DeepApp: Predicting Personalized Smartphone App Usage via Context-Aware Multi-Task Learning

TONG XIA, YONG LI, JIE FENG, and DEPENG JIN, Tsinghua University
QING ZHANG and HENGLIANG LUO, Meituan-Dianping Group
QINGMIN LIAO, Tsinghua University

Smartphone mobile application (App) usage prediction, i.e., which Apps will be used next, is beneficial for user experience improvement. Through an in-depth analysis on a real-world dataset, we find that App usage is highly spatio-temporally correlated and personalized. Given the ability to model complex spatio-temporal contexts, we aim to apply deep learning to achieve high prediction accuracy. However, the personalization yields a problem: training one network for each individual suffers from data scarcity, yet training one deep neural network for all users often fails to uncover user preference. In this article, we propose a novel App usage prediction framework, named *DeepApp*, to achieve context-aware prediction via multi-task learning. To tackle the challenge of data scarcity, we train one general network for multiple users to share common patterns. To better utilize the spatio-temporal contexts, we supplement a location prediction task in the multi-task learning framework to learn spatio-temporal relations. As for the personalization, we add a user identification task to capture user preference. We evaluate DeepApp on the large-scale dataset by extensive experiments. Results demonstrate that DeepApp outperforms the start-of-the-art baseline by 6.44%.

CCS Concepts: • **Human-centered computing** → Ubiquitous and mobile computing design and evaluation methods;

Additional Key Words and Phrases: App usage prediction, multi-task learning, deep learning

ACM Reference format:
Tong Xia, Yong Li, Jie Feng, Depeng Jin, Qing Zhang, Hengliang Luo, and Qingmin Liao. 2020. DeepApp: Predicting Personalized Smartphone App Usage via Context-Aware Multi-Task Learning. *ACM Trans. Intell. Syst. Technol.* 11, 6, Article 64 (October 2020), 12 pages.
https://doi.org/10.1145/3408325

This work was supported in part by the National Key Research and Development Program of China under grant SQ2018YFB180012; the National Nature Science Foundation of China under grants 61971267, 61972223, 61861136003, and 61621091; Beijing Natural Science Foundation under grant L182038; Beijing National Research Center for Information Science and Technology under grant 20031887521; and the research fund of Tsinghua University–Tencent Joint Laboratory for Internet Innovation Technology.

Authors' addresses: T. Xia, Y. Li (corresponding author), J. Feng, and D. Jin, Tsinghua University, Haidian District, Beijing, China; emails: xia-t17@tsinghua.org.cn, liyong07@tsinghua.edu.cn, feng-j16@mails.tsinghua.edu.cn, jindp@tsinghua.eud.cn; Q. Zhang and H. Luo, Meituan-Dianping Group, Beijing, China; emails: {zhangqing31, luohengliang}@meituan.com; Q. Liao, Graduate School at Shenzhen, Tsinghua University, Shenzhen, China; email: liaoqm@sz.tsinghua.edu.cn.

2157-6904/2020/10-ART64 $15.00
https://doi.org/10.1145/3408325

1 INTRODUCTION

With the ever-increasing prevalence of smartphones, mobile applications (Apps) become increasingly popular, and they are considered as necessary entries to access daily online services such as communication, shopping, navigation, and entertainment. As the number of available Apps increases rapidly, predicting users' App usage, i.e., which App will be used next, has become increasingly important. It can help smartphone developers and network providers better manage battery energy consumption, pre-load Apps into the memory, and cache necessary resources in advance to significantly improve user experience [28].

Smartphone App usage prediction is beneficial but challenging. First, sequential regularity in App usage is complex, as the next-used App can be influenced by previously used Apps in a long or short time interval. For example, *Taobao*^{1}*(20:05)-WeChat(20:30)-Taobao*^{2}*(21:20)-AliPay(21:30)* are four Apps used chronologically, showing that the user first views goods on *Taobao* for about 25 minutes, then talks with his friends on *WeChat* in the following 50 minutes, returns to *Taobao* for another 10 minutes, and finally pays for the selected goods by *Alipay*. Therefore, *Taobao*^{2} depends on its prior use of *Taobao*^{1} an hour ago instead of *WeChat*. But *Alipay* is more likely to be caused by *Taobao*^{2}. Second, App usage is highly correlated to spatio-temporal contexts. For instance, a user opens *Toutiao* for news *every morning* but uses *AMap* for navigation *in a metro station*. These sequential regularity and spatio-temporal correlations are complexand difficult to capture by traditional methods such as Markov [14, 15] and Bayes [8, 19].

Luckily, *deep learning* provides a promising computational framework for solving complicated tasks, which inspires us to develop deep neural networks for App usage prediction. However, through an in-depth analysis on a large-scale real-world dataset, we find that App usage is not only spatio-temporally correlated but also highly personalized, i.e., users install their preferred Apps of each category and have different usage preference on the same Apps. This personalization yields a challenging problem for deep learning. When we train one deep neural network for all users, it often fails to uncover user preference and thus leads to low accuracy. But if we train one network for each individual, it would be unreliable because the personal data is often scarce, so that the insufficient data will aggravate the influence of random factors such as noise on the model [18, 23].

In this article, we seek to build an accurate smartphone App usage prediction model through deep learning for large-scale users. Specifically, we propose a multi-task learning–based framework, named *DeepApp*, to predict the next-used Apps from historical App usage with spatio-temporal contexts. To tackle the challenge of data scarcity, we train one general network for multiple users with insufficient personal data, where the embedding module and feature-extracting module to capture the complex sequential regularity are shared. To better utilize the spatio-temporal contexts, we supplement a location prediction task, i.e., to predict the location visited next, as a fake task in the multi-task learning framework. By supervising location prediction at the output side rather than merely using it as an input feature, the relation among time, location, and App is learned more explicitly. With regard to the personalization, we design a user identification task, i.e., to distinguish one from others by the historical App usage, as another fake task to capture user preference. As such, although being shared by all users, our model can sensitively recall user preference with limited input of historical usage.

Our contributions can be summarized as follows:

- We conduct an in-depth analysis on a large-scale real-world App usage dataset and find that smartphone App usage is spatio-temporally correlated and personalized. Based on these findings, we propose a context-aware multi-task learning framework—DeepApp—to achieve accurate App usage prediction.

Table 1. Examples of App Usage Data

User ID	Timestamp	Base Station ID	App ID
0000001	20160420090604	3610*****33	1 (*WeChat*)
0000002	20160420130513	3610*****F2	10 (*AMap*)
0000002	20160421233406	3610*****C1	1034 (*TaoBao*)

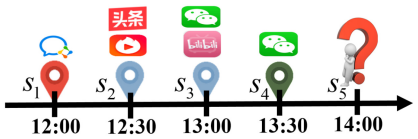


Fig. 1. An illustration of the prediction problem, where a user's four consecutive App usage sessions are known as history to predict the Apps that will be used in s_5.

- We propose a novel solution to tackle the dilemma raised by data scarcity and personalization, which is training one model for multiple users with user identification task. The general model is more robust, as the designed fake task enables it to better capture user preference.
- We perform extensive experiments to evaluate our model via the real-world dataset. Results demonstrate that DeepApp achieves significant improvements of 6.44% compared with the state-of-the-art baseline.

2 DATASET AND PROBLEM DEFINITION

2.1 Dataset

We use a real-world App usage dataset collected in Shanghai, China, with a duration of 1 week (April 20–26, 2016) [4, 21, 23]. Each record of the dataset is characterized by an anonymous user ID, timestamp, base station ID, and the used App ID. After filtering the abnormal users whose records are missed more than 30%, we finally obtain a dataset covering 10,000 users, 9,800 base stations, and 1,785 Apps. Table 1 describes the data format, where three records from two users are given as examples.

2.2 Problem Definition

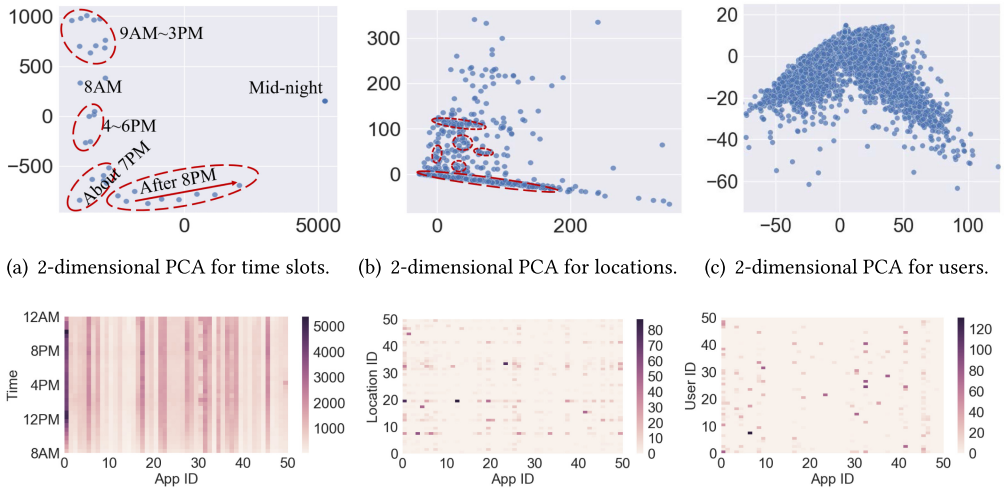
Our goal is to predict the Apps that a user will use in the next time slot given his historical records. To better illustrate this problem, we first formally give the relevant definitions as follows.

Definition 1 (App Usage Session). An App usage session refers to the Apps user u used in a specific time slot. We denote his n-th App usage session as $s_n = (t_n, l_n, a_n)$, where l_n is the location visited and a_n is the App sets used in time t_n.

Definition 2 (App Usage Sequence). App usage sequence is the consecutive App usage sessions of one user in a given time window, e.g., 1 day. The user's w-th App usage sequence is denoted by $S^w = (s_1^w, s_2^w, s_3^w, \ldots, s_n^w)$, where s_n^w is n-th the App usage session in the w-th time window.

As illustrated in Figure 1, we formally define our problem as follows.

App Usage Prediction Problem: *Given the historical App usage sequence* $S^w = (s_1^w, s_2^w, s_3^w, \ldots, s_n^w)$ *with previous* n *sessions of one user, to predict the Apps* a_{n+1}^w *to be used next in the* $(n+1)$*-th time slot.*



(a) 2-dimensional PCA for time slots. (b) 2-dimensional PCA for locations. (c) 2-dimensional PCA for users.

(d) Time-App usage matrix for weekdays. (e) Location-App usage matrix. (f) User-App usage matrix.

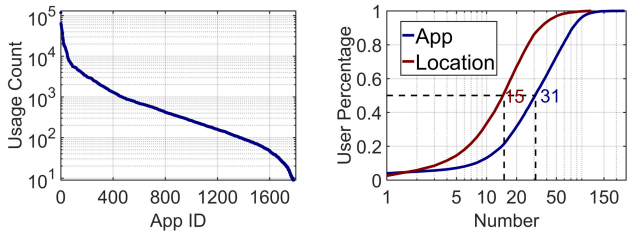
Fig. 2. Visualization for spatio-temporal characters and user preference.

2.3 Motivation

To inspire the solution of an accurate prediction model, we conduct in-depth analysis on our dataset. The findings are summarized as follows:

(1) *App usage has complex correlations with spatio-temporal contexts.* To inspect whether time and location are helpful for App usage prediction, we first analyze their relationship. We represent each time slot and location by an App usage vector, which denotes the corresponding usage count of each App. Then we conduct principal component analysis (PCA) to reduce these vectors to two dimensions and show the results in Figure 2(a) and (b), respectively. Each point denotes one time slot or one location. We can observe that as marked by the red circle, there are some clusters in both time slots and location distribution because they have a similar App usage pattern. Especially, time slots in the morning, afternoon, evening, and night form different clusters in Figure 2(a), indicating that App usage is spatio-temporally correlated. At the same time, we make a more detailed comparison to see how App usage varies with time and location. Figure 2(d) and (e) depict the aggregated App usage for 50 Apps and 50 locations that are randomly selected. In this figure, the darker the color, the more frequently the corresponding App is used [20]. Difference can be observed among different rows, although they have some similar patterns. In summary, the spatio-temporal correlations are complex. To improve prediction performance, we need to better utilize them.

(2) *App usage is also highly personalized.* To understand individual App usage patterns, we analyze user–Apps interaction by connecting overall App usage sessions into one vector. We also conduct PCA on the vectors and show the result in Figure 2(c). Unlike time and location, there are no obvious separate clusters, which means diverse App usage behaviors. In addition, we show the summed App usage of 50 selected users in Figure 2(f), from which we can observe that usage count across different Apps varies with users; even for the most popular App *WeChat,* as shown in the first column, its usage counts presented by the depth of color are different, thereby indicating that App usage is highly personalized. This also has been demonstrated by previous works [2], in that App usage depends on one's career to a great extent [29]. Those results tell us that when training one general model for multiple users, it is important to capture user preference.



(a) App usage count. (b) CDF of distinct App and location.

Fig. 3. Data statistic.

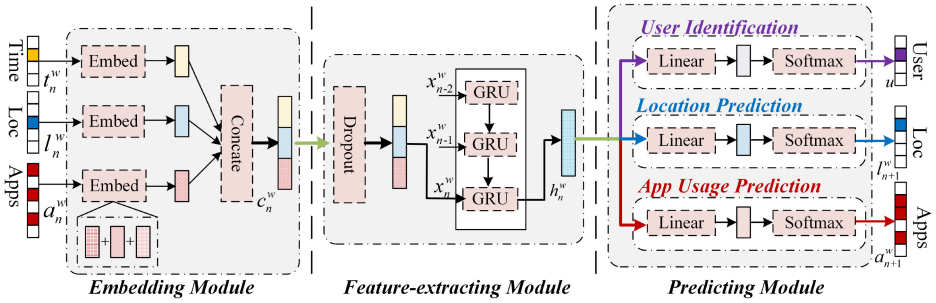


Fig. 4. Main architecture of DeepApp.

(3) *Personal data is insufficient.* From Figure 2(d) through (f), we can observe that the time-App, location-App, and user-App matrix are sparse. This is because the the interaction between users and Apps in different time slots and locations is uneven, which leads to data scarcity. Indeed, as shown in Figure 3(a), the usage count of each App follows Zipf's law, where the most frequently used 10 Apps including *WeChat* take up 19.7% of the total usage. However, as presented in Figure 3(b), more than 50% of the users are recorded in more than 15 locations and 31 Apps. This means that for most of the Apps and locations, the number of records for each user is extremely limited. Therefore, training one model for each individual is difficult to achieve high accuracy because personal data is too scarce to learn meaningful usage patterns.

To conclude, the spatio-temporal correlations and the difficulty to capture user preference from sparse individual data need to be better handled when designing the prediction model.

3 METHOD

3.1 Model Design

Inspired by the preceding findings, we propose DeepApp, a context-aware App usage prediction model, as shown in Figure 4. First of all, considering the complex temporal patterns, we integrate the time, location, and App in the embedding space, where high-order sequential regularity can be better modeled by the following GRU module. For the spatial context, besides using historical locations as input, we predict the next location to enhance the relations. In addition, to extract personalized features from the scare data, we design a user identification task combined with App usage prediction at the same time.

3.1.1 Embedding Module. The input of this module is historical App usage session sequence $[\{(t_1^w, l_1^w, a_1^w), \ldots, (t_n^w, l_n^w, a_n^w)\}]$, where t_n^w and l_n^w are one-hot vector, whereas a_n^w is multi-hot vector because multiple Apps would be used during each time slots. For each session, time t_n^w, location

l_n^w, and Apps a_n^w are first embedded into dense-represented vectors [28]. Compared with the limited one-hot or multi-hot representation, the dense representation can better capture the precise semantic relationship of different contexts. For App embedding, we adopt Doc2Vec [11], which treats each App as a word and each session as a sentence. We embed each App separately and then add them up. The embedding vector of historical time, location, and Apps is contacted, which is denoted by c_n^w as the output of this module.

3.1.2 Feature-Extracting Module. This module aims to capture the high-order sequential regularity hidden in the historical App usage sequence. We select the GRU as the basic recurrent unit because of its effectiveness to memory both long-term and short-term dependence. This module takes the contacted App usage embedding vector c_n^w as input and outputs the hidden state h_n^w step by step. It is worth noting that before the GRU, the contacted App usage embedding vector is first fed into a drop-out layer in the training process to avoid overfitting. To transfer the relation among App usage, time context, and location context with each other, we share the recurrent module for user identification, location prediction, and App prediction tasks instead of training independent GRUs.

3.1.3 Predicting Module. The predicting module is designed to complete multi-tasks with three separate branches: App usage prediction, location prediction, and user identification. Since App usage prediction is our target, the other two tasks are used as fake tasks for context awareness and personalization. These three tasks are based on the same embedding module and feature-extracting module to share spatio-temporal App usage patterns but have separate output layers in parallel because of the different outputs. Specifically, to utilize the learned features from the formal module, each branch consists of a linear layer and a nonlinear activation layer. The linear layers are fully connected layers to process the features into a smaller and more expressing vector, and changes the size of the hidden state to match the needed output size. Yet the nonlinear activation layers are softmax or sigmoid, respectively. The activation function rescales the outputs to correct values for prediction.

3.2 Loss and Training

Due to data scarcity, we train one general model for multiple users with three tasks simultaneously. The training goal is to minimize the divergence between the predicted distribution and the true distribution. All of these tasks can be regarded as a classification problem. Thus, we adopt cross-entropy loss for each task [16] and denote the loss for App usage prediction, user identification, and location prediction as J_a, J_u, and J_l, respectively. To train them jointly, we take the overall loss J of these three tasks with different weights as follows:

$$J = J_a + \lambda_1 J_u + \lambda_2 J_l, \tag{1}$$

where λ_1 and λ_2 are the weight factor for user identification and location prediction, respectively. Algorithm 1 outlines the training process of DeepApp. We first generate the training instance by taking each user's App usage sequence in a given time window as one mini-batch, then we use backward propagation through time (BPTT) and Adam to train the model step by step.

4 EXPERIMENTS

4.1 Experiment Setup

To evaluate DeepApp, we compare it with the following five baselines by the real-world dataset:

- *MRU* [19]: This method takes the *m*ost *r*ecently *u*sed Apps, i.e., the Apps used in last time slots, as prediction. It assumes that most Apps are used across several time slots continually.

ALGORITHM 1: Training algorithm for DeepApp

Input: App usage sequences: $\{S^{1,u_1}, S^{2,u_1}, \ldots, S^{w,u_1}, \ldots, S^{W,u_{M_u}}\}$;
Output: Trained Model $\mathcal{M}$.
//construct training instances: $\mathcal{D} \longleftarrow \emptyset$
for $u \in \{u_1, \ldots, u_{M_u}\}$ **do**
  for $w \in \{1, 2, \ldots, W\}$ **do**
    put a training instance (u, w, S^w) into $\mathcal{D}$
// train the model: initialize the parameters θ
for $i \in \{1, 2, \ldots, EPOCH\}$ **do**
  select instance $\mathcal{D}_w^u$ with w for user u from $\mathcal{D}$;
  update θ by minimizing the objective with $\mathcal{D}_w^u$;
  stop training when criteria is met;
output trained model $\mathcal{M}$

- *MFU* [19]: This method counts the users' App usage history and selects the *m*ost *f*requently *u*sed K Apps. This is the straightforward method for prediction, which does not use time and location context.
- *HA* [8]: This method takes the average App usage in the target time slots of historical days as prediction. This is also a straightforward method but considers the time context.
- *Bayes* [8]: Bayes takes the time of day, last-used App, and last visited location as features to learn a Bayesian network, and these features are modeled independently. Apps with the highest probability of the given time and location are taken as prediction.
- *AppUsage2Vec* [28]: AppUsage2Vec models App usage by considering the contribution of different apps, user personalized characteristics, and temporal context jointly, where personalization is handled by combining the user vector and App sequence vector. It is a state-of-the-art deep learning model for App usage prediction, but it does not use location context.

We also compare DeepApp with its variations:

- *Individual*: This trains one model for each user.
- *Unified*: This trains one model for all users without the user identification task.
- *Forward*: This trains one model for all users without the user identification task but combines user embedding with App usage characters for personalization as previous works do [6, 12, 13].
- *Backward*: This trains one model for all users with user identification, i.e., the complete DeepApp.
- *DeepApp-Loc*: This is a DeepApp model without location throughout.
- *DeepApp-LocPre*: This is a DeepApp model with location prediction in the last module removed.

We rank the Apps by the output distribution of the App usage prediction task and take the top K Apps as predictions. We use mean average precision (MAP) and area under the ROC curve (AUC), two commonly used global evaluations for ranking task [13], as metrics. We also use recall score of top K Apps (R@K) [28] for comparison. In a parameter study, we report the accuracy of top K locations and users [6]. For all of these metrics, the larger the value, the better the performance.

We pre-process our dataset into App usage sequence according to Definition 2 with a time interval of 30 minutes and time window of 24 hours for the 5 working days. After that, we sort each user's sequence by date and take the first 3 days as the training set, the fourth day as the validation set, and the last day as the test set. Experiments are implemented on a GPU by PyTorch,

Table 2. Default Parameter Settings

Training Settings	Value	Feature (Size)	Input	Output	Training Settings	Value	Feature (Size)	Input	Output
Learning rate (*lr*)	1e-4	Location	≈10000	256	λ_1, λ_2	0.2, 0.2	App	≈2000	512
Decay of *lr*	0.1	Time	48	16	Typical user number	10,000	Hidden state	{544,800}	{512, 640}

Table 3. Comparison Among Different Baselines with Our Model, Where ΔMAP Is the Relative Improvement

Model	R@5	AUC	MAP	ΔMAP
MRU	0.4180	0.7017	0.3747	31.38%
MFU	<u>0.5916</u>	0.8070	0.3770	30.98%
HA	0.5393	0.7957	0.4804	12.05%
Bayes	0.5439	0.8038	0.4957	9.24%
AppUsage2Vec	0.5604	<u>0.9566</u>	<u>0.5110</u>	6.44 %
DeepApp	**0.6095**	**0.9725**	**0.5462**	—

The best result in each column is bold, while the second is underlined.

Table 4. Comparison Among Variants of DeepApp

Metric	Individual	Unified	Forward	**Backward/DeepApp**	DeepApp-Loc	DeepApp-LocPre
R@5	0.4676	0.5082	0.5698	**0.6095**	0.5721	0.5828
AUC	0.9523	0.9679	0.9611	**0.9725**	0.9643	0.9706
MAP	0.4302	0.4522	0.5179	**0.5462**	0.5288	0.5305

and the default training settings are summarized in Table 2. For every method, we carefully tune the parameters to report the best performance.

4.2 Results

4.2.1 Overall Comparison. We first give the overall performance comparison in Table 3. We can observe that DeepApp achieves the best performance across all metrics. Among these baselines, MFU achieves the highest R@5 and AppUsage2Vec achieves the highest AUC and MAP, whereas DeepApp achieves a relevant improvement of 3.03%, 1.67%, and 6.44%, respectively. MRU and HA only utilize the temporal context directly and thus perform the worst. MFU achieves the higher R@5 because of the *head effect* of App usage as shown in Figure 3(a). Although Bayes utilizes different contexts, it treats them independently, leading to limited improvement compared with other traditional methods. With regard to the deep learning method AppUsage2Vec, its global metrics AUC and MAP are higher than other baselines, but its local metric R@5 is lower. This is because it captures more complex sequential regularity in App usage. However, since it trains one model for large-scale users, its ability to capture personalized features is weaker than an individual model like MFU. In addition, it does not utilize location, so it cannot achieve higher accuracy. In contrast, the remarkable performance improvement achieved by DeepApp shows the superiority of our model for better modeling complex spatio-temporal correlations and capturing user preference.

4.2.2 Ablation Study. To demonstrate the effectiveness of our design of the user identification task to capture user preference, we compare DeepApp with its variants in Table 4. From the results,

Table 5. Effect of λ_1 and λ_2

λ_1	λ_2	MAP(App)	ACC@1(User)	ACC@1(Loc)
0.1	0.05	0.5377	0.9164	**0.4744**
0.2		**0.5402**	0.9182	0.4599
0.5		0.5355	0.9224	0.4581
1		0.5368	**0.9263**	0.4621
0.2	0	0.4964	0.8922	0.3599
	0.2	**0.5462**	0.9292	0.4802
	0.5	0.5345	0.9256	0.4820
	1.0	0.5200	**0.9303**	**0.4881**

we can observe that individual model performance is worse because of data scarcity. The unified model also cannot achieve high accuracy because it is user independent. By utilizing the user embedding as a proxy of user preference, which is the same as most previous works do, forward model outperforms the individual and unified model. It trains one network for all users, but user embedding vectors are not shared, so it can deal with data scarcity and user preference. However, as user App usage behaviors are dynamic and dependent with spatio-temporal contexts, the ability to model user preference by a static user embedding vector is limited. In contrast, DeepApp models such dynamic preference in the backward propagation, which enforces RNN parameters to be more sensitive to different users.

When removing location from the input and output of our model, performance declines significantly as shown by DeepApp-Loc, which means that spatial context does help predict users' App usage behaviurs. However, it is worth noting that even without location, our model outperforms AppUsage2Vec, which also demonstrates the superiority of the user identification task. We can further observe that DeepApp-LocPre fails to achieve noticeable performance gain compared with DeepApp-Loc. This delivers the message that the location prediction task holds the key to fully exploit spatial-temporal correlations for better App usage prediction.

4.2.3 Parameter Tuning for Multi-Task Learning. Joint training of multi-task can become biased toward one specific task. Considering that App usage prediction is our goal, we carefully tune the weighting factors λ_1 and λ_2 in Equation (1). In this study, we adjust these two weights one after the other. Table 5 shows the performance of App prediction in terms of MAP, user identification, and location prediction in terms of ACC@1. When fixing the loss weight of location prediction λ_2 at a small value of 0.05, the performance of user identification shown by ACC@1(User) gets better with the increase of λ_1. However, when λ_1 exceeds 0.2, MAP(APP) degenerates dramatically because the model becomes biased to the user identification task. Intuitively identifying users can help improve the performance of App prediction, and an excessive user identification accuracy may reduce the performance of App usage prediction especially when new Apps are used. Therefore, we select $\lambda_1 = 0.2$ and then traverse different values of λ_2. Similarly, the accuracy of location prediction increases with λ_2 and MAP(App) rises first and then falls with a maximum of 0.5862 when $\lambda_2 = 0.2$. Therefore, the best setting of λ_1 and λ_2 is 0.2 and 0.2.

5 RELATED WORK

App usage prediction. Existing App usage prediction methods can be divided into Markov, Bayes, and deep learning three main lines [2]. Markov [14, 15] learns the order of different Apps used and predicts the next App based on the last-used App. It aims to model the sequential regularity in historical records but fails to utilize temporal and spatial context, i.e., when and where to use

these Apps. Compared with Markov, Bayes [8, 19] exploits spatio-temporal contexts by learning the probability of using different Apps given the last-used Apps, the time, and the location. However, its improvement is limited by modeling different contexts independently. The deep learning model—AppUsage2Vec [28]—measures the contribution of Apps previously used to the target one by an attention mechanism and considers user characters by a dual-DNN module. In addition, Shen et al. [17] recently proposed to predict App usage by reinforcement learning. However, most of the models are evaluated on a small-scale dataset (tens or hundreds of users), except in the work of Zhao et al. [28] (8,739 users). In addition, existing deep learning models [17, 28] all ignore utilizing helpful spatial contexts. In contrast, we propose a context-aware prediction model and evaluate it on 10,000 users.

Location prediction. Location prediction is a longstanding problem and has received consistent attention from researchers and industries. It also can be divided into traditional and deep learning–based methods. In early years, individual and group Markov methods were proposed to model location transition [3, 7, 25]. Recently, deep learning–based methods utilized the RNN [6, 13, 27], attention mechanism [5, 6], and memory network [10] to better capture the spatial-temporal correlations. The success of these models encourages us to design a deep learning model for App usage prediction. However, they are limited in extracting personalized features, and to the best of our knowledge, we are the first to propose user identification to tackle this problem, thereby improving performance.

Multi-task learning. Multi-task learning is a sub-field of transfer learning in which multiple learning tasks are solved at the same time while exploiting commonalities and differences across tasks. This can result in improved learning efficiency and prediction accuracy for the task-specific models when compared to training the models separately [1, 26]. It has been applied in image recognition [9], NLP [24], and travel time estimating [22]. For user modeling, knowledge learned from different but relevant tasks can be promoted mutually. For instance, Liao et al. [12] predict the next-visited location and point of interest simultaneously to achieve higher accuracy. Similarly, in our problem, the location prediction task assists App usage prediction with a helpful spatial context. In addition, to tackle the dilemma raised by data scarcity and personalization, to the best of our knowledge, we are the first to propose a user identification task at the output side to capture user preference. This suggests that in parallel to designing new architectures for user modeling, exploring multi-task learning for personalization is a promising avenue for research.

6 CONCLUSION

In this article, we propose a context-aware multi-task learning framework—DeepApp—to predict personalized App usage, which can exploit spatio-temporal contexts and better capture user preference to achieve better performance. To tackle the dilemma raised by data scarcity and personalization, we train one model for multiple users with a user identification task to capture user preference. Extensive experiments on a large-scale real-world dataset demonstrate its effectiveness. In the future, we plan to exploit the location property (e.g., point of interest) and App attribute (e.g., category) as semantic contexts to enhance the representation of different units and further improve prediction accuracy.

REFERENCES

[1] Jonathan Baxter. 2000. A model of inductive bias learning. *Journal of Artificial Intelligence Research* 12 (2000), 149–198.
[2] Hong Cao and Miao Lin. 2017. Mining smartphone data for app usage prediction and recommendations: A survey. *Pervasive and Mobile Computing* 37 (2017), 1–22.
[3] Meng Chen, Yang Liu, and Xiaohui Yu. 2014. NLPMM: A next location predictor with Markov modeling. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 186–197.

[4] Xinlei Chen, Yu Wang, Jiayou He, Shijia Pan, Yong Li, and Pei Zhang. 2019. CAP: Context-aware app usage prediction with heterogeneous graph embedding. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 4.

[5] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1555–1564.

[6] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 Conference on World Wide Web*. 1459–1468.

[7] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2012. Next place prediction using mobility Markov chains. In *Proceedings of the 1st Workshop on Measurement, Privacy, and Mobility*. ACM, New York, NY, 3.

[8] Ke Huang, Chunhui Zhang, Xiaoxiao Ma, and Guanling Chen. 2012. Predicting mobile application usage using contextual information. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, New York, NY, 1059–1065.

[9] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7482–7491.

[10] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2341–2347.

[11] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*. 1188–1196.

[12] Dongliang Liao, Weiqing Liu, Yuan Zhong, Jing Li, and Guowei Wang. 2018. Predicting activity and location with multi-task context aware recurrent neural network. In *Proceedings of the 27th International Conference on Artificial Intelligence*. 3435–3441.

[13] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

[14] Nagarajan Natarajan, Donghyuk Shin, and Inderjit S. Dhillon. 2013. Which app will you use next? Collaborative filtering with interactional context. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, New York, NY, 201–208.

[15] Abhinav Parate, Matthias Böhmer, David Chu, Deepak Ganesan, and Benjamin M. Marlin. 2013. Practical prediction and prefetch for faster access to applications on mobile phones. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, New York, NY, 275–284.

[16] Yi Shen. 2005. *Loss Functions for Binary Classification and Class Probability Estimation*. Ph.D. Dissertation. University of Pennsylvania.

[17] Zhihao Shen, Kang Yang, Wan Du, Xi Zhao, and Jianhua Zou. 2019. DeepAPP: A deep reinforcement learning framework for mobile application usage prediction. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 153–165.

[18] Hongzhi Shi, Chao Zhang, Quanming Yao, Yong Li, Funing Sun, and Depeng Jin. 2019. State-sharing sparse hidden Markov models for personalized sequences. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1549–1559.

[19] Choonsung Shin, Jin-Hyuk Hong, and Anind K. Dey. 2012. Understanding and prediction of mobile application usage for smart phones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, New York, NY, 173–182.

[20] Zhen Tu, Yali Fan, Yong Li, Xiang Chen, Li Su, and Depeng Jin. 2019. From fingerprint to footprint: Cold-start location recommendation by learning user interest from app data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 26.

[21] Zhen Tu, Runtong Li, Yong Li, Gang Wang, Di Wu, Pan Hui, Li Su, and Depeng Jin. 2018. Your apps give you away: Distinguishing mobile users by their app usage fingerprints. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (Sept. 2018), Article 138, 23 pages. DOI: https://doi.org/10.1145/3264948

[22] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. 2018. When will you arrive? Estimating travel time based on deep neural networks. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[23] Huandong Wang, Yong Li, Sihan Zeng, Gang Wang, Pengyu Zhang, Pan Hui, and Depeng Jin. 2019. Modeling spatio-temporal app usage for a large user population. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 27.

[24] Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. arXiv:1603.06270

[25] Chao Zhang, Keyang Zhang, Quan Yuan, Luming Zhang, Tim Hanratty, and Jiawei Han. 2016. GMove: Group-level mobility modeling using geo-tagged social media. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 1305–1314.

[26] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. arXiv:1707.08114
[27] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzheng Zhuang, S. Sheng Victor, and Zhou Xiaofeng. 2019. Where to go next: A spatio-temporal gated network for next POI recommendation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.
[28] Sha Zhao, Zhiling Luo, Ziwen Jiang, Haiyan Wang, Feng Xu, Shijian Li, Jianwei Yin, and Gang Pan. 2019. AppUsage2Vec: Modeling smartphone app usage for prediction. In *Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE'19)*. IEEE, Los Alamitos, CA, 1322–1333.
[29] Sha Zhao, Julian Ramos, Jianrong Tao, Ziwen Jiang, Shijian Li, Zhaohui Wu, Gang Pan, and Anind K. Dey. 2016. Discovering different kinds of smartphone users through their application usage behaviors. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, New York, NY, 498–509.

Received January 2020; revised April 2020; accepted June 2020