Anonymous Yet Alike: A Privacy-Preserving DeepProfile Clustering for Mobile Usage Patterns

Cheuk Yee Cheryl Leung[1], Basem Suleiman[1,2](✉), Muhammad Johan Alibasa[3], and Ghazi Al-Naymat[4]

[1] School of Computer Science, University of Sydney, Sydney, Australia
`{cheuk.y.leung,basem.suleiman}@sydney.edu.au`
[2] School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
[3] School of Computing, Telkom University, Bandung, Indonesia
`alibasa@telkomuniversity.ac.id`
[4] Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman, UAE
`g.alnaymat@ajman.ac.ae`

Abstract. The ubiquity of mobile devices and unprecedented use of mobile apps have catalyzed the need for an intelligent understanding of user's digital and physical footprints. The complexity of their interconnected relationship has contributed to a sparsity of works on multi-contextual clustering of mobile users based on their digital and physical patterns. Moreover, with personalization the norm in users' lives and corporations collecting a multitude of sensitive data, it is increasingly important to profile users effectively while preserving their privacy. In this paper, we propose DeepProfile: a Multi-context Mobile Usage Patterns Framework for predicting contextually-aware clusters of mobile users and transition of clusters throughout time, based on their behaviors in three contexts - app usage, temporal and geo-spatial. Our DeepProfile framework preserves users' privacy as it intelligently clusters their mobile usage patterns and their transition behaviors while maintaining users' anonymity (i.e., without their gender, GPS location and high-level granularity application usage data). Our experimental results on a mobile app usage dataset show that the predicted user clusters have distinct characteristics in app usage, visited locations and behavioral characteristics over time. We found that on average, 18.6% to 23.6% of a cluster moves together to the next time segment, and other interesting insights such as over 90% of cluster transitions where users moved together, moved from a period of activity to inactivity at the same time.

Keywords: Deep learning · Clustering · Mobile usage · Behavioral patterns · Privacy

S. Longfei and P. Bodhi (Eds.): MobiQuitous 2022, LNICST 492, pp. 81–100, 2023.
https://doi.org/10.1007/978-3-031-34776-4_5

1 Introduction

Mobile phones are now ubiquitously embedded into people's daily lifestyles. As such, to better profile what lifestyle an individual leads and their behavioral patterns, it has become imperative to look at their actions in both the physical and digital sense. Despite this, there has been little academic research examining the intersection of the contexts of mobile app usage, mobility and behavioral patterns over time.

This level of user understanding is critical for personalization in the services used everyday - for instance, personalized content on Social Networking [16], targeted advertising [27] and many other applications. However, due to its nature, many corporations collect an abundance of sensitive data about their users. Consequently, privacy concerns are on the rise, posing a critical challenge in the ability to provide personalization without undermining user privacy [18]. This issue will only become more profound as the population of technologically adept and thus data-aware individuals grow. This is furthered by the difficulty of effectively mining patterns in users mobility, app usage patterns and their changing behavior without impeding on privacy, as location information is most easily detailed as a specific GPS location or longitude/latitude.

This complexity also exists in applications such as personalized content on social networking. For instance, Facebook gathers information such as your GPS location, what purchases you make, SMS log history and much more [5], in order to provide personalization in their News Feed, event suggestions and other features. Such data collection can have further impacts through other means, as seen in the Facebook data privacy scandal where the data analytics firm, Cambridge Analytica, harvested millions of Facebook users' personally identifiable profile information to influence election results [3].

In this study, we investigate the research question: how can we predict meaningful clusters of mobile users using multi-contextual information over time, in a way that maintains user privacy and allows for further analysis of behavioral transitions? To answer this question, we propose a novel DeepProfile framework for multi-context mobile usage clustering and transition patterns that enables the personalization of services while addressing the aforementioned challenges. Our DeepProfile framework enables effective profiling of users using only high-level data which is not composed of any granular details such as the user's gender or GPS location, with the assistance of unsupervised deep feature extraction to learn behavioral patterns. It can achieve personalization through user profiling in a way that provides a richer multi-contextual meaning to the resultant groups without using sensitive data.

The contributions of this paper are twofold: (1) a novel DeepProfile framework for multi-context mobile usage clustering and transition patterns, and (2) an extensive evaluation of our framework through a real-world dataset of 4.2 million app usage records from 871 unique users. To the best of our knowledge, we are the first to propose a framework that produces multi-contextual clusters of mobile users based on three contexts jointly: user mobile app usage, mobility and temporal behavior. Further, we believe that our proposed framework is the first designed to support effective analysis into mobile user cluster transitions throughout time. The framework includes a dynamic approach of time series segmentation that produces unique time segments based on the overall behavior of a dataset's population. Within the framework, we developed a deep feature extraction model that automatically extracts meaningful features over time. Finally, we extensively evaluated the framework by analyzing the clusters it produced and validating our framework's support for further examination into how users transition from cluster to cluster over time.

The rest of this paper is structured as follows. Section 2 discusses various studies related to our work. The details of our proposed approach, including our methods and algorithms, are introduced in Sect. 3. The experimental evaluation and result analysis are discussed in Sect. 4. A discussion of the results in the context of our research contributions is presented in Sect. 5. Finally, key conclusions and future work are described in Sect. 6.

2 Related Work

Many related works in the past decade used data mining on mobile app usage pattern to understand mobile user behaviors [4,14,15,20]. Common across these works is the lack of consideration for the inter-connected multi-contextual relationship between mobile apps usage, physical mobility and time. This observation was similarly made in works mining mobility patterns [1,8,17,23].

It is only in applications of predictive models where the multi-contextual aspects that influence user behavior are considered. For instance, to predict next app usage, Wang et al. [22] used a Bayesian mixture model to predict future app and location using information such as the specific visited places, their Point of Interest (PoI) distribution, and the categories of apps used. Xu et al. [24] took this further by also considering data from mobile sensors such as the accelerometer. Alternatively, Yu et al. [25] approached the app prediction task with limited information only, by just using PoI information of visited locations and app usage. Of interest is the work by Feng et al. [7] who used three distinct types of mobility data - call records, app data and social media data - and successfully showed a deep learning approach can effectively capture meaningful mobility patterns.

In the area of clustering and building user profiles of mobile users, past studies typically lacked multi-contextual consideration. For instance, Zhao et al. [28] clustered mobile users based on a vector of their overall app usage over four time periods on weekdays and weekends. Despite finding some interesting clusters, their results were limited to only a broad understanding of the different types of users. This is due to their aggregation of usage over large static time periods, and as such is unable to capture behavioral patterns in usage over time.

Conversely, Jones et al. [11] focused on a single behavioral characteristic only, by clustering users on their app re-visitation patterns. Their positive results amplify the notion that behavioral characteristics in terms of how users actually use their apps is critical in forming a better grasp of the individual. As such, if combined with other contextual information such as mobility patterns, could provide a much richer understanding of the different types of mobile users.

As such, we observe that most related studies tend to focus on single contexts - such as app usage alone, or mobility patterns alone. Moreover, these works typically do not consider user privacy in what kind of data is collected and used for these tasks.

3 Method: DeepProfile Framework

Our DeepProfile framework aims to produce multi-contextual deep clustering of mobile users. Particularly, the framework is designed to produce clusters of mobile users based on their app usage, mobility patterns, and behavioral patterns over a temporal context. Thus, we first present the modules that make up our novel DeepProfile framework and discuss their rationale, purpose and constitution. Afterwards, we detail the explorative analysis undertaken before reaching the final framework to help substantiate the proposed components.

3.1 DeepProfile Framework

Data gathered on mobile user's app usage and locations in past studies are typically highly granular, for instance, the users' longitude and latitude, the particular words typed by the users, or the users' demographic attributes. Such data collection led data privacy concerns, as such, our proposed DeepProfile framework tackles this privacy issue by centralizing its design to only use highly abstracted information. For instance, we use a distribution of the types of PoIs visited (e.g. Scenic Spot) instead of GPS location, and app category instead of specific app name. Our framework aims to produce rich and meaningful clusters of users with similar behavioral patterns in terms of their app usage, mobility, and temporal behavior. The proposed DeepProfile framework is shown in Fig. 1. In summary, the architecture of our framework is composed of five modules:

1. **Adaptive Time Segmentation Method** based on the overall activity behavior of users in the dataset to split the time series into meaningful segments.
2. **User Activity Partitioning** to transform data into usage dataframes for all active users in each time segment that will be used in the Deep Feature Extraction Model.
3. **Deep Feature Extraction Model** that is trained and validated to produce the Deep Feature Encoder.
4. **Deep Feature Encoder** which produces feature vectors that describe each active user's behavior in each time segment.
5. **Spatio-Temporal-App Usage Clustering** for each time segment.

3.2 Adaptive Time Segmentation Method

Mobile usage data is commonly recorded as a time-series thus time segmentation is critical to analyze it effectively. Each individual's behavior in mobile usage

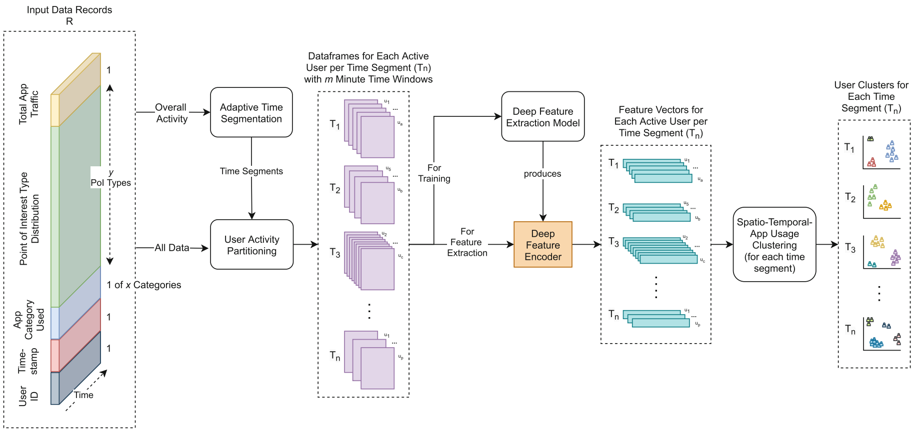

Fig. 1. DeepProfile: Proposed Framework Architecture

varies over time, and devices record the exact timestamp at which events or actions occur. Similar routine behaviors may occur at similar but still differing times everyday, particularly when considering multiple users. Whilst many past studies have utilized large static equal time segments [4,10,15,19], and others static unequal time segments [6,24,26,29], these methods fail to consider the broader cultural, geographic and socio-economic nuances that can affect the appropriateness of their static time segments. For instance, some countries may have a longer work week due to their corporate work culture.

In our DeepProfile framework, we use an adaptive change point algorithm which can detect change points in the behavior of a time series, which indicates the start of a new time segment. More specifically, we utilize the Pruned Exact Linear Time (PELT) algorithm [12]. PELT is an efficient change point detection algorithm that can achieve higher accuracy than other well-known methods such as Binary Segmentation [12]. Running the PELT algorithm on different datasets can produce a different set of change points which is unique based on the behavior of the users in the data provided. This is critical, as factors such as the cultural, geographical or socio-economic context of a population can have significant impacts on their general behavior.

The PELT algorithm is based on an Optimal Partitioning approach using Dynamic Programming by of Jackson et al. [9], but involves an extra pruning step to reduce computational cost within the dynamic program whilst unaffecting exactness [12].

In order to use this method, an appropriate penalty value must be determined. This is done by graphing an elbow plot which plots the number of change points identified against the penalty value used. The optimal value would be where the 'elbow' of the plot is, where change points induced by noise are no longer detected. Using the optimal penalty value, the PELT algorithm can be executed on the time series data. This will identify the optimal time segments based on the behavior of the input.

3.3 User Activity Partitioning

This module in the DeepProfile framework performs pre-processing on the input data for the Deep Feature Extraction Model. The data transformation here focuses on partitioning the data to be suitable for deep learning, thus no particular feature engineering is conducted. We partition the input data records R into each time segment previously identified. As such, this module will transform the data so that every time segment will have a series of dataframes, each consisting of the usage and activity of only the active users in that time segment. In other terms, for every Time Segment T_i, there will be a dataframe for each user u satisfying the condition:

$$\mid \{r \in R \mid r_{\text{userId}} = u, T_i^{start} <= r_{\text{time stamp}} < T_i^{end}\} \mid > 0 \tag{1}$$

where R is the entire input data of user activity records and r is a specific record in the input data.

As mobile usage data is typically recorded at the exact timestamp whenever events or actions occur, the time series results is irregularly spaced (or arbitrarily sampled). Therefore, this module transforms the data such that it is evenly sampled to be appropriately input into our Deep Feature Extraction Model. Each dataframe will be evenly sampled according to the time window m selected (e.g. if $m = 1$, each row of the dataframe would correspond to 1 min). The number of usage records per app category will be counted for each time window. Then, total app traffic is also summed up for each time window. For mobility information, we take the most occurring point of interest (PoI) distribution in each time window, to best reflect the key locations visited by the user.

Given the process of even sampling to a specified time window, shorter time segments will have less number of time windows. Therefore, all time segments will be padded with rows of zeroes to W, the number of time windows in the longest time segment. Therefore, each dataframe can be represented as

$$d_{\text{user, time segment}} \in \mathbb{R}^{W \times (x+y+1)} \tag{2}$$

where W represents the number of time windows in the longest time segment, x represents the number of app categories, y represents the number of Point of Interest types and the final 1 is for the total app traffic for each time window. Thus $(x + y + 1)$ represents the total number of variables per time window.

3.4 Deep Feature Extraction Model

This module is a convolutional autoencoder consisting of two main components: the encoder and decoder. The autoencoder can effectively learn features as the encoder component comprises a bottleneck which creates a latent representation of the input in a smaller dimension. The decoder then attempts to reconstruct the original input from this smaller representation, thus requiring the encoder to capture and retain only the most meaningful features.

Each dataframe from the user activity partitioning is used as input to the convolutional autoencoder, thus the autoencoder handles inputs ($d_{\text{user, time segment}}$) of dimensions $W \times (x + y + 1)$. The first layer receives the input dataframe $d_{\text{user, time segment}}$. It is then passed into a series of 1D convolution, max pooling, and dropout layers.

3.5 Deep Feature Encoder

After the convolutional autoencoder is trained, the encoder component can be extracted out for use. The encoder is used to 'predict' feature vectors for all users in each time segment for clustering in the next module. The encoder can be represented as:

$$e : d_{\text{user, time segment}} \mapsto v \in \mathbb{R}^b \tag{3}$$

where b is the vector size of the bottleneck in the convolutional autoencoder (the final layer of the encoder component) and v is the resultant feature vector.

3.6 Spatio-Temporal-App Usage Clustering

In the Spatio-Temporal-App Usage Clustering module, for each identified time segment, the following is executed. Firstly, for all active users in the time segment, we first input their dataframe ($d_{\text{user, time segment}}$) into the trained encoder e to extract its feature vector v. Once an encoded feature vector for the active users is produced, agglomerative hierarchical clustering is used to group similar users together. Users that did not have a single usage record in a time segment are automatically allocated to cluster '0'.

This clustering routine is conducted for each time segment. Subsequently, this means that every user is allocated into a cluster for every time segment. As such, this information can be then used for further analysis into the transition patterns of users and of clusters over time, as we demonstrate later in Sect. 4. The clustering results can also be used in algorithms for application areas such as advertising, in order to provide personalization that better maintains user privacy.

4 Experiments and Results

4.1 Dataset

Tsinghua App Usage Dataset [25] was used in our study, consisting of data collected from a mobile cellular network in Shanghai, China. The data spans a period of 7 days from 20 April 2016 to 26 April 2016, containing 4,171,950 records from 871 users and 9,851 base stations. Each record represents a data request on the mobile network by a user, and contains an anonymized user ID, timestamp, base station ID, used app ID and traffic. Data detailing the app category of each app, which were determined by Android Market and Google Play, and the number of Point of Interests (PoIs) in each PoI category for each base station is also provided.

4.2 Adaptive Time Segmentation

Our experiment splits the time according to the overall activity of the population in the dataset, since this will provide the clearest indication as to the general 'sections' of a day for their particular cultural, geographical, and social context. Since the Tsinghua App Usage Dataset consists of 1 week's data only, there were a few approaches we could take. Firstly, we could produce time segments for each day of the week individually. We could also average out weekday and weekend activity, and thus produce two sets of time segments. Finally, we could simply segment the entire week as one time series.

By examining the overall activity for the dataset per day, our first observation is that not all weekdays had similar activity patterns. Similarly, weekends were not seen to be much different from weekdays. As such, this means that we cannot simply average out activity for weekdays and then weekends, since there does not appear to be such a pattern in this dataset. In order to determine whether we should produce time segments for each day individually or for the entire week as one time series, we conducted experiments on both these approaches using the PELT algorithm. We use the Ruptures[1] [21] implementation.

Segmenting per Day. In order to determine the time segments based on the overall activity level for each day individually, we utilise a transformed version of the dataset that holds the number of app usage records per minute. We use the PELT algorithm to determine all change points in behavior, which act as the start/end of each time segment, to find all-time segments for each individual day. From producing the elbow plots for each day as described in Subsect. 3.2, the ideal Penalty Values were identified for Monday to Sunday as 3, 6, 3, 3, 4, 3, and 3.

Segmenting Entire Week. In this method, we segment the entire dataset time length as one time series, and also specify for the PELT algorithm to ensure that time segments are at least 60 min long. The identified Penalty Value for the PELT algorithm was 10. As shown in Fig. 2, segmenting our 1-week long dataset results in 44 time segments. In the figure, there is a vertical dotted gray line at the start of each day.

Conclusions. When examining the time segments from our second experiment using the entire week as one time series, we found that a new time segment often begin around 12:00AM but never exactly. Since these segments were identified

---

[1] https://centre-borelli.github.io/ruptures-docs/user-guide/detection/pelt/.

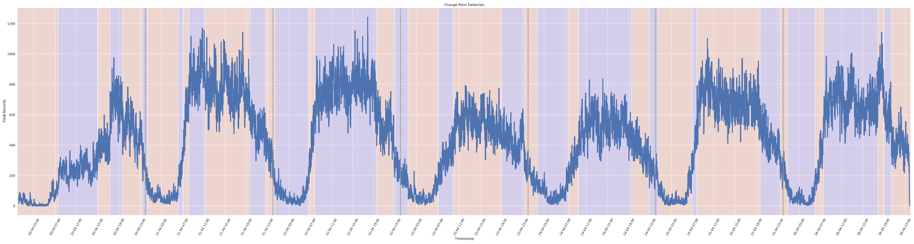

Fig. 2. Result Time Segments from our Adaptive Time Segmentation Method

from using the entire week as a single time series, these time segments reflect the natural change points of the overall activity behavior of the users. This is in contrast to the results from segmenting each day individually where a new time segment always starts at 12:00AM, as individuals are unlikely to always begin a new behavioral routine at exactly 12:00AM. Thus, we conclude that performing PELT change point detection to determine time segments on the entire week as one time series.

4.3 Deep Feature Extraction Model and Clustering

After performing adaptive time segmentation, we perform experiments on developing the Deep Feature Extraction component of our framework. We used Keras[2] to build the different Convolutional Autoencoders to evaluate for our Deep Feature Extraction Model. The models were trained on the training subset of data. The training to validation split was 80% : 20%. For all evaluated models, the number of epochs used to train is 25, with learning rate 0.001 using Adam optimization. These values were selected after empirical evaluation from experiments in training our model, by observing the validation and training losses to ensure a well-trained model without overfitting. In general, each model takes approximately 15 min to finish training on CPU. For our hierarchical clustering, we use the SciPy[3] implementation and for internal cluster validation metrics we use scikit-learn[4].

Evaluation Method. We conducted hierarchical clustering using the resultant feature vectors for each user in each time segment, which are the output from our Deep Feature Extraction Model and Encoder. To perform evaluation, we firstly use internal validation metrics of Calinski and Harabasz Score and Davies-Bouldin Score to observe the cluster quality of the resultant clusters produced using the different Deep Feature Extraction Models.

The Calinski and Harabasz Score is the ratio between the within-cluster dispersion and the between-cluster dispersion, thus a higher value indicates denser

---

[2] https://keras.io/.

[3] https://scipy.org/.

[4] https://scikit-learn.org/stable/index.html.

and better defined clusters. The Davies-Bouldin Score compares the average similarity of each cluster with another cluster most similar to it. Therefore, a lower Davies-Bouldin Score indicates clusters are more ‘different’ from each other and better separated.

We also examine the proportion of clusters in each result that consists of more than 1 individual, as we would like to ensure our model effectively discovers groups of similar people. In determining the optimal model however, we opt for qualitative assessment. We manually observe the produced dendrograms and resultant clusters to determine whether the groups formed have clearly identifiable behavioral characteristics and patterns. It is important that we use this qualitative assessment, as internal validation metrics cannot adequately measure the quality of the following critical features of our clustering. Users should show multi-contextual similarity, including what app categories they used, the behavioral characteristics of usage exhibited, and their mobility in terms of the types of locations visited, all considered over a temporal context. There are two components considered to qualitatively assess the cluster results.

1. **The ease of cluster description**: the clusters formed should be easily interpreted and described in terms of their content, showing aspects that clearly distinguish themselves from others. Clusters should be different from each other by at least 1 contextual aspect.
2. **How balanced the clusters are**: clusters should not be extremely unbalanced. No more than 25% of the resultant clusters can contain only 1 individual. Further, clusters should not necessarily be equal in size or be extremely large. A careful balance must be struck to ensure distinguishable and meaningful groups of users.

Given the expensive nature of the qualitative assessment, we conducted our evaluations on two time segments - one long time segment of Wednesday 7:35AM - 3:00PM (Time Segment 1), and one short time segment of Wednesday 3:00PM - 5:20PM (Time Segment 2). We performed ablation studies examining two main aspects: the impact of different time windows used for User Activity Partitioning and also different layer architecture in the convolutional autoencoder of our Deep Feature Extraction Model.

4.4 Evaluation Results

We first investigated the impact of different time windows for the User Activity Partitioning module (Subsect. 3.3) on the performance of the autoencoder-extracted features when clustering. The time window is used when evenly sampling our input data to form the rows of a user’s dataframe for each time segment. The Base Convolutional Autoencoder used here is composed of an encoder with 4 sets of 1D Convolution, Max Pooling, and Dropout layers. The parameters of the 1D Convolution Layers are all set to filters = 16 and kernel size = 3. Following these sets, there is a flatten and dense layer to produce the feature vector. The decoder component is designed symmetrically to the encoder. We evaluate the impact of using 1 min, 5 min, 10 min and 15 min time windows.

Table 1. Ablation Study - Internal Metrics for Different Time Windows

Time Window	Time Segment	Calinski and Harabasz Score	Davies-Bouldin Score
1 min	Wed 7:35AM - 3:00PM	59.84838	0.38686
	Wed 3:00PM - 5:20PM	181.10552	0.40569
5 min	Wed 7:35AM - 3:00PM	97.31339	0.62215
	Wed 3:00PM - 5:20PM	653.044089	0.70322
10 min	Wed 7:35AM - 3:00PM	178.07984	0.87969
	Wed 3:00PM - 5:20PM	842.32609	0.65256
15 min	Wed 7:35AM - 3:00PM	483.40322	0.812233
	Wed 3:00PM - 5:20PM	2540.53143	0.61834

The results of our internal quality metrics evaluation on the produced clusters can be seen in Table 1. We saw that as the time windows increase, so does the Calinski and Harabasz Score, suggesting higher intra-cluster similarity. While we observe its highest value for the 15 min time window, we noted that this score was a significant spike. From a manual observation of the resultant clusters, the consequent aggregation from such a large time window reduced the model's ability to identify more detailed behavioral patterns, such as their frequency of app usage and other behavioral habits.

Moreover, we observed that moving from a time window of 10 min to 15 min did not provide any improvement on more balanced clustering - the proportion of clusters which had more than 1 user did not really increase (84.745% → 87.692% for Wed 7:35AM - 3:00PM, 75% → 78.571% for Wed 3:00PM - 5:20PM). This is likely due to the low granularity of 15 min time windows. Given majority of our time segments are between 60–120 minutes, 15 min time windows means that each time segment would only have 4–8 timesteps. Since the convolution layers in our autoencoder itself slides across the timesteps to identify patterns over time, overly summarised data will undermine the benefits of our framework. As such, a 15 min time window is not appropriate.

Finally, when observing the resultant clustering using a 10 min window, we observed a good balance between focusing on the temporal aspect of users' app usage behavior and other factors such as the types of locations visited and app categories used. In contrast, 5 min time windows caused the model to focus too strongly on the temporal aspect of the users' app usage, causing the resultant clusters to only bear strong similarity in what time they used apps. In the 10-minute window, we observed a good compromise between the results from the 5 min time window and 15 min time window. Thus, we select the 10 min time window for our final best implemented model.

Similar to the previous ablation study, we also conducted experiments for various layer architecture in the convolutional autoencoder for our Deep Feature Extraction Model. The parameters of the models used, determined through our empirical evaluation, for this ablation study are shown in Table 2. After assessing the internal metrics and qualitative examinations of resultant clusters, the outcome of these additional experiments found the architecture of Model B to generate the best quality clusters.

Table 2. Definition of Models for Evaluation

Model	Layers	Convolution Layer Parameters
A	4 sets of: 1D Convolution, Max Pooling and Dropout Layers	All: filters = 16, kernel size = 3
B	4 sets of: 1D Convolution, Max Pooling and Dropout Layers	All: kernel size = 3 First 2 sets: filters = 32 Last 2 sets: filters = 64
C	3 sets of: 1D Convolution, 1D Convolution, Max Pooling, Dropout	All: kernel size = 3 First 2 sets: filters = 32 Last set: filters = 128

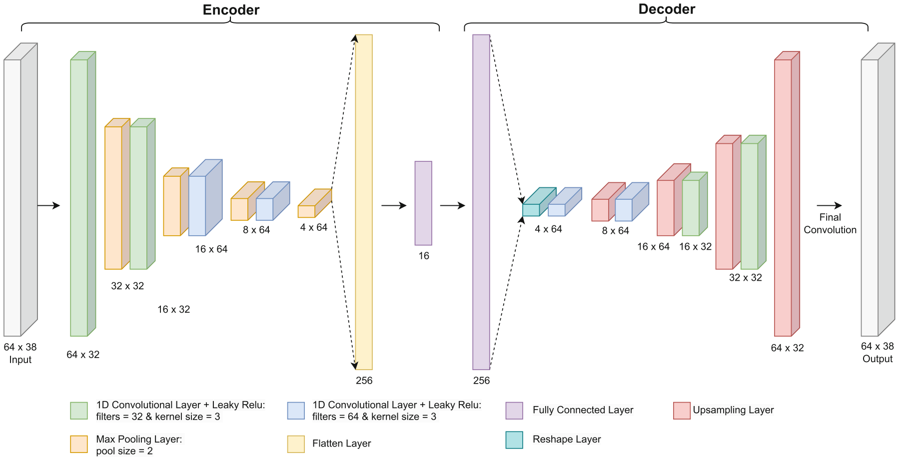

Fig. 3. Best Implemented Convolutional Autoencoder Model for Deep Feature Extraction

Best Model. As a result from our ablation studies, our best implemented model is using 10 min time windows using the layer architecture of Model B. This structure of the Deep Feature Extraction Model's Convolutional Autoencoder can be seen in Fig. 3.

4.5 Clustering Results from the Best Model

In this subsection, we provide examples for the cluster results produced by our implementation of the proposed framework and examine them to identify the nature of those users. The results provide evidence in our achievement of rich and meaningful clusters based on users' app usage, mobility, and temporal patterns. Our implementation of the proposed framework produced different clusters of users for each of the 44 time segments of the week's data. The two clusters we discuss in this paper are from Time Segment 1, Wednesday 7:35AM - 3:00 PM. Our framework produced a total of 64 clusters for this time segment.

Cluster 3. As seen in Fig. 4, Users in this cluster showed consistent usage of Utilities apps, recording around 100–150 instances of such usage in this time segment. Their behavioral characteristics are composed of consistent usage throughout these several hours, generally with frequent 'blocks' of usage . The locations these users visited tended to have several Business Affairs Point of Interests in the vicinity, with occurrences of Life Services, Residential, Government, Shopping and Restaurant PoIs (see Fig. 5) We can infer the locations these users visited reflect mixed-use development that blends residential, commercial, cultural, and institutional establishments into one space.

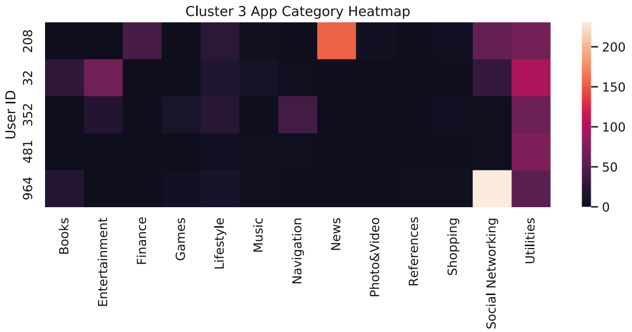

Fig. 4. Cluster 3 App Category Usage Heatmap, Wednesday 7:35AM - 3:00PM

Cluster 15. Users in this cluster were extremely heavy users of Social Networking apps - each user recorded over 1000 usage records in this category, with little of anything else (see Fig. 6).

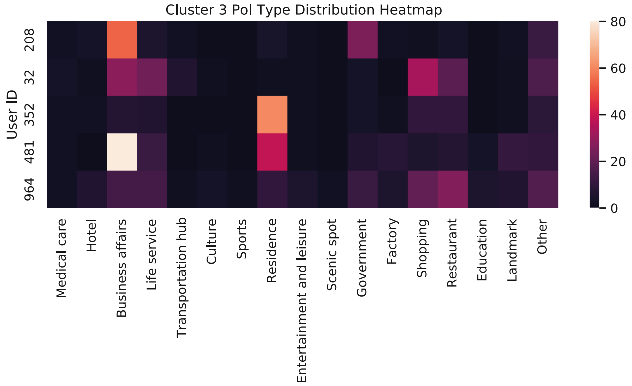

Fig. 5. Cluster 3 PoI Type Heatmap, Wednesday 7:35AM - 3:00PM

This is a significant contrast to the previous cluster that had approximately 200 usage records at most in any app category. The PoI distribution of Cluster 15's most visited locations included residential PoIs with some shopping and restaurants, as seen in Fig. 7. Their usage was also extremely consistent throughout the time period, with very heavy blocks of Social Networking use.

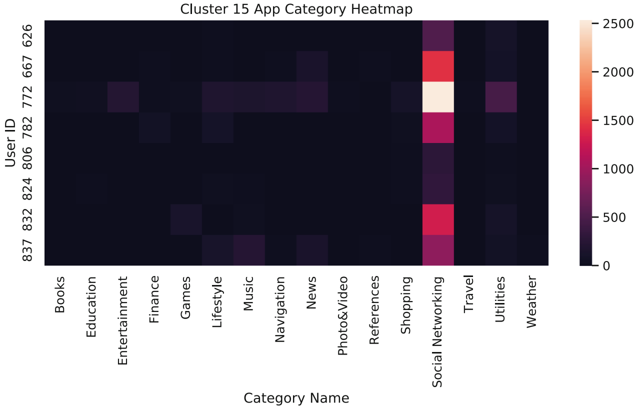

Fig. 6. Cluster 15 App Category Usage Heatmap, Wednesday 7:35AM - 3:00PM

The differences in the various contextual aspects of each cluster can significantly impact the optimal way to personalize for the users in each group. For instance, from our examination of each cluster's behavioral patterns in their app category usage through categorical Scatterplots over time, Cluster 15 demonstrated extremely heavy usage that was characterized by a pattern of high-frequency blocks of use throughout the entire time segment. If a firm wishes to provide mobile advertising to the users from this cluster, ads which exploit

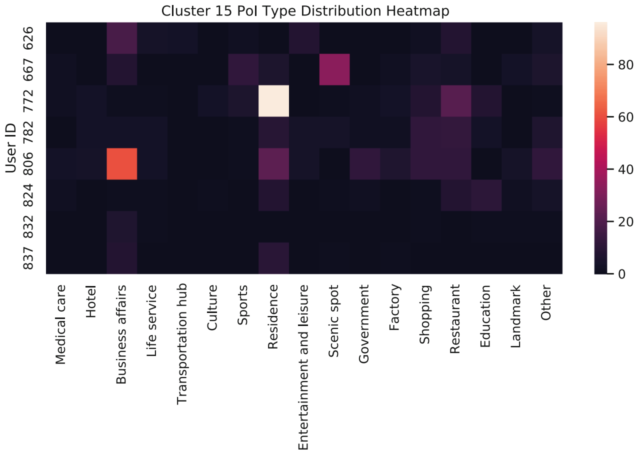

Fig. 7. Cluster 15 PoI Type Heatmap, Wednesday 7:35AM - 3:00PM

the technique of repetition can be optimal. This is contrast to the both sparse and significantly lighter usage of Cluster 3, which would be more motivated by ads that have a strong call to action. As such, the discovered clusters demonstrate the capturing of similar behavior across users' mobile app usage, visited locations, and behavioral patterns without utilizing more sensitive data.

4.6 Cluster Transitions Analysis

Using our best implemented deep feature extraction model, we clustered the users in all 44 time segments, as according to the method described in Subsect. 3.6. Any inactive users in each time segment are automatically assigned to cluster '0'. The number of clusters in each time segment is variable, since users were clustered for each time segment individually. This subsection details our analysis into how clusters transitioned from one time segment to the next . We analyze whether there are any transition patterns that certain clusters or individuals take part in. For instance, we check if there are specific clusters that all move together to the same cluster in the next time segment, or, are there individuals who perform the same cluster transitions for several time segments. By conducting this analysis and the previous examination of resultant clusters, we showed that our proposed framework can enable rich analysis into the behavioral characteristics and transitions of the produced clusters through time.

Transitions Between 2 Consecutive Time Segments. In order to examine the transitions of clusters between 2 consecutive time segments, we create a transition probability matrix for each transition. These matrices hold the likelihoods of someone in a specific cluster in one time segment, moving to another cluster in the next time segment. We found the median value to be 0.1052, indicating that in more than half of our transitions between 2 consecutive time segments, at least 10.52% of a cluster moves together to the next time segment. The mean value is 0.1987, indicating that on average, 19.87% of a cluster moves together to the next time segment, however further statistical testing is required to see if this is consistent across transitions between different pairs of time segments.

We then analyze the mean likelihoods of each transition. When computing the mean, we ignore transitions where the cluster is only comprised of 1 person, as 1 person moving to the next time segment would undoubtedly give a likelihood of 1.0. When observing the histogram of these mean likelihoods and plotting the data against a theoretical normal distribution using a Probability Plot, we found the data is normally distributed. Since the mean likelihoods is normally distributed, we can compute the confidence interval. The 95% confidence interval is (0.18627, 0.23584). Hence, we can say that we are 95% confident that on average, between 18.627% and 23.584% of a cluster transitions together to the next time segment.

We also observed specific cluster transitions where more than 40% of the cluster moved together to the next time segment and more than 2 people transitioned together. There were a total of 380 of such transitions, where the majority (344/380) are moving from an active cluster to cluster ‘0’ (inactive), 18 transitions are moving from cluster ‘0’ to cluster ‘0’, and 18 transitions are moving from an active cluster to another active cluster. This suggests a strong tendency for clusters to move towards inactivity at the same time.

As an example, there is a cluster transition between Time Segment 23 (Saturday 10:05-17:30) and Time Segment 24 (Saturday 17:30-23:20), where 3 of 6 (50%) of users in cluster 87 moved together to cluster 9. The three users who transitioned together exhibited behavior of consistent app usage throughout Time Segment 23, with clear similarities in their Utilities and Social Networking usage. They then transitioned together to Cluster 9, where they all had app usage at only one point in time during this time segment. In terms of the PoI types at their visited locations, in Time Segment 23 all 3 users were mainly in a residential area, whilst in the next time segment, they moved to a more business-oriented area, with some shopping and restaurants.

5 Discussion

Our evaluation using real-world app usage dataset found the predicted clusters of individuals had similarities in several multi-contextual aspects - the app categories used, the behavior with which they used apps, the locations they visited in terms of what types of PoIs were in the vicinity, and the temporal characteristics of their mobility. The input data into our proposed framework was extremely high level. It did not consist of any granular information such as a user’s ethnicity, age, gender, or GPS location. Despite this, the clusters predicted were still meaningful and distinguished from each other, with users in a cluster bearing multi-contextual similarities. Each cluster had distinct characteristics in their mobile app usage, mobility, and behavioral patterns over time.

This is in contrast to previous works, which tended to cluster users based on individual aspects only. For instance, some works [28] clustered users based only on their aggregated app usage over static time segments with aggregated data based on weekdays and weekends. As such, any rich insights into the behavioral characteristics they exhibited when using the apps or the temporal patterns undertaken could not be reflected. Some [11] took a granular approach, clustering users based only on their app re-visitation patterns. Again, this was based on aggregated data for each user and thus cannot provide insights into their temporal behavior. Other studies [2,13] clustered users based on their mobility patterns only, using specific GPS locations and undermining user privacy.

Finally, we provided an extensive analysis into the patterns of cluster transitions. We discovered several insights, such as how likely a cluster would transition together into the next time segment, along with the proportion of users who shared the same cluster transitions with another individual over several time segments. We found that on average, between 18.627% and 23.584% of a cluster typically transitions together to the next time segment. Moreover, over 90% of such cluster transitions where users moved together, moved from a period of activity to inactivity at the same time.

6 Conclusion and Future Work

In this work, we have presented DeepProfile: a multi-context mobile usage patterns framework that clusters users across time using multi-contextual information without undermining user privacy. Our framework enables further analysis on the transition of users' behavior throughout time to discover insights into the behavioral patterns of different groups. Through our extensive evaluation of the framework, we found that the framework was able to produce rich and meaningful clusters that captured the several aspects of app usage, mobility and behavioral characteristics over time.

In the future, it could be interesting to explore the potential of a hybrid model that uses more than one Deep Feature Extraction Model for each type of contextual pattern. This could allow each feature extraction model to excel at learning patterns for the particular context it has been trained for. It could be also interesting to investigate modifications to the clustering method undertaken in this work to enable real-time clustering in an unsupervised manner, even while mobile data is being gathered.

References

1. Becker, R., et al.: Human mobility characterization from cellular network data. Commun. ACM. **56**(1), 74–82 (2013). https://doi.org/10.1145/2398356.2398375
2. Ben-Gal, I., Weinstock, S., Singer, G., Bambos, N.: Clustering users by their mobility behavioral patterns. ACM Trans. Knowl. Discov. Data **13**(4) (2019). https://doi.org/10.1145/3322126, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/3322126
3. Cadwalladr, C., Graham-Harrison, E.: Revealed: 50 million Facebook profiles harvested for Cambridge analytica in major data breach. The Guardian (2018). https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election
4. Do, T.M.T., Gatica-Perez, D.: By their apps you shall understand them: mining large-scale patterns of mobile phone usage. In: Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia. MUM 2010, ACM, New York, NY, USA (2010). https://doi.org/10.1145/1899475.1899502, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/1899475.1899502
5. Facebook: Data policy (2021). https://www.facebook.com/privacy/explanation
6. Farrahi, K., Gatica-Perez, D.: Probabilistic mining of socio-geographic routines from mobile phone data. IEEE J. Sel. Top. Signal Process. **4**(4), 746–755 (2010)
7. Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., Jin, D.: Deepmove: Predicting human mobility with attentional recurrent networks. In: Proceedings of the 2018 World Wide Web Conference. p. 1459–1468. WWW 2018, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). https://doi.org/10.1145/3178876.3186058
8. Herder, E., Siehndel, P.: Daily and weekly patterns in human mobility. In: UMAP Workshops, pp. 338–340. Citeseer (2012)
9. Jackson, B., et al.: An algorithm for optimal partitioning of data on an interval. IEEE Signal Process. Lett. **12**(2), 105–108 (2005). https://doi.org/10.1109/LSP.2001.838216
10. Jayarajah, K., Kauffman, R., Misra, A.: Exploring variety seeking behavior in mobile users. In: Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, pp. 385–390 (2014)
11. Jones, S.L., Ferreira, D., Hosio, S., Goncalves, J., Kostakos, V.: Revisitation analysis of smartphone app use. In: Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 1197–1208. UbiComp 2015, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2750858.2807542, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/2750858.2807542
12. Killick, R., Fearnhead, P., Eckley, I.A.: Optimal detection of changepoints with a linear computational cost. J. Am. Stat. Assoc. **107**(500), 1590–1598 (2012)
13. Lu, E.H.C., Tseng, V.S., Yu, P.S.: Mining cluster-based temporal mobile sequential patterns in location-based service environments. IEEE Trans. Knowl. Data Eng. **23**(6), 914–927 (2011). https://doi.org/10.1109/TKDE.2010.155
14. Ma, H., Cao, H., Yang, Q., Chen, E., Tian, J.: A habit mining approach for discovering similar mobile users. In: Proceedings of the 21st International Conference on World Wide Web, pp. 231–240. WWW 2012, ACM, New York, NY, USA (2012). https://doi.org/10.1145/2187836.2187868, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/2187836.2187868

15. Mukherji, A., Srinivasan, V., Welbourne, E.: Adding intelligence to your mobile device via on-device sequential pattern mining. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, pp. 1005–1014. UbiComp 2014 Adjunct, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2638728.2641285, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/2638728.2641285
16. Park, J.H.: The effects of personalization on user continuance in social networking sites. Inf. Process. Manage. **50**(3), 462–475 (2014). https://doi.org/10.1016/j.ipm.2014.02.002, https://www.sciencedirect.com/science/article/pii/S0306457314000120
17. Pavan, M., Mizzaro, S., Scagnetto, I.: Mining movement data to extract personal points of interest: a feature based approach. In: Lai, C., Giuliani, A., Semeraro, G. (eds.) Information Filtering and Retrieval. SCI, vol. 668, pp. 35–61. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-46135-9_3
18. Rafieian, O., Yoganarasimhan, H.: Targeting and privacy in mobile advertising. Mark. Sci. **40**(2), 193–218 (2021)
19. Rawassizadeh, R., Momeni, E., Dobbins, C., Gharibshah, J., Pazzani, M.: Scalable daily human behavioral pattern mining from multivariate temporal data. IEEE Trans. Knowl. Data Eng. **28**(11), 3098–3112 (2016)
20. Srinivasan, V., Moghaddam, S., Mukherji, A., Rachuri, K.K., Xu, C., Tapia, E.M.: Mobileminer: mining your frequent patterns on your phone. In: Proceedings of ACM International Joint Conf. on Pervasive and Ubiquitous Computing, pp. 389–400. UbiComp 2014, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2632048.2632052, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/2632048.2632052
21. Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. Signal Process. **167**, 107299 (2020). https://doi.org/10.1016/j.sigpro.2019.107299, https://www.sciencedirect.com/science/article/pii/S0165168419303494
22. Wang, H., et al.: Modeling spatio-temporal app usage for a large user population. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **3**(1) (2019). https://doi.org/10.1145/3314414, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/3314414
23. Xie, R., Ji, Y., Yue, Y., Zuo, X.: Mining individual mobility patterns from mobile phone data. In: Proceedings of International Workshop on Trajectory Data Mining and Analysis, pp. 37–44. ACM (2011). https://doi.org/10.1145/2030080.2030088, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/2030080.2030088
24. Xu, Y., et al.: Preference, context and communities: a multi-faceted approach to predicting smartphone app usage patterns. In: Proceedings of the International Symposium on Wearable Computers, pp. 69–76. ISWC 2013, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2493988.2494333, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/2493988.2494333
25. Yu, D., Li, Y., Xu, F., Zhang, P., Kostakos, V.: Smartphone app usage prediction using points of interest. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **1**(4) (2018). https://doi.org/10.1145/3161413, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/3161413
26. Yu, K., Zhang, B., Zhu, H., Cao, H., Tian, J.: Towards personalized context-aware recommendation by mining context logs through topic models. In: Tan, P.N., Chawla, S., Ho, C.K., Bailey, J. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 431–443. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-30217-6_36

27. Zhang, T., Cheng, X., Yuan, M., Xu, L., Cheng, C., Chao, K.: Mining target users for mobile advertising based on telecom big data. 2016 16th International Symposium on Communications and Information Technologies (ISCIT), pp. 296–301 (2016)
28. Zhao, S., et al.: Discovering different kinds of smartphone users through their application usage behaviors. In: Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 498–509. UbiComp 2016, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2971648.2971696, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/2971648.2971696
29. Zhu, H., Chen, E., Xiong, H., Yu, K., Cao, H., Tian, J.: Mining mobile user preferences for personalized context-aware recommendation. ACM Trans. Intell. Syst. Technol. **5**(4) (2014). https://doi.org/10.1145/2532515, https://doi-org.ezproxy.library.sydney.edu.au/10.1145/2532515