gary White, Andrei Palade, Christian Cabrera, Siobhán Clarke, Autoencoders for qos prediction at the edge, in: 2019 IEEE International Conference on Pervasive Computing and Communications, PerCom, IEEE, 2019, pp. 1–9.
[25] Muhammad Ateeq, Farruh Ishmanov, Muhammad Khalil Afzal, Muhammad Naeem, Predicting delay in IoT using deep learning: A multiparametric approach, IEEE Access 7 (2019) 62022–62031.
[26] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, Hamid Laga, A comprehensive survey of deep learning for image captioning, ACM Comput. Surv. 51 (6) (2019) 1–36.
[27] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, Khaled Shaalan, Speech recognition using deep neural networks: A systematic review, IEEE Access 7 (2019) 19143–19165.
[28] Sylvie Thiria Patrick Gallinari, Francoise Fogelman, Multilayer perceptrons and data analysis, in: IEEE 1988 International Conference on Neural Networks, Vol. 1, 1988 pp. 391–399.
[29] Ruoxi Wang, Bin Fu, Gang Fu, Mingliang Wang, Deep & cross network for ad click predictions, in: Proceedings of the ADKDD'17, in: ADKDD'17, Association for Computing Machinery, New York, NY, USA, 2017.
[30] Xiong Luo, Ji Liu, Dandan Zhang, Xiaohui Chang, A large-scale web QoS prediction scheme for the industrial internet of things based on a kernel machine learning algorithm, Comput. Netw. 101 (2016) 81–89.
[31] Huaizhou Yang, Bowen Lv, A simplified prediction method of IoT service response time, in: Recent Developments in Intelligent Computing, Communication and Devices, Springer, 2019, pp. 963–971.
[32] Gary White, Andrei Palade, Siobhán Clarke, Qos prediction for reliable service composition in iot, in: International Conference on Service-Oriented Computing, Springer, 2017, pp. 149–160.
[33] Huaizhou Yang, Li Zhang, Response time prediction of IoT service based on time similarity, J. Comput. Sci. Eng. 11 (3) (2017) 100–108.
[34] Tongwen Huang, Zhiqi Zhang, Junlin Zhang, FiBiNET: Combining feature importance and bilinear feature interaction for click-through rate prediction, in: Proceedings of the 13th ACM Conference on Recommender Systems, in: RecSys '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 169–177.
[35] Wentao Ouyang, Xiuwu Zhang, Li Li, Heng Zou, Xin Xing, Zhaojie Liu, Yanlong Du, Deep spatio-temporal neural networks for click-through rate prediction, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, in: KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 2078–2086.
[36] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah, Wide & deep learning for recommender systems, in: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, in: DLRS 2016, Association for Computing Machinery, New York, NY, USA, 2016 pp. 7–10.
[37] Ruslan Salakhutdinov, Andriy Mnih, Probabilistic matrix factorization, in: Proceedings of the 20th International Conference on Neural Information Processing Systems, in: NIPS'07, Curran Associates Inc. Red Hook, NY, USA, 2007, pp. 1257–1264.
[38] Gary White, Siobhan Clarke, Short-term qos forecasting at the edge for reliable service applications, IEEE Trans. Serv. Comput. (2020).
[39] Huiying Jin, Pengcheng Zhang, Hai Dong, Yuelong Zhu, Athman Bouguettaya, Privacy-aware forecasting of quality of service in mobile edge computing, IEEE Trans. Serv. Comput. (2021).
[40] Y. Zhang, Z. Zheng, M.R. Lyu, WSExpress: A QoS-aware search engine for web services, in: 2010 IEEE International Conference on Web Services, 2010, pp. 91–98.
[41] Y. Yin, W. Zhang, Y. Xu, H. Zhang, Z. Mai, L. Yu, QoS prediction for mobile edge service recommendation with auto-encoder, IEEE Access 7 (2019) 62312–62324.
[42] Meisam Mirahsan, Rainer Schoenen, Halim Yanikomeroglu, HetHetNets: Heterogeneous traffic distribution in heterogeneous wireless cellular networks, IEEE J. Sel. Areas Commun. 33 (10) (2015) 2252–2265.
[43] Lingshuang Shao, Jing Zhang, Yong Wei, Junfeng Zhao, Bing Xie, Hong Mei, Personalized qos prediction forweb services via collaborative filtering, in: Ieee International Conference on Web Services, Icws 2007, IEEE, 2007 pp. 439–446.
[44] Zibin Zheng, Hao Ma, Michael R. Lyu, Irwin King, Qos-aware web service recommendation by collaborative filtering, IEEE Trans. Serv. Comput. 4 (2) (2010) 140–152.
[45] S.U. Kai, M.A. Liang-Li, S.U.N. Yu-fei, G.U.O. Xiao-ming, Non-negative matrix factorization model for web service QoS prediction, J. ZheJiang Univ. (Eng. Sci.) 49 (7) (2015) 1358–1366.
[46] Yueshen Xu, Jianwei Yin, Wei Lo, Zhaohui Wu, Personalized location-aware QoS prediction for web services using probabilistic matrix factorization, in: International Conference on Web Information Systems Engineering, Springer, 2013, pp. 229–242.

Yuanyi Chen received his BSc degree from Sichuan University in 2010, the MSc degree from Zhejiang University in 2013 and PhD degree from Shanghai Jiao Tong University in 2017. From 2014 to 2015 and 2018 to 2019, he was a visiting scholar at Hong Kong Polytechnic University and University of California, Irvine. He is currently a distinguished research fellow at the Department of Computer Science and Engineering at Zhejiang University City College. His research interest includes the Internet of Things, mobile computing and ubiquitous computing. He published more than 20 technical papers in major international journals and conference proceedings.

Yu Peng received his B.S. degree in Computer Science from Soochow University in 2020. He is now a master candidate at College of Computer Science and Technology, Zhejiang University and is studying at Zhejiang University City College. His research topic includes the Internet of Things and Mobile Edge Computing.

Zengwei Zheng received the B.S and M.Ec. degrees in Computer Science and Western Economics from Hangzhou University, China in 1991 and 1998, respectively. He received the Ph.D. degree in Computer Science and Technology from Zhejiang University, China in 2005. He is currently a professor of the Department of Computer Science and Engineering at Zhejiang University City College. His research interests include wireless sensor network, location-based service, internet of things, digital agriculture and pervasive computing. He is a member of the ACM and the CCF.



Jiaxing Shen received B.E. in Software Engineering from Jilin University in 2014 and Ph.D. in Computer Science from PolyU in 2019. He is currently a Research Assistant Professor in the Department of Computing at The Hong Kong Polytechnic University. His research interests include Mobile Computing, Data Mining, Social Computing, Affective Computing, and Internet of Things. He has published several papers in high-impact journals and top conferences. He served as a reviewer for many international conferences and journals like TMC, IJCAI, and PERCOM.

Minyi Guo received his B.Sc. and M.E. degrees in computer science from Nanjing University, China, and his Ph.D. degree in computer science from the University of Tsukuba, Japan. He is currently a Zhiyuan Chair Professor and a chair of the Department of Computer Science and Engineering, Shanghai Jiao Tong University, china. He received the National Science Fund for Distinguished Young Scholars award from NSFC in 2007. He is a Fellow of IEEE.