Hindawi
Mobile Information Systems
Volume 2022, Article ID 6434598, 17 pages
https://doi.org/10.1155/2022/6434598

Research Article

Fine-Grained Urban Functional Region Identification via Mobile App Usage Data

Lei Deng and Hangyu Hu

School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China

Correspondence should be addressed to Lei Deng; denglei@nwpu.edu.cn

Received 2 January 2022; Accepted 17 January 2022; Published 1 March 2022

Academic Editor: Hye-jin Kim

Understanding fine-grained urban function for different regions is essential for both city managers and residents in terms of strategy design, tourism recommendation, business site selection, etc. A huge amount of data from the mobile network in the past several years provides the possibility for fine-grained urban function identification since it provides the opportunity to extract useful information about urban functions. However, challenges remain: (i) there is no prior knowledge about the existence of App usage patterns relating to urban functional regions; (ii) collected data are very noisy and data from different cellular towers have different noise levels. Therefore, it is difficult to extract unique patterns to identify urban functional regions. This article proposes a fine-grained urban functional region identification system, which utilizes mobile App usage data from cellular towers. To address challenge (i), we first extract three key variables for each cellular tower, App number, user number, and traffic. Then, we design a Davies–Bouldin index (DBI)-based filtering method to automatically select the most distinguishable features for multi-classification. To address challenge (ii), we first reduce cellular tower level noise by designing a clustering-based method to select the most representative cellular tower data. The data from these cellular towers share similar patterns for the same urban functional region and different patterns between different urban functional regions. Then, we reduce feature level noise by designing a Fourier transform-based method to reconstruct the features with several key frequency components, which preserves the most important information and removes the unnecessary noise. We evaluate our system and selected features with three representative supervised learning models, all of which achieve more than 95% classification accuracy.

1. Introduction

The last decades have witnessed the fast development of modern cities composed of different *functional regions*. These functional regions provide citizens with various urban functions for socioeconomic activities, such as living, working, and shopping [1]. Some functional regions are designed by urban planners, whereas others are naturally formulated due to citizens' actual lifestyle [1]. According to [2], both territories and functional regions can be reformed during the evolution of a city, especially in developing countries whose cities evolve fast. For example, the old wharf region in Shanghai used to be a shipping area. Due to the nice view and location, it has become a tourist area, which consists of restaurants, museums, and shopping malls [3]. Therefore, understanding fine-grained urban function regions is essential for both city managers and residents. In addition, many valuable applications can be realized, such as calibration for urban planning, business site selection, and tourism recommendations.

A huge amount of data from the cellular network in the past several years provides the possibility for fine-grained urban function detection. According to Cisco white paper, monthly global mobile data traffic increased from 7.2 exabytes (10^{18}) at the end of 2016 to 11 exabytes at the end of 2017 [4]. Mobile App usage data, which describes which Apps are used by how many users within the given time, provides a good opportunity for different kinds of human and city investigations. This is because the type of region affects users' Apps usage patterns. For example, people in the residential areas tend to use Apps more frequently during the night, whereas people in the office areas tend to use Apps

more frequently during the daytime of workdays. This might not be absolutely correct for an individual user, whereas the type of urban functional regions does affect the Apps usage pattern of overall users in the area. A common method is to collect data from mobile devices for investigating human activities [5], application usage [6], and human communication activities [7]. However, the limited number of sampled users is not able to represent the global characteristics of the whole area. Some research studies focus on investigating land usage with Call Description Records (CDRs), such as data of phone call [8] and text message [9]. Since people tend to use the application more often than phone calls, these datasets may have some bias or missing information. In addition, more and more users prefer applications such as WhatsApp, WeChat, and Line to send text messages. Therefore, these CDR data lose a lot of key information for the investigation.

The mobile APP usage data provide more information to extract unique patterns for urban functional region identification. However, it is still difficult to identify unique patterns for different urban functions due to the following three challenges: (i) There is no prior knowledge about the existence of App usage patterns relating to urban function regions. We do not know how different urban functional regions affect the overall mobile App usage within the cellular tower range. As a result, it is difficult to extract features based on experience [10]. (ii) The collected data are usually very noisy, and data from different cellular towers have different noise levels [11]. As a result, the extracted features within the same urban functional regions have high standard deviations, which leads to low classification accuracy [12].

This article proposes a fine-grained urban functional region identification system, which utilizes mobile App usage data from cellular towers. To address challenge (i), we first extract three key variables for each cellular tower, App number, user number, and traffic. Then, we design a Davies–Bouldin index (DBI)-based filtering method to automatically select the most distinguishable features for multiclassification. To address challenge (ii), we first reduce cellular tower level noise by designing a clustering-based method to select the most representative cellular tower data. The data from these cellular towers share similar patterns for the same urban functional region and different patterns between different urban functional regions. Then, we reduce feature level noise by designing a Fourier transform-based method to reconstruct the features with several key frequency components, which preserves the most important information and removes the unnecessary noise. We evaluate our system and selected features with three representative supervised learning models, all of which achieve more than 95% classification accuracy. Our core contributions are listed as follows:

(i) We design a hierarchical clustering-based method and a Fourier transform-based method to reduce cellular tower level noise and feature level noise, respectively, which improves the classification accuracy of urban functional regions.

(ii) We design a DBI-based method to automatically figure out the most distinguishable features based on *mobile App fingerprint* (App number, user number, and traffic) and correlate them to different urban functional regions.

(iii) We evaluate our system based on App usage dataset from real cellular networks in Shanghai. Three learning models supervised learning models are tested in terms of classification accuracy, type I and type II error. We also investigate the classification accuracy of different feature combinations. In addition, the computational complexity of key parts in our system is also discussed.

We structure this article as follows. In Section 2, we introduce the importance of urban functional regions analysis and the possibility to adopt mobile App accessing data to analyze urban functional regions. After the problem statement, in Section 3, we check raw mobile App usage data and design the system. In Section 4, we describe the feature extraction and analysis, which is the key part of our system. In Section 5, we evaluate our system and validate the system design and extracted features. Finally, we discuss the related work in Section 6 and conclude this work with a summary of our main findings in Section 7.

2. Dataset and Visualization

This section first introduces the details of how we collect the data of mobile App usage, based on which we extract valid information for urban functional region identification. Then, for better understanding, we visualize raw data in daily and weekly patterns. Finally, we define the region and urban functional region.

2.1. Dataset Description. This article utilizes mobile App usage information to identify urban function regions, which is collected by a major cellular network operator China Telecom with Deep Packet Inspection (DPI) appliances [13, 14]. DPI records mobile subscribers' temporal and spatial information when they connect to the cellular network.

We extract the information of mobile App usage with a systematic framework SAMPLES, which classifies network traffic generated by mobile Apps [15]. It utilizes constructs of conjunctive rules against the App identifier, which is discovered in a snippet of the HTTP header. This framework operates automatically with a supervised methodology and has been proved to identify over 90% of these Apps with 99% accuracy on average [15]. In addition, we crawl 2000 most popular mobile Apps on Apple App Store (iOS Apps) and Google Play (Android Apps) and apply SAMPLES to generate conjunctive rules to match each App's network traffic. As a result, SAMPLES achieves about 97% accuracy on matching these Apps [16].

Each extracted mobile App usage log contains a starting time (t_s) and an ending time (t_e) of App data accessing, an anonymized user ID (u), an App ID (a) used during the

start-end time, the amount of traffic flow (f) consumed by a during the connection time (t_s–t_e), and the connected cellular tower ID (c). Each cellular tower ID is associated with a location expressed by latitude and longitude (x_c, y_c).

The dataset includes 2084 cellular towers, 2000 commonly used Apps in China, and more than 2.1 million App usage logs. The data were collected from April 20, 2016 to April 26, 2016 in Shanghai, one of the largest cities in China. Each cellular tower contains more than 1000 logs on average. The spatial resolution of the dataset is decided by the cellular tower granularity, and the temporal resolution of the dataset is decided by the sampling rate, which is 60 seconds. Small cells lead to the higher spatial resolution of urban functional region identification, but at the cost of larger number of cells, the large scale and high resolution of the dataset provide a vast amount of information for urban functional region identification.

2.2. Data Preprocess and Visualization. In order to obtain regional useful information for urban function identification, we first discretize the logs into small time chunks: 1, 2, . . ., N, where N is the number of time chunks. Then, within each chunk i, we aggregate three variables of mobile usage logs from the same cellular tower c and derive *total number of unique Apps used* ($a^c[i]$), *total number of unique connected users* ($u^c[i]$), and *total amount of traffic flow consumed* ($f^c[i]$), based on which we extract features for urban function identification. For simplicity, we utilize *App number*, *user number*, and *traffic*, respectively, to represent these three variables in the following contents of the article.

After the preprocess, we derive *mobile App fingerprint* logs, each of which contains a cellular tower ID (c), a time chunk index (i), a *total number of unique Apps used* ($a^c[i]$), a *total number of unique connected users* ($u^c[i]$), and the *total amount of traffic flow consumed* ($f^c[i]$). For simplicity, we utilize App number, user number, and traffic to represent $a^c[i]$, $u^c[i]$, and $f^c[i]$, respectively.

To provide an intuitive understanding for feature extraction at a macro level, we first aggregate and visualize *mobile App fingerprint* from all 2084 cellular towers at each time chunk.

Figure 1 shows the temporal distribution of *mobile App fingerprint* in the entire week. Similar traffic flow patterns are observed on different days of the week. A similar phenomenon is also observed in the App number pattern and user number pattern. This indicates that information on different days in a week may be similar.

To check the details of daily *mobile App fingerprint* in a day, we plot the temporal aggregated mobile App fingerprint on a typical day (Monday, April 25, 2016) in Figure 2. The trends of the three variables in the figure are highly similar, which are tightly coupled with human activity patterns. High peaks and low valleys are observed during the day and night, respectively. Two peaks at 12:00 PM and 6:00 PM correspond to lunch and dinner time when people use mobile apps more often. This illustrates the possibility to extract features according to different human activity patterns at different urban functional regions.

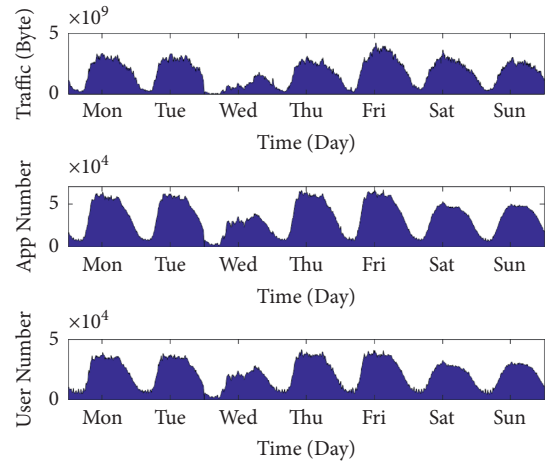

Figure 1: The temporal distribution of mobile App fingerprints (traffic, App number, and user number) in a week. Similar patterns of mobile App fingerprint are observed on everyday of the week. This indicates that information of different days in a week may be similar.

To further analyze the similarity between different days in a week, we calculate the correlation coefficients between different days. First, at each time chunk i, we aggregate traffic, App number, and user number from all cellular towers, respectively. Then, we separate them into 7 segments, which derives $A_1, \ldots, A_7$ for App number, $U_1, \ldots, U_7$ for user number, and $F_1, \ldots, F_7$ for traffic. We utilize subscript 1 to 7 to represent Monday to Sunday, respectively, seven days a week. Finally, we calculate the correlation coefficients between different days:

$$\begin{aligned} \mathrm{corr}A_{ij} &= \frac{A_i^T A_j}{\left|A_i\right|\left|A_j\right|}, i, j = 1, 2, \ldots, 7, \\ \mathrm{corr}U_{ij} &= \frac{U_i^T U_j}{\left|U_i\right|\left|U_j\right|}, i, j = 1, 2, \ldots, 7, \\ \mathrm{corr}F_{ij} &= \frac{F_i^T F_j}{\left|F_i\right|\left|F_j\right|}, i, j = 1, 2, \ldots, 7, \end{aligned} \tag{1}$$

where $\mathrm{corr}A_{ij}$, $\mathrm{corr}U_{ij}$, and $\mathrm{corr}F_{ij}$ denote the correlation coefficients of App number, user number, and traffic between two different days, respectively.

We calculate the correlations for all days versus the other days, which show similar patterns as those in Monday versus the others. Therefore, for simplicity but without loss of generality, we only show the correlation coefficients between Monday and other days in the way in Table 1. The *mobile App fingerprint* on Monday has high correlations with those on other days for all three variables (App number, user number, traffic).

It is noticed that the correlation of traffic between Monday and Wednesday is 0.58, which is the only one lower than 0.9. This is because Wednesday (April 20, 2016) is the first day of our data collection, which leads to missing data on App usage due to technical issues. As a result, a lot of

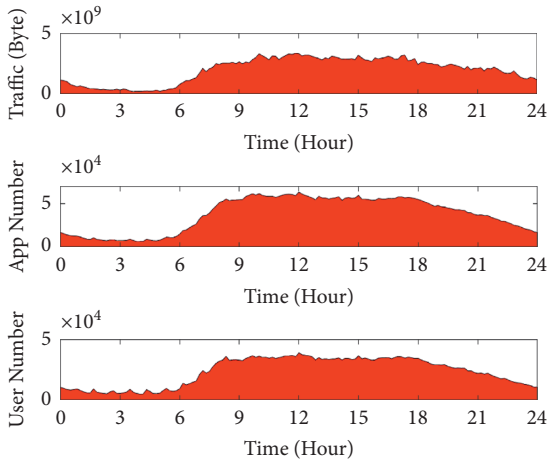

Figure 2: The temporal distribution of mobile App fingerprints (traffic, App number, and user number) in a typical day. The daily patterns of mobile App fingerprints are highly similar and tightly coupled with human activity patterns.

Table 1: Correlation coefficients between Monday and other days in the week.

	Traffic	App Number	User Number
Tuesday	0.97	0.99	0.98
Wednesday	0.58	0.91	0.90
Thursday	0.98	0.99	0.99
Friday	0.97	0.99	0.99
Saturday	0.97	0.98	0.97
Sunday	0.96	0.96	0.95

traffic flow consumed is not counted. However, that does not affect the user number and App number too much since a subscriber use multiple Apps and an App is used by multiple subscribers. The similarity between different days shows that the information of the entire week has high redundancy and we can use data on Monday to represent the information of the entire week. This helps to reduce data dimension to one-seventh of original data.

2.3. Definition of Region and Functional Region. In this article, a region is defined as a closed area, which is formed by level 3 roads of China, where no other level 3 road exists in the area. This definition is similar to the related work, which utilizes roads to segment the city map into small regions [1]. To derive final regions, morphological operators are required, which contains dilation, thinning, and aggregating. More details can be found in [1].

A functional region is a region that provides residents with various urban functions to meet their different needs of socioeconomic activities [1]. The socioeconomic activities include catering, shopping, going to school, and living. For example, the Songjiang University Town in Shanghai is a region consisting of many universities for college education [17]. The Gubei in Shanghai is a region consisting of many residential communities, where people from Japan, South Korea, Hong Kong, Macau, and Taiwan live [18].

3. System Design

This section first introduces how we design the system to address challenges mentioned in Section 1. Then, we provide details of the first module *pattern identification*.

3.1. System Architecture. Figure 3 shows our system architecture, which consists of 4 modules: *initial cleaning* (in green), *pattern identification* (in red), *offline training* (in blue), and *online classification* (in grey). Our system first cleans the noise from raw App usage logs in the *initial cleaning* module. Then, it reduces cellular tower level noise and feature level noise to derive effective features in the *pattern identification* module. Finally, the extracted features are used to do offline training and online classification.

Initial Cleaning. The raw dataset of mobile App usage logs cannot be directly utilized to extract effective patterns for urban functional region identification due to three reasons. First, the raw dataset includes redundant and conflicting logs due to collection technical issues. Second, the collected data are based on individual subscribers, while the urban functional region identification requires regional information. Third, as shown in Section 2, the dataset contains redundant information, which increases the complexity of the problem. Therefore, the *initial cleaning* module first cleans raw mobile App usage logs to remove redundant and conflicting logs and then prepares vectorized regional data according to cellular towers for pattern identification. The details will be discussed in Section 3.2.

Pattern Identification. It is difficult to identify unique patterns directly from vectorized regional data after initial cleaning for different urban functional regions. First, the regional data at different cellular towers have different noise levels. Second, the extracted features are noisy. Finally, there is no prior knowledge about the existence of mobile App usage patterns relating to urban functional regions. We do not know whether cellular towers of the same urban functional regions share the same patterns and how these patterns look like. All these factors make it challenging to identify the hidden pattern for urban functional region identification. To address these issues, the *representative filtering* first selects the most representative cellular tower data at different urban functional regions for *raw feature extraction*. Then, we extract the major frequency components to reconstruct the feature in *frequency component extraction*, which reduces the noise while keeping the useful information of features. Finally, *feature filtering* automatically selects the most distinguishable features for offline supervised model learning and online classification. The details will be discussed in Section 4.

Offline Training and *Online Classification.* The selected and filtered features from the *pattern identification* module are used for offline supervised model training and online classification. Three representative supervised machine learning models are adopted to check the validation of these features. POIs in the corresponding regions are utilized as

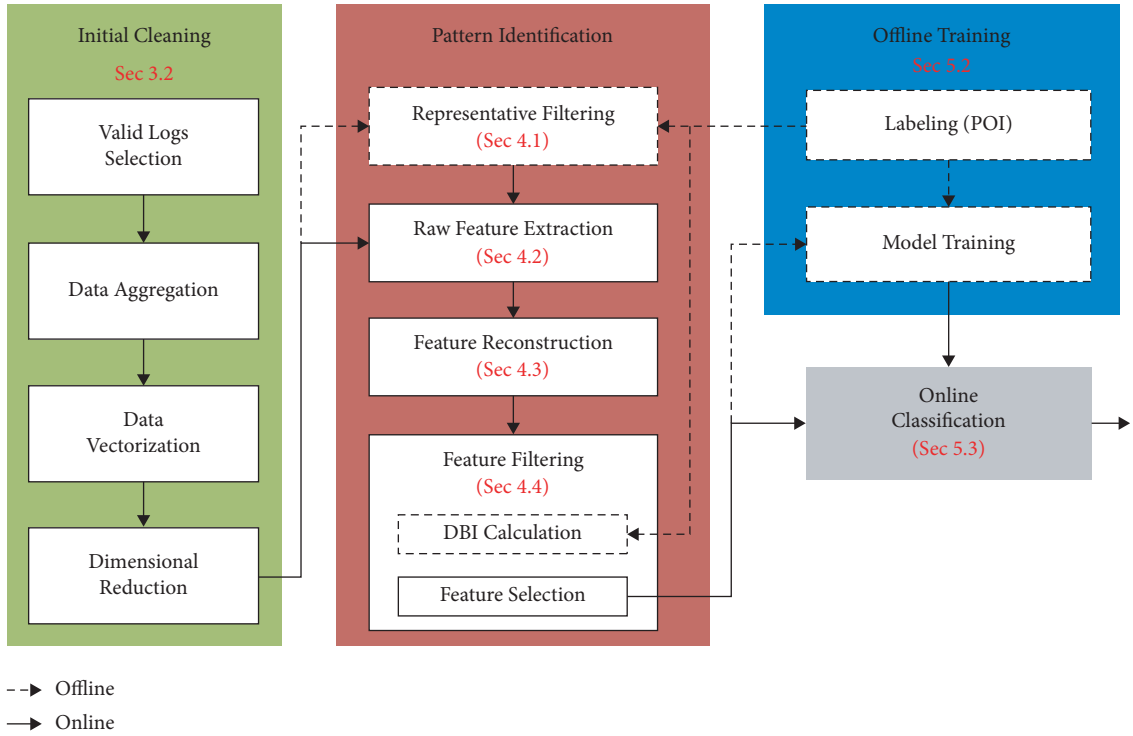

FIGURE 3: The architecture of our system for identifying urban functional regions with mobile App usage data, which consists of 4 modules: *initial cleaning* (in green), *pattern identification* (in red), *offline training* (in blue), and *online classification* (in grey).

labels and ground truth. A POI is a specific point location that may be useful or interesting to people. It is a term most often used on a map or guidebook that reflects the function of a region. Our system can identify five urban functional regions: catering, shopping, school, office, and residential area, which cover most human activity areas. The details will be discussed in Section 5.

3.2. Initial Cleaning. The *initial cleaning* module is composed of 4 parts: *valid logs selection* to remove redundant and conflicting logs, *data aggregation* and *vectorization* to get regional information and structure the regional information, and *dimensional reduction* to reduce data redundancy.

Valid Logs Selection. As mentioned in Section 2, we extract mobile App usage logs with SAMPLES, where each log contains a starting time (t_s) and an ending time (t_e) of App data accessing, an anonymized user ID (u), an App ID (a) used during the start-end time, the amount of traffic flow (f) consumed by a during the connection time (t_s–t_e), and the connected cellular tower ID (c). However, there exists redundant and conflicting logs, which will probably lead to the potential error in the *pattern identification*. Therefore, we first remove redundant logs with same t_s or t_e to present repetitive calculation in the *data aggregation* part. Then, we remove the logs that switch between multiple cellular towers in a very short time, that is, close t_s and t_e but different c. This is because the subscribers stay at the overlap areas of multiple cellular towers and their mobile phones frequently switch between these cellular towers. As a result, this part selects valid logs to ensure that at each time period (t_s–t_e), one subscriber only connects to one or no cellular tower.

Data Aggregation. Since urban function identification requires regional information, we first discretize the mobile App usage logs into small time chunks (10 minutes). Then, within each chunk, we aggregate three variables of mobile usage logs from the same cellular tower c and derive *App number* ($a^c[i]$), *user number* ($u^c[i]$), and *traffic* ($f^c[i]$), which compose the *mobile App fingerprint*, as described in Section 2.

Data Vectorization. In order to get structured data for *pattern identification*, we vectorize the three aggregated variables of *mobile App fingerprint*. For the cellular tower c, we derive $X^c = (a^c[1], \ldots, a^c[N])^T$, $Y^c = (u^c[1], \ldots, u^c[N])^T$, and $Z^c = (f^c[1], \ldots, f^c[N])^T$, where X^c, Y^c, and Z^c represent vectors of *App number*, *user number*, and *traffic*, respectively. In addition, to eliminate the differences caused by amplitude, we normalize each vector with daily maximum values.

Dimensional Reduction. Based on the visualization and analysis from Section 2, we know that the information of the entire week has high redundancy. Therefore, we first calculate the correlation coefficients of *App number*, *user number*, and *traffic* between different days according to (1), respectively. Then, a threshold (0.7) is set to decide whether the data include similar information for dimensional reduction. In the case of our dataset, we only keep the data on Monday due to its high correlation with other days.

4. Pattern Identification

In order to figure out unique patterns at different urban functional regions, we first visualize the raw *mobile App fingerprint* in Figure 4. We randomly select 60 cellular towers from two different urban functional regions, office, and residential areas and plot them on top and bottom rows. First, for cellular towers located at office or residential areas, the peak App traffic values vary significantly (from 7:00 AM to 1:00 AM). Although the peak values for App number and user number mostly appear from 9:00 AM to 10:00 PM, it is still difficult to identify unique patterns with peak values. In addition, the daily patterns at the same urban functional regions also vary significantly. Second, we do not observe the obvious difference between cellular towers in office and residential areas. This shows that simply cleaning mobile APP usage logs with the *initial cleaning* module cannot help identify unique patterns for feature extraction, which echoes the analysis in Section 3. We will introduce how we design 4 parts of the *pattern identification* module, *representative filtering*, *raw feature extraction*, *frequency component extraction*, and *feature filtering* to address these issues in the following 4 subsections, respectively.

4.1. Representative Filtering. Figure 4 illustrates that not all the cellular towers for the same urban functional regions share similar patterns. Therefore, we design the *representative filtering* to remove the data from outlier cellular towers and keep the most representative cellular towers for urban functional regions. The outlier cellular towers are those who do not share similar *mobile App fingerprint* patterns with most cellular towers for the same urban functional regions. Unlike normal cellular towers, the *total number of unique Apps used*, *total amount of traffic flow consumed*, and *total number of unique connected users* in these outlier cellular towers show different patterns, which does not correspond to regular human activities.

To pick up representative cellular towers, which shares the similar pattern for the same urban functional regions, the hierarchical clustering method is adopted [19] to cellular towers, which belongs to the same type of urban functional regions. As shown in Algorithm 1, we first select App fingerprint data of cellular towers with the same labels (POIs). Then, treating each input (cellular tower) as a cluster, the hierarchical clustering iteratively merges the nearest two clusters. In the clustering, we take correlation (computed as shown in (1)) as the distance metric, since we care about the trend and pattern of the data. This process will stop until the distance between any two clusters is larger than a predefined threshold value d_{th}. The threshold value is decided based on our observation of the raw dataset, which makes sure that any two clusters are distinguishable. Finally, we select the clusters with more than M cellular towers. This is because clusters of a small number are not representative. In this article, based on our observation of the raw data, we M as 100, which not only prevents too many noisy and outlier data being selected but also keeps data of most representative cellular towers who share similar patterns. It is noticed the *representative filtering* is only used for offline model learning, which contains POI labeling information. For the online classification, this step will be skipped.

Figure 5 shows the selected cellular tower data after *representative filtering*. We plot *mobile App fingerprint* at two kinds of urban functional regions (office areas and residential areas) at the top and bottom row, respectively. It is noticed that *mobile App fingerprint* are all normalized by their daily maximum values and we show 60 randomly selected cellular tower data for better comparison. Figure 5 shows the consistent pattern on three variables of *mobile App fingerprint* for the same urban functional regions. In addition, *mobile App fingerprint* shows different patterns between different urban functional regions. For example, for user number, the residential area shows two high peaks at 8:00 AM and 7:00 PM, whereas office area shows high values between these two time. This corresponds to human daily activity pattern that people go to office from home around 8:00 AM and leave office for home around 7:00 PM.

4.2. Raw Feature Extraction. In order to show detailed pattern differences between five urban functional regions, we show traffic, user number, and App number in Figure 6. Each column represents one urban functional region. Values in each subfigure are obtained by averaging data from filtered cellular towers. For better comparison, we normalize the values of each subfigure by their maximum values, respectively, whose values are shown in Table 2. Different patterns are observed at different urban functional regions, based on which features can be extracted. Figure 6 gives us reference on how to select features for urban functional region identification, and we discuss the details of raw feature selection in this subsection. In addition, we provide insight into these features for different urban functional regions. Due to the page limitation and avoiding redundant description of features, we just discuss three representative features from App number in detail. Peak position and negative slope contain information of important value position and data changing rate, respectively, whereas weekday-weekend ratio represents the difference between weekday and weekend. In addition, to illustrate their differences between five urban functional regions, we plot the average values and standard deviation values of the same three features in Figure 7.

Peak Position: In Figure 6, daily peaks of App numbers appear at different times for different urban functional regions. We plot the average and standard deviation of daily App number peak positions in the left of Figure 7. The school area shows the earliest average peak time (just after 12:00 PM), whereas the shopping area shows the latest average peak time (around 4:00 PM). This is because Chinese students tend to use larger number mobile Apps, such as gaming and social Apps, during the lunch, while people tend to go shopping during afternoon when they use most Apps, such as map, recommendation, and social Apps.

Negative Slope: In Figure 6, both increasing and decreasing rates differ among urban functional regions. This is also observed in the middle of Figure 7. Negative slopes in school areas decrease fastest, which indicates that students

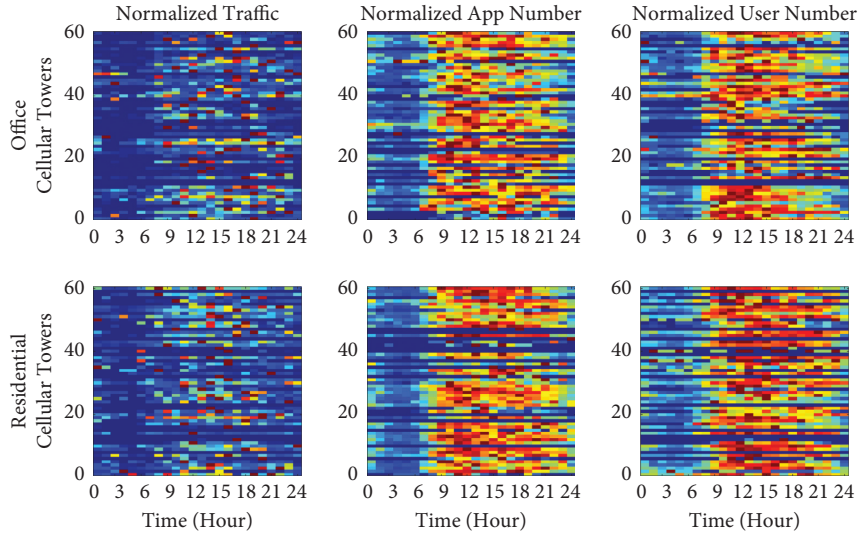

FIGURE 4: The heatmap of normalized *mobile App fingerprint* from the *initial cleaning* module. The 60 cellular towers are randomly selected at two urban functional regions (office and residential areas), respectively. It is difficult to identify unique patterns for different urban functional regions.

Input: App fingerprint data belongs to the same type urban functional regions $D = \{x_1, x_2, \ldots, x_m\}$;
cluster distance function d;
distance threshold value d_{th}
Output: Clusters $C = \{C_1, C_2, \ldots, C_k\}$
for $j = 1, 2, \ldots, m$ **do**
  $C_j = \{x_j\}$
end
for $i = 1, 2, \ldots, m$ **do**
  for $j = 1, 2, \ldots, m$ **do**
    $M(i, j) = d(C_i, C_j)$;
    $M(j, i) = M(i, j)$
  end
end
Set current number of clusters: $q = m$
while *the distance of any two clusters* $< d_{\text{th}}$ **do**
  Find the nearest two clusters C_{i^*} and C_{j^*}
  Union C_{i^*} and C_{j^*}: $C_{i^*} = C_{i^*} \cup C_{j^*}$
  for $j = j^* + 1, j^* + 2, \ldots, q$ **do**
    Renumber clusters set C_j to C_{j-1}
  end
  Delete the j^*th row and the j^*th column of the distance matrix M;
  for $j = 1, 2, \ldots, q - 1$ **do**
    $M(i^*, j) = d(C_{i^*}, C_j)$;
    $M(j, i^*) = M(i^*, j)$
  end
  $q = q - 1$
end

ALGORITHM 1: Hierarchical clustering: AGNES (AGglomerative NESting).

switch between different Apps more often. In contrast, negative slopes in shopping areas decrease the slowest. This is because the Apps used when people go shopping is usually more stable.

Weekday-Weekend Ratio: Although the information on Monday has a high correlation with the other days, the absolute value of different days still varies, especially between weekdays and weekends. We plot the used App number on the weekday over that on the weekend in the right of Figure 7. The results show that schools have the highest ratio ($> r\text{bin}1$) since students leave school during weekends and fewer Apps are used. On the contrary, the shopping area has the lowest ratio (< 1) since citizens go shopping more during weekends and more Apps are used.

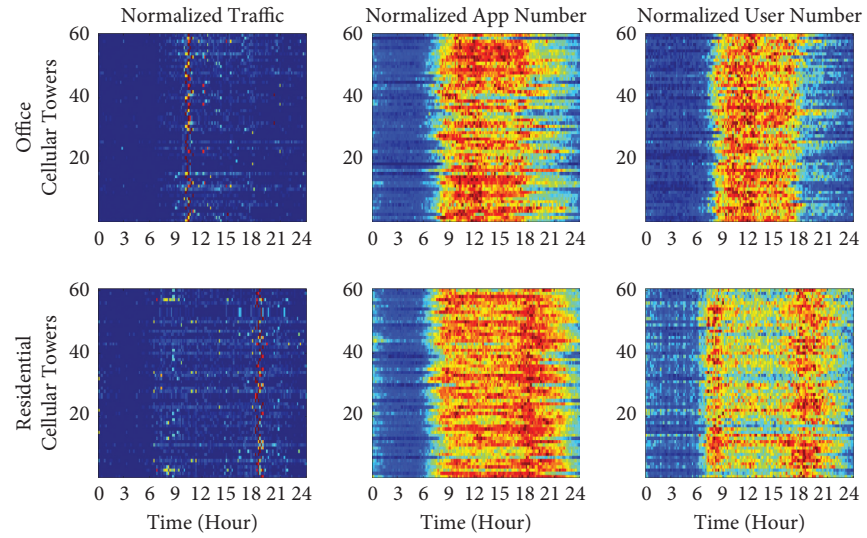

Figure 5: This figure shows the heatmap of normalized *mobile App fingerprint* after *representative filtering*. The 60 cellular towers are randomly selected at two urban functional regions (office and residential areas). Clear unique patterns for different urban functional regions are observed after *representative filtering*.

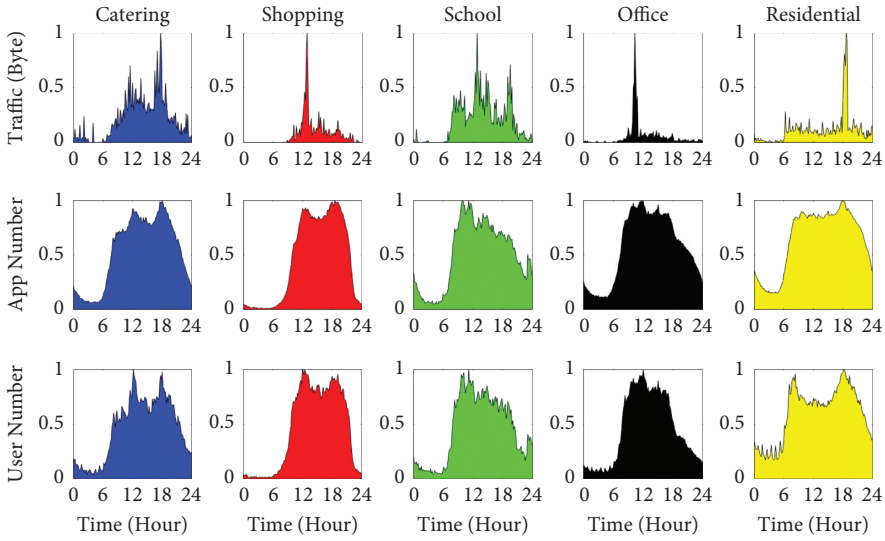

Figure 6: The normalized daily patterns of user number, App number, and traffic after *representative filtering*.

Table 2: Maximum values of *mobile App fingerprint* for five urban functional regions.

	Traffic (Byte)	App number	User number
Catering	$7.7 * 10^6$	524	372
Shopping	$5.7 * 10^6$	286	1057
School	$5.8 * 10^6$	456	281
Office	$1.4 * 10^7$	238	7534
Residential	$7.5 * 10^6$	348	3440

In total, we first consider 15 raw features for each variable of mobile App fingerprint, which include weekday peak value, weekend peak value, weekday peak position, weekend peak position, weekday valley value, weekend valley value, weekday valley position, weekend valley position, weekday valley-peak ratio, weekend valley-peak ratio, weekday positive slope, weekend positive slope, weekday negative slope, weekend negative slope, and weekday-weekend ratio. It is noticed that not all these features can be used for classification since some of them have large noise and some of them contain redundant information. The details of feature noise reduction and filtering will be addressed in the following two subsections.

4.3. Frequency Component Extraction. This subsection addresses the problems of feature noise reduction. We first show that the directly extracted feature cannot be used for classification due to the high noise level. Then, we reduce the feature noise by extracting the key frequency component of these features, based on which these features are reconstructed.

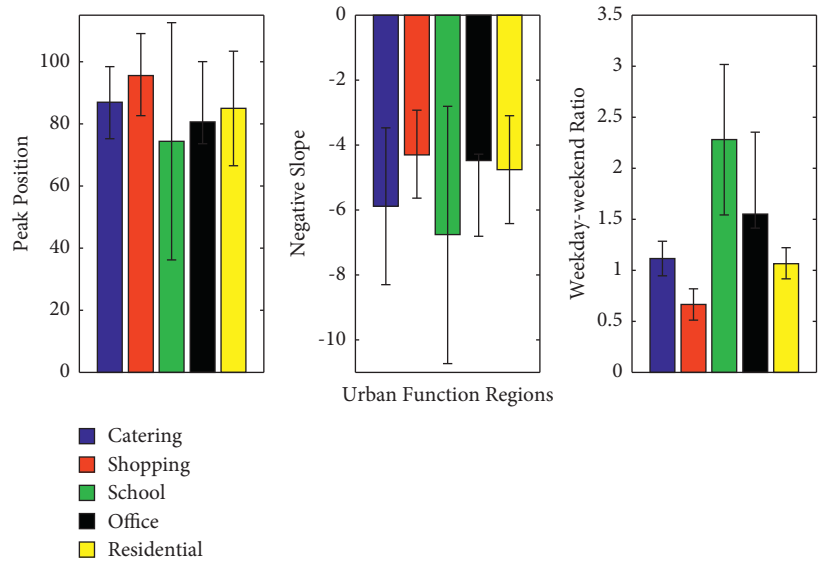

Figure 7: Three representative features directly extracted from App number data after *representative filtering*.

For all the three features shown in Figure 7, the standard deviation is also large despite different average values among different urban functional regions. The large standard deviations mean that within each urban functional region, features from different cellular towers vary significantly, which causes classification errors with these features. Therefore, we need further reduce the feature noise to improve the distinguishability of these features.

In order to lower the variation of features within the same urban functional region, we analyze the frequency pattern of the daily fingerprint. The discrete Fourier transform (DFT) [20] is applied on the original data after *representative filtering*, which is calculated as

$$\widehat{D}[k] = \sum_{n=1}^{N} d[n] e^{-2\pi i k n/N}, \tag{2}$$

where N is the number of samples. $\widehat{D}[k]$ is the frequency spectrum of time-domain data d. Based on the analysis on the spectrum after DFT, we find that most energy is gathered at 4 frequency components, which represent 24-, 12-, 8-, and 6-hour periods, which correspond well to the human life patterns. Therefore, we utilize these four frequency components to reconstruct the data as

$$\begin{cases} \widehat{D}^r[k] = \begin{cases} \widehat{D}[k], & \text{if k is the selected frequency,} \\ 0, & \text{otherwise,} \end{cases} \\ d^r[n] = \dfrac{1}{N} \displaystyle\sum_{k=1}^{N} \widehat{D}^r[k] e^{-2\pi i k n/N}, \end{cases} \tag{3}$$

where $d^r[n]$ is the reconstructed time-domain data. Figure 8 shows the original data ($d[n]$) after *representative filtering* and the reconstructed data ($d^r[n]$) with 4 key frequency components of one randomly selected cellular tower in blue and red line. One typical cellular tower is randomly selected for each urban functional region. It is observed that the red lines show the same trends with the blue lines, while the high oscillations in the blue lines do not exist in the red lines. This proves that the selected 4 key frequency components represent the most important information of the data as well as remove the unnecessary noise.

Figure 9 shows the average and standard deviation of 3 features extracted from the reconstructed data. These 3 features are the same as those in Figure 7.

The average values of peak position of 5 urban functional regions in Figure 9 (left) do not change too much from original features showing in Figure 7. This is because the reconstruction does not change the signal trend a lot, thus keeping the daily peak position at the similar place. To be noticed, the standard deviations are much smaller than the original features; that is, the standard deviation of school area decreases from 38.3 to 13.4, which helps reduce the classification errors.

Different from average values of the peak position, the average negative slope of 5 urban functional regions in Figure 9 (middle) varies significantly. This is because that high-frequency oscillation of the original signal leads to a high decreasing rate, which does not represent the really stable decreasing rate. In addition, the standard deviations are also reduced a lot. For example, the standard deviation of the school area decreases from 4.0 to 0.6. Since both standard deviations and average values are reduced on a similar scale, the differentiation between different urban functional regions does not improve a lot.

The average values of weekday-weekend ratios of 5 urban functional regions in Figure 9 (right) do not change a lot from Figure 7. This is because the reconstruction does not change the signal trend a lot, thus keeping the weekday-weekend ratio at similar levels. However, the standard deviations are much smaller than raw features in Figure 7. For example, the standard deviation of the school area decreases from 0.74 to 0.14. The reduction in the standard deviations reduces the distance within the same urban functional regions, which reduces the classification errors.

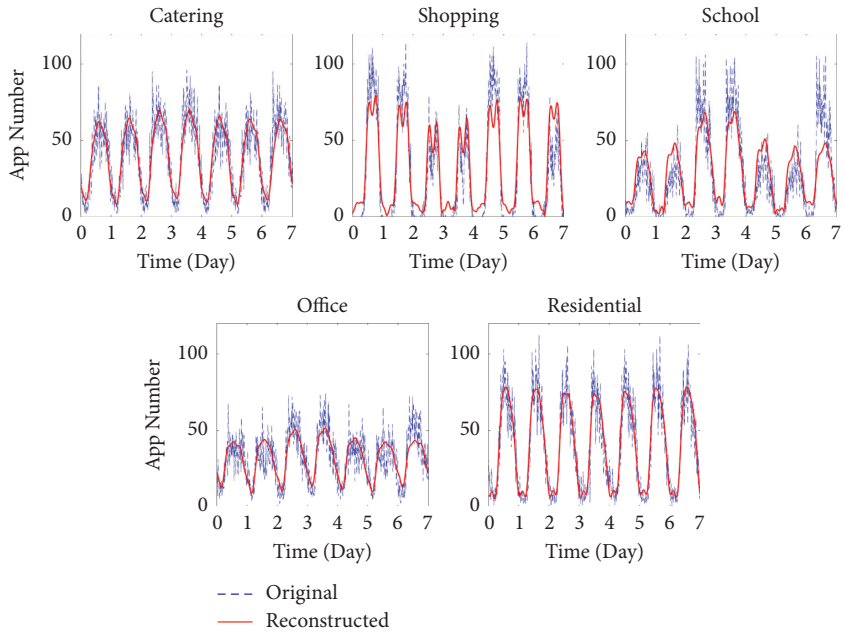

Figure 8: The original used App number data after *representative filtering* and reconstructed used App number data with 4 major frequency components.

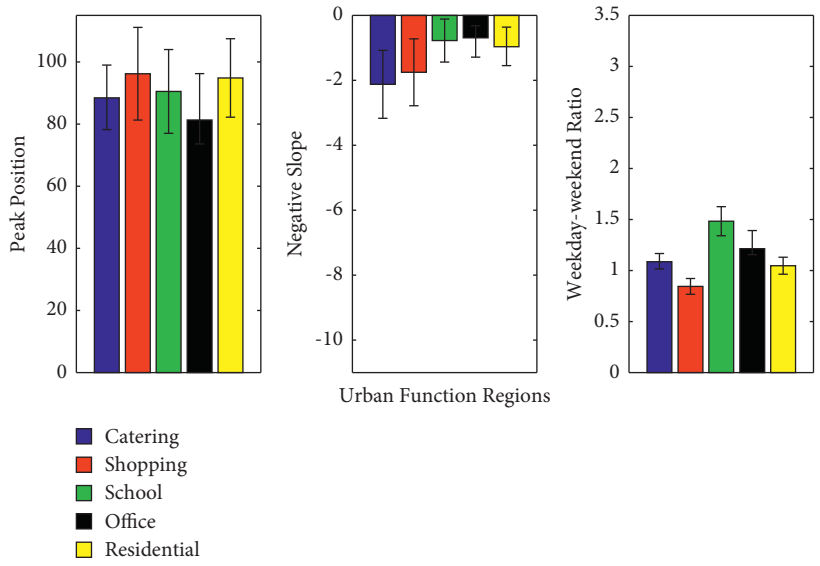

Figure 9: Reconstructed features of 5 urban functional regions. The figure shows the same 3 App number features extracted from reconstructed data. First, the reconstructed data reduce the oscillation from the original data while keeping the major trends. This means that the reconstructed data reduce the noise and keep the useful information. Second, the features extracted from reconstructed data have lower standard deviations while keeping the trends of average values. This means the features extracted from reconstructed data increase the differentiation between urban functional regions, which helps reduce the misclassification errors.

To conclude, adopting 4 frequency components to reconstruct the original data keeps the most important information for feature extraction while reducing the feature noise, which helps improve the distinguishability of these features.

4.4. Feature Filtering. The next question lies on how to select distinguishable features from all 45 feature candidates for accurate classification. In our system, we adopt the Davies–Bouldin index (DBI) as a selection metric [21], which is defined as

$$\frac{1}{N}\sum_{i=1}^{N}\max_{j=1,j\neq i}^{N}\frac{S_i+S_j}{E_{i,j}}, \tag{4}$$

where N is the number of classes and $E_{i,j}$ is derived by

$$E_{i,j}=\left\|A_i-A_j\right\|_2, \tag{5}$$

and S_i is calculated as

$$S_i=\frac{1}{N_i}\sum_{k=1}^{N_i}\left\|X_k-A_i\right\|_2, \tag{6}$$

where A_i denotes the centroid of each class and N_i is the number of ith class. The DBI not only measures the separation between classes but also considers the cohesion within each class. Based on the calculated DBI values in the offline training period, we select 10 features (most DBIs are between 3 and 19) from 45 features for urban functional region classification: 1 weekday App traffic valley position, 2 weekend App traffic negative slope, 3 weekday App number valley-peak ratio, 4 weekend App number valley-peak ratio, 5 weekday App number peak position, 6 weekend App number valley position, 7 weekday App user number valley-peak ratio, 8 weekend App user number peak position, 9 weekday App user number valley position, and 10 weekend App user number valley position.

5. Evaluation

This section evaluates our system design and feature extraction. We first introduce the evaluation setup in Section 5.1. Then, we describe details of 3 machine learning methods in Section 5.2. Finally, we evaluate the performance of online classification in Section 5.3. Our evaluations focus on (1) evaluating the overall performance of our system design and feature extraction in Section 5.3.1; (2) evaluating the effect of the key parts of the system, that is, *representative filtering*, *feature reconstruction,* and *raw feature extraction* in Section 5.3.2; (3) validating the features selected by the DBI-based method of *feature filtering* in Section 5.3.3; and (4) evaluating computational complexity of *pattern identification* module and three machine learning methods in Section 5.3.4.

5.1. Experiment Setup. We introduce the experimental configuration, including testing set, ground truth, machine learning methods, and comparing sets, respectively.

Tenfold cross-validation is adopted. We first randomly divide the dataset, as described in Section 2.1, into ten groups with equal sample numbers. Each time, nine groups are used to train the machine learning model, whereas the remaining one group is used to test the accuracy.

Ground Truth: We separate the city of Shanghai into small regions and correlate them to point of interest (POI) provided by BaiduMap API [22]. We adopt POIs of the regions as the ground truth of urban functional regions. It is noticed that there usually exist multiple numbers and type POIs within the same region due to its multiple functions. Since we focus on identifying the regions with the single function, we select the regions with one dominant POI ($>r$bin80%), which represents the function of that region. Only the cellular towers located in these regions will be used for evaluation. We focus on 5 types of urban functional regions: *catering*, *shopping*, *school*, *office*, and *residential*.

Dataset and Testing Method: Since our dataset has the problem of imbalanced data, that is, data points of classes are not approximately equally, we first adopt oversampling method [23] to avoid misclassification. Then, we adopt tenfold cross-validation to evaluate the classification performance. Specifically, we randomly divide the dataset described in Section 2 into ten groups with equal sample numbers. Each time, nine groups are used to train the machine learning models, whereas the remaining one group is used to test the accuracy.

Performance Metric: The major metric we consider is *Accuracy*, which is calculated as

$$\begin{cases}\text{Accuracy}=\sum_{i=1}^{n}\frac{M_{ii}}{N_T}*100\%,\\ N_T=\sum_{i=1}^{n}\sum_{j=1}^{n}M_{ij},\end{cases} \tag{7}$$

where M_{ij} is the number of the real urban function region i classified as urban function region j. As mentioned before, we have 5 classes in our evaluation, that is, $n=5$. We tune the parameters of machine learning models to optimize classification accuracy. In addition, we also consider *type I error* and *type II error* with confusion matrix [24], as well as running time. To be noticed, in our multiclassification problem, since there exist the similar number of samples for each class in the testing sets, high *Accuracy*.

Machine Learning Methods: To investigate how our system works with different machine learning methods, we adopt three commonly used methods to compare the performance difference. We first select support vector machine (SVM) [25], a low complexity classier, which is based on statistical learning theory. Then, we select decision tree (DT) [26], a low complexity classier, which is based on the multistage or hierarchical decision. The SVM and the DT represent parametric and nonparametric supervised learning, respectively. Finally, we adopt random forest [27], a more complicated classier and ensembled classifier, which combines more than one same or different basic classifiers, such as SVM and DT.

Baseline: We compare our method with a most recent related work [28], which adopt mobile traffic data to classify urban functional regions. To be simple, we utilize MTC to represent this baseline in the following part of the article.

5.2. Offline Training of Machine Learning Methods. SVM is a discriminative classifier, which is formally defined by a separating hyperplane [25]. The algorithm outputs an optimal hyperplane based on the calculation on labeled training data. The hyperplane is used to classify new examples. In order to derive hyperplanes with more complicated forms, kernel functions are used [29]. SVM can be used to deal with multiclassification problem by combining multiple SVMs [30]. In our *offline training* module, we adopt one-versus-one method to construct a multiclass SVM

classifier. A polynomial kernel function with order 3 is utilized. The regularization parameter C is set to 1.

Decision tree is a nonparametric effective machine learning modeling, which utilizes a tree-like model of decisions and their possible consequences [26]. The decision tree classifier repetitively divides the working area into subparts according to the information gain [31]. In addition, the pruning technique can be used to reduce the complexity of the final classifier and prevent overfitting [32]. In our *offline training* module, in order to reduce the computational complexity, we adopt Gini impurity to calculate information gain and trained classifier is not pruned [33].

Random forest is an ensemble algorithm that combines more than one algorithm of the same or different kinds for classification [27]. A random forest classifier creates a set of decision trees and aggregates the votes from different decision trees to make the final classification decision. In our *offline training* module, the random forest consists of 3 decision trees, each of which adopts Gini impurity to calculate information gain.

5.3. Performance Analysis of Online Classification

5.3.1. System Performance. Figure 10 shows the classification accuracy of our system and the baseline using three different machine learning models. For all three methods, our method achieves 47%, 20%, and 10% improvements over baseline. In addition, with three different methods, our method achieves similar accuracy (97.3%, 95.14% 97.37%), whereas the baseline achieves various accuracy (50.27%, 76.76%, 88.42%). This shows that our method on feature cleaning and extraction brings performance improvement and robustness with different machine learning methods. The improvements and robustness come from (1) adopting extra information of App number and user number and (2) our feature cleaning technique in the *pattern identification* module.

Figure 11 shows the confusion matrix of our system with three machine learning models. There is no misclassification on *catering*, *shopping,* and *school* due to different patterns of Traffic, user number, and App number in these three kinds of urban functional regions. All three methods have errors on classifying *office* since the feature patterns have the largest overlaps with other classes based on the DBI calculation. In addition, there exist 3 misclassifications in *residential* with decision tree methods, while that does not happen with random forest. This is because random forest utilizes 3 decision trees and aggregate the votes from these trees, which prevent the misclassification.

5.3.2. Influence of Pattern Identification. In order to check the effect of key parts, we evaluate the accuracy of 4 comparing sets:

(i) Initial Cleaning: In this testing configuration, we adopt the raw data from all the cellular towers after the *initial cleaning* module as the feature for classification.

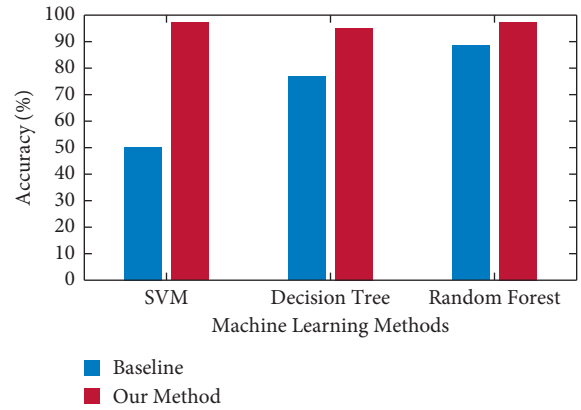

Figure 10: This figure shows the classification accuracy of our method and the baseline with three different machine learning models.

(ii) Representative Filtering: In this testing configuration, we adopt the raw data from selected cellular towers after *representative filtering*. Outlier cellular tower data are removed from the previous comparing set. By comparing *Representative Filtering* with *Initial Cleaning*, we check the performance improvement from *representative filtering*.

(iii) Raw Feature Extraction: In this testing configuration, we adopt the features extracted from *Representative Filtering*. As shown in Figures 7 and 9, the extracted features include significant noise. By comparing raw feature extraction with *Representative Filtering*, we can the effect of feature extraction plus noise.

(iv) Feature Reconstruction. In this testing configuration, we adopt the features extracted from reconstructed data with 4 major frequency components, which also represent our system performance. Since the reconstructed data keep the most useful information as well as remove the noise, the *Features Reconstruction* is supposed to be better than *raw feature extraction*. By comparing feature reconstruction with *raw feature extraction*, we can check the effect of frequency component extraction. It is noticed that for both raw feature extraction and feature reconstruction sets, we use the selected 10 features selected based on DBI calculation.

We plot the classification accuracy of 4 comparing sets with three methods in Figure 12. For all methods, the initial cleaning set shows the lowest accuracy, whereas the feature reconstruction set shows the highest accuracy. This shows that directly applying raw data for urban functional region classification does not achieve satisfying accuracy. In contrast, through *representative filtering*, *raw feature extraction,* and *feature extraction*, our system improves the classification accuracy from ~50% to ~95%.

In addition, for the raw feature extraction set, three methods achieve 86%, 90%, and 92% accuracy, respectively. Performance from all methods is close and acceptable. This validates our selected features, which preserve the most

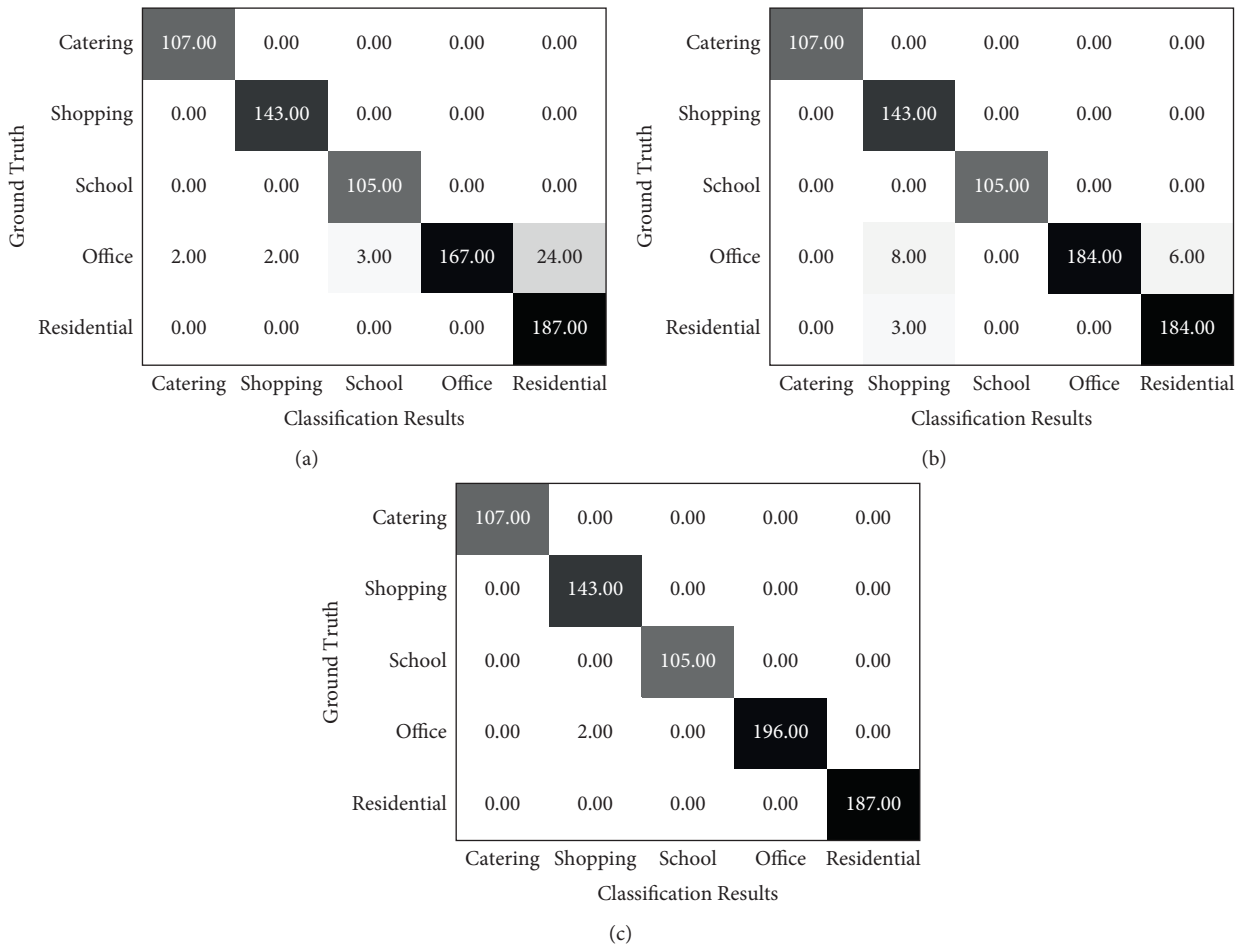

Figure 11: Classification confusion matrix with three different machine learning models.

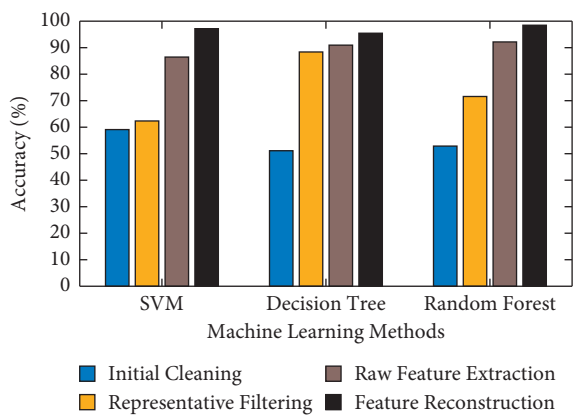

Figure 12: The classification accuracy of 4 comparing sets with 3 machine learning methods.

important information to classify different urban functional regions. However, directly extracted features still contain high-level noise that does not bring very high accuracy.

The *feature reconstruction* brings further accuracy improvements on three methods (11% for SVM, 5% for decision tree, and 6% for random forest). This also illustrates that *feature reconstruction* helps reduce the feature noise while preserving the key information for classification.

Therefore, three key parts of our system (*representative filtering*, *raw feature extraction*, and *feature reconstruct*) work cooperatively to ensure both high classification accuracy (up to 98%) and high robustness (97%, 95%, and 98%) in different machine learning methods.

5.3.3. Influence of Feature Selection. In order to check how different features affect our system performance, we show the classification accuracy with different feature combinations in Figure 13. We first separate the features into three groups: 2 App traffic features (No. 1 and No. 2), 3 used App number features (No. 3–No. 6), and 2 App user number features (No. 7–No. 10). Then, we increase the feature number for classification from 1 to 10 to check the classification accuracy with three machine learning methods.

From the results, we can observe that three methods have a large accuracy gap with only 2 App traffic features. SVM achieves only 50% accuracy, but decision tree and random forest achieve 76% and 88% accuracy, respectively. This illustrates that only adopting App traffic features does not offer enough information for accurate classification. In addition, it is also sensitive to machine learning methods.

When 4 App number features are added, SVM has around 40% accuracy improvement. This illustrates App

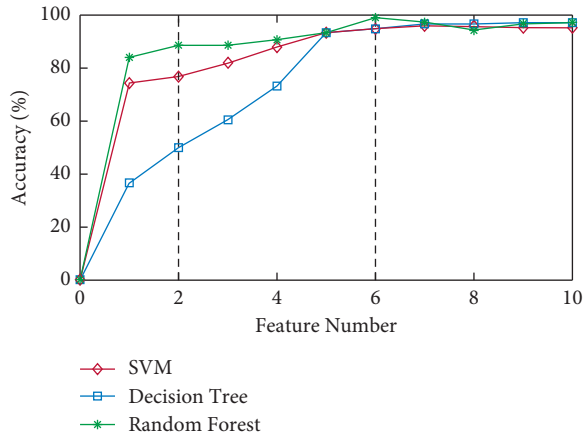

Figure 13: The classification accuracy under different feature combinations.

number features contain new valid information for classification. Decision tree and random forest show around 22% and 10% accuracy improvement, respectively. This illustrates that App traffic and App number feature combinations are less sensitive to machine learning methods compared to only App traffic features.

When the remaining 4 user number features are added, accuracy with SVM and decision tree is improved by 5% and 2%, respectively. However, accuracy with random forests does not improve. This shows that user number brings limited new information for classification.

In order to check the effectiveness of features in detail, we show the classification accuracy of 7 different combinations with three machine learning methods in Table 3. With only one feature set, the user number brings the highest accuracy for all three methods. This is because the user number is the most direct factor that reflects human activity, which is highly related to urban functional region type. When adding another feature set, we see accuracy improvements vary from ~10% to ~40% with different feature combinations and different machine learning models. With all three feature sets, all the machine learning models achieve the highest classification accuracy. However, the improvement is limited. This is because these three sets of features have information overlap. This shows the possibility of using any combination of two feature sets for urban functional region classification.

5.3.4. Computational Complexity Analysis. In order to check the computational complexity of key parts in our system, we run the system 20 times and record the running time to calculate the average running time. Table 4 shows the average running time of 5 key parts in our system. We run the system on a 64 bit windows server, whose CPU is Intel(R) Xeon(R) CPU E5-2637 v4 @ 3.5 GHz 3.5 GHz (dual processor) and memory is 128 GB. Initial cleaning and representative filtering rank in the top two on time consumption. The high computation of *initial cleaning* comes from aggregating mobile usage logs from individual users to derive user number, App number, and traffic of each cellular towers, which requires calculation over all mobile usage logs. The high computation of *representative filtering* comes from the expensive computation of hierarchical clustering in Algorithm 1, time complexity of which is $O(n^3)$.

In order to check the computational complexity of training different features, we plot the training time and classification accuracy in Figure 14. For simplicity but without loss of generality, we only show the training time and classification accuracy of SVM. SVM achieves high accuracy with 6 features, which corresponds to the analysis above. The accuracy does not change too much after 7 features. The training time does not change too much with less than 16 features, which is around 42 s. The training time increases dramatically after 17 features. Therefore, considering the trade-off between classification accuracy and training time, adopting features from any two feature sets for urban functional region classification can achieve high accuracy with low training time.

In conclusion, with the help of 4 key parts of our system and 10 extracted features within on App number, user number, and traffic, our system achieves up to 98% classification accuracy on urban functional region identification. In addition, the key parts and features are also robust to different kinds of supervised learning methods. It is noticed that adopting combinations of any two sets of features can achieve high classification accuracy (95%).

6. Related Work

In this section, we discuss the existing research related to our work, including urban function detection, and urban analysis with data from the cellular network or mobile devices.

Some studies have been carried out on urban function detection. A common method is to collect data from mobile devices for investigating human activities [5], application usage [6], and human communication activities [7]. However, the limited number of sampled users is not able to represent the global characteristics of the whole area. Some research studies focus on investigating land usage with Call Description Records (CDRs), such as data of phone call [8] and text message [9]. Classical time series analysis methods are used to infer land use. Reades et al. [34] adopt the principal component analysis approach and Yuan et al. [35] use the dynamic time warping approach. Soto et al. [36] propose the fuzzy C-means approach to classify places based on usage. Since people tend to use the application more often than phone calls, these datasets may have some bias or missing information. In addition, more and more users prefer applications such as WhatsApp, WeChat, and Line to send text messages. Therefore, these CDR data may lose a lot of key information for investigation.

People have adopted data from mobile devices and cellular networks for different aspects of urban analysis. The first category is population distribution estimation. Ahas et al. [37] and Becker et al. [38] design a new method to detect human living and working locations. De Jonge et al. [39], Ratti et al. [40]. and Sohn et al. [41] identify user mobility on high-level properties with coarse-grained GSM

Table 3: Classification accuracy with different feature combinations.

	SVM	DecisionTree	RandomForest
Traffic	50.27	77.30	81.43
AppNumber	68.78	85.95	88.86
UserNumber	74.05	87.57	92.43
AppNumber + UserNumber	96.22	94.22	97.73
UserNumber + Traffic	94.05	92.43	96.05
AppNumber + Traffic	94.73	93.11	95.32
AllFeatures	97.03	95.27	98.37

Table 4: Average running time of different parts.

InitialCleaning	Represen – tativeFiltering	RawFeatureExtraction	FeatureReconstruction	FeatureFiltering
15.1101 s	29.9022 s	394 ms	4.137710 s	440 ms

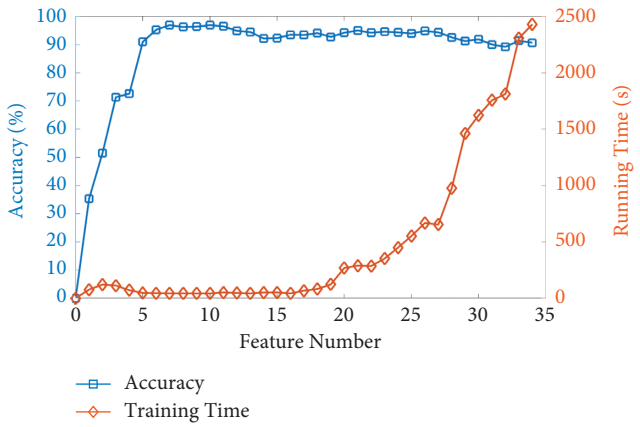

Figure 14: The training time of different feature numbers with SVM.

data. Krisp et al. [42] adopt mobile phone density calculation and visualization to help fire and rescue services. Soto et al. [43] identify the socioeconomic levels of a population with the information extracted from the aggregated cell phone usage record. The second category lies in activity estimation. Girardin et al. [44] track the evolution of attractiveness in New York with the cell phone data and Flickr georeferenced photos. Reades et al. [34] propose to monitor city dynamics and derive clusters of geographical areas. Frias-Martines et al. [36] propose a time series analysis method to automatically identify land use. Moreover, different aspects of mobility patterns are analyzed, such as human trajectories by Song et al. [45], human migration by Simini et al. [46], and road usage patterns by Wang et al. [47]. Several works aim at detecting human mobility patterns during special events like earthquake [48, 49].

7. Conclusion

This article proposed a fine-grained urban functional region identification system, which utilized mobile App usage data from cellular towers. We first design a hierarchical clustering-based method and a Fourier transform-based method to reduce cellular tower level noise and feature level noise, respectively. Then, we designed a DBI-based method to automatically figure out the most distinguishable features based on *mobile App fingerprint* (App number, user number, and traffic) and correlated them to different urban functional regions. We evaluated our system and selected features with three representative supervised learning models, all of which achieved more than 95% classification accuracy.

Data Availability

No data were used in the study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant Numbers 61976178 and 62076202, the Aero Science Foundation of China under Grant Number 202051053002, and the National College Students' Innovation Project under Grant Number 202110699037 (corresponding author: Lei Deng). Lei Deng and Hangyu Hu are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China, and Lei Deng is also with the Science and Technology on Electro-optic Control Laboratory(e-mail: denglei@nwpu.edu.cn).

References

[1] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, "Discovering urban functional zones using latent activity trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 712–725, 2015.

[2] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 186–194, ACM, Beijing China, August, 2012.

[3] Y. A. Xu, Z. A. Qiang, B. Pca, and N. Qiang, "Fine-grained Predicting Urban Crowd Flows with Adaptive Spatio-Temporal Graph Convolutional Network," *Neurocomputing*, vol. 446, 2021.

[4] Cisco, *Cisco Visual Networking index: Global Mobile Data Traffic Forecast Update, 2016–2021 white Paper*, 2017, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[5] A. Noulas, C. Mascolo, and E. Frias-Martinez, "Exploiting foursquare and cellular data to infer user activity in urban environments," in *Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management (MDM)*, pp. 167–176, IEEE, Milan, Italy, June, 2013.

[6] J. Toole, M. Ulm, and D. González, "Inferring land use from mobile phone activity," in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pp. 1–8, ACM, New York, NY, USA, August, 2012.

[7] B. Cici, A. Markopoulou, E. Frías-Martínez, and N. Laoutaris, "Quantifying the potential of ride-sharing using call description records," in *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, February, 2013.

[8] T. Pei, S. Sobolevsky, C. Ratti, S. Shaw, T. Li, and C. Zhou, "A new insight into land use classification based on aggregated mobile phone data," *International Journal of Geographical Information Science*, vol. 28, no. 9, pp. 1988–2007, 2014.

[9] M. Shafiq, L. Ji, A. Liu, J. Pang, and J. Wang, "Characterizing geospatial dynamics of application usage in a 3g cellular data network," in *Proceedings of the 2012 Proceedings IEEE INFOCOM*, pp. 1341–1349, IEEE, Orlando, FL, USA, March, 2012.

[10] H. Shi, J. Li, J. Mao, and K. Hwang, "Lateral transfer learning for multiagent reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 10, pp. 1–13, 2021.

[11] H. Shi, H. Wu, C. Xu, J. Zhu, M. Hwang, and K. S. Hwang, "Adaptive image-based visual servoing using reinforcement learning with fuzzy state coding," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 12, pp. 3244–3255, 2020.

[12] J. Li, H. Shi, and K.-S. Hwang, "An explainable ensemble feedforward method with Gaussian convolutional filter," *Knowledge-Based Systems*, vol. 225, Article ID 107103, 2021.

[13] H. Abelson, K. Ledeen, and C. Lewis, "Just deliver the packets," in *Essays on Deep Packet Inspection, Ottawa" Office of the Privacy Commissioner of Canada*, pp. 1–8, 2010, http://dpi.priv.gc.ca/index.php/essays/just-deliver-the-packets/.

[14] China TeleCom Webpage, "China TeleCom Webpage," 2018, http://www.chinatelecom-h.com/en/global/home.php/.

[15] H. Yao, G. Ranjan, A. Tongaonkar, Y. Liao, and Z. M. Samples, "Self adaptive mining of persistent lexical snippets for classifying mobile application traffic," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pp. 439–451, ACM, Paris, France, September 2015.

[16] X. Chen, Y. Wang, J. He, S. Pan, Y. Li, and P. Zhang, "Cap: context-aware app usage prediction with heterogeneous graph embedding," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, p. 89, 2019.

[17] Songjiang University Town, "Songjiang University Town," 2019, https://exploreshanghai.com/metro/pedia/station/songjiang-university-town/.

[18] Gubei Residential Community, "Gubei Residential Community," 2019, https://en.wikipedia.org/wiki/Gubei.

[19] F. Corpet, "Multiple sequence alignment with hierarchical clustering," *Nucleic Acids Research*, vol. 16, no. 22, Article ID 10881, 1988.

[20] F. N. Kong, "Analytic expressions of two discrete Hermite-Gauss signals," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 1, pp. 56–60, 2008.

[21] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, 2002.

[22] B. Baidu, "Map," 2019, https://map.baidu.com/.

[23] H. Han, W.-Y. Wang, and B. H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 2005.

[24] M. D. Lieberman and W. A. Cunningham, "Type i and type ii error concerns in FMRI research: re-balancing the scale," *Social Cognitive and Affective Neuroscience*, vol. 4, no. 4, pp. 423–428, 2009.

[25] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[26] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[27] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[28] H. Wang, F. Xu, Y. Li, P. Zhang, and D. Jin, "Understanding mobile traffic patterns of large scale cellular towers in urban environment," in *Proceedings of the 2015 Internet Measurement Conference*, pp. 225–238, ACM, Tokyo, Japan, October 2015.

[29] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for svms," *Advances in Neural Information Processing Systems*, vol. 13, pp. 668–674, 2001.

[30] J. Weston and C. Watkins, "Multi-class Support Vector Machines," Technical Report, 1998.

[31] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters," *Robotics: Science and Systems*, vol. 2, pp. 65–72, 2005.

[32] J. Mingers, "An empirical comparison of pruning methods for decision tree induction," *Machine Learning*, vol. 4, no. 2, pp. 227–243, 1989.

[33] J. L. Grabmeier and L. A. Lambe, "Decision trees for binary classification variables grow equally with the Gini impurity measure and Pearson's chi-square test," *International Journal of Business Intelligence and Data Mining*, vol. 2, no. 2, pp. 213–226, 2007.

[34] J. Reades, F. Calabrese, A. Sevtsuk, and C. Ratti, "Cellular census: explorations in urban data collection," *IEEE Pervasive Computing*, vol. 6, no. 3, 2007.

[35] Y. Yuan and M. Raubal, "Extracting dynamic urban mobility patterns from mobile phone data," *Geographic Information Science*, vol. 7478, pp. 354–367, 2012.

[36] V. Soto and E. Frias-Martinez, "Robust land use characterization of urban landscapes using cell phone data," in *Proceedings of the 1st Workshop on Pervasive Urban Applications, in Conjunction with 9th Int. Conf. Pervasive Computing*, San Francisco, CA, USA, October 2011.

[37] R. Ahas, S. Silm, O. Järv, E. Saluveer, and M. Tiru, "Using mobile positioning data to model locations meaningful to users of mobile phones," *Journal of Urban Technology*, vol. 17, no. 1, pp. 3–27, 2010.

[38] S. Isaacman, R. Becker, R. Cáceres et al., "Identifying important places in people's lives from cellular network data," *Lecture Notes in Computer Science*, Springer, Berlin, Germany, pp. 133–151, 2011.

[39] E. De Jonge, M. van Pelt, and M. Roos, "Time patterns, geospatial clustering and mobility statistics based on mobile phone network data," in *Proceedings of the Paper for the Federal Committee on Statistical Methodology Research Conference*, Washington, DC, USA, January, 2012.

[40] A. Sevtsuk and C. Ratti, "Does urban mobility have a daily routine? learning from the aggregate data of mobile networks," *Journal of Urban Technology*, vol. 17, no. 1, pp. 41–60, 2010.

[41] T. Sohn, A. Varshavsky, A. LaMarca et al., "Mobility detection using everyday gsm traces," *Lecture Notes in Computer Science 2006*, Ubiquitous Computing, Orange County, CA, USA, pp. 212–224, 2006.

[42] J. M. Krisp, "Planning fire and rescue services by visualizing mobile phone density," *Journal of Urban Technology*, vol. 17, no. 1, pp. 61–69, 2010.

[43] V. Soto, V. Frias-Martinez, J. Virseda, and E. Frias-Martinez, "Prediction of Socioeconomic Levels Using Cell Phone Records," in *Proceedings of the User Modeling, adaption and personalization*, pp. 377–388, Girona, Spain, July, 2011.

[44] F. Girardin, A. Vaccari, A. Gerber, A. Biderman, and C. Ratti, "Quantifying Urban Attractiveness from the Distribution and Density of Digital Footprints," *Journal of Spatial Data Infrastructure Research*, vol. 4, pp. 175–200, 2009.

[45] C. Song, T. Koren, P. Wang, and A. Barabási, "Modeling the scaling properties of human mobility," 2010, https://arxiv.org/abs/1010.0436.

[46] F. Simini, M. González, A. Maritan, and A. Barabási, "A universal model for mobility and migration patterns," 2011, https://arxiv.org/abs/1111.0586.

[47] P. Wang, T. Hunter, A. M. Bayen, K. Schechtner, and M. C. González, "Understanding road usage patterns in urban areas," *Scientific Reports*, vol. 2, no. 1, p. 1001, 2012.

[48] L. Ferrari, M. Mamei, and M. Colonna, "People get together on special events: discovering happenings in the city via cell network analysis," in *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 223–228, IEEE, Lugano, Switzerland, March 2012.

[49] X. Lu, L. Bengtsson, and P. Holme, "Predictability of population displacement after the 2010 Haiti earthquake," *Proceedings of the National Academy of Sciences*, vol. 109, no. 29, Article ID 11576, 2012.