A Large-Scale Study in Predictability of Daily Activities and Places

Gordon E. Moon
Department of Computer Science and Engineering
The Ohio State University
Columbus, Ohio 43210, U.S.A
moon.310@osu.edu

Jihun Hamm
Department of Computer Science and Engineering
The Ohio State University
Columbus, Ohio 43210, U.S.A
hamm.95@osu.edu

ABSTRACT

Modeling human activity has a wide range of applications, including customized service for mobile users and resource management for mobile communications. Predicting future activities and places is a challenging task, as the activity pattern shows both regularity and randomness simultaneously. Time information such as time-of-day or day-of-week can partially explain this regularity. Furthermore, current activity and place may depend strongly on the activity and place history far back into the past. To capture this long-range temporal dependency, we used Long Short-Term Memory (LSTM) network, a state-of-the-art sequential model. Instead of predicting a future state of the immediate next step, we proposed multi-step look-ahead predictions to evaluate how far we could accurately predict future states up to 6 hours. Experiments on large-scale American Time Use Survey dataset showed that our model outperforms previous approaches, including Markov-based models, consistently achieving 0.3-7% relative improvement. We divided subjects into four groups by employment status, and we showed qualitatively that daily activity patterns are different among groups, and quantitatively that group-specific training is essential to accurately predict the future activity of individuals.

Keywords

daily activity prediction; daily place prediction; sequential model; Recurrent Neural Network; LSTM

1. INTRODUCTION

Analysis of human activity patterns has garnered a lot of attention since the advent of localization capabilities through global positioning systems. Recent developments in sensing technology and smart devices offer diverse types of mobility data accessible to a wide range of fields [47, 28, 29]. Human activity pattern can be further studied by employing mobility data from GPS trajectories [48, 27], cellular towers [13], location check-ins [4, 40], and WiFi signals [39, 43]. A deeper understanding is especially vital in order to solve anticipatory resource management problems such as monitoring public emergency, intelligent urban planning, and preventing the spread of diseases. Therefore, it is important to examine potential problems associated with predictability of human activity patterns and methods applied to prediction models.

Predicting daily human activities is challenging, because they have regularity and randomness in the pattern. Time information is important for predicting future activity and place. For example, regular workday schedules are likely to be explained by the circadian rhythm. Certain activities, however, such as working out in a gym and eating in a restaurant, can happen semi-regularly but not on the same time everyday. The variability of semi-regular patterns is affected by the past sequences of activity and place up to when a certain activity occurs. Therefore, prediction models should be able to take long ranges of past observations to make predictions.

In the last few years, a number of studies such as the standard Markov model and Hidden Markov model [26, 24, 7, 2] have proposed probabilistic models for predicting daily places. While the Markov model is popular for modeling temporal sequences, it should increase the order of model to capture long-term dependency in the sequences. In practice, the Markov model is limited to use small number of order, since parameter space exponentially increases as the number of order grows. Therefore, using the Markov model leads to difficulty in learning extremely long ranges of dependencies in activity patterns.

In this paper, we used Long Short-Term Memory (LSTM) [17], a special type of Recurrent Neural Network (RNN) to build human activity predictors and verify long-term dependencies in the activity patterns. Compared to the other temporal models, the LSTM network can contain very long ranges of temporal dependencies of data, which is known to be one of the greatest methodological advantages of this model. With its memory cells that store past information, LSTM is capable of capturing a long range of past sequences over a long period of activity patterns. Recently, LSTM has produced state-of-the-art results in many domains including automatic speech recognition [14] and handwriting recognition [15]. A previous study has employed a RNN approach to predict fine-level actions of activity using a wearable sensory dataset [34] (refer to Section 2 for more details). Using LSTM network, we proposed and evaluated multi-step many hours later of future prediction. In the performance of prediction, shorter time intervals between consecutive time steps can artificially generate arbitrary results, because activity patterns do not change frequently. To avoid this problem, we evaluated the performance of multi-step look-ahead predictions up to 6 hours, observing how far we could accurately predict future state of the subjects with no respect to the size of time intervals between two consecutive units in the sequence. Due to the randomness in human activity patterns, multi-step look-ahead prediction is a challenging task. Rather than other studies that made immediate time step predictions, the task in our study is more difficult as we addressed many hours of multi-step look-ahead predictions. A more detailed explanation of the arbitrary problem is described in Section 4.2.

There are several public mobility datasets available, but a majority of the real mobility datasets consists of pre-defined labels which are somewhat ambiguous. For example, the exact meanings of some user-defined place labels such as attractions and cultural places are difficult to recognize. Also, the number of labels is insufficient to precisely identify the semantic information of locations. Human mobility datasets used in [9, 7] are limited only to place information, and the frequency of place labels is dominant in specific places such as home and workplace. Moreover, there are only limited number of subjects in datasets in [9] and [7], which respectively is only 80 and 10. These numbers are insufficient to analyse various types of human activity patterns. Consequently, we used a large-scale American Time Use Survey (ATUS) dataset in our experiments, as it has been collected from a large number of heterogeneous populations with their demographic information [33]. Furthermore, the ATUS dataset consists of not only place information but also activity information that is densely categorized to help understand more detailed dependencies between places and activities in daily activity patterns.

In a recent study, which reported that individuals' human mobility patterns are independent of travel distances [42], they found that individuals' mobility patterns have a 93% predictability. This predictability was measured by the entropy of individual's trajectories across 50,000 cellphone users, which is highly predictable. As each activity pattern can be clearly distinguishable from others according to the demographic information, we analysed the differences of daily activity patterns and predictability of future activity and place by dividing activity traces of 6,765 subjects into four different groups in terms of their current employment status[1]. We compared four groups in terms of daily activity patterns and visualization of frequency of places and activities using t-distributed stochastic neighbor embedding (t-SNE) technique [25]. In addition, we conducted cross-prediction tasks in order to examine the compatibility between the four different groups.

Experiments have showed that every group has its unique activity patterns, and group-specific training is essential in order to accurately predict the future activity of individuals. Furthermore, the performance of prediction accuracy and the activity pattern of each group are highly correlated to each other in a way that as groups with more regular activity pattern result in higher prediction accuracy. LSTM networks consistently outperformed the Markov-based models by a margin of 0.3-7% relative improvement on multi-step look-ahead prediction. These results suggest that the LSTM networks have strength in capturing long-term dependencies of one's historical trajectories. The main contributions of our work are:

- We built human activity prediction models and comprehensively compared traditional and state-of-the-art temporal models to validate long-term dependency on daily human activity traces.
- We proposed and evaluated multi-step look-ahead predictions to resolve the problem of predicting immediate next time step along with arbitrary rescaled time intervals between consecutive samples.
- We investigated the differences in activity patterns among groups with various qualitative and quantitative analyses using a large-scale daily human activity dataset.

2. RELATED WORK

Location prediction.

The location predictors that have been studied so far can be categorized into two large groups based on whether the prediction models are trained with a single subject separately or multiple subjects collectively. The temporal prediction models applied in most of the individual mobility models recognize the mobility pattern of a single subject's past trajectories in advance to predicting the future locations. In [39] the author proposed a prediction technique based on nonlinear time series analysis to extract important locations that subjects spent most of their time. They estimated the time of the future visits and expected residence time in predicted locations for single subjects. Monreale et. al [27] attempted to train a decision tree, called T-pattern Tree, to find the best matching pattern and to predict the next locations. Many studies have built individual location predictors based on the Markov chain model [26, 24, 7, 2, 10, 43]. Using GPS trajectories data, [2, 26] clustered GPS data derived from an extended period of time into significant locations, and incorporated those classified locations into a prediction model. Song et al. [43] evaluated the next cell prediction using mobility patterns of more than 6,000 users on Dartmouth's campus Wi-Fi network. In their experiments, the median of prediction accuracy of second-order Markov predictor was shown to be about 72% for users with long sequences. In another study where the travel patterns of 500,000 individuals in Cote d'Ivoire were used, Markov-based models performed a prediction accuracy of 87% for stationary trajectories and 95% for non-stationary trajectories [24], which means individual mobility pattern is highly predictable. Gambs et al. [10] developed n-Mobility Markov Chain (n-MMC) to look at the n previous visited locations. The proposed model, n-MMC can be considered as a regular nth-order Markov model. For a specific user, [31] analyses the spatio-temporal patterns of a owner of devices over his/her movements and forecasts future Wi-Fi connectivity using the second-order Markov model. As many results related to the Markov model indicated previously, low-order Markov predictors outperformed high-order Markov models in predicting the immediate state of the next time step [43, 10]. In our experiments, the prediction accuracy of low-order Markov models was measured as baselines of our work, and we compared it with prediction accuracy obtained from LSTM networks for the purpose of verifying the assumptions we formulated of capturing long-range temporal dependency.

Collective mobility models, as opposed to individual models, are learned with multiple subjects to predict an individual's future locations [38, 6, 32, 46]. These models assume that different people can have a tendency to follow similar mobility patterns with others. Under this hypothesis, [38, 6, 32] utilized the users' social relationships and incorporated it into the model of human mobility. Sadilek et al. [38] built a scalable probabilistic model of human mobility in a dynamic Bayesian network, inferring social ties with considering patterns in friendship information for predicting future locations. Furthermore, Cho et al. [6] built Periodic and Social Mobility Model with a mixture of Gaussians, and they found that compared to the long-distance travel, a short-ranged travel is periodic in terms of both spatial and temporal features and not influenced by the social network structure. By combining all users' features in supervised

[1] Due to the unbalance in heterogeneous populations in the sizable ATUS dataset which includes a total of 49,666 subjects, we randomly and uniformly selected 1,945 students with job, 2,000 non-students with job, 820 students without job and 2,000 non-students without job.

learning model based on linear regression, [32] addressed the problem of predicting the next location a mobile user will visit.

The most closely related works to ours are those that predict physical location using neural network architectures and their variants for time-series domains. To solve the problem of next place prediction, [9] considered the problem as a classification task. They learned mobility predictors using feedforward artificial neural networks to predict the next location. However, their model has not taken into account the sequential model on this type of time-series prediction problem. Given start point associated metadata such as date, time, and client information of taxi trajectory, [8] used multi-layer perceptron (MLP), bidirectional recurrent neural networks and memory networks to predict the destination of a taxi. Their best model was used by bidirectional recurrent neural networks that encode the prefix, and several embeddings are employed to encode the metadata and destination clusters for generating the output.

Activity recognition.

Previous research on activity recognition have used various classifiers including Fuzzy Logic, Neural Network, Naive Bayes, Bayesian Network, Nearest Neighbor, Decision tree, Support Vector Machines, boosting and bagging (refer to [22]). More recently, continuous recognition of activity was posed as a sequence labeling problem [41, 45], using temporal models such as Hidden Markov Model [36], Conditional Random Field (CRF) [21], and structured Large-Margin classifiers [1]. The continuous nature of daily activities makes the temporal models potentially more appropriate for handling noisy multisensory streams and labels from smartphones. In [16], the author compared the performance of various classifiers that represent temporal vs. non-temporal approaches and generative vs. discriminative approaches, and the feasibility of automatic annotation of unconstrained daily activities was demonstrated. By exploiting convolutional neural networks (CNNs) [23] and LSTM recurrent units, [34] studied the activity recognition problem with wearable sensor data. The combination of CNNs and LSTMs architecture gave rise to capturing temporal dependencies on features extracted by convolutional operations. While they used a fine-grained sub-second level of sensory dataset, in this study we used a larger unit of temporal daily activity traces to predict long periods of future activities up to 6 hours.

3. BACKGROUND

3.1 Markov Model with Time Information

A Markov model is a subclass of Bayesian networks known as a primitive dynamic Bayesian networks, which are simply Bayesian networks to model time-series type of data. A first-order Markov model assumes that the label state of a sequence is conditionally independent of other label states given the state of the preceding sequence. The observed feature at a sequence is conditionally independent of other observations given the state of the sequence. And the probability distribution of one state is only dependent on the preceding state. In Figure 1, each state $L_{1:T}$ indicates a sequence of state and $O_{1:T}$ is a sequence of corresponding absolute time information. Hereafter, we denote the state L_t as the label state of activities or places. The joint probability of the state of sequence is composed of two probabilities: initial probability of the state $P(L_1 = l_1|O_1 = o_1)P(O_1 = o_1)$, and probability of transition $P(L_t = l_t|L_{t-1} = l_{t-1}, O_t = o_t)$. The joint probability of a Markov model under these assumptions is

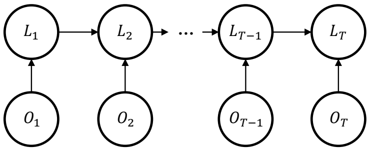


Figure 1: Graphical representation of Markov model with time information

$$L(p) = P(L_1 = l_1|O_1 = o_1)P(O_1 = o_1)\prod_{t=2}^{T} P(L_t = l_t|L_{t-1} = l_{t-1}, O_t = o_t) \tag{1}$$

The transition probability of Markov model is trained from maximum likelihood estimation given the state of sequence.

$$logL(p) = logP(L_1 = l_1|O_1 = o_1)P(O_1 = o_1) + \sum_{i,j,k} n_{ijk} log p_{ijk} \tag{2}$$

where p_{ijk} is a probability of transition $p_{ijk} = P(L_t = j|L_{t-1} = i, O_t = k)$ and n_{ijk} is a total count of transitions from i to j given a k. By taking the derivative and setting the Equation 2 to zero, we obtain following equation,

$$\frac{\partial logL(p)}{\partial p_{ijk}} = \frac{n_{ijk}}{p_{ijk}} = 0 \tag{3}$$

However, when we solve Equation 3, all p_{ijk} should be ∞. To prevent arbitrarily changing the probability of transition, the method of Lagrange multipliers [19] is used, which allows to optimize differentiable likelihood function with a given constraint as $\forall i, k$, $\sum_j p_{ijk} = 1$.

$$p_{ijk} = \frac{n_{ijk}}{\sum_j n_{ijk}} \tag{4}$$

Accordingly, the transition probability, p_{ijk}, the parameter of Markov model is learned from Equation 4, dividing transition counts of state i followed by state j given an input k over total transition counts coming out from state i with k. While we are estimating the transition parameter by counting the frequency of each transitions, the problem of overfitting can be occurred if the number of observations is insufficient compared to the dimension of transition parameter. A certain transition probability becomes zero when the observation has not been previously seen in the finite training set. Consequently, the Markov model leads to numerical errors as unseen transitions increase. If the log-likelihood equals to zero, then it is impossible to compute log-likelihood because the result is likely to converge towards $-\infty$. To prevent this numerical errors, we used the Dirichlet prior [44] that enables Markov model to improve numerical probability and reduce numerical errors through assuming the prior distribution with respect to the transition parameter. By doing this, the point estimation problem is changed into the posterior probability estimation problem. We employed a Laplace smoothing method [30], assigning the non-zero probabilities to the unseen observations so that it allows Markov model to become more accurate in estimating the probability of the states.

3.2 Long Short-Term Memory

Traditional RNN was developed as a neural network structure for sequential processing tasks such as speech recognition and language modeling. RNN has cyclic path of synaptic connections in which feeding the activations from previous time steps as input to the network to determine for the current input [18]. However, [17] showed that RNNs trained with a variant of back-propagation or other gradient-based methods can cause vanishing or exploding gradients problem when the input sequence is very long, since the gradients propagate down through a large number of layers for every timestep in the recurrent structure.

In the mid-90s, Long Short-Term Memory networks, a special type of RNN, was orginally introduced by [17] as a solution to the vanishing gradient problem. LSTM networks are capable of learning long-term dependencies by incorporating memory cell that makes the network to forget or add information about previous hidden states in current cell state. Recently, variants of LSTM networks have been developed according to the particular objective of sequential tasks. We used one of the LSTM variant, augmented by "peephole connections" that is known as better at exploiting the long-range dependencies and nonlinear periodic patterns in the sequence than conventional LSTM networks [12]. A single LSTM memory block is shown in Figure 2. The LSTM memory block in the hidden layer contains memory cells, cell state c_t at time t. The idea behind of memory cell is to control and protect the error back flow by multiplicative gating units. Each memory block consists of sigmoidal gates: input gate i_t, forget gate f_t, output gate o_t, and memory cell c_t. Given an input sequence, $x = < x_1, ..., x_T >$, an LSTM networks update sequence of hidden vector $h = < h_1, ..., h_T >$ and cell state vector $c = < c_1, ..., c_T >$ to generate output vector sequence by iteratively calculating the following equations, where W is weight vectors and $t = 1, ..., T$:

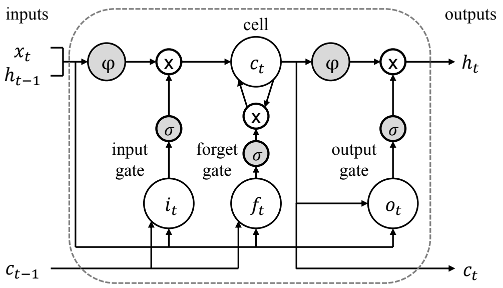


Figure 2: A single memory block of Long Short-Term Memory

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t \varphi(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \varphi(c_t)
\end{aligned}
$$

Essentially, the four gates in an LSTM unit take an input x_t and the previous hidden state h_{t-1} at each time step t. Additionally, an input gate and forget gate receive an output of previous cell state c_{t-1}. The input gate manages the flow of input activations into the memory cell, and the forget gate scales the internal state of the memory cell by the self-recurrent connection called "Constant Error Carousel" (CEC) of memory cell [11]. The output gate controls the flow of memory cell activations into the hidden state. The memory cell unit c_t at time step t is computed by summing up the previous cell state c_{t-1}, which is modulated by f_t and the internal state, modulated by i_t. The internal state of memory block is computed by passing the modulation of input x_t through the hyperbolic tangent non-linearity function, where $\varphi(x) = tanh(x)$ squashes the input x into the range of $[-1, 1]$. Since both i_t and f_t are associated with logistic sigmoid non-linearity function $\sigma(x)$, which squashes real-valued input x to lie within the range of [0, 1], allowing memory cell unit c_t to adaptively control which previous information to kept or which new information to write in the current memory cell. The output of hidden state at time step t can be obtained by multiplying the output gate o_t and the activation of memory cell state $\varphi(c_t)$. For the multi-layer LSTM networks, the output from hidden state in $n-1$ hidden layer at time step t is used as the input of n hidden layer at same time step t. Given the hidden state h_t from the top of hidden layer in the network, the final output L_t is computed by applying sigmoid activation function to h_t.

4. METHODOLOGY

4.1 LSTM Predictor

For LSTM prediction networks, the model takes an input vector x_t at every time step t, and outputs an output vector L_t. Each output vector has information for the likelihood of different states. The number of hidden layers and that of hidden units in each hidden layer is determined through a number of experiments with different settings. We used 'one-hot'encoding to convert discrete data into vector representation. For example, if we assume that class l is an actual target value at time step t, size of L target vector $target_t$ composed of a single one value for l-th index and all the remainders zero.

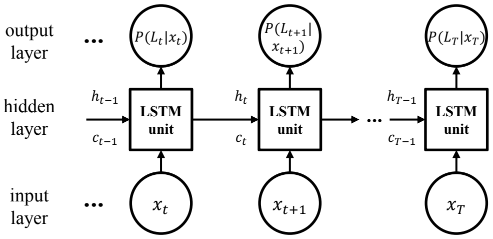


Figure 3: LSTM architecture

Since the LSTM networks are continuous model, a softmax activation function such that the real-valued outputs always sum to 1, is adopted at the output layer to convert the continuous type of output into discrete class of output. Given an output vector $L_t = \sigma(W_{hL}h_t + b_L)$ at time step t, a softmax activation function parameterizes each output vector L_t into multinomial distribution, $P(L_t = l|x_t)$ where l is a certain class among the output classes. Given by the softmax function, predictive distribution at time step t is:

$$P(L_t = l|x_t) = \frac{e^{L_t(l)}}{\sum\limits_{l=1}^{L} e^{L_t(l)}} \tag{5}$$

The model therefore outputs the state label of activity or place that has a higher likelihood among the state labels. After the model completes forward pass, the backward pass computes error signals for all weights. The objective function used in our model is the cross-entropy cost function [37], since our task can be considered as a multi-class classification problem. The cross-entropy cost function over a sequence can be calculated as:

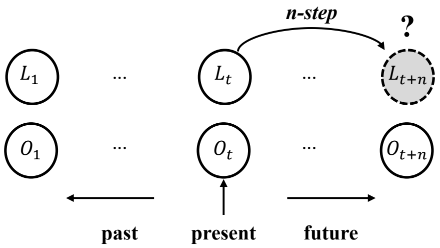
$$-\sum_{l=1}^{L} P(target_t = l) log P(L_t = l|x_t) \tag{6}$$

The network was trained with stochastic gradient descent-based optimization algorithm, truncated backpropagation through time (BPTT) and customized version of real time recurrent learning (RTRL) [18].

4.2 Multi-step look-ahead prediction

To predict future states, some previous studies have exploited Markov model using a dataset with short data sampling rates between two consecutive samples. When this type of dataset with an arbitrary size of time interval is used to track the subjects' transitions, the probability of next sample staying at the same state compared to one second earlier will be about 99%, as the time interval between two consecutive samples reduces. Intuitively, it is clear that smaller time units used in temporal models definitely lead to higher prediction accuracy, since people do not change their activity suddenly. For instance, people regularly stay and spend most of their time in usual places, such as home and workplace, during the day. Consequently, actual prediction performance and time intervals between two consecutive samples are closely correlated. This indicates that, for time-series prediction, arbitrary scaled datasets are erroneous.

To avoid artificially generated results from arbitrary scaled datasets, we trained separate multi-step look-ahead prediction models for each look-ahead time in minutes and hours from 15 minutes up to 6 hour. Multi-step look-ahead prediction model enables us to examine how far we could accurately predict future state of the subjects' regardless of the size of time intervals between two consecutive units. The task of our study is similar to those in several studies on multi-step-ahead prediction with various algorithms [35, 3, 5]. These studies also have built prediction models to investigate the effectiveness of predicting multi-steps, other than a single step, ahead of current state. Our approach differs from previous works in that we employed LSTM networks, the state-of-the-art temporal model to capture long ranges of dependency over the sequence. To define our task more concretely, multi-step look-ahead prediction is defined as:



$L_{1:T}$: sequence of semantic location
$O_{1:T}$: sequence of absolute time information

Figure 4: Multi-step look-ahead prediction

Multi-step look-ahead prediction: Given a sequence of state $L_{1:T}$ along with a sequence of regularly assigned absolute time information $O_{1:T}$, predict a state L_{t+n} after n-steps at time step t.

Markov model with time information is specified by the probability of transition $P(L_{t+n} = l_{t+n}|L_t = l_t, O_{t+n} = o_{t+n})$ that models the probability of observing n-step ahead of state L_{t+n} given a current state L_t and a n-step ahead of time information O_{t+n}. The only difference between a standard Markov model and the Markov model with time information we used for multi-step look-ahead prediction is that, our Markov model does not require to compute the sum of products of transition probabilities by multiplying transition probability matrix by itself. We trained the Markov model separately according to the number of look-ahead time, from 15 minutes up to 6 hour. With the trained parameter of transition probability, a predicted n-step ahead of state is derived from:

$$\hat{L}_{t+n} = \underset{l}{\arg\max} P(L_{t+n} = l|L_t = l_t, O_{t+n} = o_{t+n}) \tag{7}$$

Using Equation 7, we take out the probabilities over all transition probabilities conditioned on a current state L_t and a n-step ahead of time information O_{t+n}, for independently predicting n-step ahead of state L_{t+n} at time step t. The index number with the largest probability across all transition probabilities indicates the predicted label of n-step future state. The main idea behind the prediction scheme is to compare the predicted n-step future state and ground-truth n-step future state. Prediction accuracy is measured by dividing the number of correct predictions by the number of predictions. When the model predicts n-step ahead of the current state, the number of predictions is $T-n$, where T is the total length of a given sequence.

In LSTM prediction networks, a target vector $target_t$ is set to n-step ahead of an input vector x_{t+n} at each time step t, i.e. $target_{1:T-n} = x_{1+n:T}$. As with the Markov model, we trained the LSTM predictor independently according to the number of look-ahead time. With the trained prediction network, the accuracy of n-step look-ahead prediction at time step t is calculated by comparing correctness between l-th index of target vector $target_t$ and the index number that has the largest probability under an output vector $P(L_t|x_t)$ parameterised by softmax activation function.

5. EXPERIMENTS AND RESULTS

5.1 American Time Use Survey

The ATUS dataset is accessible every year on the U.S. Census Bureau of Labor Statistics website [33]. The survey is conducted entirely by telephone, and subjects are asked to recall their daily activities and trips of the previous day. ATUS measures the amount of time people spend traveling to various semantic places such as home, workplace, restaurants, and stores. Each sample also provides various types of activity information, such as personal care, household activities, and socializing. In addition, the actual time information with regard to the arrival time from a previous activity and place, is contained in each sample. Activity traces are collected 24 hours of continuous recordings, and all traces start from 4:00 am for all subjects.

5.2 Data Preprocessing

We discarded subjects with a total daily duration of less than 17 hours because a duration less than 17 hours is not considered to fully represent the daily activity pattern. We were able to finally select 6,765 subjects among 49,666 subjects generated from 2008 to 2015, with balancing proportion of heterogeneous populations by current employment status. Each subject was divided into four groups: student with job, student without job, non-student with job, and non-student without job. Table 1 describes statistics for the four groups on which we used in our experiments. The subjects sampled from even years are combined to use as a training set, and the subjects obtained from odd years are used as a test set to conduct cross-validation evaluations.

Table 1: ATUS dataset Statistics

	Student with job	Non-student with job	Student without job	Non-student without job
Training set	1,010	1,000	418	1,000
Test set	935	1,000	402	1,000
Total subjects	1,945	2,000	820	2,000

Table 2: Details of 16 place categories and 18 activity categories

Common Place Categories

Label	Description
p1	Respondent's home or yard
p2	Respondent's workplace
p3	Someone else's home
p4	Restaurant or bar
p5	Place of worship
p6	Grocery store
p7	Other store/mall
p8	School
p9	Outdoors away from home
p10	Library
p11	Other place
p12	Transportation-related
p13	Bank
p14	Gym/health club
p15	Post office
p16	Unspecified place

Common Activity Categories

Label	Description
a1	Personal care
a2	Household activities
a3	Caring for and helping household members
a4	Caring for and helping nonhousehold members
a5	Working and work-related activities
a6	Educational activities
a7	Consumer goods purchases
a8	Professional and personal care services
a9	Household services
a10	Government services
a11	Eating and drinking
a12	Socializing, relaxing and leisure
a13	Sports, exercise, and recreation
a14	Organizational, civic, and religious activities
a15	Volunteering
a16	Telephone calls, mail, and e-mail
a17	Unknown activities
a18	Travel-related activities

Table 3: Details of top-20 place–activity pair labels

Top-20 Place-Activity pair Categories

Label	Description			
	Place		Activity	
pa1	p16	Unspecified place	a1	Personal care
pa2	p1	Respondent's home or yard	a12	Socializing, relaxing and leisure
pa3	p2	Respondent's workplace	a5	Working and work-related activities
pa4	p1	Respondent's home or yard	a2	Household activities
pa5	p12	Transportation-related	a18	Travel-related activities
pa6	p1	Respondent's home or yard	a11	Eating and drinking
pa7	p1	Respondent's home or yard	a3	Caring for and helping household
pa8	p8	School	a6	Educational activities
pa9	p1	Respondent's home or yard	a6	Educational activities
pa10	p1	Respondent's home or yard	a5	Working and work-related activities
pa11	p3	Someone else's home	a12	Socializing, relaxing and leisure
pa12	p4	Restaurant or bar	a11	Eating and drinking
pa13	p7	Other store/mall	a7	Consumer goods purchases
pa14	p11	Other place	a12	Socializing, relaxing and leisure
pa15	p1	Respondent's home or yard	a18	Travel-related activities
pa16	p2	Respondent's workplace	a11	Eating and drinking
pa17	p1	Respondent's home or yard	a16	Telephone calls, mail, and e-mail
pa18	p6	Grocery store	a7	Consumer goods purchases
pa19	p11	Other place	a8	Professional and personal care
pa20	p11	Other place	a6	Educational activities
pa21	None of the above			

Labeling places / activities and time information.

The primary motivation of discretization is to convert the provided activity and place features, and actual time information into the finite states so as to use discrete type of data in our model. In the original ATUS dataset, each sample contains two main features: pre-defined identification code for visited activities and places, and actual time information. As shown in Table 2, places and activities are categorized into 16 and 18 labels, respectively. Furthermore, we selected the top-20 place-activity pair labels, indicating the top-20 jointly occurred place and activity labels, to integrate categorized place and activity labels. The top-20 place-activity pair labels are sorted in descending order of frequency of co-occurrence as shown in Table 3. In order to discretize the continuous set of actual time information, we first categorized time information into two main classes: weekday and weekend. Afterwards, we subdivided time information into 8 discrete times of day, i.e. time information is discretized into the total of 16 different classes, and each class indicates the departure time at the current state in 3 hours. For LSTM model, instead of directly transforming the discretized time information into 'one-hot'encoding vector, we converted actual time information into continuous vector representation with probability distribution over every hour of the day from 00:00 to 23:59. This lets the LSTM network to be able to train with more precise time information compared to discretized time information with the same number of time bins.

Rescaling in time.

In the original ATUS dataset, it is easy to identify a transition between current and next state but difficult to identify how long subject stays at each state. If we transform the dataset through rescaling it in terms of even time interval, a time spent on a state can be measured by counting the number of the same states appeared consecutively. Since the original dataset was not sampled on even time interval, we should preprocess dataset so that the samples in the dataset are displayed on even time interval. Figure 5 shows the one-dimensional grid data structure which is divided by the units with even time interval 15 minutes, and the samples are preprocessed on that structure. The value of start point of this structure is a time stamp of the first sample in the dataset, and this first sample is assigned to it. The rest of the samples in dataset are assigned on the end point of each unit of the structure based on their time stamps. As explained in Figure 6, there are two rules for assigning each sample on the structure in Figure 5.

- *Rule 1*: Inside a unit, if any time stamp of sample is not located on it, sample with smallest time stamp later than the end point of corresponding unit is assigned on its end point.
- *Rule 2*: If a time stamp of a sample is located inside a unit, we investigate majority of a time stamp measured by time-difference from closer one between the previous time stamp and the start point of the unit to its time stamp, then assign a sample with the largest majority to the end point of the unit.

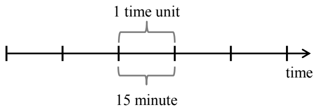
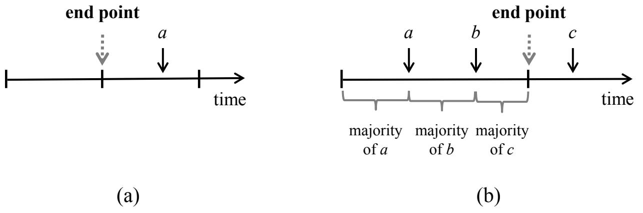


Figure 5: Data structure where samples of our dataset will be assigned



(a) (b)

Figure 6: The end point of a unit we are considering for assigning a sample is marked as dotted arrow. Characters *a*, *b*, and *c* are samples with corresponding time stamps. (a) A case where *Rule 1* is assigned. Sample *a* is assigned to the end point by *Rule 1*. (b) A case where *Rule 2* is assigned. Sample with the largest majority will be assigned to the end point of the unit by *Rule 2*.

For the portion from the most recent time stamp inside the unit to its end point as like the most right majority one in (b) of Figure 6, it will be the majority of a sample whose time stamp passes

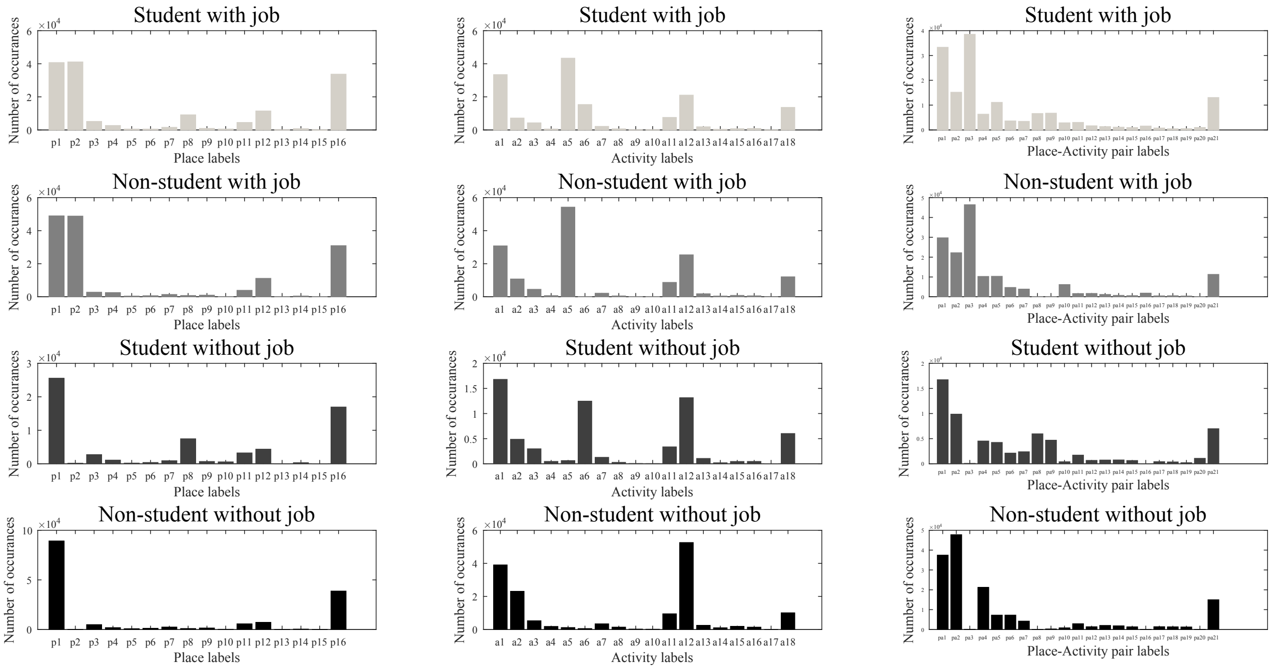


Figure 7: Frequency of each label according to the four groups and three label types

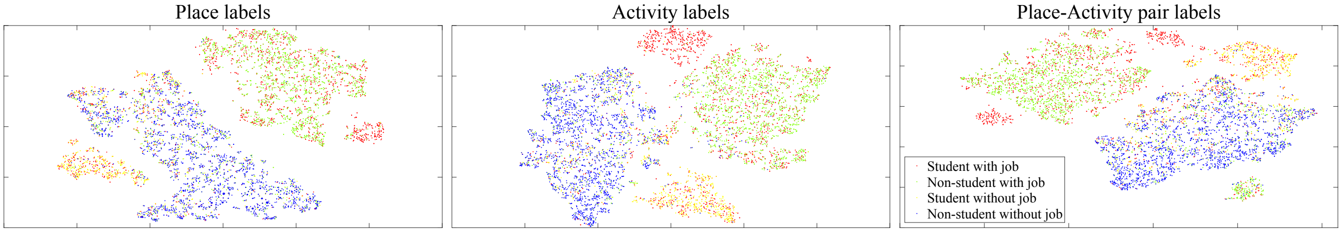


Figure 8: t-SNE results of three label types

and is close to the end point. By applying *Rule 1* and *Rule 2*, we obtained preprocessed dataset in which the time interval between two consecutive samples are even as 15 minutes. In general, the subject has a tendency to stay in a one place for a long time (e.g. sleeping at home, working at workplace, etc.). Accordingly, there is no occasion to set a short time interval between samples. An unit with 15 minutes interval is sufficient for preprocessing the dataset. In addition, varying the size of time interval of an unit does not impact on our evaluation, since we evaluated prediction accuracy of the future state by looking at absolute look-ahead of time in minutes and hours. When the time interval between two samples get shorter, the length of sequence will be increased, and that would only increases the running time of our evaluations.

5.3 Models Compared

We built Markov-based and LSTM predictors to comprehensively compare performance of traditional and state-of-the-art temporal models upon the multi-step look-ahead prediction. The predictors are mainly separated under two categories according to whether it incorporates time information or not. Eight predictors we used in our experiments are as follows:

- $LSTM+TI$: LSTM with time information
- $LSTM$: LSTM without time information
- MM^2+TI: 2nd-order Markov model with time information
- MM^2: 2nd-order Markov model without time information
- MM^1+TI: 1st-order Markov model with time information
- MM^1: 1st-order Markov model without time information
- MM^0+TI: frequency based model with time information
- MM^0: frequency based model without time information

Baseline - Frequency based.

Chance level accuracy is selected as one of the baseline models which we refer to as 0-th order Markov model, MM^0. In spite of this baseline seems very simple, it is a very strong baseline, because some activities and/or places are dominant in a daily life pattern (e.g. home and workplace) as illustrated in Figure 7. As far as the uniqueness of daily human activity pattern of four different groups is concerned, this baseline is capable of identifying the most likely activity and/or place for a specific group. It is measured by ratio of the number of most frequented activity and/or place to the total number of predictions. In addition, MM^0+TI, a frequency based predictor with time information, predicts multi-step ahead of activity and/or place with the probability distribution over corresponding discrete

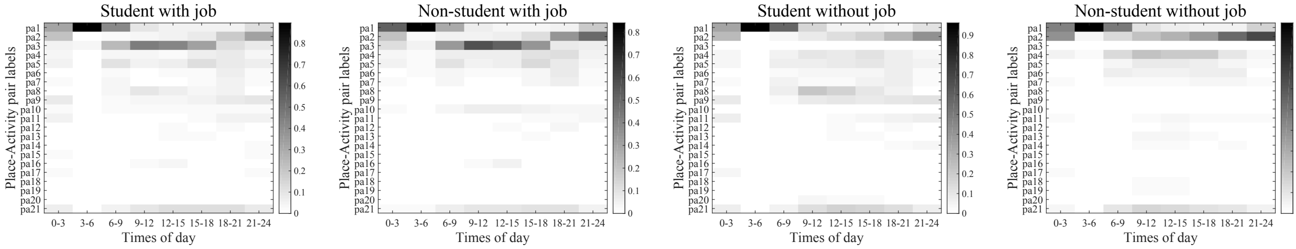


Figure 9: Probability of time-based daily activity in four groups. X-axis is the time in 3 hours and y-axis is the 21 top-20 place–activity pair labels. Colorbar on the right side indicates probability.

time information. $MM^0 + TI$ is trained by counting the number of occurrences for each place label in terms of the corresponding class of time information, and then normalized by the number of occurrences for each class of time information separately. Finally, $MM^0 + TI$ picks out future activity and/or place which has a highest probability at a specific look-ahead time.

$$\hat{L}_{t+n} = \arg\max_{l} P(L_{t+n} = l | O_{t+n} = o_{t+n}) \tag{8}$$

In testing stage, using an Equation 8, $MM^0 + TI$ forecasts the n-step ahead of activity and/or place by simply extracting a target of activity and/or place L_{t+n} which has a maximum probability with corresponding time information O_{t+n} at time step t.

5.4 Result 1 – Patterns of daily activities and places

Daily activity patterns of each subject are purely dependent on their current employment status. To qualitatively analyse the differences of daily activity patterns among the four groups using unlableled data, a total frequency of labels over all groups in the dataset, we applied t-distributed stochastic neighbor embedding (t-SNE), unsupervised machine learning technique. Figure 7 shows the frequency of each label for all groups in three different types of label. t-SNE is a non-linear dimensionality reduction technique for embedding high-dimensional dataset into two-dimensional spaces to visualize a grouped scatter plot [25]. Our initial embedding spaces each have 21 dimensions. As shown in Figure 8, the four groups are markedly different in their daily activity patterns. Especially, the histograms are distinctly clustered by job status. Even though the subjects had fewer students than non-students, the student groups were uniquely divided against non-student groups. According to the visual patterns, in order to predict a specific subject's future activities and places, it is crucial to separately build each group's predictor.

The uniqueness of daily activity patterns among the four groups can also be explained by visualizing the probability of time-based daily activity patterns. As shown in Figure 9, we can easily capture that each group has their unique daily activity pattern. The subjects tend to behave and move differently according to whether they are currently a student or not. Moreover, subjects usually follow regular activity patterns in terms of job status. Based on the observations from daily activity patterns in the student groups, they spent their day time at school for educational activities. In case of the students who have a job, the activity patterns are somewhat unusual compared to students without job. By contrast, the group of non-students with job spent most of their time in major places, such as workplace and home, and rhythmically traveling back and forth between those major places. We observed that daily activity patterns of non-student groups are relatively more consistent compared to student groups in terms of the top-20 place-activity pair labels. In this respect, it is completely inconsistent to predict future activity and place of specific subjects from different groups collaboratively. Such a collaborative prediction might adversely affect the prediction of specific subjects' performances. Thus, to examine the predictability of future activity and place for a subject in a particular group, the predictors should have to be separately trained by each group and unaffected by other groups.

5.5 Result 2 – Multi-step look-ahead prediction

With the preprocessed dataset, each groups' dataset is subdivided into the number of subjects as multiple sub-sequences, and thereby each sub-sequence indicates sequential activity or place traces of a single subject. The two-fold cross-validation technique [20] is used to measure average prediction accuracy for each group. Performance is measured by leave-one-held out years-out cross-validation. All prediction models are trained with sub-sequences from even years, and the remaining sub-sequences from odd years are used as a test set, and vice versa. The results of two evaluations are then averaged to estimate prediction accuracy for each group.

In LSTM predictors, the model complexity, such as setting the number of hidden layers and that of memory units, was determined by a number of experiments with various settings. We conducted experiments on 12 different network configurations, which are 1, 2 and 3 hidden layers, and 5, 10, 15 and 20 units for each hidden layer, and finally chose the combination of 1 hidden layer with 20 memory units based on the performance. Furthermore, during training, weight updating was set to stop at 500 epochs or when the model did not obtain the lowest testing error within consecutive 200 epochs. The momentum and learning rate were set to 0.9 and 0.00003, respectively.

Recall that prediction accuracy is defined as the number of correct predictions divided by the number of predictions. Figure 10 illustrates the performance of eight predictors, and the evaluations are individually performed for each group in three different types of labels. The highest prediction accuracy among the groups can differ due to the regularity in their activity pattern. It is seen that daily activity patterns of student groups are relatively irregular compared to non-student groups. As a result of this, average prediction accuracy of student groups shows to be relatively lower than non-student groups.

Another important fact is that the predictors which incorporated time information produced higher accuracy than the predictors without considering time information. The absolute time information indeed aided to improve performance for all predictors. It is interesting to see that the $LSTM$ is not necessarily worse than $LSTM + TI$. The memory cell in $LSTM$ predictor has a capability for memorizing previous observations by counting steps from the initial to current.

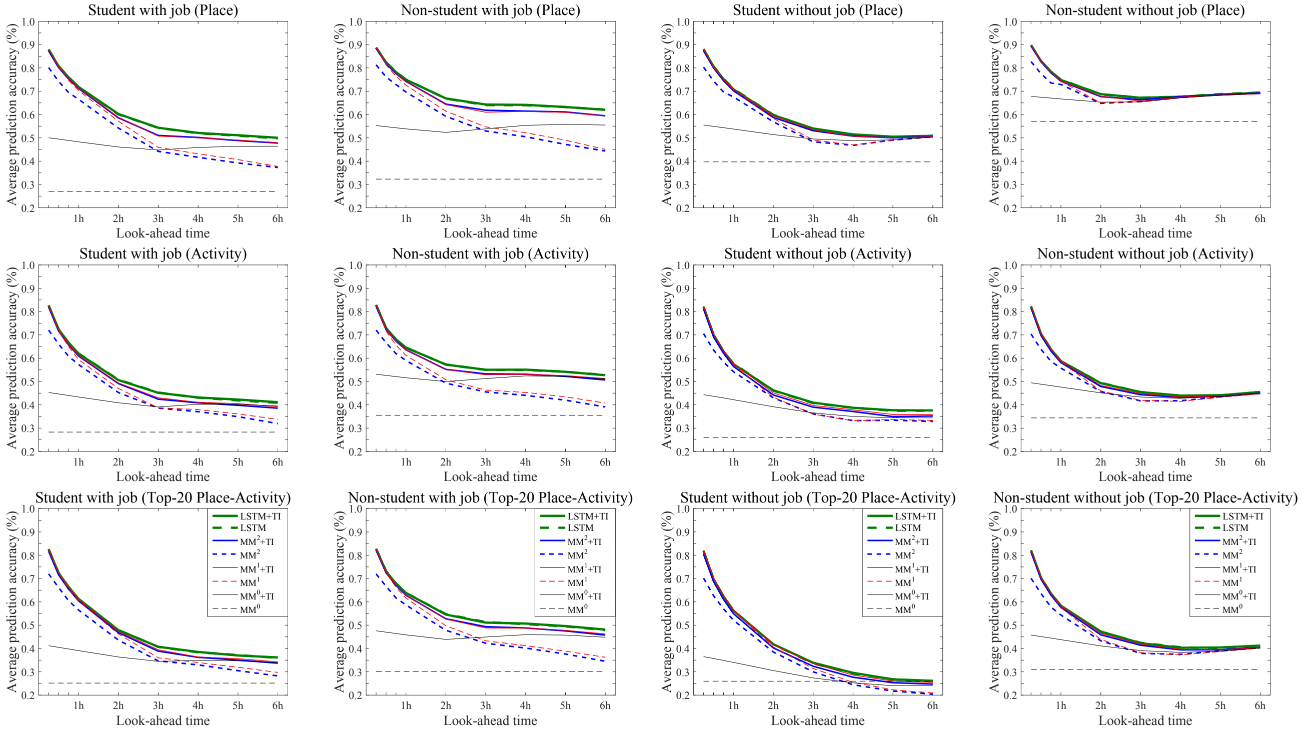


Figure 10: Average multi-step look-ahead prediction accuracy of eight prediction models in three different label types

This makes the network able to internally learn long-term dependencies when the data dependency is present given a sequence. Another possible reason is that every sub-sequence included in training and test sets starts with the same time information as 4:00 am. The strength of using time information was not fully being utilized in this setting. Overall, the result implies that absolute time information is an important feature for recognizing semi-regular patterns in daily activities.

We also found that people tend to travel and behave according to their regularly schedule within an hour. The performances among all predictors are almost the same for within an hour look-ahead prediction. As the look-ahead time increases up to 6 hours, the differences of performance among predictors become larger. Another plain fact is that as the look-ahead time increases, prediction accuracy of most predictors' deceases. In most case, prediction accuracy of 6 hour look-ahead prediction converges to both frequency based predictor MM^0 and MM^0+TI. It seems to be a common observation. This observation indicates that, it is a challenging problem to predict future activities and places far away from the current time. In terms of prediction accuracy among all predictors, $LSTM+TI$ has a noticeable improvement compared to the best performed Markov-based predictor MM^1+TI, and the degree of falling on its curve is relatively small over other predictors. Overall, $LSTM+TI$ consistently performed 0.3-7% better relative improvement than MM^1+TI. From this result, in order to accurately predict future activity and place, we are likely to assume that the predictors capture long-term dependencies of past observations. However, MM^1 is an insufficient model to consider long-term dependencies since it can only captures short-term dependencies, between two states. Accordingly, Markov-based predictors require to increase the order of its model because high-order Markov model uses more past observations for estimating the current state. However, as shown in Figure 10, MM^1+TI performed almost the same as MM^2+TI at every look-ahead time. Although MM^2+TI is more precise than MM^1+TI in terms of using more information during training, high-order Markov model has a limitation of high dimensional parameter space. For example, having k different states, the Markov model requires extremely large k^{n+1} dimensions to explicitly estimate the next state with n past observations. If the length of sequence is relatively small compared to the models' dimensional spaces, the high-dimensional spaces can negatively affect the performance of Markov model, since the model complexity will be exponentially increased. Without abundant datasets, the high-order Markov model raises the problem of overfitting due to its large parameter space.

The predictors using place label show the highest prediction accuracy among the three types of labels. This is because daily activity patterns are simple when we only consider location information. If we use activity labels for the same sequence, the activity pattern will be relatively dispersed. Also, the number of place labels is less than activity labels and top-20 place-activity pair labels. Hence, it is relatively easy to predict future places compared to future activities and top-20 pair labels. Different from the labels with only using places and activities, the top-20 place-activity pair label is very detailed in the aspect of context. It is seen that although it appears dispersed patterns because of the largest number of labels, it performed similar to the predictors that only used activity labels.

In order to analyse the effectiveness of a LSTM predictor, we computed the conditional probability of correctly predicted true labels given the number of true labels based on the aggregated confusion matrices from 15 minutes to 6 hour look-ahead predictions. In Figure 11, diagonal terms indicate recall for each label. Overall, the predictor has a tendency to predict labels that have high frequency. Since we sorted the place-activity pair labels in high-frequency order, the predictor was also trained with a large number of labels such

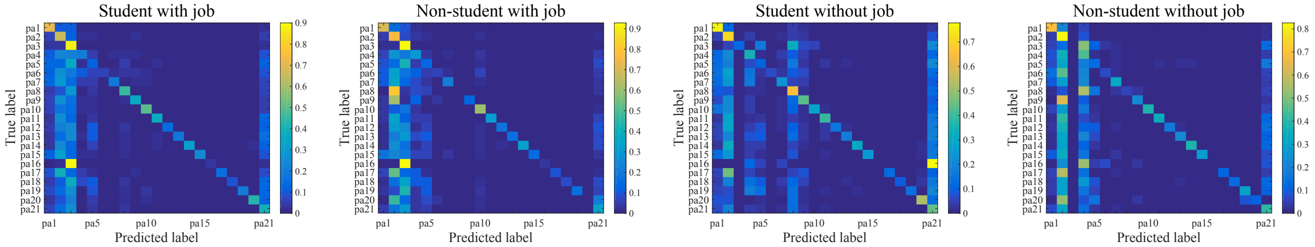


Figure 11: Probability of correctly predicted true labels in four groups. X-axis is predicted 21 top-20 place–activity pair labels and y-axis is true 21 top-20 place–activity pair labels. Colorbar on the right side indicates probability.

Table 4: Prediction accuracy of cross-prediction among four groups. 1-hour look-ahead prediction task and top-20 place–activity pair label are used with $LSTM+TI$ predictor. Boldface indicates the best result in cross-prediction for each group.

		Test set			
		Student with job	Non-student with job	Student without job	Non-student without job
Training set	Student with job	**0.6114**	0.5981	0.5544	0.5657
	Non-student with job	0.5535	**0.6408**	0.4401	0.5681
	Student without job	0.5341	0.5550	**0.5662**	0.5527
	Non-student without job	0.2354	0.2552	0.3360	**0.5881**

as pa1, pa2 and pa3, which have higher prior probability. Hence, the weights of prediction naturally appear to be congregated in the left side of matrices. Diagonal terms were much more highlighted than off-diagonal terms due to their higher prediction accuracy. For example, recalls for the educational activities related labels such as pa8 and pa9 in the students group were highlighted, implying that activity patterns are different from non-student groups. However, traveling and eating-related labels, pa5 and pa16, had higher recall on off-diagonal term compared to diagonal term. Even if we used the trained prediction network with a specific group, it was a challenging task to predict specific activities such as traveling and eating-related activities.

5.6 Result 3 – Cross-Prediction among different groups

The aim of the cross-prediction task was to quantitatively analyse how the activity patterns of other groups can affect the predictability of activities in a specific group. Different from the experiments above, LSTM predictor was trained on sub-sequences of a certain group and measured prediction accuracy with sub-sequences from a different group. As we have seen daily activity patterns for different groups, each group has their unique activity pattern. Intuitively, the performances of cross-prediction deeply depend on how different the activity patterns are between training and test sets. If there exists a significant difference, the performance of cross-prediction will be down. As results in Table 4 show, the group-specific predictions outperformed the cross-predictions that use training and test sets from different groups. To conclude, it is the group-specific training that has the capacity to capture regular activity patterns latent in a specific group.

6. CONCLUSION

In this paper, we studied predictability of daily human activities using large-scale ATUS dataset. Compared to the dataset used in previous studies, this dataset contains a large number of subjects from heterogeneous populations with individuals' detailed demographic information, and both activity and place information. We proposed and evaluated multi-step look-ahead prediction up to 6 hours. The main findings of our experiments are as follows. LSTM-based predictors constantly produced better performances than Markov-based predictors for every look-ahead time up to 6 hours. Due to the problem of overfitting, increased Markov-based model complexity led to difficulty in model learning and inference. Unlike the Markov-based model, the LSTM network, the non-parametric model, has an advantage to efficiently capture long-term temporal dependencies of regularity in daily human activity patterns. Additionally, we divided subjects in ATUS dataset into four groups based on their current employment status to identify the differences of daily activity patterns among different groups. According to the experiments, the activity patterns of each group are unique, in terms of analysing differences of daily activity patterns and visualization of frequency of places and activities using t-SNE technique. Quantitatively, we evaluated cross-prediction tasks to examine the compatibility of four groups and the importance of group-specific training.

7. REFERENCES

[1] Y. Altun, I. Tsochantaridis, T. Hofmann, et al. Hidden markov support vector machines. In *ICML*, volume 3, pages 3–10, 2003.

[2] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[3] R. Boné and M. Crucianu. Multi-step-ahead prediction with neural networks: a review. *9emes rencontres internationales: Approches Connexionnistes en Sciences*, 2:97–106, 2002.

[4] J. Chang and E. Sun. Location 3: How users share and respond to location-based data on social networking sites. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 74–80, 2011.

[5] H. Cheng, P.-N. Tan, J. Gao, and J. Scripps. Multistep-ahead time series prediction. In *Advances in knowledge discovery and data mining*, pages 765–774. Springer, 2006.

[6] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[7] Y. Chon, H. Shin, E. Talipov, and H. Cha. Evaluating mobility models for temporal prediction with high-granularity mobility

data. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pages 206–212. IEEE, 2012.

[8] A. de Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio. Artificial neural networks applied to taxi destination prediction. *arXiv preprint arXiv:1508.00021*, 2015.

[9] V. Etter, M. Kafsi, and E. Kazemi. Been there, done that: What your mobility traces reveal about your behavior. In *Mobile Data Challenge by Nokia Workshop, in conjunction with Int. Conf. on Pervasive Computing*, number EPFL-CONF-178426, 2012.

[10] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, page 3. ACM, 2012.

[11] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[12] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *The Journal of Machine Learning Research*, 3:115–143, 2003.

[13] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[14] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[15] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552, 2009.

[16] J. Hamm, B. S. M. Belkin, and S. Dennis. Automatic annotation of daily activity from smartphone-based multisensory streams.

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

[18] H. Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*. GMD-Forschungszentrum Informationstechnik, 2002.

[19] D. Klein. Lagrange multipliers without permanent scarring. *University of California at Berkeley, Computer Science Division*, 2004.

[20] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[21] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.

[22] O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

[23] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[24] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson. Approaching the limit of predictability in human mobility. *Scientific reports*, 3, 2013.

[25] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[26] W. Mathew, R. Raposo, and B. Martins. Predicting future locations with hidden markov models. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 911–918. ACM, 2012.

[27] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 637–646. ACM, 2009.

[28] R. Montoliu, J. Blom, and D. Gatica-Perez. Discovering places of interest in everyday life from smartphone data. *Multimedia Tools and Applications*, 62(1):179–207, 2013.

[29] R. Montoliu and D. Gatica-Perez. Discovering human places of interest from multimodal mobile phone data. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, page 12. ACM, 2010.

[30] A. Ng. Generative learning algorithms, 2008.

[31] A. J. Nicholson and B. D. Noble. Breadcrumbs: forecasting mobile connectivity. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 46–57. ACM, 2008.

[32] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In *Data mining (ICDM), 2012 IEEE 12th international conference on*, pages 1038–1043. IEEE, 2012.

[33] U. S. B. of Labor Statistics. American time use survey. Downloaded from http://www.bls.gov/tus/home.htm, 2016.

[34] F. J. Ordóñez and D. Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

[35] A. G. Parlos, O. T. Rais, and A. F. Atiya. Multi-step-ahead prediction using dynamic recurrent neural networks. *Neural networks*, 13(7):765–786, 2000.

[36] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[37] R. Y. Rubinstein and D. P. Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[38] A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 723–732. ACM, 2012.

[39] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Pervasive computing*, pages 152–169. Springer, 2011.

[40] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. *ICWSM*, 11:329–336, 2011.

[41] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Conditional models for contextual human motion recognition. *Computer Vision and Image Understanding*, 104(2):210–220, 2006.

[42] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327:1018–1021,

2010.

[43] L. Song, D. Kotz, R. Jain, and X. He. Evaluating location predictors with extensive wi-fi mobility data. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1414–1424. IEEE, 2004.

[44] H. Steck and T. S. Jaakkola. On the dirichlet prior and bayesian regularization. In *Advances in neural information processing systems*, pages 697–704, 2002.

[45] D. L. Vail, M. M. Veloso, and J. D. Lafferty. Conditional random fields for activity recognition. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 235. ACM, 2007.

[46] Y. Wang, N. J. Yuan, D. Lian, L. Xu, X. Xie, E. Chen, and Y. Rui. Regularity and conformity: Location prediction using heterogeneous mobility data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1275–1284. ACM, 2015.

[47] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pages 1029–1038. ACM, 2010.

[48] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 312–321. ACM, 2008.