App Usage On-the-Move: Context- and Commute-Aware Next App Prediction

Yufan Kang[a], Mohammad Saiedur Rahaman[a], Flora D. Salim*[a], Yongli Ren[a], Mark Sanderson[a], Ryen W. White[b]

[a]*School of Computing Technologies, RMIT University, Melbourne, VIC, Australia*
[b]*Microsoft Research Redmond, Seattle, WA, USA*

Abstract

The proliferation of digital devices and connectivity enables people to work anywhere, anytime, even while they are on the move. While mobile applications have become pervasive, an excessive amount of mobile applications have been installed on mobile devices. Nowadays, commuting takes a large proportion of daily human life, but studies show that searching for the desired apps while travelling can decrease productivity significantly and sometimes even cause safety issues. Although app usage behaviour has been studied for general situations, little to no study considers the commuting context as vital information. Existing models for app usage prediction cannot be easily generalised across all commuting contexts due to: 1) continuous change in user locations; and 2) limitation of necessary contextual information (i.e. lack of knowledge to identify which contextual information are necessary for different commuting situations. We aim to address these challenges by extracting essential contextual information for on-commute app usage prediction. Using the extracted features, we propose AppUsageOTM, a practical statistical machine learning framework to predict both destination amenity and utilise the inferred destination to contextualise the app usage prediction with travelling purposes as crucial information. We evaluate our framework in terms of accuracy, which shows the feasibility of our work. Using a real-world mobile and app usage behaviour dataset with more than 12,495 trajectory records and more than 1046 mobile applications logged, AppUsageOTM significantly outperformed all baseline models, achieving *Accuracy*@k 46.4%@1, 66.4%@5, and 75.9%@10.

*Corresponding author
Email address: `flora.salim@rmit.edu.au` (Flora D. Salim*)

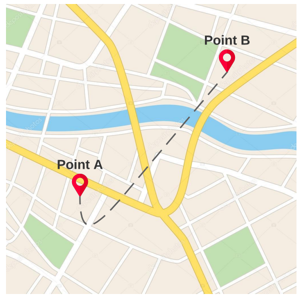


Figure 1: Example of a user commuting between location A and B

Keywords: app usage, mobile, information behaviour, app behaviour, prediction, recommendation, context-aware, transportation, mobility

1. Introduction

With the increasing prevalence of smart devices and the increasing number of applications in online markets, users can take advantage of a wide range of them anywhere at any time [1]. For example, the growing popularity of social network service apps such as Twitter and location-based service apps like Google Maps allows users to connect with others and retrieve information about their environment. While users can exploit abundant applications, the average number of apps installed on mobile devices also increases. For example, as [2] stated, on average, the number of apps installed on a smartphone can reach 56, and the highest number of apps installed is about 150. Given the number of apps installed on the smartphone and the number of apps that can be contained on a home screen, it is often tedious for a user to find the desired app. In order to overcome this issue, companies that target smart devices (i.e. Apple and Google) tend to have several home screens that show a set of shortcuts to enable users to find their desired applications efficiently. However, managing and selecting applications are still cumbersome and may require users to browse all screens.

In order to overcome the issue, various approaches exploiting different algorithms such as Machine Learning, Deep Learning, and Reinforcement Learning to predict what apps tend to be used next [3, 4, 5]. However, existing methods mainly focus on predicting the next applications while users are static at a specific location such as Point A or Point B by exploiting available contextual information such as time and location. However, app prediction while commuting, such as travelling from Point A to Point B, remained under-explored. When the methods are applied to app prediction while users are moving, the methods show limitations due to different contextual information such as continuously changing locations. Additionally, while increasing users tend to use applications when they are travelling, managing and searching desired applications can decrease production and cause distractions. With all the issues in mind, it is becoming increasingly important to predict what applications will be used next, focusing on travelling context, to make applications more accessible for users.

To address the challenge, we first analyse contextual information that is essential for app prediction while commuting. We then propose AppUsageOTM that exploiting analysed contextual information to predict the next apps during commuting. Thus, our proposed question is unique with the following features provided:

- **Contextual Information**: Existing studies show that contextual information is important for app prediction [6, 7, 8]. While existing studies focus on examining which contextual information is important to improve the accuracy of app prediction under the context that users are static, the contextual information essential for app prediction while users are on the move remains uncovered. Additionally, the available contextual information under moving conditions is different [9, 10]. For example, location information is accessible as contextual information while users are static, but it is unrealistic to keep extracting all locations given the users are travelling. In our study, we collect different contextual information while users are using applications when travelling. By exploiting the collected contextual information, we examine what contextual information is essential to predict next used apps considering users are moving and analyse the differences comparing with app prediction while users are static.

- **Travelling Purpose Prediction**: By examining available contextual information, we discover the purposes of commuting bring significant impact on app prediction on the move. Thus, our proposed approach predict travelling purpose and conduct a filtering system accordingly to improve app usage prediction accuracy.

- **App Prediction While People On the Move**: Most existing papers about app

Table 1: Comparison between App Usage Prediction and our problem.

	App Usage Prediction[24, 25, 27, 29, 30, 34, 41]	**App Usage Prediction On The Move** [this paper]
Question	What is the next app that will be used at a certain point of time and/or location?	What is the next app that will be used on the trajectory with the inferred destination?
Input	*Time*, *Location*, *App Sequences*	*Travelling Start Time*, *Travelling origin amenity*, *App Sequences*
Output	k apps with the highest probability to be used next	Destination Amenity (work, home, shopping, etc.); k apps with the highest probability to be used next
Sample Inference	The user have the highest probability to use Facebook as the next mobile application around 7:00pm and/or while he/she is at home	The user have the highest probability to use Google Map as next mobile application while he/she is driving back to home

usage prediction only predict the next app while users are static. As Verkasalo *et al.* stated that users tend to have different app usage behaviours while they are moving from they are static [11], our approach can enrich studies on the app usage behaviour with an extensive analysis on app usage behaviour under moving contexts.

Table 1 summarises the differences and the unique aspects of our approach. Compare with app usage prediction that predicts the next application on a static location, we target app usage prediction while people commute. Table 1 also illustrates the sample outputs for a particular scenario. For example, assume the user is on his/her way driving back home from a shopping mall and will use Google Map as the next app, the prediction can help to pre-load Google Map so that the user can pay less attention and spend less time on searching for the desired app.

A framework for App Usage Prediction On the Move requires frequent re-training to adapt to new data since users can change their app usage or commuting habits. Additionally, we also need to consider the potential increase of new users. Thus, we propose AppUsageOTM, which is a personalised and lightweight framework. AppUsageOTM is constructed using

Hierarchical Linear Modelling (HLM) and Support Vector Machine (SVM) that have comparably simple structures and achieve high performance with the limited amount of training data [12]. Through data processing, we found out that app usage patterns are different based on different purposes of travelling. Hence, AppUsageOTM first uses HLM to predict travelling purposes. Using the predicted travelling purpose to build a filtering system, AppUsageOTM then uses SVM to predict the next app while users are on the move.

There are three major contributions in this paper as follows:

- We propose a novel app usage prediction on the move problem: *Given a user is travelling on a weekday, can we predict the next mobile application that the user will use.*
- We construct a set of features exploiting the collected data. We also examine the essential contextual information for our prediction task.
- We develop a framework that exploits contextual information for app usage prediction on the move.

2. Related Work

The section first describes the previous work on productivity and attention management to show the significance of studying app usage prediction while commuting. We then demonstrate previous studies about applying contextual information to improve the accuracy of app usage prediction. Finally, we show existing methods for app usage prediction.

2.1. Productivity and Attention Management

People spend a large proportion of time commuting for multiple purposes, and studies have revealed that people often engage with their mobile phones to carry out different activities during their commutes [13, 14, 15]. However, studies show that engagement with mobile phone activities can cause distraction and compromise safety under certain circumstances such as driving [16, 17, 18]. In 2015, Mark *et al.* conducted a study that indicates the negative impact of multitasking which shows that when people engage in a second activity (driving or keep checking train stops) while using mobile applications, the more total screen switches to search the desired app, the less productive people feel at the end of the day. In

2019, to eliminate the negative impacts and distractions, Martelaro *et al.* stated that it is essential to have intelligent assistants to support practical app usage to increase productivity for app usage while people are on the move[16]. Thus, app usage prediction on the move as part of intelligent assistants benefits people's daily commuting time.

2.2. Contextual Information for App Usage Prediction

Modern applications use different contextual dimensions following the 5W1H approach (what, when, why, who, where, how) [19, 20]. Studies suggested that the dependency of spatio-temporal information also plays an important role in app use prediction. Eagle *et al.* show understandings on diverse app usage patterns based on different locations such as home and office and different times such as morning and lunchtime [21]. Later Verkasalo *et al.* analysed contextual patterns on app usage based on different contexts such as people are static, or on the move, [11]. Currently, there exist multiple contextual information analyses for app usage prediction while people are static. In 2012, Shin *et al.* discovered that the latest used app, Cell ID, and the hour of the day have a strong impact on app usage prediction. Later, Yan *et al.* discover that time of day and location clustering have a strong correlation to app usage prediction [22]. In 2018, Yu *et al.* further investigate spatial information as contextual information. In addition to the time of the day and location information, they use Point of Interest information to improve the app usage prediction accuracy [23].

However, there exist few studies on contextual information for app usage prediction on the move. While in 2019, Shen *et al.* exploit semantic location such as home and on the way as contextual information and predict app usage while users are on the move, the study only considers whether the user is moving or static in different locations without further discovery based on different types of moving [3], which is the focus of our study.

2.3. App Usage Prediction

Studies focused on app usage prediction using different machine learning algorithms. Major studies based on Markov and Bayesian framework to predict the next application exploiting a sequence of previously used applications and contextual information [24].

Currently, there are several studies for app usage prediction using a single Markov model or a mixture of Markov models [7, 8, 25]. In 2013, Natarajan *et al.* proposed iConRank, which use collaborative filtering to cluster users and cluster-level Markov models for app usage prediction. iConRank is tested using two datasets: the Apps dataset collected from $17,062$ users cross one year and LastFM collected from $1,000$ users and receive around 67% and 12.8% respectively for top-5 predictions. Later, Parate *et al.* uses the dataset that consists of 22 users and achieved an accuracy of about 81.9% with top-5 predictions based on a mixture of Markov models.

Other than the Markov model, Bayesian frameworks also are popularly applied for app prediction in different studies, such as user-NB (Naive Bayesian model based on different users) [26], 2-NB (2 features based Naive Bayesian model) [11], 3-NB (3 features based Naive Bayesian model) [27], and app-NB (Naive Bayesian model based on mobile applications) [28]. The Bayesian frameworks are based on either different users or mobile applications using contextual information as priors. Among the studies, app-NB proposed by Shin *et al.* achieve 88.2% accuracy with top-9 predictions [28].

Except for the methods based on machine learning, in 2019, Zhao *et al.* construct AppUsage2Vec model using Dual DNN and attention mechanism. The conducted experiment compared with multiple baselines using Markov models and Bayesian frameworks mentioned above. The study shows that AppUsage2Vec outperforms the baselines and reaches above 80% using the criterion of recall with top-4 predictions [4]. Furthermore, Shen *et al.* develop DeepApp based on Deep Reinforcement Learning which learns a model-free predictive neural network exploiting historical app usage data. DeepApp is tested on the dataset consists 443 active users for 21 days and achieved the recall of 46.7% for top-1 prediction [3].

In summary, existing studies show different app usage patterns while users are at a static location compare to situations where users are commuting. Note that a commute between two places may involve continuous changing locations over time. Thus, we propose a novel challenge which is to predict the next app while users are on the move.

3. Preliminaries and Problem Definition

In this part, we first present a formal definition of our problem. Then, we explain our collected data, followed by an illustration of how we pre-process our data. Finally, we discuss our preliminary discovery of the data.

Table 2: Description of raw features of the collected data.

Trajectory-Related Features	**Description**
TimeStamp	Start and end timestamp for trajectories
Duration	Derived from *Timestamp*, which records the time length of trajectories in seconds
GPS Location	A sequence of latitude and longitude that indicate physical locations of trajectories.
Travelling Mode	Transportation modes for trajectories such as train and bus.
Origin, Destination Place Mode	Modes (such as home and work) of origin and destination places for trajectories to indicate whether a user goes to the place on a regular-basis or not
App-Related Features	**Description**
Timestamp	Start timestamp for mobile applications
App Package Name	Package name of apps on Android phones
Foreground Time	Duration in seconds that mobile applications run in the foreground

3.1. Problem Formulation

Given a set of required contextual information C_u of user u, the set of n mobile applications S_n that are previously used in sequence, the app usage prediction while commuting problem is to predict the app x which belong to the set of apps A with the highest probability to be used next based on C_u and S_n. Thus, the whole process can be formulated as Equation 1.

$$arg\max_{x} P(x \mid C_u,\ S_n) \ \ where\ x\ \in\ A, \tag{1}$$

The contextual information C_u of user u includes any spatial-temporal information that is potentially correlated with the app usage pattern, such as start time of commuting and semantic meaning of the trajectory.

3.2. Dataset

To acquire a better understanding of work-related tasks and activities, a survey was developed to collect weekday data between 7 am and 8 pm over 4 weeks from people with different occupations who are using Android smartphones [29]. The dataset is collected from 53 users. Among the 53 users, there are 3 users without sufficient trajectory or app

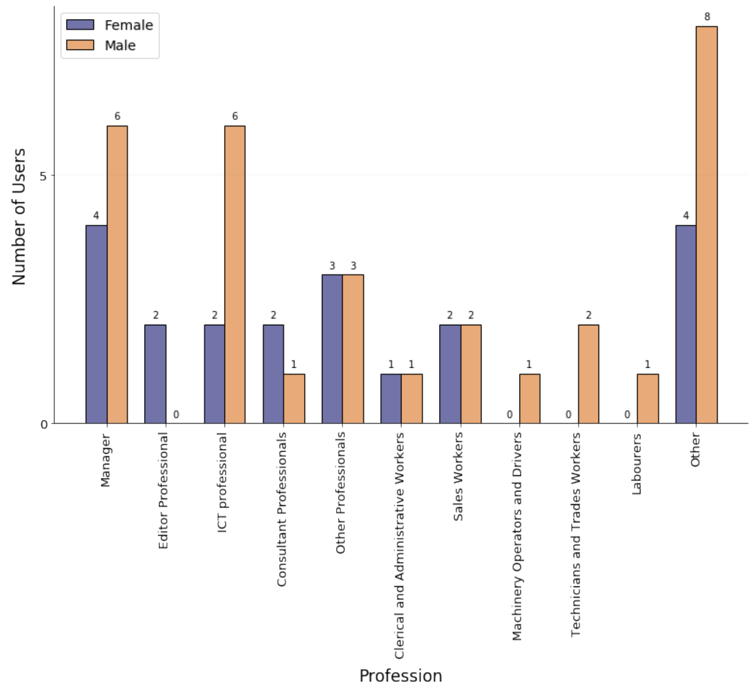


Figure 2: Number of users with different genders and different professions in the dataset.

usage information for this study. Hence, this study uses information from 50 users, including 12,495 trajectory records with more than 1,046 mobile applications recorded in the dataset. The dataset is collected from 30 males and 20 females with different professions, including Managers, Editor Professional, ICT professionals, Consultant Professionals, Clerical and Administrative Workers, Sales Workers, Machinery Operators and Drivers, Technicians and Trades Worker, Labourers, and Other professionals [1]. Figure 2 shows demographic details of the dataset.

The dataset contains different types of information that are associated with work-related activities, as shown in Table 2. The data capture human movements in daily life on weekdays via modelling and inference from multivariate time series data streamed from the mobile sensors embedded in Android smartphones [30]. In order to utilise the sensors, a variety of apps are installed on smartphones to collect cyber, physical, and social signals associated with different tasks [31]. In 2018, Liono *et al.* conducted a study to infer transportation modes based on the structured hierarchical contexts associated with human activities by

[1] Job categories are extracted from ANZSCO

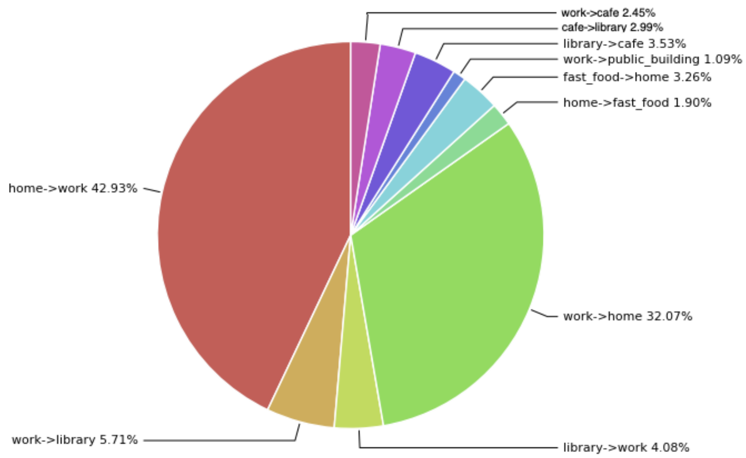


Figure 3: Top 10 travelling purposes and frequency

proposing CBAR (context-based Activity Recognition) [30].

3.3. Pre-processing

We clean and extract features from the collected data to determine what contextual information helps understand travel patterns and the corresponding app usage patterns.

3.3.1. Travelling Purpose Construction

There exist different methods to detect travelling paths for different users [32]. To analyse the purposes of different people travelling during weekdays, we extracted origin and destination amenities from GPS location to construct the travelling purposes. Figure 3 shows the travelling purposes constructed across all users with the highest frequency (i.e. Home→Work represents the purpose of the trajectory is go to work from home).

3.3.2. App Location Extraction

As the locations for mobile apps used on the move are constantly changing, we use the location when a mobile application is opened as the app location. Hence, according to Table 2, we use the timestamp in App-Related Features and match it with GPS location in Trajectory-Related Features to find app locations.

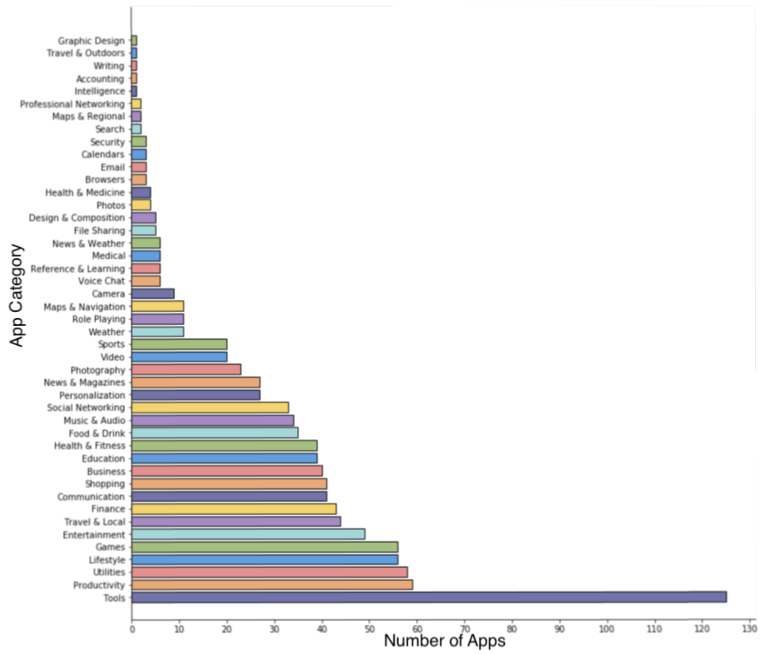


Figure 4: Categories of all mobile applications users used and the number of apps in each category

3.3.3. App Categorisation

In order to get a further understanding of the data, we extracted categories of apps from Google Play. The apps are categorised into 45 categories which are shown in Figure 4.

By pre-processing the collected data, the extracted features are listed in Table 3.

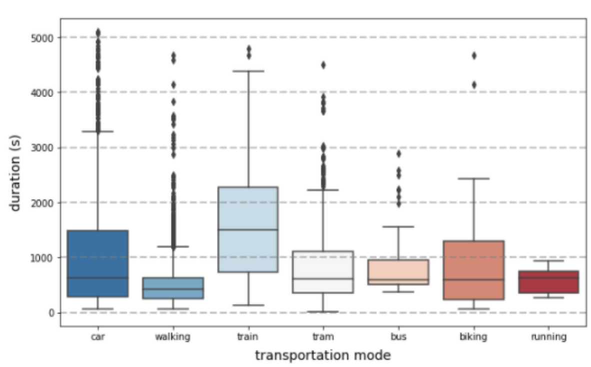
3.4. Preliminary Analysis

A brief insight into the data is essential for further analysis and app usage prediction while commuting. For example, Figure 5a shows the distribution of time spent for each trip based on different transportation modes across all users in the collected dataset. In general, people tend to use more apps when travelling on public transportation than other travelling methods.

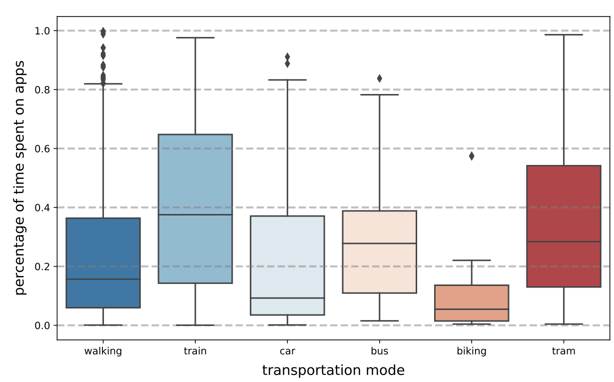
Figure 6 shows a general analysis based on collected data. Figure 6a shows the number of apps used during different transportation modes versus the percentage of travelling based on different transportation modes. According to the figure, people tend to use more apps on a train than other transportation modes, and people use at least 3 apps in about 60% of train travellings according to the data. Additionally, Figure 6b shows an analysis of the percentage of all users versus the number of mobile applications through the data collection

Table 3: Features available after pre-processing the data.

Trajectory-Related Features	**Description**
Day of Week (day_of_week)	The day of week of the trajectory started
Start Hour ($start\_hour$)	The start time (in hour) of the trajectory started
Duration ($duration$)	Duration (in seconds) of the trajectory
Travelling Mode ($mode$)	Transportation modes (train, tram, bus and etc.) for the trajectory
Travelling Purpose ($purpose$)	The purpose for travelling
Origin Amenity ($depart\_mode$)	The amenity of the origin location of the trajectory
Destination Amenity ($destination\_mode$)	The amenity of the destination of the trajectory
App-Related Features	**Description**
App Start Day of Week ($day\_of\_week\_app$)	Start time (day of week) of the app
App Start Hour ($start\_hour\_app$)	Start time (in hour) of the app
App Name (app_name)	Name of the app
Foreground Time ($foregrnd\_time$)	Foreground running time of the app
App Location ($app\_location$)	The location where the app started
App Category ($app\_category$)	The category of the app

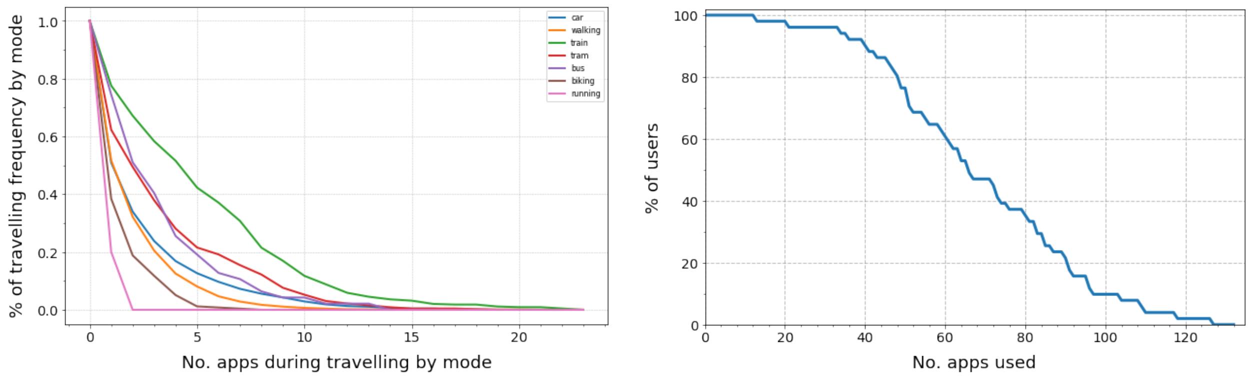


(a) Distribution of time spent on different transportation modes across all users.



(b) Distribution of time spent on apps during different transportation modes.

Figure 5: Preliminary analysis on transportation information across all users.



(a) Number of apps used during different *mode* of travelling.

(b) Number of apps used versus percentage of users.

Figure 6: App usage analysis across all users.

period. The number of apps ranges from 0 to 130, and most of the people use at least 40 different apps in 4 weeks, and when the number of apps used increases from 40 to 120, the percentage of the users decreases rapidly.

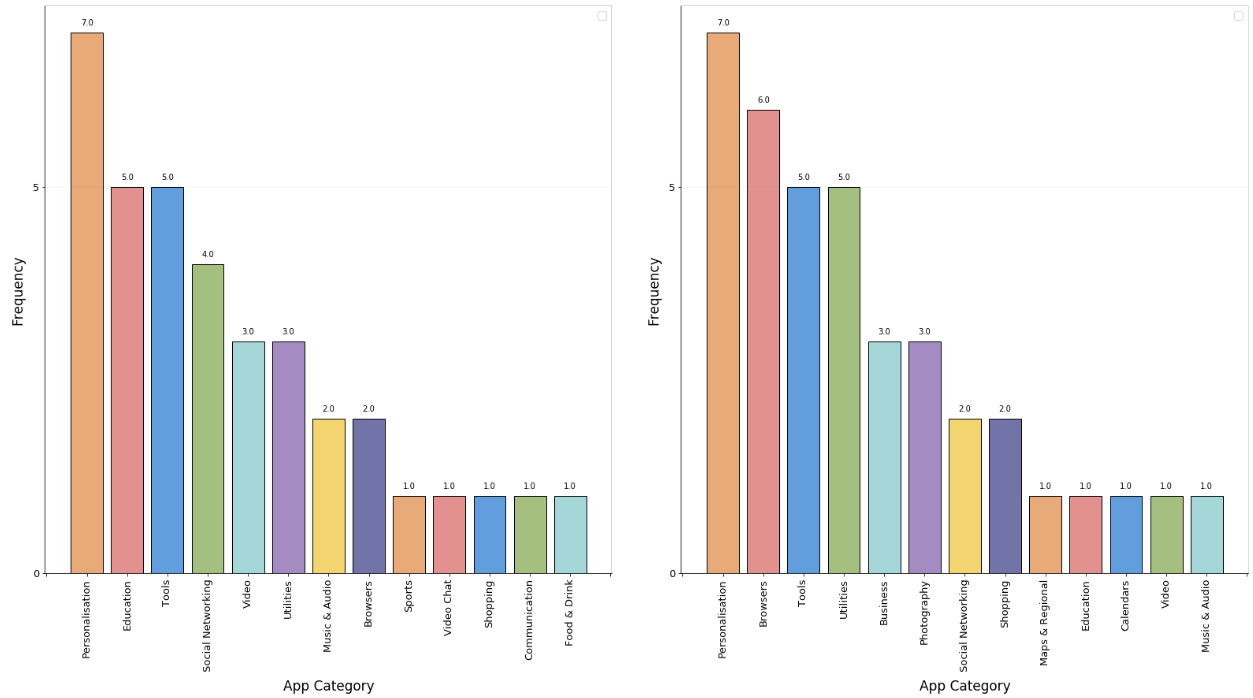
There are different app usage patterns based on different contextual information. According to the app categorisation, Figure 7 shows an analysis of app usage patterns of a user based on different times of the day while commuting – morning (from 6 am to 2 pm) and afternoon (from 2 pm to 7 pm). It shows the user prefers to use **Personalisation** and **Tools** throughout the whole day during travelling while he/she prefers to use **Education** in the morning and **Browsers** and **Utilities** in the afternoon. The preliminary analysis gives inspiration on the correlation between types of trajectories and app usage patterns.

4. Methodology

As shown in Figure 8, the framework(AppUsageOTM) requires contextual information analysis to determine the correlated features. We discuss the method for contextual information analysis and the overall construction of AppUsageOTM in the first and second subsections. Then we discuss each part of the framework in detail.

4.1. Contextual Information Analysis

To analyse and extract contextual information that is essential for app usage prediction while commuting, we applied Information Gain Ratio(IGR) to measure: (1) relationship between contextual features and destination amenities, (2) relationship between contextual



(a) In the morning.

(b) In the afternoon.

Figure 7: App usage patterns during commuting (frequency represents average app usage on weekly basis).

features and app usage prediction while commuting. According to the study by Kullback *et al.*, IGR for each feature is calculated by dividing target classes with their discrete values according to Equation 2 [33]:

$$IGR(X,a) = \frac{H(X) - \sum_{v \in values(a)} \left(\frac{|\{x \in X \mid value(x,a) = v\}|}{|X|} \times H(\{x \in X \mid value(x,a) = v\}) \right)}{- \sum_{v \in values(a)} \frac{|\{x \in X \mid value(x,a) = v\}|}{|X|} \times \log\left(\frac{|\{x \in X \mid value(x,a) = v\}|}{|X|}\right)} \tag{2}$$

where X represents the training set, $value(x, a)$ with $x \in X$ defines the value of a certain training sample x for attribute $a \in Attr$. $values(a)$ indicates all possible values of attribute $a \in Attr$, and H specifies the entropy.

4.2. AppUsageOTM: Framework Construction

Through the whole structure, we first used HLM to predict the Destination Amenity for a trajectory given the information of Day of Week, Start Hour and Origin Amenity, which are information that can be inferred from different sensors on smartphones [31]. Then, after getting the Destination Amenity and Transportation Mode, we build a filtering system to eliminate unnecessary app usage patterns to make the patterns consistent as inputs for SVM.

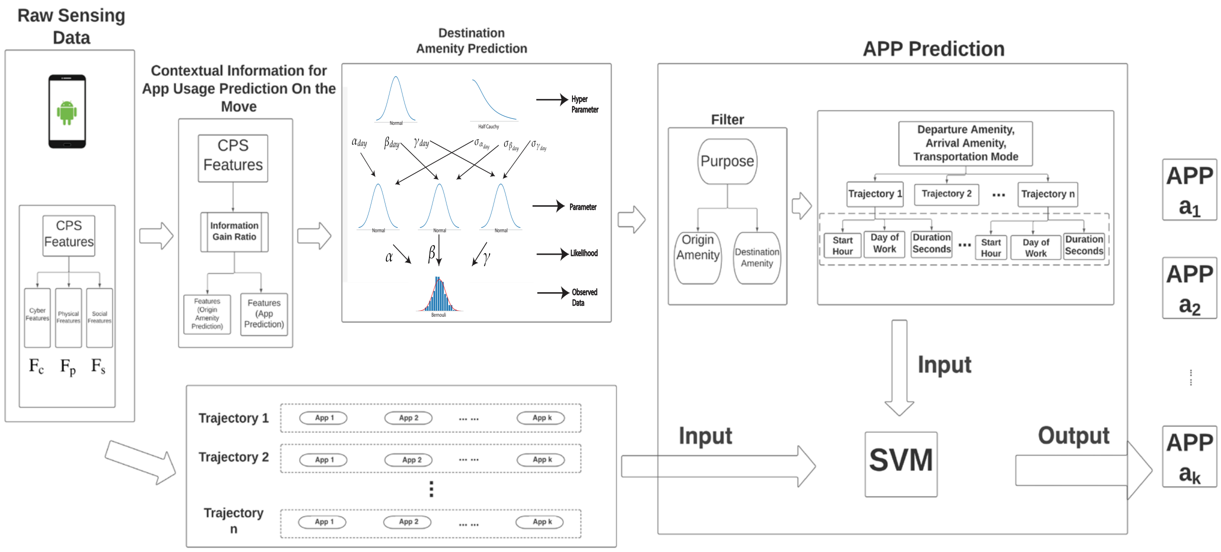


Figure 8: **AppUsageOTM**. By analysing Raw sensing data, we extract features for Destination Amenity Prediction and App Prediction accordingly, used in AppUsageOTM. In Destination Amenity Prediction, we use *Day of Week*, *Start Hour*, *Origin Amenity* as input features for prediction of *Destination Amenity*. Then, in App Prediction, by using *Origin Amenity* and *Destination Amenity* along with *Transportation Mode*, we can extract historical records of a certain portion of trajectories. With the contextual information of trajectories and the corresponding sequence of apps as SVM input, we predict the top-k apps that will be used next.

HLM for predicting the Destination Amenity can well adapt the hierarchical character of the data. HLM is a statistical method with low complexity and interpretable structure. The simplicity of the model can help to reduce the training time of our proposed model.

According to the study from Xing *et al.*, different studies have proved that Support Vector Machine (SVM) is an effective method for sequence classification [34]. Furthermore, SVM is a popular method for pattern recognition, and it requires less amount of training data compared to other Machine Learning algorithms [35].

To explain the first part in detail, each destination amenity was predicted separately using one stochastic process. Assume we use S^u_{end} represents the set of destination amenities for user u, hence m_i is the i-th element in S^u_{end}. h represents the start hour of the trajectory, and n_j is the origin amenity of the trajectory, which is an element in S^u_{start}, the set of origin amenities. To simplify the equation, we dropped the superscript u. We aim to predict

Table 4: Factors at each hierarchical level that affects destination amenities.

Hierarchical Level	Example of Hierarchical Level	Example Variables
Level-3	Day Level	day_of_week
Level-2	Time Level	$start\_hour$
Level-1	Trajectory Level	$depart\_mode$, $destination\_mode$

the destination amenity for the trajectory by calculating the max probability as shown in Equation 3.

$$\underset{m_i}{arg\ max}(Pr(m_i \mid h,\ n_j), \forall m_i\ \in\ S_{end} \tag{3}$$

As for the second part, by using HLM as a filtering system, we can get corresponding trajectories as contextual information. The sequences of applications used under the context construct our training data, and the output is the prediction of the next app.

4.3. Step 1: Hierarchical Logistic Regression

4.3.1. HLM: Background

HLM is a complex form of regression used when the variables are at varying hierarchical levels. It has been found that by using HLM, the classification model is simple with lower execution time and fewer computation units required [36].

Hierarchical Linear Modeling (HLM) is an ordinary least square (OLS) regression-based analysis that considers the hierarchical structure of the data. For example, in our data, trajectories occur on different days of the week through all weekdays. In this case, the structure of the data conflicts with the independence assumption of OLS regression because the clusters of observations are not independent of each other. HLM is a statistical method, and as its development occurred across different fields, it is now frequently used in the education, social work, business sectors, and health sectors [37].

HLM considers the shared variance in hierarchically structured data: it can accurately estimate lower-level slopes and their influences on estimating higher-level outcomes [37]. Table 4 shows an example using our collected data.

4.3.2. HLR for destination amenity prediction

Hierarchical Logistic Regression (HLR) is a model that is a part of HLM. It is proposed for studying data with a hierarchical structure and a binary response variable [38]. It can be applied to hierarchical levels of grouped data [37].

Consider if we only used logistic regression, the probability for each destination amenity can be predicted by Equation 4.

$$\begin{aligned} &value = \alpha \times h + \beta \times n_j + \gamma \\ &\Pr(m_i = 1 \mid h,\ n_j\ ) = \frac{1}{1 + e^{-value}}\ ,\ \forall m_i \in S_{end} \end{aligned} \tag{4}$$

where $value$ is the intermediate output of logistic regression, α and β are the parameters for feature h and n_j, and γ is the bias.

However, different days of the week have different patterns, hence, the set of weights for each feature should be different for a different day of the week. At the same time, as we only considering weekday data, while a different day of the week has different patterns, they still share some shared characteristics. In this case, we have the hierarchical logistic regression to predict destination amenity as shown in Figure 8 (Destination Amenity Prediction).

We distribute the set of weights for a different day of the week using a shared group distribution as the hyperparameters sampled by Normal distributions (N):

$$\alpha_{day} \sim N\left(\mu_\alpha,\ \sigma_\alpha^2\right), \beta_{day} \sim N\left(\mu_\beta,\ \sigma_\beta^2\right), \gamma_{day} \sim N\left(\mu_\gamma,\ \sigma_\gamma^2\right), \tag{5}$$

where $\mu_\alpha, \mu_\beta, \mu_\gamma$ represent the means for the three Normal distributions accordingly, and $\sigma_\alpha^2, \sigma_\beta^2, \sigma_\gamma^2$ represent variance of the three Normal distributions accordingly.

To model the uncertainty of the shared group distribution, we define with a HalfCauchy (HC) with its parameter is 4:

$$\sigma_{\alpha_{day}} \sim HC(4), \sigma_{\beta_{day}} \sim HC(4), \sigma_{\gamma_{day}} \sim HC(4) \tag{6}$$

Accordingly, we can define parameters α, β, γ with Normal distribution (N) as follow.

$$\alpha \sim N\left(\alpha_{day},\ \sigma^2_{\alpha_{day}}\right), \beta \sim N\left(\beta_{day},\ \sigma^2_{\beta_{day}}\right), \gamma \sim N\left(\gamma_{day},\ \sigma^2_{\gamma_{day}}\right) \tag{7}$$

Finally, by using the parameters, according to logistic regression, we use Sigmoid function to reveal the likelihood of different destination amenities:

$$Pr(m_i \mid h,\ n_j\ )\ \sim\ Ber(p), \tag{8}$$

where $p\ =\ Pr(m_i\ =\ 1 \mid h,\ n_j\ )\ =\ \frac{1}{1\ +\ e^{-value}}$.

4.4. Step 2: SVM for app usage prediction

According to Figure 8 (App Prediction), we aim to use a sequence of apps along with contextual information to predict the next app.

The basic idea is to map a sequence into a feature space, find the maximum-margin hyperplane to separate classes, and find the probability of classifying the record into the class. By using SVM, we aim to find the probability for each app in S_{end} as the next app:

$$Pr\ (x\ =\ 1 \mid C_u,\ S_k),\ \forall x\ \in A \tag{9}$$

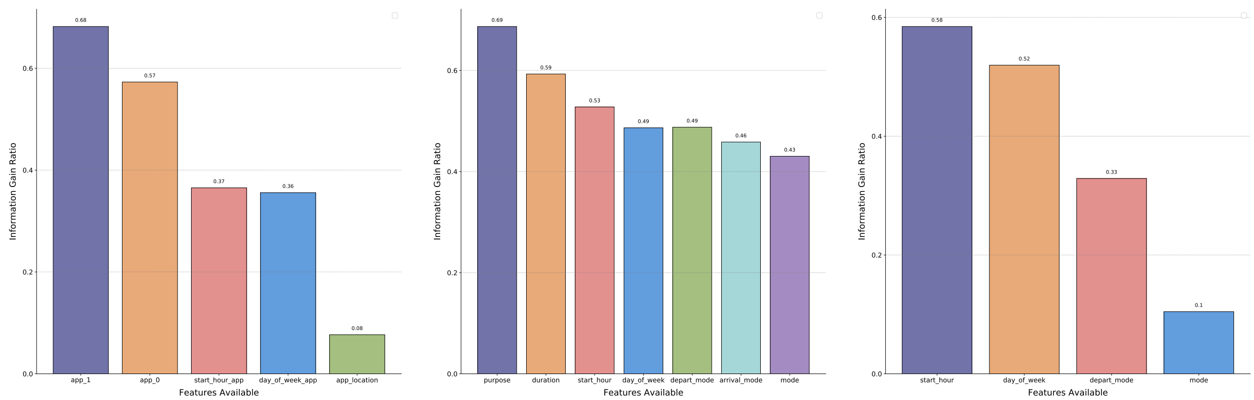
5. Experiments

To evaluate AppUsageOTM, we first examine which contextual information is essential for app usage prediction while commuting, then we construct baselines using existing methods for app usage prediction. To make the experiments sufficient for the study, we conduct an ablation study to show the necessity of different features as contextual information. Finally, we conduct further discussion based on travelling modes and app categorisations.

5.1. Contextual Information for App Usage Prediction On the Move

In order to get an understanding of what contextual information is strongly correlated to App Usage Prediction On the move, we use IGR to evaluate different parts in AppUsageOTM (as shown in Figure 8), and the results are shown in Figure 9.

According to previous studies on App Usage Prediction, we use *day_of_week_app*, *start_hour_app*, and *app_location* as spatial-temporal contextual information to calculate the IGR, and we also use n apps in sequence that are previously used [22, 23]. Figure 9a shows the result where we use $n = 2$. Based on the result, apps in sequence have high IGR for App Prediction in AppUsageOTM, in which *app_1*, the most recent used app, have the highest IGR followed



(a) Average IGR of app-related features to App Prediction in AppUsageOTM across all users.

(b) Average IGR of trajectory-related features to App Prediction in AppUsageOTM across all users.

(c) Average IGR of availiable features to Destination Amenity Prediction in AppUsageOTM across all users.

Figure 9: Average IGR of different parts in AppUsageOTM across all users.

with *app_0*. However, app-related spatial-temporal information show weaker correlation to App Prediction. According to Figure 9a, though *day_of_week_app* and *start_hour_app* show comparably higher calculated IGR, *app_location* show that there is little correlation between locations and the next app used.

We conducted an additional experiment to calculate IGR of trajectory-related features for App Prediction in AppUsageOTM. According to Figure 9b, *purpose* shows the highest IGR followed by *duration*, *start_hour*, *day_of_week*, *depart_mode*, *destination_mode*, and *mode*.

Additionally, the values of *start_hour_app* is similar to the values of *start_hour*, which shows users tend to start using apps at the same hour they start travelling. As *start_hour* shows higher correlation compare to *start_hour_app*, we keep *start_hour* as the feature in AppUsageOTM. Based on Figure 9a and Figure 9b, the features ranked by IGR are in the sequence of *purpose*, *duration*, *start_hour*, *day_of_week*, *day_of_week_app*, *mode*, and *start_hour_app*.

As *purpose* is highly correlated to App Usage Prediction On the Move, and it is unrealistic to acquire *destination_mode* before the travelling is completed, AppUsageOTM require to predict *destination_mode* (Destination Amenity Prediction). Figure 9c show IGR of trajectory-related features for prediction of *destination_mode*. We use *depart_mode*, *start_hour*, and *day_of_week* to predict *destination_mode*.

Table 5: Selected features for AppUsageOTM.

	Features
Features for Destination Amenity Prediction	*depart_mode*, *start_hour*, *day_of_week*
Features for Filter	*purpose* (constructed using *depart_mode* and *destination_mode*)
Features for App Prediction	sequence of apps (e.g. *app_1*, *app_0*), *duration*, *start_hour*, *day_of_week*, *mode*, *start_hour_app*

In summary, we discover that the contextual information for App Usage Prediction On the Move is different from App Usage Prediction. Compare to app-related features, trajectory-related features have a higher correlation to predict the next app while commuting. As *purpose* have the highest IGR for App Prediction in AppUsageOTM, and it is constructed using the output from Destination Amenity Prediction in AppUsageOTM, we use *purpose* to construct the filter (see Figure 8) to eliminate unnecessary data so that the records for App Prediction in AppUsageOTM have similar patterns. Table 5 shows the features for AppUsageOTM.

5.2. AppUsageOTM Prediction Performance and Evaluation

5.2.1. Evaluation Metrics

We used the criterion of accuracy to measure the performance of AppUsageOTM. The accuracy was computed when top k apps with the highest probability were selected, named as *Accuracy@k*. *Accuracy@k* is a typical metric for evaluating app usage prediction, such as used in [39], which is calculated as shown in Equation 10. We tested with k ranges from 1 to 10 in the following experiments.

$$Accuracy@k = \frac{\sum_{i=1}^{|D^{Test}|} 1\left(y_{real} \in Y_{predict}\right)}{|D^{Test}|}, \tag{10}$$

where $|Y_{predict}| = k$ and D^{Test} represents testing dataset.

5.2.2. Baseline Approaches

The following algorithms were selected as baselines. We tested with *Accuracy@k* against our proposed framework (AppUsageOTM) using MRU, MFU, App-NB, Markov Chain, and

AppUsage2Vec. We further investigated the methods of LR+SVM (Logistic Regression and SVM) and SVM to examine the performance of our proposed framework.

- **MRU**: We use a fixed length of sequences of apps where the length is window size (n) [28]. In this case, the output for MRU is the last app in the app sequence.
- **MFU**: Similar to MRU, we used a sequence of n apps as input for MFU [28]. However, in this study, if every app in the sequence has the same frequency (e.g. frequency is 1), the output of MFU is the same as MRU.
- **App-NB (App Usage Prediction)**: We use App-NB based on its implementation for App Usage Prediction to examine its performance on the collected data eliminating its characteristic that the data the app usage information while users are moving. According to the study by Shin *et al.*, we use n apps in sequence, $app\_location$, $start\_hour\_app$ (see Table3) to implement the model [28].
- **App-NB (App Usage Prediction On the Move)**: We implement App-NB using the features for AppUsageOTM to examine its performance on the data [28].
- **Markov Chain (App Usage Prediction)**: We constructed Markov Chain for app usage prediction using joint probability of the sequence of n apps, app-related features, and the target app based on Markov chain rule [40] to examine its performance on the collected data.
- **Markov Chain (App Usage Prediction On the Move)**: We constructed Markov Chain for app usage prediction using joint probability of the sequence of n apps, the selected features for AppUsageOTM, and the target app based on Markov chain rule [40] to examine its performance on the collected data.
- **LSTM (App Usage Prediction)**: According to the structure of LSTM [41], we aim to use n apps in sequence (app_name, $foregrnd\_time$ for each app), $day\_of\_week\_app$, $start\_hour\_app$, $app\_location$, and user id (unique id assigned to each user) to examine its performance using the collected data eliminating the characteristic.

- **LSTM (App Usage Prediction On the Move)**: According to the structure of LSTM [41], we aim to use n apps in sequence (app_name, $foregrnd\_time$ for each app), the selected features for AppUsageOTM, and user id (unique id assigned to each user) to examine its performance using the collected data considering the characteristic.

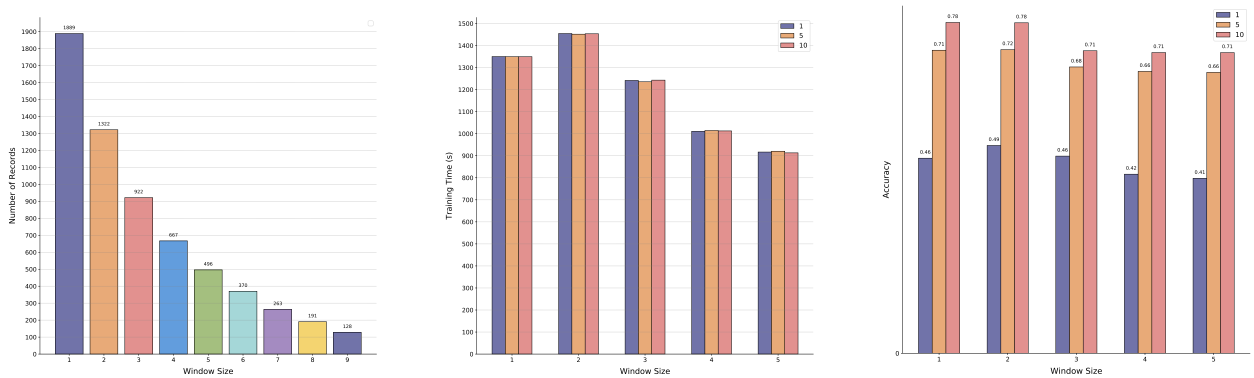
- **AppUsage2Vec (App Usage Prediction)**: According to the study by Zhao *et al.*, we constructed the model AppUsage2Vec using n apps in sequence (app_name, $foregrnd\_time$ for each app), day_of_week, $start\_hour\_app$ (see Table 3), and user id (unique id assigned to each user) to examine its performance using the collected data eliminating the characteristic [4].

- **AppUsage2Vec (App Usage Prediction On the Move)**: According to the study by Zhao *et al.*, the model consider temporal features as contextual information [4]. Thus, to maintain the structure of AppUsage2Vec, we consider the temporal features designed for App Usage Prediction On the Move. We use n app in sequence (app_name, $foregrnd\_time$), day_of_week, $start\_hour$ (see Table 5), and user id (unique id assigned to each user) to examine its performance using the collected data considering the characteristic of the data.

5.2.3. Performance and Results

We first investigated the performance of AppUsageOTM based on different window size n shown in Figure 10c. Window size n refers to the number of apps in a sequence used as input for predicting the next app. Hence, when we use a certain window size n for experiments, we use in total $n+1$ apps (the n apps in sequence for training and the last 1 app be the prediction result). We further investigated with execution time of AppUsageOTM with varying window sizes, and the result is shown in Figure 10. Training time refers to the data training time using the first 80% of data in the temporal domain with certain window size.

As shown in Figure 10c, we tested *Accuracy@k* where k ranged from 1 to 10, by varying window size from 1 to 5 according to the number of apps used in different trajectories. It can be seen that accuracy increases when window size n increases from 1 to 2. It shows that adding more recently used apps leads to more information for app usage patterns which helped for improving the accuracy of app usage prediction. However, when window size



(a) Illustration of number of records based on window size n.

(b) Illustration of variation of training time based on different window size.

(c) Prediction Performance with different window sizes.

Figure 10: Training time illustration based on window size and number of records.

increasing to 5, the accuracy slightly decreases. It can be because of the limited amount of training data. Another reason can be that when the window size n increases, it increases the difficulty in fitting the model due to increased computation complexity.

The model achieves the highest accuracy when n is 2 for all different *Accuracy*@k, and there is a significant increase from *Accuracy*@1 to *Accuracy*@5. On the other hand, accuracy starts to decrease when $n = 3$, which shows a longer sequence of apps does not always lead to higher accuracy, which further examined the conclusion from a study by Parate *et al.* [42].

Figure 10 shows the training time of AppUsageOTM based on different n. According to the figure, the execution time for $k = 1, 5, 10$ is almost the same for different value of n. Through the figure, we can see that training time is not necessarily increases based on the increase of window size n. However, it is correlated to the number of records as shown in Figure 10a. In general, when the number of records decreases, the training time decreases accordingly.

Based on Figure 10c and 10b, when n is 2, the accuracy for our model is the highest while the training time is acceptable. Hence, about 1250 sampled app sequences were used for the following experiments; each of the sequences has 3 apps, and we use the first 2 apps in sequence to predict the last one.

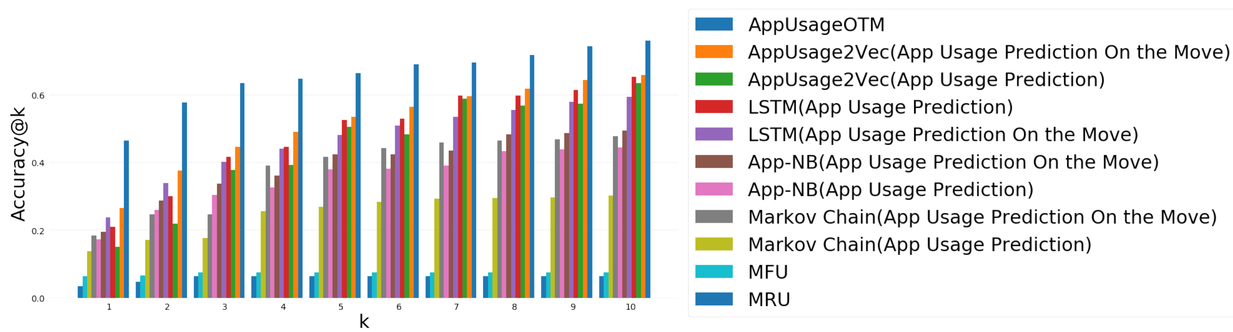


Figure 11: The performance of the proposed method (AppUsageOTM) in comparison to the baselines.

5.2.4. Comparison with baselines

Figure 11 shows AppUsageOTM compare with the baselines. We aim to have the highest accuracy with the smallest number of predictions.

Figure 11 shows that for any existing models, our selected features outperforms app-related features, which means the collected data for app usage prediction on the move has unique characteristics compare to app usage prediction. Additionally, AppUsageOTM outperforms other models for any number of predictions. As shown in the figure, when k increases, the improvement of accuracy becomes less significant. When $k = 8$, the accuracy of app usage prediction reaches above 0.8, and there are minor improvements when k is larger than 8, which shows the performance is getting stable.

Among all the baselines, AppUsage2Vec (App Usage Prediction On the Move) achieves the highest accuracy. For further analysis, we have done Paired T-Test to compare AppUsageOTM and AppUsage2Vec over the range of k. The *p-value* is about **2.441e-11**, which shows a significant improvement using our proposed framework in comparison to AppUsage2Vec. The lower accuracy for AppUsage2Vec can be resulted by 1) AppUsage2Vec is constructed using Deep Neural Network (DNN), and it requires a comparably larger amount of data for training, and 2) the features for App Usage Prediction On the Move is consist of app features and trajectory features, but we can only use parts of the features in AppUsage2Vec due to its model structures. According to [4], AppUsage2Vec exploits temporal features as contextual information. Hence, by substituting the features from App Usage On the Move from App Usage Prediction, we still cannot exploit essential features such as *purpose* (see Table 3) as it is a spatial feature.

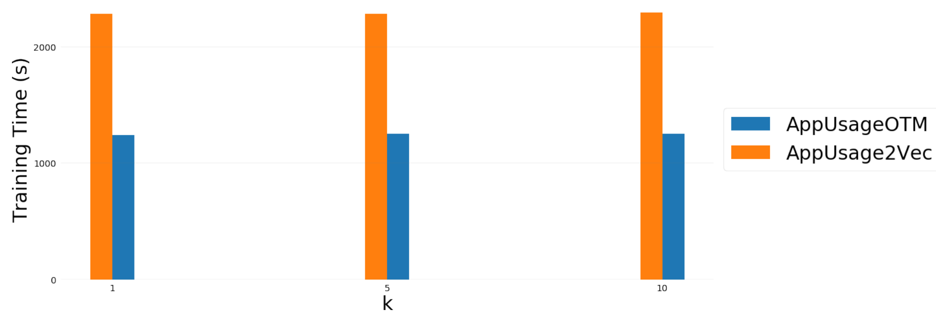


Figure 12: Training time (s) for AppUsageOTM and AppUsage2Vec based on various top k predictions.

We further test the training time compare AppUsageOTM to AppUsage2Vec shown in Figure 12 using the first 80% of the data for training based on a temporal split with window size $n = 2$. The average training time is calculated using the total training time divided by the total times of training. We conduct all experiments under the same environment, Ubuntu 16.04, with Intel 5820k and 32GB memory. According to Figure 12, training time is almost the same for different values of k, which coincident with Figure 10b. It also shows that the training time for AppUsageOTM is about half of the training time for AppUsage2Vec.

5.3. Contributions of the different features

In order to evaluate the contribution of different parts in AppUsageOTM, we accomplished a large set of experiments based on subsets of the selected features shown in Table 5. Therefore, we first use subsets of the selected features to test Destination Amenity Prediction.

Destination Amenity Prediction: we conduct experiments for Destination Amenity Prediction use all the features listed in Table 5 to compare with the results of using subsets of the features. **LR** (*start_hour*): according to Table 4, *day_of_week* is the hyper-parameter for HLR (we use HLR for Destination Amenity Prediction as shown in Figure 8). Thus, by only use *start_hour* for Destination Amenity Prediction, we can only use LR (Logistic Regression) instead of HLR. **LR** (*depart_mode*): according to LR (*start_hour*), we use LR to test the feature. **LR** (*start_hour*+*depart_mode*): according to LR (*start_hour*), we use LR (*start_hour* + *depart_mode*) to test the effect of the two features, and it tests the effect of HLR structure for Destination Amenity Prediction. **HLR** (*day_of_week*+*start_hour*): we eliminate *depart_mode* to test its importance. **HLR** (*day_of_week*+*depart_mode*): we

Table 6: Ablation study on Destination Amenity Prediction in AppUsageOTM.

	Accuracy
Destination Amenity Prediction	0.8645
LR (*start_hour*)	0.7273
LR (*depart_mode*)	0.6441
LR (*start_hour+depart_mode*)	0.7337
HLR (*day_of_week+start_hour*)	0.8182
HLR (*day_of_week+depart_mode*)	0.7606

Table 7: Ablation study on App Prediction in AppUsageOTM.

	Accuracy@1
App Prediction	0.4644
SVM (without Filter)	0.3691
SVM (without sequence of apps)	0.3629
SVM (without *duration*)	0.3927
SVM (without *start_hour*)	0.4563
SVM (without *day_of_week*)	0.4116
SVM (without *mode*)	0.4078
SVM (without *start_hour_app*)	0.4644

eliminate *start_hour* to test its importance.

Table 6 lists sample results of training Destination Amenity Prediction in AppUsageOTM based on subsets of the available features compare with using all the features. In general, the accuracy of Destination Amenity Prediction is higher with HLR than LR, which shows the importance of building the model according to the hierarchical structure of the data. Furthermore, *start_hour* shows a more significant impact on Destination Amenity Prediction than *depart_mode*, and by adding *depart_mode* into the model, the performance shows moderate improvement. We then use subsets of the selected features to test App Prediction.

App Prediction: we conduct experiment for App Prediction use all the features listed in Table 5 to compare with the results of using subsets of the features. **SVM (without Filter)**: we eliminate Filter (see Figure 8). **SVM (without sequence of apps)**: we eliminate sequence of apps. **SVM (without *duration*)**: we eliminate *duration*. **SVM (without *start_hour*)**: we eliminate *start_hour*. **SVM (without *day_of_week*)**: we eliminate *day_of_week*. **SVM (without *mode*)**: we eliminate *mode*. **SVM (without**

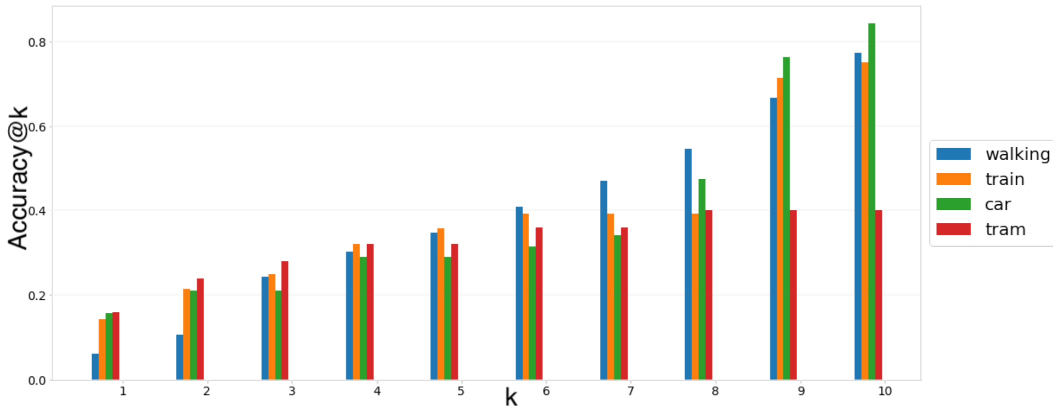


Figure 13: *Accuracy@k* across different transportation modes

start_hour_app): we eliminate *start_hour_app*.

Table 7 lists sample results using Accuracy@1 for training App Prediction in AppUsageOTM based on subsets of the available features compare with using all features. The result shows that both Filter and the sequence of apps have the most significant impact on the result. Additionally, the accuracy does not significantly decrease when eliminating *start_hour_app*.

For further analysis, the significant improvement of Destination Amenity Prediction compare to LR (*start_hour* + *depart_mode*) and App Prediction compare to SVM (without Filter) shows: (1) it is necessary to build a framework according to the hierarchical structure of the data, and (2) eliminating unnecessary app usage pattern using Filter leads to more consistent patterns that help to improve app prediction accuracy.

5.4. Discussion

5.4.1. Performance study based on different transportation modes

In this section, we investigated the prediction performance for different transportation modes. Specifically, for every transportation mode shows in Figure 5, we extracted data accordingly. However, we cannot train the framework for cycling mode due to the data limitation and the pattern variation (10 records from 4 users).

In Figure 13, walking remains stable based on *Accuracy@k*, where $\forall k \in [1, 10]$, $k \in N$. From the observation, the walking trail usually remains the same every day for every user. Thus, the walking patterns are also similar. As shown in Figure 5a, both walking and tram tend to be done as short trips. Trams are often performed as inner-city trips for wider

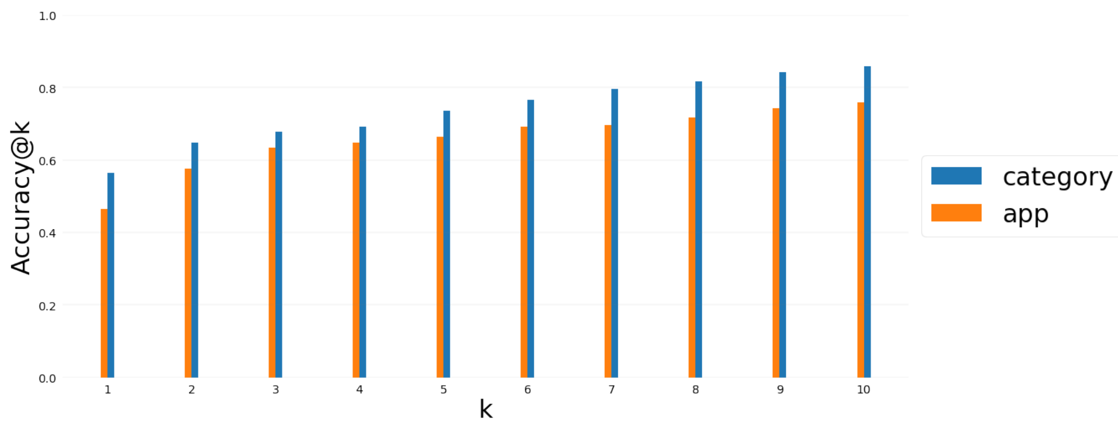


Figure 14: Accuracy@k comparison between category and app prediction

purposes, apart from the regular morning or afternoon commute. With a wider variety of trip purposes on short trips, a broader range of apps could be used, which increases the difficulties of accurately predicting the next app.

5.4.2. Performance based on app-categories prediction

We further investigated the performance of our model on the granular level. Instead of a prediction of the next app, we focused on predicting what category the next app falls in according to Figure 4. We compared category prediction and app prediction using our framework shown in Figure 14. The prediction results on a granular level are better, as shown in Figure 14. It shows that our predicted apps are in the same categories as the actual apps used. Specifically, there is a substantial improvement for our top-1 result for category prediction, around 10% higher than app prediction (57% for category prediction and 46% for app prediction). The apps that fall in the same category have similar functionality, which means if we use our predicted app as a recommendation for the user, the user may use our prediction instead of what he/she tends to use (such as both our predicted app and the actual app fall in Game category). It means our proposed framework is applicable in the real world. For example, Figure 15 shows that Messaging is the next app after the first four apps, but our prediction is WhatsApp. The reason for our prediction is that WhatsApp was used previously in the sequence, which gives us more weight to predict. As WhatsApp and Messaging are in the same category, it shows the intention to use communication applications.

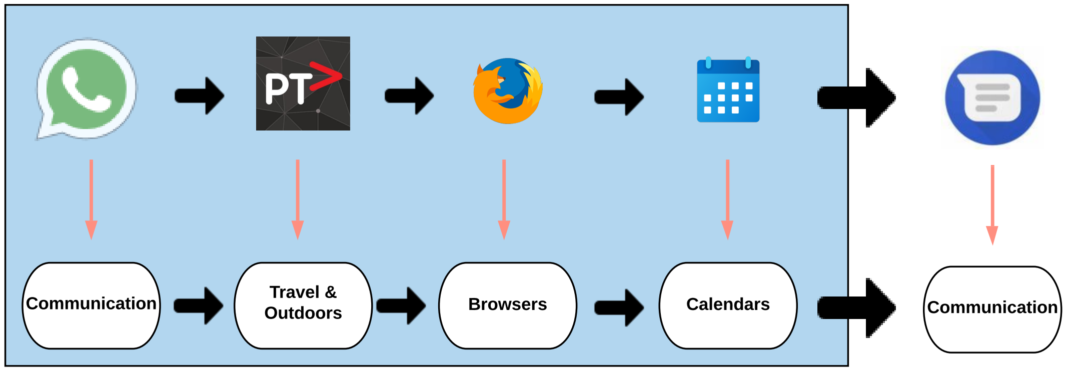


Figure 15: A case study where the first four apps are used as features and the last app suppose to be our app prediction. The categorisation of the five apps are illustrated at the lower part of the figure.

6. Conclusion

We proposed AppUsageOTM, a framework that consists of a statistical model: Hierarchical Logistic Regression(HLR), and Support Vector Machine (SVM), to predict the next app while commuting. We introduced a filtering system using HLR to eliminate unnecessary app usage patterns to improve app usage prediction accuracy. Additionally, by using HLR, we made our framework fit for hierarchically structured data.

Extensive experiments were conducted on a real-world dataset that consists of weekday-travels and app usage behaviours from 50 users over 4 weeks. We found that users spend much time on mobile devices while commuting, even by riding a bike, making our study necessary to improve productivity and help with attention management during their travelling time. On weekdays, users tend to go to the same place regularly, and app usage patterns are consistent throughout the week if the purposes of the trajectories are the same. We identified that several contexts such as start time and start location of trajectories and the most recently used apps have a significant influence on subsequent app usage. Finally, we conducted a baseline comparison to show that our framework outperforms the baselines.

AppUsageOTM can predict the next app and capture the app usage patterns based on different travel modes. With further evaluation, our proposed framework can predict the categories of the next app with higher accuracy in comparison to app usage prediction. Additionally, as apps that fall in the same category share similar functionalities, it shows that our proposed framework can accurately predict app usage intention.

7. Acknowledgements

This research was partially supported by Microsoft Research through Microsoft-RMIT Cortana Intelligence Institute.

References

[1] M. Satyanarayanan, Swiss army knife or wallet?, IEEE Pervasive Computing 4 (2) (2005) 2–3.

[2] Z.-X. Liao, S.-C. Li, W.-C. Peng, S. Y. Philip, T.-C. Liu, On the feature discovery for app usage prediction in smartphones, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 1127–1132.

[3] Z. Shen, K. Yang, W. Du, X. Zhao, J. Zou, Deepapp: a deep reinforcement learning framework for mobile application usage prediction, in: Proceedings of the 17th Conference on Embedded Networked Sensor Systems, 2019, pp. 153–165.

[4] S. Zhao, Z. Luo, Z. Jiang, H. Wang, F. Xu, S. Li, J. Yin, G. Pan, Appusage2vec: Modeling smartphone app usage for prediction, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE), IEEE, 2019, pp. 1322–1333.

[5] F. Colace, M. De Santo, M. Lombardi, F. Pascale, D. Santaniello, A. Tucker, A multilevel graph approach for predicting bicycle usage in london area, in: Fourth International Congress on Information and Communication Technology, Springer, 2020, pp. 353–362.

[6] K. Huang, X. Ma, C. Zhang, G. Chen, Predicting mobile application usage using contextual information, UbiComp'12 - Proceedings of the 2012 ACM Conference on Ubiquitous Computing (2012) 1059–1065doi:10.1145/2370216.2370442.

[7] N. Natarajan, D. Shin, I. S. Dhillon, Which app will you use next? Collaborative filtering with interactional context, RecSys 2013 - Proceedings of the 7th ACM Conference on Recommender Systems (2013) 201–208.

[8] A. Parate, M. Böhmer, D. Chu, D. Ganesan, B. M. Marlin, Practical prediction and prefetch for faster access to applications on mobile phones, UbiComp 2013 - Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (2013) 275–284doi:10.1145/2493432.2493490.

[9] W. Shao, F. D. Salim, A. Song, A. Bouguettaya, Clustering big spatiotemporal-interval data, IEEE Transactions on Big Data 2 (3) (2016) 190–203.

[10] W. Shao, A. Prabowo, S. Zhao, S. Tan, P. Koniusz, J. Chan, X. Hei, B. Feest, F. D. Salim, Flight delay prediction using airport situational awareness map, in: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2019, pp. 432–435.

[11] H. Verkasalo, Contextual patterns in mobile service usage, Personal and Ubiquitous Computing 13 (5) (2009) 331–342. doi:10.1007/s00779-008-0197-0.

[12] A. Abdiansah, R. Wardoyo, Time complexity analysis of support vector machines (svm) in libsvm, International journal computer and application 128 (3) (2015) 28–34.

[13] H. Alm, L. Nilsson, The effects of a mobile telephone task on driver behaviour in a car following situation, Accident Analysis & Prevention 27 (5) (1995) 707–715.

[14] V. Briem, L. R. Hedman, Behavioural effects of mobile telephone use during simulated driving, Ergonomics 38 (12) (1995) 2536–2562.

[15] F. Amato, F. Moscato, V. Moscato, F. Pascale, A. Picariello, An agent-based approach for recommending cultural tours, Pattern Recognition Letters 131 (2020) 341–347.

[16] N. Martelaro, J. Teevan, S. T. Iqbal, An exploration of speech-based productivity support in the car, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, pp. 1–12.

[17] S. T. Iqbal, Y.-C. Ju, E. Horvitz, Cars, calls, and cognition: Investigating driving and divided attention, in: Proceedings of the SIGCHI conference on human factors in computing systems, 2010, pp. 1281–1290.

[18] F. Colace, M. De Santo, M. Lombardi, F. Mercorio, M. Mezzanzanica, F. Pascale, Towards labour market intelligence through topic modelling, in: Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019.

[19] G. Annunziata, F. Colace, M. De Santo, S. Lemma, M. Lombardi, Appoggiomarino: A context aware app for e-citizenship., in: ICEIS (2), 2016, pp. 273–281.

[20] F. Clarizia, S. Lemma, M. Lombardi, F. Pascale, A mobile context-aware information system to support tourism events, in: International Conference on Green, Pervasive, and Cloud Computing, Springer, 2017, pp. 553–566.

[21] N. Eagle, A. Pentland, Reality mining: Sensing complex social systems, Personal and Ubiquitous Computing 10 (4) (2006) 255–268. doi:10.1007/s00779-005-0046-3.

[22] T. Yan, D. Chu, D. Ganesan, A. Kansal, J. Liu, Fast app launching for mobile devices using predictive user context, in: Proceedings of the 10th international conference on Mobile systems, applications, and services, 2012, pp. 113–126.

[23] D. Yu, Y. Li, F. Xu, P. Zhang, V. Kostakos, Smartphone app usage prediction using points of interest, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1 (4) (2018) 1–21.

[24] H. Cao, M. Lin, Mining smartphone data for app usage prediction and recommendations: A survey, Pervasive and Mobile Computing 37 (2017) 1–22. doi:10.1016/j.pmcj.2017.01.007.

[25] V. Kostakos, D. Ferreira, J. Goncalves, S. Hosio, Modelling smartphone usage: a markov state transition model, in: Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing, 2016, pp. 486–497.

[26] M. Matsumoto, R. Kiyohara, H. Fukui, M. Numao, S. Kurihara, Proposition of the context-aware interface for cellular phone operations, Proceedings of INSS 2008 - 5th International Conference on Networked Sensing Systems (Kdd 96) (2008) 233. doi:10.1109/INSS.2008.4610872.

[27] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, G. Bauer, Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage, Mobile HCI 2011 - 13th International Conference on Human-Computer Interaction with Mobile Devices and Services (2011) 47–56.

[28] C. Shin, J. H. Hong, A. K. Dey, Understanding and prediction of mobile application usage for smart phones, UbiComp'12 - Proceedings of the 2012 ACM Conference on Ubiquitous Computing (2012) 173–182.

[29] J. R. Trippas, D. Spina, F. Scholer, A. H. Awadallah, P. Bailey, P. N. Bennett, R. W. White, J. Liono, Y. Ren, F. D. Salim, et al., Learning about work tasks to inform intelligent assistant design, in: Proceedings of the 2019 Conference on Human Information Interaction and Retrieval, 2019, pp. 5–14.

[30] J. Liono, Z. S. Abdallah, A. K. Qin, F. D. Salim, Inferring transportation mode and human activity from mobile sensing in daily life, in: Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018, pp. 342–351.

[31] J. Liono, J. R. Trippas, D. Spina, M. S. Rahaman, Y. Ren, F. D. Salim, M. Sanderson, F. Scholer, R. W. White, Building a benchmark for task progress in digital assistants, Proceedings of WSDM (2019).

[32] F. Colace, M. De Santo, M. Lombardi, R. Mosca, D. Santaniello, A multilayer approach for recommending contextual learning paths., J. Internet Serv. Inf. Secur. 10 (2) (2020) 91–102.

[33] S. Kullback, R. A. Leibler, On information and sufficiency, The annals of mathematical statistics 22 (1) (1951) 79–86.

[34] Z. Xing, J. Pei, E. Keogh, A brief survey on sequence classification, ACM SIGKDD Explorations Newsletter 12 (1) (2010) 40–48. doi:10.1145/1882471.1882478.

[35] J. Wang, P. Neskovic, L. N. Cooper, Training data selection for support vector machines, in: International Conference on Natural Computation, Springer, 2005, pp. 554–564.

[36] M. Antunes, V. Andreozzi, M. Amaral Turkman, A note on the use of bayesian hierarchical models for supervised classification, Tech. rep., CEAUL Research Report 13/2006 (2006).

[37] J. Osborne, J. W. Osborne, A Brief Introduction to Hierarchical Linear Modeling, Best Practices in Quantitative Methods 8 (1) (2011) 444–450. doi:10.4135/9781412995627.d35.

[38] G. Y. Wong, W. M. Mason, The hierarchical logistic regression model for multilevel analysis, Journal of the American Statistical Association 80 (391) (1985) 513–524. doi:10.1080/01621459.1985.10478148.

[39] X. Chen, Y. Wang, J. He, S. Pan, Y. Li, P. Zhang, Cap: Context-aware app usage prediction with heterogeneous graph embedding, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3 (1) (2019) 1–25.

[40] T. M. T. Do, D. Gatica-Perez, Where and what: Using smartphones to predict next locations and applications in daily life, Pervasive and Mobile Computing 12 (2014) 79–91.

[41] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[42] A. Parate, M. Böhmer, D. Chu, D. Ganesan, B. M. Marlin, 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, 2013, pp. 275–284.