MobiDict – A Mobility Prediction System Leveraging Realtime Location Data Streams

Vaibhav Kulkarni*
vaibhav.kulkarni@unil.ch

Arielle Moro*
arielle.moro@unil.ch

Benoît Garbinato
benoit.garbinato@unil.ch

Distributed Object Programming Laboratory
University of Lausanne
CH-1015 Lausanne, Switzerland

ABSTRACT

Mobility prediction is becoming one of the key elements of location-based services. In the near future, it will also facilitate tasks such as resource management, logistics administration and urban planning. To predict human mobility, many techniques have been proposed. However, existing techniques are usually driven by large volumes of data to train user mobility models computed over a long duration and stored in a centralized server. This results in inherently long waiting times before the prediction model kicks in. Over this large training data, small time bounded user movements are shadowed, due to their marginality, thus impacting the granularity of predictions. Transferring highly sensitive location data to third party entities also exposes the user to several privacy risks. To address these issues, we propose MobiDict, a realtime mobility prediction system that is constantly adapting to the user mobility behaviour, by taking into account the movement periodicity and the evolution of frequently visited places. Compared to the existing training approaches, our system utilises less data to generate the evolving mobility models, which in turn lowers the computational complexity and enables implementation on handheld devices, thus preserving privacy. We test our system using mobility traces collected around lake Geneva region from 184 users and demonstrate the performance of our approach by evaluating MobiDict with six different prediction techniques. We find a satisfactory prediction accuracy as compared to the baseline results obtained with 70% of the user dataset for majority of the users.

Categories and Subject Descriptors

H.4.2 [**INFORMATION SYSTEMS APPLICATIONS**]: Spatial-temporal systems

Keywords

Realtime Mobility Prediction; Mobility Behaviour; Location based Services

[*] Co-first Authors

IWGS'16, October 31-November 03 2016, Burlingame, CA, USA
© 2016 ACM. ISBN 978-1-4503-4579-8/16/10...$15.00
DOI: http://dx.doi.org/10.1145/3003421.3003424

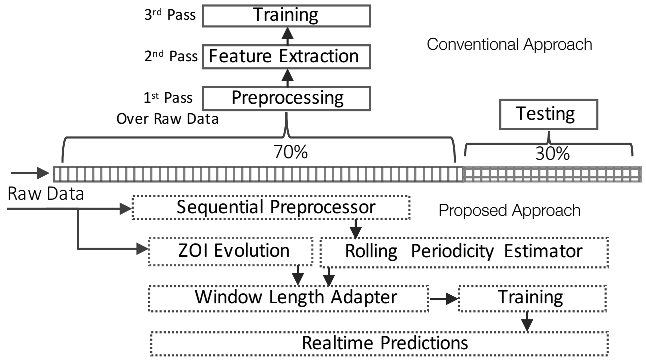

Figure 1: Traditional Prediction Systems *vs.* MobiDict. The process on the top depicts the traditional mobility prediction approach, while the process chain shown at the bottom gives an overview of our technique.

1. INTRODUCTION

In recent years, we have seen a rapid proliferation in the number of applications offering location-based services. Popular applications such as *Google Now*,[1] collect and utilise sensitive data such as, location history, agenda and contact list, to infer and assist users in everyday activities. Another well-known application, *Moves*[2] enables to automatically identify the transportation mode from collected data and display relevant information on the fly, such as the number of burnt calories. On the similar lines, *Google Maps* is equipped to predict where the user wants to go next based on the location history[3]. As evident from the above examples, mobility prediction is becoming a key paradigm of location-based services.

Problem. The services described above, demand a large volume of data in order to provide relevant mobility predictions. Existing works in this domain utilise more than 70% of the entire dataset, exclusively for the training purpose [10, 1, 21] as depicted in Figure 1 under conventional approach. The duration of the datasets, used in the literature usually lasts for more than a year, which amounts for a considerable time, explicitly for model training [19, 34]. This results in a substantial waiting time until the model is able to produce usable predictions in real deployment scenarios.

Another issue associated with learning on a large dataset is the shadowing effect on small user movements that appears insignificant, but affects the granularity of predictions.

[1] Google Now: https://www.google.com/intl/fr/landing/now/
[2] Moves: https://www.moves-app.com.
[3] Google Maps Predictions: https://www.searchenginejournal.com

Existing works attempting to address the above problem, link user behaviour with forecasting models, which on the hindsight only results in statistical prediction models without truly capturing the inherent nature associated with user movements [8].

Collecting a substantial quantity of user locations also leads to a privacy issue. A malicious entity can infer sensitive information related to the user, making it relatively easy to discover a particular place by using simple heuristiques [12] and identifying the user [6]. The algorithmic cost of making predictions on a mobile device in a real deployment scenario is relatively high due to the expensive ensemble techniques, combined with complex and extreme learning models, which makes it essential to have a centralized server [14].

Contributions. The fundamental goal of our approach is to restrict the amount of data required for training the mobility models, to small time windows usually lasting for a couple of weeks. Our solution analyzes the substantive user mobility behavioural changes in realtime and incorporates the associated changes to adapt the length of the time window required for training. We explore the evolution of the frequently visited places by the user according to the time and the associated periodicities among those places as a means to quantify user behaviour and couple it with the prediction process to give rise to quick realtime predictions as shown in Figure 1 under proposed approach . This process takes place in realtime, over sequential location data that is operational on a mobile device, thus ensuring that no personal location data is transferred to the location-based services. However, in order to utilise these services, only the predicted locations can be shared to maintain the utility/privacy tradeoff space. More specifically, the paper makes the three contributions listed hereafter.

- We propose MobiDict, a mobility prediction system on realtime sequential data in order to forecast user location. This system adapts user mobility model constantly, according to the user behavioural changes. Consequently, utilising considerably less data as compared to the conventional approaches of formulating predictive models and achieving satisfactory prediction accuracy.
- The lower computational complexity, resulting due to the lesser data involved leads to implementation on hand held devices feasible. Thus, eliminating the need to transfer highly sensitive user raw data to third party entities and ensuring user privacy. This enables to avoid the usual long and strenuous training period involved in generating the prediction models, obtaining quicker predictions.
- The reactive zone of interest computation scheme, incorporated in MobiDict, helps to model the mobility behaviour, restricted to small time periods as compared to modelling on long duration data, where the true nature of user behaviour is lost. This enables to make predictions during those small periods with higher accuracies as compared to the conventional approaches.

The rest of the paper is organised as follows. Section 3 presents our system model and introduces some formal definitions and notations used in the paper. Section 4 describes our approaches of quantifying user behaviour. In Section 5, we present the different prediction techniques used in the MobiDict system. Section 6 then presents MobiDict as a whole, showing how the elements presented in the two previous sections fit together to make up the complete system. We discuss the results of a thorough experimental evaluation of our approach in Section 7, based on mobility traces collected on our campus by Nokia Research, from 184 users, between October 2009 and March 2011. Finally, Section 2 discusses research efforts similar to ours and Section 8 concludes the paper by sketching future research directions around MobiDict.

2. RELATED WORK

We breakdown the literature review in areas concerning mobility modelling, and mobility prediction.

A domain of works apply sequential mining to extract frequently visited regions with mean travel time, to formulate the mobility models [3, 26]. The above approaches rely on clustering visits to form a visit region. We reviewed several location based clustering works [5, 18, 12, 33, 9]. As opposed to their approach of analysing the entire dataset to cluster the individual regions, we form the models by obtaining the zones in realtime, and characterising the mobility behaviour, dependent on the evolution of the number of zones with time. There exists several studies regarding stream data clustering including real-time analysis (see [4, 16]) but they fail to realistically monitor the evolution of the zones according to time. Other domain of this work falls under modelling the movements as a whole, such as the continuous-time random-walk (CTRW) [25], Levy-flight nature [29] and daily activity analysis and dissimilarities within them as presented in[17]. Although the above models aid in predicting human mobility, the mobility models are derived by offline analysis are computationally expensive to be applied to raw location data thus not feasible for making swift online predictions.

Detecting periodicity in time series data is a widely studied problem to predict trends in the data stream. [28] computes the periodicity by analysing the user's visit frequency of places and aggregating total time spent at those places followed by applying Fourier transform to this series. This knowledge is used to predict the users next visit as shown in [2, 24]. However, the analysis are based on the computations on the complete dataset, as opposed to our work in real time location log preprocessing and retrieving non-stationary and non-frequent periodic patterns lasting only for small time intervals. We did not find existing literature to formulate prediction models in realtime based on periodicities, whose instances may be shifted or distorted.

We focus the literature review regarding prediction techniques that first formulate a mobility model and consequently use it to make predictions. More specifically, prediction tasks that address the task of forecasting the next user move based on the users current location. The results of the works based on this technique [15, 8, 21, 8] show that it is possible to attain accuracies in the range of 60-80%. Several approaches have been used to make the predictions, ranging from Markov based predictors, neural networks, dynamic bayesian schemes, decision trees having several tradeoffs as compared to each other for next place prediction as summarised in [27, 22]. The learning based predictors fall under the category of predictive modelling, association analysis and cluster analysis. The next place predictions derived using the above approaches by having a trained model mapped to 70% of the dataset are presented in the works of [1, 31, 30]. [7] discusses several approaches for learning over sequential data including sliding window methods, conditional random fields and graph transformer networks. Further, Kalman filter based prediction approaches cannot be applied to non-stationary data, involves higher complexity and thus results in higher latency as discussed in [20]. Our approach falls under learning over streaming location data using a recurrent sliding window technique where we adapt the window length for training depending on the the mobility behaviour.

3. SYSTEM MODEL

Hereafter, we introduce our system model, together with formal definitions and notations used in the paper.

User and Locations. We assume a moving user carrying a mobile device whose locations are tracked by Global Positioning System (GPS) and/or Wi-Fi positioning system (WPS). The device regularly receives user's raw location logs as a sequence $L = \langle loc_1, loc_2, \cdots, loc_n \rangle$, where $loc_i = (\phi, \lambda, t)$ is a 3-item tuple representing a location in the format (latitude,longitude,timestamp). Rest of the paper uses the notation $loc.\phi$, $loc.\lambda$ and $loc.t$ for the tuple elements.

Zone of Interest. Everyday activities of a user might consist of some location points that she might find useful or spend considerable amount of time. A Zone of Interest (ZOI) is a similar concept that depicts a region encapsulating several of these points. A ZOI is not strongly bounded to any location due to the temporal constraints. It begins when the human activity at a location is initiated and ends when the activity decays. At which stage the ZOI is tied up to the relocated location.

User Mobility Model. Collecting real-life mobility data of users, which is complex and chaotic, yields mobility traces of individuals. Statistical analysis of these trajectories unfolds hidden patterns to turn this raw data into mobility knowledge. This results in abstracting away from the cluttered data and discover general movement patterns respective to individuals. Simply put, mobility models are these generalisations of movement patterns representing a user.

4. MOBILITY BEHAVIORS

4.1 Zone of Interest Evolution

This section highlights the complete process of discovering the ZOIs and their evolution which forms an integral part of the user mobility behaviour.

4.1.1 *ZOI Discovery*

The discovery of ZOIs can be divided into three distinct steps. We read the user dataset sequentially so as to simulate the realtime streaming of user locations.

Cluster Discovery. A cluster intuitively contains, locations having common spatial and temporal characteristics. $\Delta d_{max} \in \mathbb{R}$ and $\Delta t_{min} \in \mathbb{N}$ represents a distance in meters and a minimum time threshold respectively.

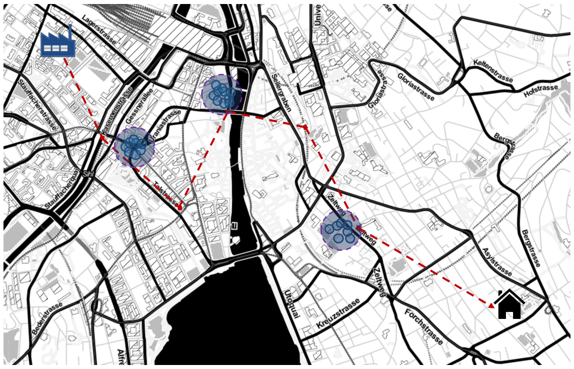

Figure 2: ZOI Construction from Cluster of Location Points.

The two following functions are considered: $centroid(\langle loc_1, loc_2, \ldots, loc_n \rangle)$ computing and returning the centroid, which maps the individual locations into the geometrical centroid based on the set distance, and $distance(loc_i, loc_j)$, which computes and returns the Euclidian distance between the two locations loc_i and loc_j.

A subset $l \subseteq L$ becomes a cluster iff the following conditions in Equations 1, 2 and 3 are satisfied[4]:

$$\forall loc_i, loc_{i+1} \in l : \\ distance(centroid(loc_1, \cdots, loc_i), loc_{i+1}) \leq \Delta d_{max} \quad (1)$$

$$loc_n.t - loc_1.t \geq \Delta t_{min} \quad (2)$$

$$\nexists l' \neq l : l \subset l' \quad (3)$$

A cluster is a 4-item tuple $c = (\phi, \lambda, \Delta r, l)$, where $\phi \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $\Delta r \in \mathbb{R}$ and l are a latitude, a longitude, a radius in meters and a subset of locations respectively. Here, $\Delta d_{max} > 0$ and $\Delta r > 0$. The mean of all ϕ and λ of the locations contained in the subset l is the centroid (ϕ, λ) of the cluster, which is designated as *c.centroid*. We consider the set C, containing all user's clusters, where $C = \{c_1, c_2, \ldots\}$. Equations 1, 2 and 3 do not guarantee disjointness of clusters which is in turn used to form cluster groups as explained further.

Cluster Group. A cluster group includes all the clusters that can be assembled iff an intersection exists between these clusters. Thus, two clusters $c_i, c_j \in C$ are included in the same cluster group g iff the next condition in Equation 4 is met:

$$distance(c_i.centroid, c_j.centroid) \\ - \ (c_i.\Delta r + c_j.\Delta r) \ < \ 0 \quad (4)$$

A cluster group is a 4-item tuple $g = (\phi, \lambda, \Delta r, \{c_1, c_2, ...\})$, where $\phi \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $\Delta r \in \mathbb{R}$, $\{c_1, c_2, \ldots\} \in C$ are latitude, longitude, radius and array of clusters constituting g respectively. The centroid of the cluster group is defined by (ϕ, λ), being the mean of all the centroids of the clusters included in g. The following set G contains all the discovered cluster groups, such as $G = \{g_1, g_2, \ldots\}$.

ZOI. A ZOI is a frequently and recently visited zone by a user in everyday life. The two constants $visitThreshold \in \mathbb{N}$ and $maxTimeDuration \in \mathbb{N}$ represent a maximum threshold of visits and a maximum duration threshold between two

[4] This clustering process is inspired by a technique called *DT cluster* and presented in [12].

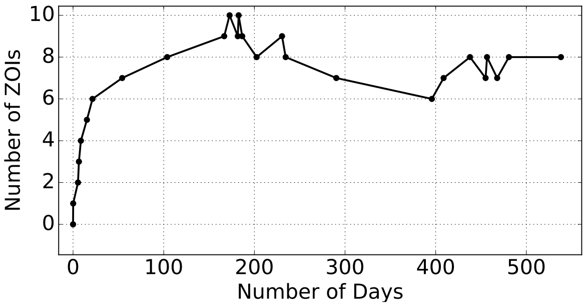

Figure 3: ZOI Evolution Over Time.

dates respectively, while $minVNB \in \mathbb{N}$ is a variable representing the minimum number of visits. Then, $size(g)$ is a function which computes and returns the number of clusters of the cluster group g, $meanVNB(G)$ is a function computing and returning the mean number of visits amongst the set of all cluster groups G and $timeDuration(G)$ is a function which returns the duration between the current date and the last visited date of the cluster group contained in G. $meanVNB(G)$ returns values which dynamically change over time according to the mobility behaviour of the user due to the realtime nature of the process. $minVNB$ is equal to the value returned by $meanVNB(G)$ until reaching the $visitThreshold$, which is the maximum number of visits that converts a cluster group into a ZOI. A cluster group $g \in G$ is transformed into a ZOI z iff the conditions in Equation 5 and 6 are satisfied:

$$\begin{aligned} & size(g) \geq minVNB \\ \vee\ \ & minVNB = meanVNB(G) \\ \wedge\ \ & minVNB <= visitThreshold \qquad (5) \\ timeDuration(G) & <= maxTimeDuration \qquad (6) \end{aligned}$$

A ZOI z is formally, a four item tuple $z = (\phi, \lambda, \Delta r, g)$, where $\phi \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $\Delta r \in \mathbb{R}$ and g are the latitude, longitude, radius and the cluster group becoming a ZOI respectively. The tuple (ϕ, λ) is the centroid of z computed from group g. The set Z is finally the set of ZOIs of the user, such that $Z = \{z_1, z_2, \cdots, z_n\}$ as shown in Figure 2.

4.1.2 ZOI Evolution

A user's ZOIs may change over time and space. Figure 3 shows an example of ZOI updates occurring over time for a certain user having location data of more than 500 days. We see a surge of ZOI updates at the beginning, minor variations intermediary and attains a flat tail towards the end. Monitoring this trend of ZOI evolution according to time reflects the changing user behaviours. Thus the number of ZOIS and their evolution can be used to quantify user mobility behaviour.

4.2 Periodicity of Movement

Human mobility is characterised by a high degree of periodicity, contrary to the popular assumption that the mobility patterns are highly stochastic. Detecting these periodic behaviours can assist to generate quick predictions, evading the complex training procedure. However, one of the challenges is to identify periods which do not repeat precisely at the same times, in addition to having multiple interlaced patterns in the non-stationary time series. As a result, standard period estimation techniques such as autocorrelation or Fourier transform cannot be directly applied. We describe the steps involved to accurately detect the movement periodicity.

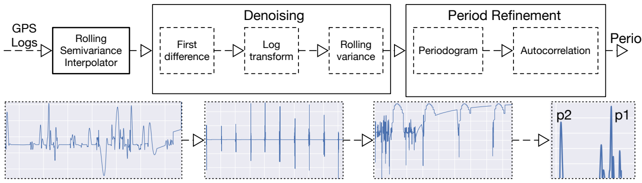

Figure 4: Realtime Periodicity Estimation Chain.

Uniform Location Sampling. One of the fundamental drawbacks of the periodicity detection algorithms is the prerequisite that the incoming location stream should be uniformly sampled. However, location logs coming in at non uniform rate is common in communications due to imperfect geolocation sensors or network unavailability to stream logs online. When the sampling is nonuniform, a common technique is to resample/interpolate the signal onto a uniform grid. We use semivariance interpolation on the incoming stream using moving average construction.

In a nutshell, the semivariance conceals the incoming data stream about the spatial variance at a specified distance. We find that Gaussian model provides accurate fitting to the missing data after calculating the semivariance. The semivariance along with Gaussian model allows to model the similarity between points in a filed as a function of changing distance. The semivariance can be mathematically expressed in Equation 7 as:

$$\delta_h = \frac{1}{2N_h} \sum_{N_h} (R. \sqrt{(\delta_x . \cos_\theta)^2) + \delta_y^2})^2 \qquad (7)$$

where δ_h is maximum distance separation among the location logs, N_h are the number of points separated by the distance h. The semivariance is than the sum of the squared difference between these values. To calculate the distance, we utilise, equirectangular distance approximation, which is faster as compared to the Harversine formula. In addition, as the distances traversed are usually small, the performance is superior compared to great circle distance approximation.

Dealing with Non Stationary data. Applying signal processing techniques, directly to estimate the user movements and periodicity to non-stationary data puts forth several challenges. The interpolation step is followed by taking the first difference of the streaming interpolated location logs. This step brings forth the trends present in the movement data by exposing the variance for further processing. Next, in order to estimate the magnitude of day-to-day variations, log transform is applied to the series. The rolling variance applied to the logged series, results in a series of constant variance.

Periodicity Estimation. For the final step of the periodicity detection, we compute the rolling autocorrelation over the precessed stream. Next, we calculate the power spectral density to get the candidate periods and feed them into the autocorrelation estimator so as to rectify false alarms resulting due to the spectral leakage. The robust autocorrelation routine, results in the computation of statistically sig-

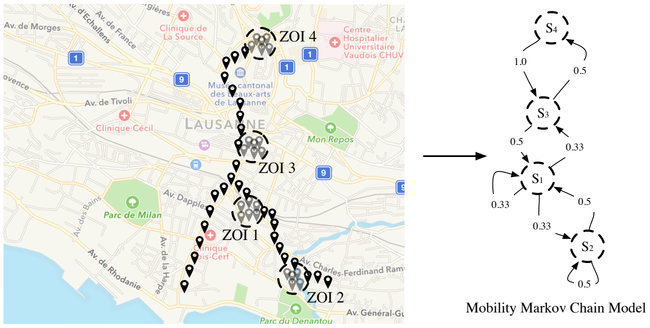

Figure 5: From User's ZOIs to MMC Model.

nificant period(s) contained in the non-stationary and noisy location stream. The complete processing chain is shown in Figure 4. Through this process, we are able to detect weekly periodic patterns. We focus on estimating short repetitive patterns and detecting larger periods such as holiday vs. non holiday pattern is beyond the scope due to the data limitations we impose.

The above two demeanours, i.e., evolution of ZOIs and movement periodicities, serve as a basis to decipher user mobility behaviours, which play a key role to alter the real-time model formulation and assist in prediction.

5. MOBILITY PREDICTORS

In this section, we describe the different families of prediction techniques used to forecast the next user location. We specify the procedure involved in training these predictors in the context of mobility forecasting. In the plethora of predictor options available, we select the ones that have been already proved successful for spatiotemporal prediction by published works [11, 21]. The starting time at a ZOI and the total spent duration serve as input features to the prediction techniques. The prediction task is then modelled to perform one-step-ahead forecasting on the basis of available data gathered in a time window.

Mobility Markov Chain. A Mobility Markov Chain (MMC) model is described by a state-transition matrix including the user's ZOIs, which are the states, and all the transitions amongst them. These transitions, collected during a training period, feed the model. We assume a set S of n states discovered at a current time t such that $S = \{s_1, s_2, \ldots, s_n\}$ knowing that $S = Z$. This set of states is the baseline to build the matrix. Then, by exploring the user's raw locations, we can extract a sequence of vectors V, where one vector contains two successive states, visited by the user such that $V = \langle (s_1, s_2), (s_2, s_2) \ldots, (s_1, s_n) \rangle$. The matrix is then filled according to the sequence V by computing the transition probability of each vector included in V. The transition probability to move from s_i to s_j is expressed in Equation 8 as follows:

$$p_{s_i,s_j} = P(s_j | s_i) \tag{8}$$

Then, the predicted next state is the most likely state s_{next} found on the basis of all computed transition probabilities of the matrix line of the current state s_c as expressed in Equation 9:

$$\forall s_i \in S : pred_{s_{next}} = max(p_{s_i,s_c}, p_{s_{i+1},s_c}, \ldots, p_{s_n,s_c}) \tag{9}$$

Figure 5 shows the creation of a MMC model of a user as well as the state-transition matrix based on the evolution of the ZOIs of a user. Unlike the 1-order Markov chain, which is described above, the 2-order Mobility Markov Chain is slightly different, as the previous state is also taken into account in the prediction process, which increases the prediction accuracy as presented in [13].

Classification Based Learning. Predicting the next location can be viewed as a classification task, where the training and the discrete output class consist of the currently computed and active ZOIs according to the user behaviour. This set consists of permanent zones and temporary zones which may vanish with time. Thus, every learnt model bounded by a particular time instant may consists of ZOIs that may not be active in prediction models formulated at another time instant. We employ an elementary 1-NN based classifier that uses a training point closest to the query point to predict the output label. If $X_i = \{\{x_p\}, \{x_t\}\}$ is the input vector consisting of permanent and temporary zones $\{x_p\}$ and $\{x_t\}$ respectively, the training set consists of $\{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$, where y_i is the next zone, traversed according to the time series sequence. Thus, the task here is to determine y_{new} for x_{new} that is performed by finding the closest point x_j to x_{new}, w.r.t. the euclidean distance.

Artificial Neural Network. In the ANN model, the training patterns are highly dependent on the training window length to model the forecasting as predictive regression problem. Each pattern will consist of the number of ZOI movements recorded in the training window. $pattern1 = x_1, x_2, \ldots x_n$, $pattern2 = x_2, x_3, \ldots x_{n+1}$ correspond to the number of input nodes, where each node represents a pattern collected for a day, thus training the model using the sliding time window approach. The model consists of one output node and the number of nodes in the hidden layer is determined empirically. The 3 layered feed-forward neural network with back propagation can be represented in Equation 10 as below:

$$y_i = f(g((\sum_j w_{ij}.in_j) - \theta_j)) \tag{10}$$

where w_{ij} is the weight vector, θ is the bias, in_j is the input layer representing the movement patterns for a day.

Recurrent Neural Networks. Recurrent neural networks have a memory layer that is advantageous to model long term time series data, by assisting the inputs to be correlated with input/output pairs, which even lie beyond the current window length. Similar to the previous description, we use a three layered architecture where the hidden layer is recurrently connected to itself. The network can be represented in Equation 11 as below:

$$y^i = w_2.\sigma(w_1.x^i + w_r h^{i-1}) \tag{11}$$

where σ is a non-linear transfer function, here, we use sigmoid function. w_1, w_2 are the connecting weights and w_r are the recurrent weights.

Fourier Extrapolation. We also test MobiDict with Fourier extrapolation, which is capable of deconstructing the time series as a polynomial base, with bounded randomness

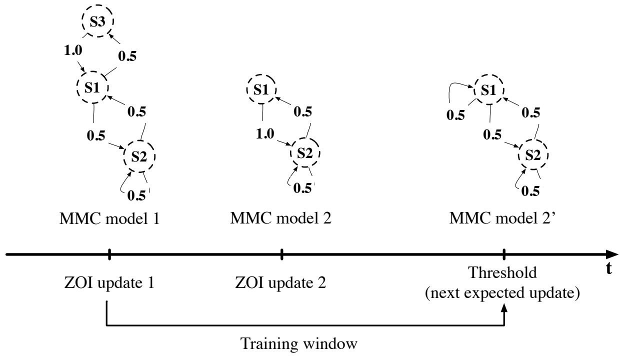

Figure 6: MMC Training Window.

and a cyclic component. The frequency domain, due to its nature, transforms the time bounded user visits in time domain into fixed cycles. The extrapolation yields a de-noised copy of the movements observed in the history. Thus performing prediction over a time window of t time units. Here, the high frequency components are used to estimate movements over regular ZOIs and the low frequency components, to predict irregular user movements restricted to small time bounds.

We implement the above machine learning prediction techniques using PyBrain [32] and the MMC models are generated within the Cocoa application where the entire realtime process is implemented.

6. THE MOBIDICT SYSTEM

In this section, we present the MobiDict prediction system design and illustrate how the mobility behaviours are coupled with the predictors to produce the next location prediction in realtime. The overview of our approach is presented in Figure 1, which depicts the fundamental elements involved in our system. As described in Section 4, we present two approaches to quantify user behaviour. Due to the nature of MMC, movement periodicity cannot be directly integrated into the model, thus we base MMC only on the ZOI evolution aspect. However, in case of the machine learning techniques, we involve the periodicity associated, with the movement within the evolving ZOIs. We perform a systematic evaluation of MobiDict by testing it with all the described prediction approaches to analyse the prediction accuracies. We now describe how the MMC and the machine learning based system individually integrate the respective mobility behaviours to produce next place predictions. The common goal being, formulation of a robust predictive system on streaming location data.

6.1 MMC-based System

A Mobility Markov Chain model only depends on the states and the transition probabilities amongst them. This property bears similarity with the evolving ZOIs of the user over time representing the behaviour. Therefore, to implement MobiDict, we combine the evolution of ZOIs with the creation of the user's MMC model. Figure 5 presents an intuitive description of what is a ZOI significant update, which basically triggers the adaptation of the user model every time there is a new significant update. This preserves the freshness of the user mobility model.

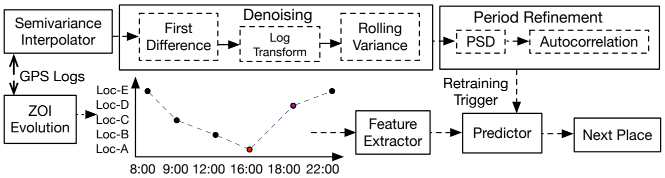

Figure 7: Realtime Periodicity Aware Training Process.

Figure 6 shows the successive training windows, the ZOI updates and the updates of the mobility model. The training window is initiated when there is a significant ZOI update and ends when a certain threshold is exceeded. An update is considered as significant when either a new ZOI is added to the ZOI set or removed from it under the assumption that this set contains more than one ZOI. At each update, the user's MMC model is rebuilt according to the entire state sequence S that is updated in realtime by taking into account user's raw locations. As seen in Figure 3, many updates are sometimes accumulated, in such cases, the mean time between two updates is used to compute the threshold of the training window. The next expected update is triggered by adding the mean time between all the past updates u_{mean}. The next threshold t_{next} can be formally expressed in Equation 12 as below:

$$t_{next} = date_c + u_{mean} + \frac{u_{mean}}{2} \tag{12}$$

where $date_c$ is the current date of the system. If this threshold is exceeded without having detected the expected significant update, the training window is interrupted. At the end of the training window, the MMC model is also updated in order to take into account the entire state sequence collected during the window as well as the one formulated during previous training windows.

6.2 Machine Learning-based System

The system should take into consideration the recent movement histories and the associated periodicities in order to produce an updatable mobility model. The problem can be formulated as a non-stationary time series prediction, where the model needs to be retrained according to variations in the incoming data stream, which in our case are the user movements and the variations, link to changing periodicities. We empirically determine that the model accuracy is affected for an autocorrelation index change of 0.2 and greater. This severs as a trigger for periodic and incremental model retraining, where the batch size consists of movement histories, with the changed periodicity bounds.

We first describe the realtime processing chain, as shown in Figure 4 to estimate the changing periodicities. As described in Section 4, we perform Fourier analysis that expresses the function, as summation of individual periodic elements. Further, we compute the power spectral density to find the strength at each frequency, and only the dominant frequency components are selected. The periodogram highlights the periodicities lasting for short and medium terms, on the other hand, autocorrelation is suitable for large period detection. We combine the approaches so as to filter out harmonics and get refined candidate periods. This can be formally expressed in Equation 13 as below:

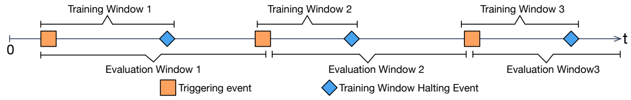

Figure 8: Realtime Evaluation Scheme.

$$P(\frac{C_k}{N}) = \left|\left|\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} x(n).e^{\frac{-j.2\pi.c.n}{N}}\right|\right|^2, c = 0, 1...\frac{N-1}{2} \tag{13}$$

where, C_k are the strength encodings at a given frequency k, $x(n)$ are the spectral coefficients associated with the sinusoids.

We track the periodicity continually, following the above approach. Regarding the training phase, as depicted in Figure 7, the ZOI evolution is tracked to form a feature vector representing the movements across them. The other features consist of the starting time and stay time at a particular zone. The extracted feature vectors are fed to the predictors described in Section 5. The periodicity feature is tracked to monitor if it changes by 0.2. At this point, the training reinitiates to reform the mobility model, taking the new periodicities, thereby adapting to current behaviour of the user.

7. EXPERIMENTAL EVALUATION

In this section, we demonstrate the experimental results of our approach based on the Nokia data set [19] consisting of mobility traces, collected from 184 users in Switzerland from October 2009 to March 2011. The participants consisted of university students and professionals with a mean duration of 14 months comprising of more than 10 million location points. Amongst the users of this dataset, we only select 168 of them having a dataset duration of at least 30 days.

7.1 Experimental settings

Here, we describe the selection of users from the dataset, as well as the choices made, behind the algorithmic parameters. In order to obtain the ZOIs of a user, we set a value of 60 meters for Δd_{max}, 900 seconds for Δt_{min} to cluster the individual points of interest with respect to space and time. The *visitThreshold* parameter is set to 6 visits and the *maxTimeDuration* of three months. In order to determine the above parameters, we analyse the complete dataset to compute the average of the mean number of visits of all cluster groups of each user per month. This choice was based on selecting users having a dataset duration of at least 30 days. In order to simulate realtime incoming data, we read the data-points sequentially according to the logged timestamps.

7.2 Real-time Evaluation Scheme

Figure 8 describes the evaluation approach, followed to compute the prediction accuracy over time for each user. This scheme shows the successive training windows, as well as the consecutive evaluation windows. In the case of the 1-order and 2-order MMC, this trigger is a significant update about the set of the user's ZOIs, while in the case of learning based approaches, we rely on a significant change in the autocorrelation index, representing the user periodicity. It is important to note that all the information collected during the previous training windows is also taken into account for the next training windows. The evaluation window commences at the very first trigger, which is when the first two ZOIs of a user are computed. During a training window, the model analyses the user's movements to construct a user specific mobility model according to the techniques described in Section 5. The MobiDict system is evaluated with respect to each family of predictors. At the beginning of the every new training window, the prediction accuracy result is computed. As the evaluation metric, we consider the prediction accuracy, which is the fraction of samples for which the model successfully predicts the next location during the evaluation window.

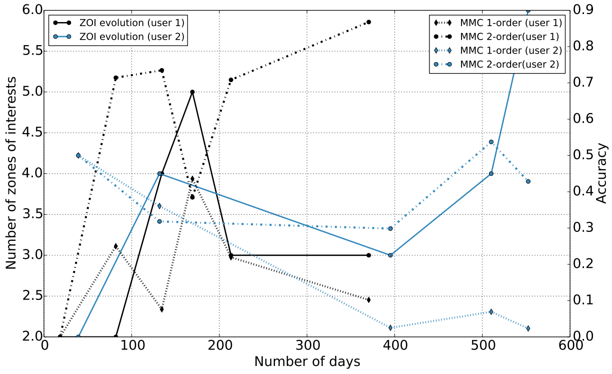

Figure 9: Evolution of ZOIs and Prediction Accuracy Over Time of 2 Users According to 1-order and 2-order MMC.

7.3 Results and Discussion

We evaluate the performance of MobiDict by comparing it against the accuracy obtained by using the conventional approach of formulating a model, trained on 70% of the dataset and evaluated on the rest. The resulting accuracy that we use for baseline comparison for all the predictor families is shown in Table 1. We also compare our baseline results with the results obtained by existing works on the same dataset and achieve similar accuracies with the same feature selection techniques.

Figure 9 depicts the evolution of the user's ZOIs and the prediction accuracy computed with the 1-order and 2-order MMC prediction technique over time. We consider two different users, contained in the Nokia dataset having different dataset durations, i.e., more than 350 days for the first user and more than 500 days for the second user.

We obtained higher prediction accuracies with 2-order MMC as compared to 1-order MMC for majority of the users as also depicted in case of these two users. This is mainly due to the fact that, 2-order MMC takes into account the current user's state and the previous state to search the next state in the model, improving the quality of the predictions. We also observe that, when the number of ZOIs has a sudden shift, the accuracy does not necessarily decrease with this drastic variation. For instance, at the end of the evolution of the number of ZOIs of user 2 (i.e., from point 4 to point 5), there is an increase of two zones, however, the prediction accuracy is unaffected for both the MMC. With the decrease of two zones (i.e., from point 3 to point 4 for user 1), the accuracy of 2-order MMC increases, while that of the 1-order decreases. Here, we assume that some variations may sometimes require longer training periods to obtain relevant

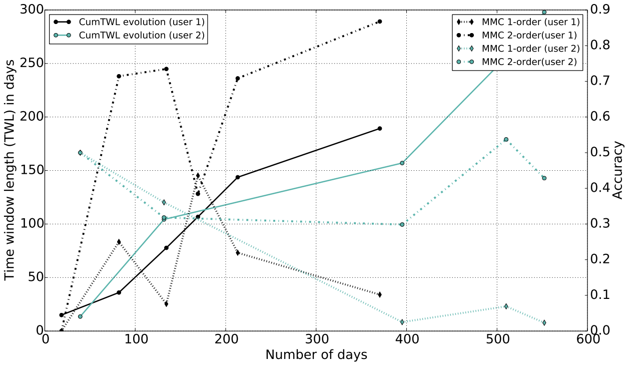

Figure 10: Evolution of Cumulative Time Window Length and Prediction Accuracy Over Time of 2 users According to 1-order and 2-order MMC.

MMC model according to the changes, as the predictions are strongly linked to the transition probabilities contained within them.

Technique	Accuracy (%)
1-order MMC	57,19
2-order MMC	61,66
1-NN	59,28
ANN	60,85
RNN	72,79
Fourier ext.	63,87

Table 1: Baseline Results.

In Figure 10, cumulative training window lengths are depicted according to the accuracy and the evolution of user movements according to time. We see, a clear trend in the number of days taken to compute the predictions at a specific time. With the considered two users, we observe that, there is no absolute requirement to use a large amount of data to obtain satisfactory prediction accuracies with the MMC prediction technique, because, with less than 100 days, we can obtain accuracy of more than 0.5. Regarding the entire dataset analyses, 34% of users reach a satisfactory accuracy with less than 100 days. We also assume that this is closely linked to the quality of the information, i.e., transitions between 2 ore more states, included into the model during the training windows. In addition, it is also important to note that, in realtime and for the MMC techniques, we use raw data without any refinement, which could affect the quality of the user's mobility model.

Next, we analyse the effect of movement periodicities, on the accuracies of the learning based predictor families. As shown in Figure 11, we see a clear correlation between the periodicity and the accuracy of classification, neural networks and the Fourier based approach. However, we also see that, recurrent neural network has no visible impact of user periodicities except for the minor variations. We observe this trend for majority of the users across the dataset. The main reason being, RNN's blend the input vector at the current state (i.e. the movement histories) with the previously learnt state vector to yield a new state. Thereby, taking the entire history into account before making a prediction, effectively combining, high level direction with low level modelling that results in high accuracy, maintained almost stable with time. On the other hand, the classification and neural network based approach weighs the current state higher than the past depicting very high correlation with periodicity. As, with respect to Fourier extrapolation, since the individual frequency components are contribute to forecasting, the higher the periodicity the better is the accuracy.

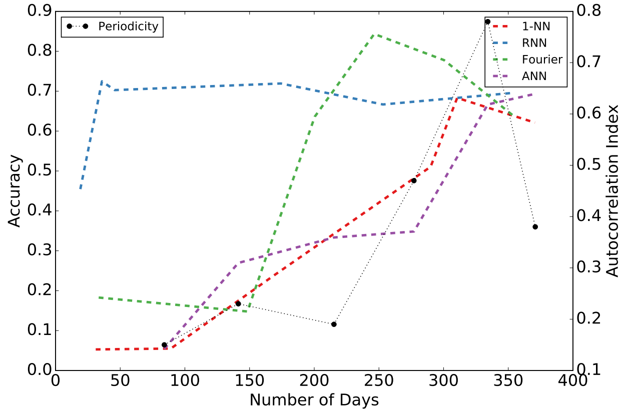

Figure 11: Variation of accuracy with time and the movement periodicity.

	Percentage of days			
Prediction technique	<=20%	>20% & <=60%	>60%	Number of satisfactory users
1-NN	168	0	0	101
ANN	103	65	0	129
RNN	37	131	0	149
Fourier ext.	65	103	0	112
1-order MMC	137	17	14	93
2-order MMC	62	41	65	142

Table 2: Dataset Analysis.

Next, we evaluate the running accuracy difference between MobiDict for all the predictors against the baseline accuracy at each training model update as shown in Figure 12. As we see, the accuracies are in general lower than the baseline accuracies however, in most of the cases represent satisfactory accuracy level above 50%. After update 3, the accuracies of 2-MMC, ANN, RNN and Fourier based predictors are often higher as compared to the baselines. Regarding 1-order MMC technique, although the baseline result is not very high compared to the other techniques (i.e., 57.19%), the prediction accuracy results of the 1-order MMC model are mainly far from it. In addition, we note that the 2-order MMC accuracy results are fairly satisfactory remaining higher than the baselines for most of the time. The high baseline accuracies may also result due to the overfitting of the model when looking directly at the 70% of the dataset as compared to realtime training. Realtime training and prediction, involves higher stochasticity in the time bounded noisy data that is not the case when formulating a prediction model over the complete dataset. We further evaluate the total number of users in the entire dataset providing accuracy levels higher than 50% as summarised in Table 2. We indicate the number of users having prediction accuracy greater than 50% in terms of total percentage of days. We observe that RNN yields the maximum number of satisfactory users whose accuracy is greater than 50% (and lower than 60%), making it an ideal predictor to be integrated in MobiDict.

Further, we approach the problem concerning the com-

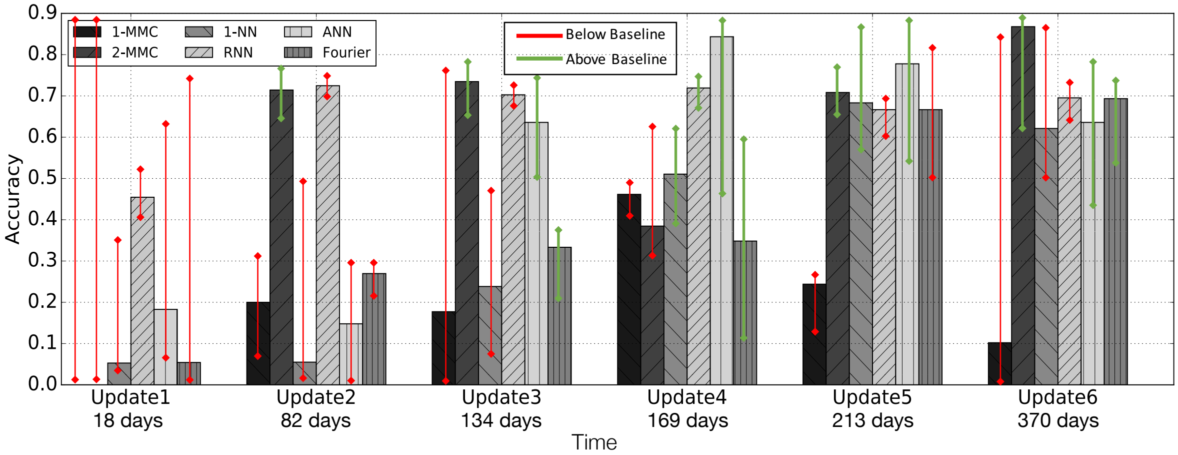

Figure 12: Comparison of MobiDict accuracy for individual predictors against the baseline accuracies.

putational complexity of learning approaches by analysing the cost involved at the training time. This complexity is directly linked to a quadratic equation that involves inverting a kernel matrix having a complexity of order n^3, where n is the size of the training data [23]. The training time to arrive at an optimal solution depends on the technique used, but generally has the order of n^2. Thus, the baseline complexity is $0.7 * D$ where D is the total number of data-points collected. Therefore, the complexity in our case is $N_{up} * n^3$ where, N_{up} is the total number of updates. Further, n in our system represents the data-points included in the individual training windows, i.e. $n = t_{n1} + t_{n2} \ldots + t_{nN}$, since we account for the user behaviour, which is constant for some time periods the total number of data-points will be lower than D, thereby having a lower complexity. The same goes for the training time.

8. CONCLUSION

With the growing ubiquity of location-aware mobile devices, the ability to analyse and predict mobility on a large scale is becoming possible, opening new opportunities but also posing new challenges. Furthermore, with mobile devices becoming more powerful every day, it becomes possible to compute mobility predictions locally, i.e., without resorting to backend servers. Yet traditional approaches to mobility prediction rely on processing large datasets on powerful backend servers. This makes mobility prediction quite tedious and slow. In addition, such centralized approaches come with a major location privacy concern, threatening the success of widespread adoption of LBS in the coming days. This enforces a real need to restrict computations involving sensitive user data on a local mobile device.

To address these issues, we introduce MobiDict, a real-time mobility prediction system, to provide swift next place predictions. Our approach couples the prediction system with dynamic user mobility behaviours to restrict the data required for model training to short durations as opposed to conventional training approaches. This achieves accuracies exceeding 50% for about 40% of the users contained in the dataset for 2-MMC and RRN predictors. We also examine periods where our system accuracy, even exceeds the baselines. Thus exhibiting that large amount of training data is not an absolute requirement to produce viable next place predictions. We also evaluate the computational cost associated with our approach and theoretically validate the feasibility to operate on a mobile device.

We observe that certain family of predictors are more suited for particular mobility behaviours. Our future work will be an attempt to have an ensemble approach in the system to select a suitable predictor in realtime according to behavioural changes, to attain higher accuracies. We will also focus to quantify the computational cost of the approach on an actual mobile device to confirm our hypothesis. Another area will be to optimise the process so as to have fewer number of model updates that will intern contribute to the cost.

Acknowledgment. This work is partially supported by the Swiss National Science Foundation grant 146714 and 157160.

9. REFERENCES

[1] T. Anagnostopoulos, C. Anagnostopoulos, and S. Hadjiefthymiades. Mobility prediction based on machine learning. In *IEEE 12th International Conference on Mobile Data Management*, pages 27–30, June 2011.

[2] H. Cao, N. Mamoulis, and D. W. Cheung. Discovery of periodic patterns in spatiotemporal sequences. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):453–467, April 2007.

[3] X. Chen, J. Pang, and R. Xue. Constructing and comparing user mobility profiles for location-based services. SAC, pages 261–266, 2013.

[4] Y. Chen and L. Tu. Density-based clustering for real-time stream data. KDD '07, pages 133–142, 2007.

[5] A. Daniel and S. Thad. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, pages 275–286, 2003.

[6] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the Crowd: The privacy bounds of human mobility. *Scientific Reports*, 3, Mar. 2013.

[7] T. G. Dietterich. Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 15–30. Springer-Verlag, 2002.

[8] T. M. T. Do and D. Gatica-Perez. Contextual conditional models for smartphone-based human mobility prediction. UbiComp '12, pages 163–172, 2012.

[9] E. Eftelioglu, X. Tang, and S. Shekhar. Geographically robust hotspot detection: A summary of results. In *ICDMW*, pages 1447–1456, 2015.

[10] V. Etter, M. Kafsi, and E. Kazemi. Been There, Done That: What your Mobility Traces Reveal about your Behavior. *the Procedings of Mobile Data Challenge by Nokia Workshop at the Tenth International Conference on Pervasive Computing*, June 2012.

[11] V. Etter, M. Kafsi, E. Kazemi, M. Grossglauser, and P. Thiran. Where to go from here? mobility prediction from instantaneous information. *Pervasive and Mobile Computing*, 9(6):784 – 797, 2013. Mobile Data Challenge.

[12] S. Gambs, M.-O. Killijian, and M. Núñez del Prado Cortez. Show me how you move and i will tell you who you are. *Trans. Data Privacy*, pages 103–126, 2011.

[13] S. Gambs, M.-O. Killijian, and M. Núñez del Prado Cortez. Next place prediction using mobility markov chains. MPM '12, pages 3:1–3:6. ACM, 2012.

[14] L. Ghouti. Mobility prediction using fully-complex extreme learning machines. In *ESANN*, 2014.

[15] G. Gidófalvi and F. Dong. When and where next: Individual mobility prediction. MobiGIS '12, pages 57–64, 2012.

[16] M. Hahsler and M. H. Dunham. Temporal structure learning for clustering massive data streams in real-time. In *SDM*, pages 664–675, 2011.

[17] S. Jiang, J. Ferreira, and M. C. González. Clustering daily patterns of human activities in the city. *Data Mining and Knowledge Discovery*, 25(3):478–510, 2012.

[18] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *Mobile Computing and Communications Review*, 9:58–68, 2004.

[19] J. K. Laurila, D. Gatica-Perez, I. Aad, B. J., O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*, 2012.

[20] J. J. LaViola. Double exponential smoothing: An alternative to kalman filter-based predictive tracking. EGVE '2003, pages 199–206.

[21] T. Le Hung, M. Michele, Catasta an Lucas Kelsey, and A. Karl. Next place prediction using mobile data. Mobile Data Challenge by Nokia, 2012.

[22] S. Lee and K. C. Lee. Context-prediction performance by a dynamic bayesian network: Emphasis on location prediction in ubiquitous decision support environment. *Expert Syst. Appl.*, pages 4908–4914, 2012.

[23] D. Levi and S. Ullman. Learning model complexity in an online environment. In *Computer and Robot Vision, 2009. CRV '09. Canadian Conference on*, pages 260–267, 2009.

[24] Z. Li and J. Han. *Mining Periodicity from Dynamic and Incomplete Spatiotemporal Data*, pages 41–81. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[25] M. Lin, W. J. Hsu, and Z. Q. Lee. Modeling high predictability and scaling laws of human mobility. In *IEEE 14th International Conference on Mobile Data Management*, volume 2, pages 125–130, 2013.

[26] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: A location predictor on trajectory pattern mining. KDD, 2009, pages 637–646, 2009.

[27] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer. Comparison of different methods for next location prediction, 2006. pages 909–918. Springer, 2006.

[28] B. Prabhala, J. Wang, B. Deb, T. L. Porta, and J. Han. Leveraging periodicity in human mobility for next place prediction. In *WCNC*, pages 2665–2670, 2014.

[29] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking*, 19:630–643, 2011.

[30] N. K. Saini and A. Trivedi. Refined cluster based mobility prediction with weighted algorithm. In *CICN*, pages 350–354, Nov 2010.

[31] N. Samaan and A. Karmouch. A mobility prediction architecture based on contextual knowledge and spatial conceptual maps. *IEEE Transactions on Mobile Computing*, 4(6):537–551, Nov. 2005.

[32] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber. PyBrain. *Journal of Machine Learning Research*, pages 743–746, 2010.

[33] Y. Yuan and M. Raubal. A framework for spatio-temporal clustering from mobile phone data. In *AGILE*, pages 22–26, 2012.

[34] Y. Zheng, X. Xie, and W.-Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33:32–39, 2010.