The following publication T. Kulshrestha, D. Saxena and R. Niyogi, "A Learning Based Human Interaction Modeling Using Mobile Sensing," 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2019, pp. 1049-1058 is available at https://dx.doi.org/10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00150.

A Learning based Human Interaction Modeling using Mobile Sensing

Tarun Kulshrestha
Department of CSE
IIT Roorkee, Roorkee, India
tarun.kul@ieee.org

Divya Saxena
University Research Facility in Big Data Analytics
The Hong Kong PolyU
Kowloon, Hong Kong
divya.saxena.2015@ieee.org

Rajdeep Niyogi
Department of CSE
IIT Roorkee, Roorkee, India
rajdpfec@iitr.ac.in

***Abstract*—Online social networks are emerging as a convenient platform where users build social relations with other individuals having similar interests, family/work background, etc. However, existing human interaction modeling is based on social graphs which are not more precise for friend suggestions in real-life. In this paper, we leverage the basic feats of deep learning for developing human interaction system, named *MyCompanion*, based on the user's lifestyle/activity information collected using the mobile crowd sensing. We collect a user's local knowledge, such as local information, ambient, and activity type, activity location and activity time. Then, the collected information is further aggregated and transferred to the deep learning enabled cloud server for user's daily schedule/activities analysis. We propose a schedule matching algorithm which finds the similarity among individuals' activities w.r.t. their activity type, activity time and activity location to recommend the most suitable friend(s) to the users. We develop a real-time testbed to perform a spatio-temporal analysis of the collected data from the users' smartphones. We also perform several experiments for evaluating the system performance. Our proof-of-concept prototype shows the usability of the proposed system.**

Keywords-Mobile Crowd Sensing*; *Crowdsourcing; Friend's Suggestion*; *Human Interactions; LSTM*; *Similarity Index*; *Social Networks*; *Recurrent Neural Networks

I. INTRODUCTION

In recent times, Smartphones have been widely accepted as a powerful and environmental sensing unit for collecting context and content-aware information, such as collection of large spatio-temporal data which allows finding the correlation between humans and locations. However, the spatio-temporal data operations are computationally and memory intensive in order to process the data in real-time [1]. People activities in the outdoor environments define more about their behavior and lifestyles. For example, individuals going for swimming and gym since long time can be more inclined for sports activities. People spending more time in libraries can infer the information that they are more interested in studies related events. People attending the same clubs and restaurants might have similar entertainments interests and preferences.

In addition, individuals sharing similar location traces and performing same activities in their daily life over a long/short period of time may have similar interests and lifestyles. The correlation among users' locations and activities can be used further for finding friends having similar lifestyles. Here, a location can be represented in absolute (latitude and longitude), symbolic (shopping mall, school, gym) and relative (100 meters north of the Space Needle) form. Social media apps have already worked a lot for suggesting friends based on the common friends between users. These friends mainly depend on graph analysis and preference similarity. One of the challenges faced by the existing social networking apps is the suggestion of a most suitable friend to a user having the high probability of being friends. Most of the existing social networking apps, like Facebook, suggests friends connected to the user's current friends.

On the other hand, according to the recent studies [2], [3], the best way of human interactions can be based on either they have same attitudes (level), or habits and/or lifestyle or tastes or moral standards or economic level or they have already some common friends. Existing social networking apps are using tastes and/or common friends features for friends suggestion. Suggesting a friend based on individuals' habits and lifestyles is the most intuitive option while it is not possible to capture it through web actions. Recently, there are some existing systems [4]–[6] for suggesting friends having the similar daily routines and activities to the user. However, none of the existing systems has exploited the power of deep learning algorithms for finding the similarity of users in terms of their daily routines, activities and their locations for suggesting an individual as a friend to the user in real-time.

Mobile crowd sensing (MCS) is an active area of interest and has been applied to a variety of human interaction related applications, such as social connectivity, user's common trajectories. Recently, a number of researches have been done for large-scale crowd sensing [7][8]. These researchers [7]–[9] use large-scale Smartphone-collected sensor data from users, including accelerometer, GPS, Wi-Fi, camera, Bluetooth, and microphone to explore social interactions. Furthermore, several researchers are keenly interested and exploring the usage of wireless probes emitted from the user's Smartphones, BT/BLE tags, Smartwatches, and wearable devices, etc. These probe data packets can be utilized for the analysis of several areas, such as human mobility behavior, social networking, security and privacy.

Moreover, probe requests with location information can be useful in user tracking and monitoring, device association history, user's physical closeness, and spatio-temporal user's behavior. Hong, et al. [10] performed a research work for analyzing the social behavior and interaction patterns of individuals using probe requests emitted by Smartphones.

In recent years, many researchers are working in the field of activity recognition using Smartphones, there is relatively less work for users' daily routines and lifestyle analysis using Smartphones. It will be the best option if Smartphones can be used for collecting the user's daily life activities and schedule with the activities' location. The app on the smartphone itself will be useful to suggest companion either among strangers or within a certain community if individuals share their daily routine/lifestyles. The system should have high accuracy, high coverage, scalability and handling of missing and noisy sensor data in order to accurately identifying human interactions and similarity among individuals.

To address the aforementioned challenges, in this paper, we model a real-time, smartphone-based human interaction system, named *MyCompanion* for suggesting friend(s) having the similar daily routine, activities and location history over the period of time. We collect the users' daily routine and activities with the activity location through users' smartphones and then we train a Long Short Term Memory (LSTM) [11] recurrent neural network (RNN) with the collected sequential localization data. We also handle security and privacy issues related to the user's daily activities data in the proposed system.

The reason for using LSTM is that the occurrence of activities consists of processes having high complexity and many dependent factors which are hard to analyze. There are obviously difficult nonlinear correlations among various activities, their locations and time, due to this traditional statistical and machine learning approaches are unable to analyze this process. Recently, deep learning algorithms, such as RNNs are capable enough to capture the nonlinear correlations among data.

We use the LSTM as it stores all the related information in a sequence to predict particular outputs. LSTMs are the state of the art sequence learning models that have been widely used for developing various applications, such as natural language processing [12] and unsegmented handwriting generation [13]. LSTM learns long-term dependencies through some gating mechanisms for storing the information. LSTM remembers who has performed a particular activity at what time and location. The input is the individuals' activities with the activities' locations over a period of time and other relevant information, such as weather, time of the day, day, week, etc., while the output is the schedule of the user for the coming days. This information is further passed to the similarity calculation module which finds the similarity index between the user and other people performing the same activities at the same time slots and at the same locations. Then, individuals having similarity index more than a threshold are suggested as a friend. The security and privacy issues related to the proposed system are handled carefully and explain in detail in section IV.

In summary, we have made following contributions in this paper.

- To the best of our knowledge, we are the first to leverage the basic feats of deep learning for developing human interaction system based on the user's lifestyle/activity information collected using the user's smartphone in real-time.
- We propose a schedule matching approach which finds the similarity among individuals' activities w.r.t. their activity type, activity time and activity location. Then, we use the calculated similarity to suggest the most suitable friend(s) to the users.
- We develop a real-time testbed to perform a spatio-temporal analysis of the real-time collected data from the users' smartphones. Our proof-of-concept prototype shows the usability of the proposed system. We also perform several experiments for evaluating the system performance.

The rest paper is organized as follows. In section II, we explain the related works. In Section III, we discuss the mathematical model related to our proposed approach. Section IV describes the proposed work and privacy concern, while Section V elaborates the experimental evaluations. In Section VI, performance results supported by thorough analyses are presented. Finally, Section VII concludes our paper and provides directions for future researchers.

II. RELATED WORKS

The MCS-based social applications in which users can share their sensed information among themselves for comparing their daily activities. Examples of which include, users can share their exercise data with the rest of the community, such as types of exercises, time for each type of exercises and timings. Some MCS-based social applications are, DietSense [14], and BikeNet [15]. In DietSense, users share their pictures about what they eat within a community for comparing their eating habits among themselves. On the other hand, in BikeNet, a rider can log the record of the location and bike route quality (air pollution on route, road conditions), and process the data to extracting the most favorable routes. Currently, with the progress of social networking, friend suggestion has achieved a great deal of consideration. Facebook, LinkedIn, and Twitter suggest friends to users on the basis of their social relations and common friends. Meanwhile, many researchers have proposed several friend suggestion mechanisms. The high availability of various location-acquisition technologies, such as GPS, Wi-Fi, Cellular facilitate users to include location data in the existing online social networks in a different way. For example, users can upload photos with location-tagging to a social networking service. Also, a stand-alone instant location of individuals can be represented into social networks, such as shopping mall at 6 PM. The user's location history over a certain time period in the real-world implies his interests and behavior. The main aim of modeling human interactions is to deeply understand the individual's behavior and interests. Individuals having similar location records are most likely to have same interests and behavior.

Eagle et al. [16] employed Bluetooth to infer the social interactions among users, whereas Bian, et al. [17] used individuals' personality matching based on the social information, connections, and contextual data of their physical interactions for recommendations. Kwon and Kim [18] proposed a friend suggestion system based on physical and social contexts to provide adaptive recommendations from enormous information. Yu, et al. [4] introduced a Cyber-Physical Social Network system based on the geographic location of people by incorporating the social network and GPS information in the real-world. Hsu, et al. [19] proposed the content-based suggestion using mutual interests and link structure information of the social network. Furthermore, Guo, et al. [20] presented a visual-based system to suggest friends based on the context of interests. They performed experiments on user's tagging behavior in music groups to explore and validate the system's results. Jiang, et al. [21] did mining of the enormous big data of social networks and suggest friendship groups and popular users based on the various social links. They used Map Reduce model for mining the social data to discover groups of frequently connected users.

On the other hand, Huang, et al. [22] presented the more precise friend recommendation technique which has two stages of processing. In the first stage, authors align various social networks and select possible friends through analyzing friendship information among users. In the next stage, authors build a topic model for refining the recommendation results by utilizing the relationship between users and image features. In [23], authors used the additional information with the social network data, such as similar interests, real-life location, and dwell time for more precise recommendations. In [5], a trajectory-matching prediction is introduced for friend recommendation in anonymous social networks. Authors make the future estimation about user's meeting based on their historical trajectory data and recommend friends by determining the similarity of their trajectories.

Further, location-based information navigation [24] is proposed for the group of friends where users are within the proximity are supposed to help each other through messaging and other information sharing. In [25] and [26], authors tried to mine location-based life routines from the large datasets of locations and find out the daily activities, such as staying at the hotel, going to home from market, etc. However, using only the location parameter for activity retrieval is not appropriate and enough. Farrahi and Gatica-Perez [27] overcame this issue in their research to find out daily activities of any user at the same place by combining location with physical proximity. Further, similar work was proposed in [28] in which topic model is used to mine activity patterns from sensor-generated data. In this research, authors used two wearables sensors instead of Smartphone for daily activity logging.

Wang, et al. [29] presented a semantic-based friend recommendation system, Friendbook, for social networks. The proposed system used Smartphones and the probabilistic topic model for discovering the lifestyles of users and finding the patterns from their daily routines instead of social graphs. Friendbook identifies the lifestyle of users through sensor-equipped Smartphones and measures the similarity of users' lifestyles. Authors proposed a similarity metric for measuring the similarity of users' lifestyles.

Overall, most of the existing works either focus on the user's preference similarity and common friends between users or rely on people`s proximity on the social graph for improving the suggestion quality. Some works considered user's lifestyle and location for more precise friend suggestion but not in real-time (based on existing social networking datasets). Our proposed approach is different from these existing systems. We model human interactions based on the user's daily routine, location and time of the activities for friend suggestion. We also leverage the deep learning for predicting the users' schedule on the collected data through users' Smartphones in real-time. Moreover, our deep learning based human interaction modeling supports incremental learning through involving online user feedback technique into the learning process to further improve suggestion accuracy.

III. MATHEMATICAL MODEL

In this section, we present the mathematical formulation of recurrent neural networks.

Recurrent Neural Networks

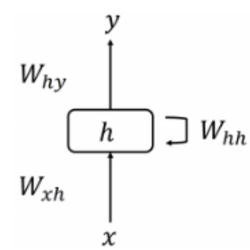


Figure 1. A recurrent neural network

RNNs are the powerful, most widely used and robust type of neural networks for processing the sequential time-series data. RNNs can predict the output based on the important information retrieved from the inputs while feed-forward neural networks determine the output on the basis of current input only. There are many possible cases where the predicted value of future output at time t+1 depends on the predicted output at the time t. RNNs are recurrent because they make the similar operations on each element of a sequence, where output depends on the past computations.

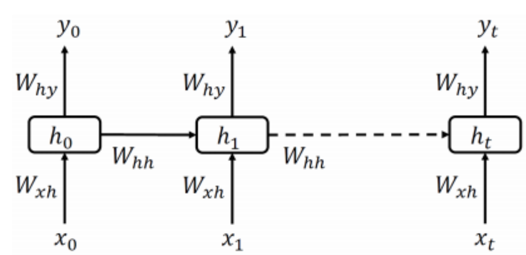


Figure 2. An unrolled recurrent neural network

Fig. 1 presents the structure of the typical RNN. At every time step, RNN takes x as an input, maintains hidden state h, and produces the y as an output. RNNs have memory for storing the previous data, however, it is possible to look back only a couple of steps. A loop passes the data from one-time step to the next time step. Fig. 2 demonstrates the unrolled network for a finite number of time steps where *Ws* are the common weights among various time steps.

An unrolled network shows how the data passes to the next steps and RNN is preferred for the sequence learning. The computation performed at every time step will be as follows: x_t is the input value at time step t, h_t refers the hidden state at time step t which is computed through the present input with a non-linearity application and the previously hidden state, such as *ReLU* or *tanh* and y_t refers the output value at time step t.

Weights W_{xh}, W_{hh} and W_{hy} are shared in the network at every unrolled time step which shows that the network performs the similar steps of computation at every time step, having different inputs x_t. As a result, it reduces parameters needed for the network. Moreover, it also prevents overfitting on small datasets. h_t is the most important feature of the RNN as it is the network memory and keeps capturing valuable information of the predictions in the earlier time steps.

Recently, to overcome the limitation of the RNN, Long Short Term Memory networks (LSTMs) are proposed by Hochreiter & Schmidhuber [11]. LSTM is a special kind of RNN, and it is one of the most popular deep learning models. Unlike RNNs, LSTMs are proficient in learning the long dependencies through the gating mechanism. We use the LSTM model for predicting the individual's schedule with the location from the large-scale continuous time-series human activity data

IV. PROPOSED WORK

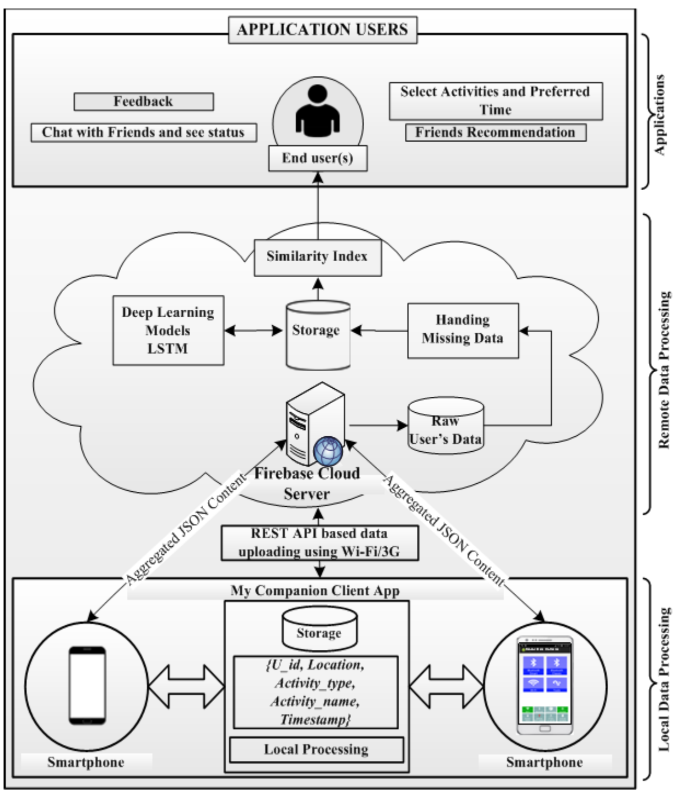


Figure 3. System architecture

In this section, we shall discuss our proposed system for suggesting a good friend based on the user's lifestyle.

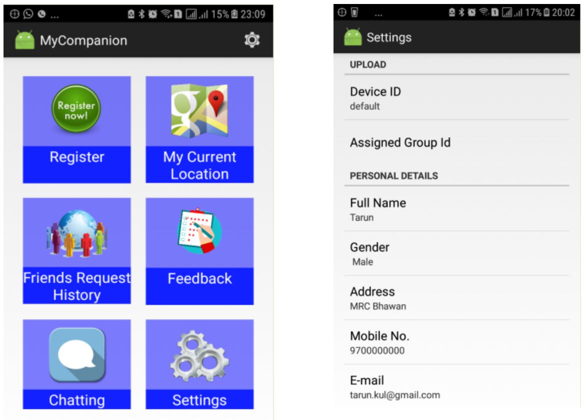
Fig. 3 shows the system architecture of the *MyCompanion*. Our proposed system keeps predicting the user's schedule and searches the persons who have the most similar schedule patterns over the period of time. We collect the users' activity type, time and location using their smartphones and upload the data to the cloud server at pre-specified intervals. At the cloud server, we use the deep learning scripts which run at regular intervals to update the schedule and also suggests new friends to users. The proposed system is divided into three modules: data collection, data processing and analysis, and similarity index calculation.

A. Data Collection

A User (U) is a person carrying a smartphone who can install an Android-based user app to register himself in the cloud server. In this app, at the very first stage, *Us* should register themselves (Name, Address, etc.) to the cloud server. The app keeps uploading the current location directly to the cloud server at the fixed intervals. The data logging and uploading interval is fixed to 15 minutes. The data uploaded by the *Us* on cloud server are stored in a time-stamped manner <*U_id*, *Location*, *Activity_type*, *Activity_name*, *Timestamp*> where *U_id* is user's name, *Location* captures the GPS coordinates of the *U. Location* comprises of *latitude* and *longitude. Activity_type* is the place visited by the user which represents the user's activity, *Activity_name* is the name of the location and *Timestamp* is the physical date and time at which a *U's* location uploaded to the cloud server. To have the continuous data in a particular sequence, we also perform data pre-processing and handle missing data.

Mining the daily routine and activities in real-time through *MyCompanion* app raises justifiable concern over users' privacy. However, the research done in this paper is conducted with human subject approval and users' consent. At the time of installing *MyCompanion*, we take permission/consent from the users. If a user does not agree to disclose his information to some particular users of *MyCompanion*, he can restrict/block those users. *MyCompanion* uses the activity location information for helping users to find friends in the nearby locations. To protect the users' location privacy, only the nearby friends' information is uploaded to the app as some users are sensitive to information leakage. On the other hand, *MyCompanion* protects suggested friends' privacy at their daily routine/activities level too. Instead of exactly showing the daily routine/lifestyle/activities of suggested friends, *MyCompanion* app shows only the similarity score (High, Medium, and Low) of suggested friends. Moreover, all 13 activities have been assigned the integer values for privacy/security of activities types. The activity type list contains *SBI ATM*, *SBI Bank*, *Gym near LBS ground*, *Hospital*, *Main library*, *Main building park and ground*, *KIH restaurant*, *Shopping mall* (MAC building), *Shops near gate number 8*, *Railway station booking center*, *Ravindra hostel*, *CSE department*, and *Cafeteria* (*CCD*) activities. The data stored in the cloud is encrypted using AES [30]. Thus, only the authenticated/admin users can access the results.

Currently, we have successfully tested and used this app in the IITR campus as well as outside areas.



(a) (b)

Figure 4. User-side app interfaces: (a) Features of *MyCompanion* app, (b) Profile interface after registration

The possibility of human-induced errors in the dataset is primarily when a user switches off the smartphone either explicitly or due to exhausting the phone battery. During this erroneous period, we apply the heuristic approach for handling the missing data. According to this approach, if a person location is not logged in the next time interval (i.e., after 15 minutes), we assume that the user is involved in the same activity for the next hour. After an hour, if data is not logged, we define it as ambiguous data. Fig. 4 shows the main user interface for *MyCompanion* and user profile interface after registration.

B. *Privacy Concern*

In MCS-based applications, there are several controversial questions related to privacy concern which are essential to address. Smartphones are sensing the surrounding data, including the individual's carrying smart devices. Continuous tracking of individuals' Smartphones can be used for user's location monitoring with the user's consent. There is a need for a strict privacy-preserving scheme (e.g., cryptography, privacy-preserving data mining) for sensitive information. In MCS, privacy and anonymity will always be a key issue for the foreseeable future. The most fundamental responsibility of MCS is respecting the user's privacy. Users should be reasonably sensitive regarding the data capturing and using.

The group sensing based applications deal with privacy by giving membership to group members only. E.g., Loopt and CenceMe [31] like social networking applications share locations, activities and other sensitive applications within a group in which users have trust with each other based on friendship or a shared common interest. Sensing and collecting data using community sensing applications can be at the risk of involuntary information leakage. Further, the risk of location-based attacks is fairly well explored in the previous researches. However, researches based on activity inferences, and social network data are in the nascent state. There are various examples concerning about reconstruction type of attacks in which the data that may appear harmless and safe to users can have invasive information.

C. *Data Processing and Analysis*

Every individual follows a pattern in his/her daily life which keeps changing from time to time. Therefore, we need to find out the features affecting the user's daily life schedule from the pre-processed data. There are many factors which affect the user's schedule calculation process. E.g., if we want to predict the activity of a user on 25th June 2018, at 16:00, Monday evening, then we must be known the user activity in past time like (a) user activity around 16:00 on past Mondays, i.e., 18th June 2018, 11th June 2018 and 4th June 2018. If that user went to 'Temple' on past 3 Mondays at that time, then there are high chances that user will repeat the same process on the next Monday, (b) user activity on previous days as if user is going to 'Temple' for past 3 days then most likely on the 4th day user may continue the same process, (c) user's today`s schedule till that time if user's daily schedule follows a pattern, i.e., if a user goes to 'Playground' early in the morning around 06:00, user does not go to 'Temple' in the evening at 16:00.

After finding the features affecting the user's schedule, we apply the deep learning (LSTM) based algorithm (see LSTM architecture in Fig. 5) for finding the most likely schedule of the user. In our case, the first layer of the LSTM generates 1024 outputs. After that, we use two dense layers which generate the probability distribution for the 13 classes (activities). We use Softmax function as our problem is of multiclass classification. Softmax function is used to "squash" a vector of *n* arbitrary real values *z* into a set of values that add up to 1, and which can be interpreted as probabilities.

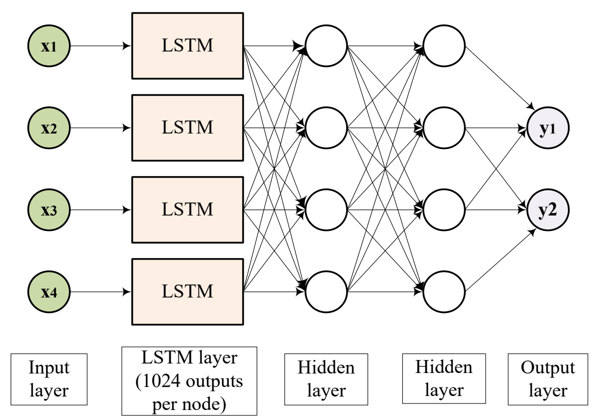


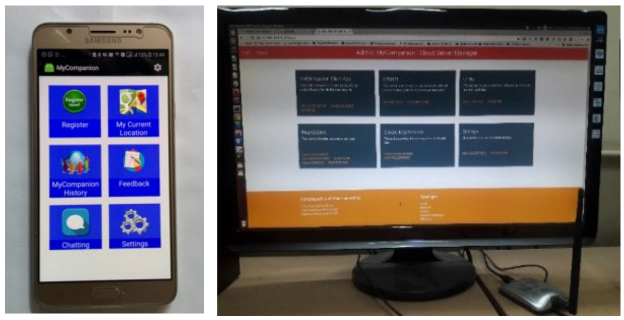
Figure 5. LSTM model architecture

$$softmax(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{n} e^{z_k}}$$

We use categorical cross-entropy as a loss function which is defined as,

$$L(x) = -\sum_{j=1}^{K} t_{i,j} \log\left(p_{i,j}\right)$$

where *p*, *t, i, and j* denotes the prediction, target, data point, and class, respectively.



(a) (b)

Figure 6 (a). Client app on smartphone, (b). Cloud server to store the client data

We use a Firebase [32], a real-time and cloud-hosted database for back end data storage for *MyCompanion* system. The captured records are stored in Firebase cloud and synchronized across connected users in real-time. The data uploading task is handled by a background service, which uploads files in the time-stamped order of creation. The stored records are deleted from the internal storage of Smartphones after uploading records to the cloud server. If uploading fails due to network unavailability, it retries to upload file when Wi-Fi or mobile data is available. The Android-based Client app and Linux-based cloud server are shown in Fig. 6 (a) and (b), respectively.

D. Similarity Index Calculation

After predicting the schedule of a user through LSTM, we need to find out the most suitable people who are having almost same schedule at the same time slots. We calculate a similarity matching index to match the predicted schedule with the other group of people schedule.

1) Similarity index calculation algorithm

In this sub-section, we present our algorithm for similarity index among users' schedule for *MyCompanion*.

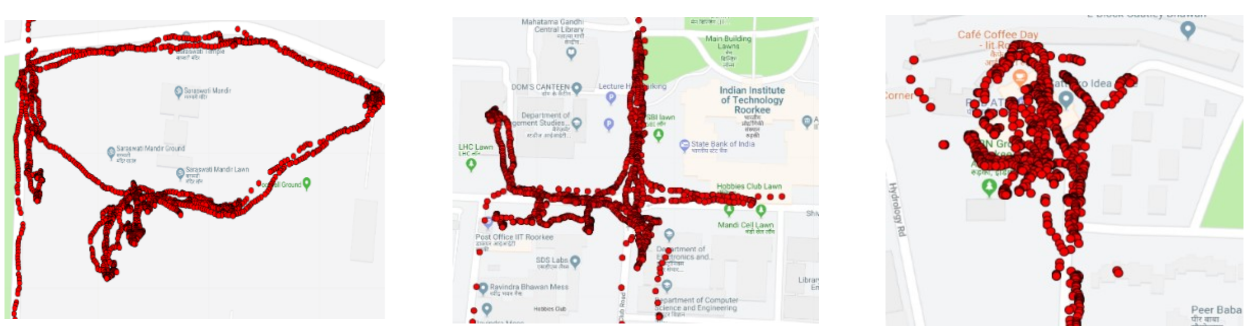
Algorithm 1 shows the similarity index calculation. First of all, the predicted schedule of users is merged with the activities' location as *MyCompanion* suggests friends who perform same activities at the same time and locations. After finding this information, we cluster them on the basis of activity type, activity time and activity location. Then, we assign a group_id (g_id) to each cluster.

Algorithm 1 Similarity Index Calculation

Input: Similarity_Index_Calc (*string A*, *string t, string l*)
Output: *vector* ***Result*** having total number of matching activities of other users.

$\boldsymbol{S_U}$ ← Schedule of users with activity *A,* time *t* and location *l*
Cluster the $\boldsymbol{S_U}$ for <*A*, *t*, *l*>
Assign a group id *g_id* to each cluster
Maintain a 2-dimensional matrix M <*t*, *user_id*> for *g_id*
for each $\boldsymbol{i} \in \boldsymbol{user\_id}$ do
  for each $\boldsymbol{j} \in \boldsymbol{user\_ids\_remaining}$ do
    for each $\boldsymbol{k} \in \boldsymbol{t}$ do
      if $\boldsymbol{M}[\boldsymbol{k}][\boldsymbol{i}] == \boldsymbol{M}[\boldsymbol{k}][\boldsymbol{j}]$
        $\boldsymbol{S_{index}}\ [\boldsymbol{k}][\boldsymbol{j}\text{-}1] = 1$
    end for
  end for
  $\boldsymbol{Result}\ [\boldsymbol{i}] = \boldsymbol{Sum\_of\_Col\_Elements}(\boldsymbol{S_{index}})$
end for
return ***Result***

We maintain a matrix of *g_ids* for each user and timestamp. We count the number of matching activities among users through the same *g_ids* and then store it in S_{index}. Then, we sum the matching activities using *Sum_of_Col_Elements*( ) and store it in the *Result* vector where high number shows higher activity correlation among users. When a *user_id* has count greater than a threshold value for other *user_id(s)*, then those *user_ids* are recommended to be friends.



(a) 21-June-2018 Thursday (b) 22-June-2018 Friday (c) 23-June-2018 Saturday

Figure 7. Location traces followed by User_1 for the particular days and time (17:30-19:30) @IITR (location detection rate: 5 mins)

V. Experimental Setup And Results

MyCompanion system is developed from the scratch. We have extensively tested the application in and around IIT Roorkee campus and shall predict the user schedule in the real-time. We also compare our proposed system with other exiting systems and show that our proposed system outperforms. We conduct experiments in IIT Roorkee campus. It is a reputed research and academic institute located in the state of Uttarakhand, India. The IITR campus has around 1.48 km^2 area housing many departments, hostels, administrative offices, libraries, shops, schools, etc.

We install our *MyCompanion* app on 50 users' smartphones and upload their activity information with the location on the cloud server for the period of six months. The selected users are mostly students whose daily life routine remains fixed during the academic sessions. The rate of data uploading is 15 minutes. The total collected records per user are 17,664. We preprocess the data and handle the missing data. The total number of features are 20, which includes user_id, activity, day, time, weather (sunny, rainy and normal), month of the year, day of the month, etc. The number of the activities (classes) associated with each record are 13. So, shape of the input matrix is (17664, 20, 13) for one user where batch and epoch size is 32 and 5, respectively.

A. Dataset Analysis

Fig. 7 (a), (b) and (c) show location traces of *User_1* on the map using client app. As already discussed in Section IV (A), each record is composed of the following fields <*U_id*, *Location*, *Activity_type*, *Activity_name*, *Timestamp*>. Data collection process is continuous and round-the-clock.

For analysing the user's activity pattern, we plot month/hour-wise heatmap of a user's visiting frequency at CSE department (see Fig. 8). Fig. 8 shows that the individual is not active during the time period of Jan. to Feb. 18. For finding the activity pattern in the department of CSE, we explore the details of 5 most visited users from the dataset as shown in Fig. 9. The result also depicts the spent time by those users in the month of April-June 2018. These results help us to find friends related to the *CSE department* activity.

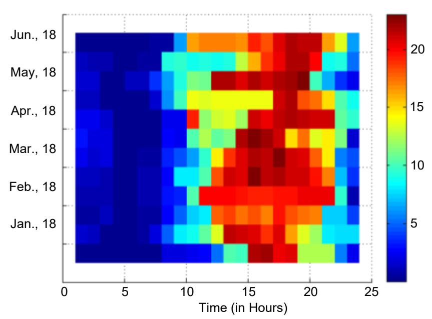


Figure 8. Month/hour-wise visiting frequency of a highly visited individual in Department of CSE

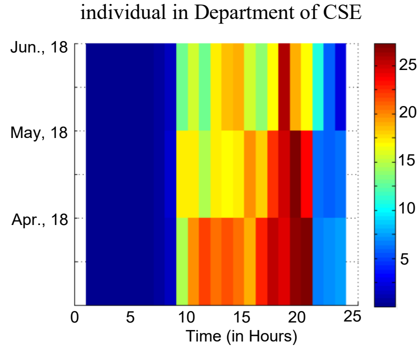
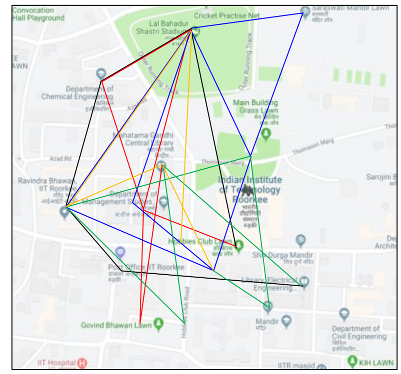


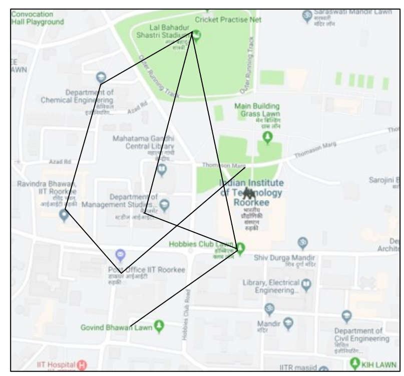
Figure 9. Month/hour-wise visiting frequency sequence of 5 individuals in the Department of CSE

B. Performance Results

We extend our analysis and plot a user's actual and estimated schedule on Google maps as shown in Fig. 10 (a) and 10 (b), respectively. In Fig. 10 (a), lines represent actual path of the user followed by him for past five Mondays, while we use different colors to represent different days. Fig. 10 (b) shows the user's estimated schedule for the next Monday.



(a) Actual schedule for past five Mondays (28 May - 30 June, 2018)



(b) Estimated schedule for next Monday (06 July, 2018)

Figure 10. Actual and estimated schedule of a User_1

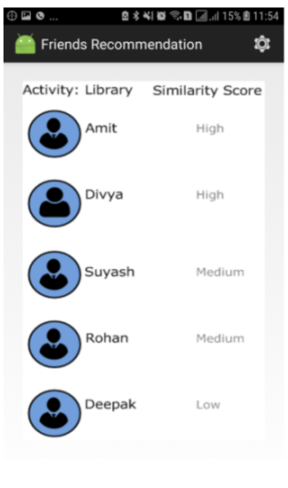
Evaluating and analyzing the location-based social network applications (e.g., community discovery, friend suggestion) are not trivial research topic due to the challenges of data, ground truth, and metrics. We compare our proposed model with Naïve statistics predictor and Fully Connected Feed-Forward Neural Network [33] which are widely used models available for prediction purpose. The naïve-based prediction depends on the mean of previous values at earlier timestamps. Whereas, Fully Connected Feed-Forward Neural Network Predictor is the simplest type of artificial neural network devised, in which connections between nodes do not have any loop or cycle. In a Feed-Forward network, information moves from the input nodes to the output nodes.

In our proposed model, we train our data for different users. Through results, we find that accuracy depends on the user's behaviour to some extent. The prediction accuracy of a disciplined person following his/her daily life schedule strictly have high accuracy while a non-disciplined person having low accuracy in comparison to the disciplined person. Overall, our LSTM based model outperforms naïve and FCFN w.r.t.

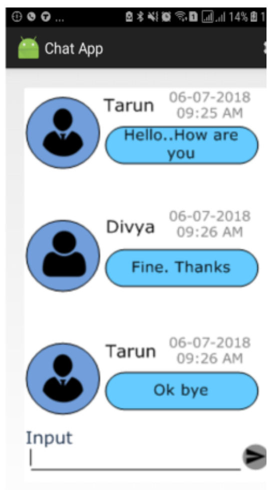
accuracy. The accuracy of our model is 92.8%. For verifying the ground truth of friends' suggestion of *MyCompanion*, we select 25 users out of 50 users, and instruct them to give the feedback against the generated results of suggested friends through our app.

Fig. 11 shows several user interfaces of *MyCompanion* app. Fig. 11 (a) shows a snapshot of the friend suggestion interface while the snapshot of chatting interface is shown in Fig. 11 (b). A user can connect to individuals through the list shown in the interface. First of all, users need to fill their personal information in the app, like Name, Photo, Mobile number, Address, etc.

We have also shown the similarity among ten users' schedule in Fig. 12. The light color represents high matching while high color represents low similarity in the schedules. Fig. 12 shows that user_id 2 schedule is highly correlated to user_id 8, user_id 2 also has a strong schedule matching with user_id 10. Similarly, user_id 7 has high schedule similarity with user_id 10.



(a) Friend suggestion interface



(b) Snapshot of chatting interface

Figure 11. User-side app interfaces

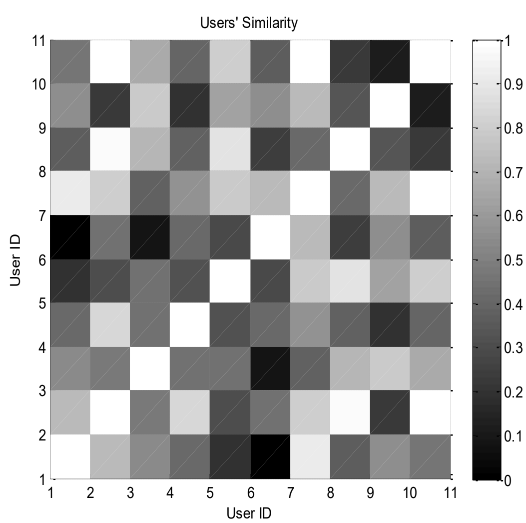
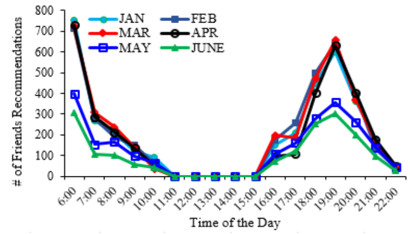
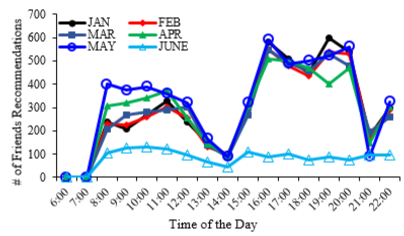


Figure 12. Similarity index among 10 users' schedule

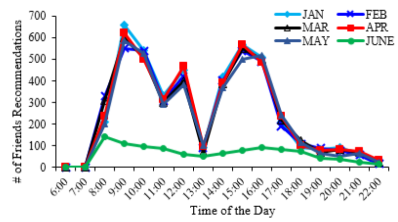
We extend our *MyCompanion* system analysis to find out how the proposed system suggests friends on the basis of only similar activities among users. Fig. 13 (a), (b), (c), and (d) show the results for the different activities (Exercise, Studying, Research, and Eating) and the total suggested friends for the period of six months (Jan. – June 2018). We disable the feature of accepting the friend's request for this analysis. We select only six activities out of 13. User_id 1 is getting the highest number of requests for activity '*Gym*' and '*Studying*'. Through this analysis, we find that our proposed system also works fine for recommending activity-specific friends.



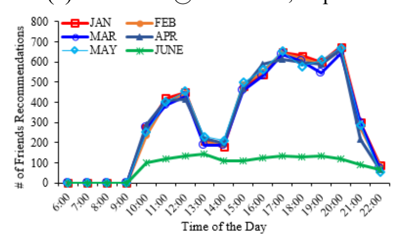
(a) Exercise@Gym



(b) Studying@Library



(c) Research@UGPC Lab, Dept. of CSE



(d) Eating@Restaurant and Canteen

Figure 13. Activity-specific Friend recommendation for User_id 1 for the period of Jan. – June, 2018 (Friend request acceptance feature is disabled)

VI. Power Consumption Analysis

We perform extensive analysis and evaluation of *MyCompanion* app through performance and power profiling application, called Trepn Profiler, Qualcomm [34]. We use three different types of Smartphones for experimental purpose: Samsung A50 (SA_50), Google Pixel 2 (GP_2), and OnePlus 5 (OP_5). Smartphones SA_50, GP_2, and OP_5 have 4000 mAh, 2700 mAh, and 3300 mAh battery capacities, respectively and have android versions v9.0 (Pie), v8.0 (Oreo), and v7.1.1 (Nougat), respectively.

Table 1. Different settings for the different smartphones

Settings	*SA_50*	*GP_2*	*OP_5*
Bluetooth state	Always on	Always on	Always on
GPS state	Always on	Always on	Always on
Display screen	Always on	Always on	Always on
Wi-Fi state	Always on	Always on	Always on
Profiling interval (milliseconds)	100	100	100
Profiling time (in min.)	15	15	15

Table 2. Comparison of *MyCompanion* with other applications

System Statistics	Nearify	Meetup	Nearby	My Companion
Battery Usage [%] per hour	2.0	1.89	1.91	**1.04**
Memory Usage [MB]	1.83	1.82	1.84	**1.32**
Power [uW]	56034.58	42383.3	38374.34	**22264.23**
CPU Load [%]	43.08	44.66	42.16	**36.25**
CPU Load Normalized [%]	6.76	6.08	5.36	**3.83**

We compare the power consumption, memory usage, and CPU load profiling of *MyCompanion* application w.r.t. the other similar types of crowd sensing applications, such as Meetup [35], Nearify [36], and Nearby [37]. After conducting several experiments to evaluate the optimal logging interval in terms of energy, memory, and CPU load, *MyCompanion* is modified to only log the environmental data once every 15 min to maintain the tradeoff between results accuracy and energy/resources consumption, providing at least 30 h of standby time under the user's normal use, such as sending emails/SMS, and making calls, etc. While continuous scanning may cause of rich depletion of energy and other resources with high accuracy in results.

Table 1 shows the settings for the power and performance profiling on the selected smartphones. Table 2 shows the performance comparison of *MyCompanion* with other three existing friends suggestion applications, Nearify, Meetup, and Nearby. The results show that *MyCompanion* outperforms existing android apps in terms of battery and memory usage, and CPU load. Moreover, the continuous usage of *MyCompanion* app drops the battery power at 0.76 % per hour only. *MyCompanion* consumes low mobile data, energy, CPU load, and memory usage, even though *MyCompanion* with GPS runs all the time due to its optimal logging interval. We calculate CPU Load (Normalized), Memory Usage [B], Battery Power [mW] (Raw), and Battery Remaining [%] for all three applications with the proposed *MyCompanion* app. Trepn Profiler collects 10 data samples per seconds (i.e., profiling interval 100 msec.). The captured records are stored in the form of CSV files for performance and energy analysis.

The four-performance metrics are defined below:

- *CPU Load (Normalized):* It is equivalent to the CPU usage w.r.t. the maximum potential of the CPU (i.e., CPU maximum frequency) for a particular Android application.
- *Memory Usage [MB]:* It refers to calculate memory usage in MegaBytes at different instances for a specific Android application.
- *Battery Power [mW] (Raw):* It estimates the used battery power in mW at different instances.
- *Battery Usage [%]:* It is for calculating the battery consumption rate/hour during specific application testing.

VII. Conclusion And Future Works

In this paper, we proposed a human interaction model, named, *MyCompanion* based on Long Short Term Memory-Recurrent Neural Networks for suggesting friends to users having similar lifestyles. By learning from historical users' daily routine and preferences data, our proposed system can predict the user's schedule and suggests friends accordingly. We collected records from the 50 users for six months in real-time to train the model. Further, they are stored and processed in the cloud server to find the user's visited location, and their working pattern/activity within a time span. Experimental results show that our prediction module can get a good accuracy of around 92.8% which is well commensurate with the high variation in the user's daily routine. We also performed power and performance profiling of the smartphones for the *MyCompanion* app. We perform real-time experiments on a proof-of-concept dataset to show the usability of the proposed system. In future, we shall explore our system at the large scale in different scenarios.

Acknowledgment

The third author was in part supported by a research grant from Google.

References

[1] U. Bellur, "On parallelizing large spatial queries using map-reduce," in *International Symposium on Web and Wireless Geographical Information Systems, Springer, Berlin, Heidelberg*, 2014, pp. 1–18.

[2] M. Tomlinson, "Lifestyle and Social Class," *European Sociological Review*, vol. 19, no. 1. pp. 97–111, 2003.

[3] G. Spaargaren and B. Van Vliet, "Lifestyles, consumption and the environment: The ecological modernization of domestic consumption," *Env. Polit.*, vol. 9, no. 1, pp. 50–76, 2000.

[4] F. Girardin, F. Dal Fiore, J. Blat, and C. Ratti, "Understanding of tourist dynamics from explicitly disclosed location information," *4th Int. Symp. LBS Telecartography*, vol. 58, 2007.

[5] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*, 2012, pp. 481–490.

[6] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha, "Understanding the coverage and scalability of place-centric crowdsensing," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13*, 2013, pp. 3–12.

[7] M. Azizyan and R. R. Choudhury, "SurroundSense: mobile phone localization via ambience fingerprinting," in *Proceedings of the ACM*

15th Annual International conference on Mobile computing and networking, 2009, vol. 13, no. 1, pp. 261–272.

[8] H. Hong, C. Luo, and M. C. Chan, "SocialProbe," in *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services - (MOBIQUITOUS)*, 2016, pp. 94–103.

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] I. Sutskever, O. Vinyals, and Q. V Le, "Sequence to sequence learning with neural networks," *Adv. Neural Inf. Process. Syst.*, pp. 3104–3112, 2014.

[11] A. Graves, "Generating Sequences With Recurrent Neural Networks," *arXiv Prepr. arXiv1308.0850*, pp. 1–43, 2013.

[12] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, "Image browsing, processing, and clustering for participatory sensing," in *Proceedings of the 4th workshop on Embedded networked sensors*, 2007, pp. 13–17.

[13] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The BikeNet mobile sensing system for cyclist experience mapping," in *Proceedings of the 5th international conference on Embedded networked sensor systems - SenSys '07*, 2007, pp. 87–101.

[14] N. Eagle, A. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proc. Natl. Acad. Sci.*, vol. 106, no. 36, pp. 15274–15278, Sep. 2009.

[15] L. Bian and H. Holtzman, "Online Friend Recommendation through Personality Matching and Collaborative Filtering," in *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, 2011, pp. 230–235.

[16] J. Kwon and S. Kim, "Friend recommendation method using physical and social context," *Int. J. Comput. Sci. Netw. Secur.*, vol. 10, no. 11, pp. 116–120, 2010.

[17] X. Yu, A. Pan, L. A. Tang, Z. Li, and J. Han, "Geo-friends recommendation in GPS-based cyber-physical social network," in *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2011, pp. 361–368.

[18] W. H. Hsu, A. L. King, M. S. R. Paradesi, T. Pydimarri, and T. Weninger, "Collaborative and Structural Recommendation of Friends using Weblog-based Social Network Analysis," in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, 2006, pp. 24–31.

[19] J. Guo, X. (Luke) Zhang, L. Wu, L. Gou, and F. You, "SFViz," in *Proceedings of the Visual Information Communication - International Symposium on - VINCI '11*, 2011, pp. 1–10.

[20] F. Jiang, C. K. S. Leung, A. G. M. Pazdor, and A. M. Peddle, "Big Data Mining of Social Networks for Friend Recommendation," *Concurr. Comput.*, vol. 28, no. 15, pp. 3994–4012, 2016.

[21] S. Huang, J. Zhang, D. Schonfeld, L. Wang, and X. S. Hua, "Two-Stage Friend Recommendation Based on Network Alignment and Series Expansion of Probabilistic Topic Model," *IEEE Trans. Multimed.*, vol. 19, no. 6, pp. 1314–1326, 2017.

[22] C. H. Chu, W. C. Wu, C. C. Wang, T. S. Chen, and J. J. Chen, "Friend recommendation for location-based mobile social networks," in *Proceedings of 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2013, pp. 365–370.

[23] Y. Duan *et al.*, "Trajectory-Matching Prediction for Friend Recommendation in Anonymous Social Networks," in *Proceedings of IEEE Global Communications Conference, GLOBECOM 2017*, 2018, vol. 2018-Janua, pp. 1–6.

[24] S. Fong, R. Luo, S. Deb, and S. M. Thampi, "Location-based mutual and mobile information navigation system: Lemmings," in *Intelligent Distributed Computing. Springer, Cham*, 2015, vol. 321, pp. 161–171.

[25] N. Eagle and A. (Sandy) Pentland, "Reality mining: sensing complex social systems," *Pers. Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, May 2006.

[26] K. Farrahi and D. Gatica-Perez, "Discovering routines from large-scale human locations using probabilistic topic models," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 1, pp. 1–27, 2011.

[27] K. Farrahi and D. Gatica-Perez, "Probabilistic mining of socio-geographic routines from mobile phone data," *IEEE J. Sel. Top. Signal Process.*, vol. 4, no. 4, pp. 746–755, 2010.

[28] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," *Proc. 10th Int. Conf. Ubiquitous Comput. (UbiComp '08)*, pp. 10–19, 2008.

[29] H. Qi, Z. Wang, Z. Wang, J. Liao, and Q. Cao, "Friendbook: A Semantic-Based Friend Recommendation System for Social Networks," *IEEE Trans. Mob. Comput.*, vol. 14, no. 3, pp. 538–551, 2014.

[30] A. U. Rahman, S. U. Miah, and S. Azad, "Advanced encryption standard," in *Practical Cryptography: Algorithms and Implementations Using C++*, 2014, pp. 91–126.

[31] E. Miluzzo *et al.*, "Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application," in *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, 2008, pp. 337–350.

[32] "Firebase: Real-time cloud-hosted database for mobile applications." [Online]. Available: https://firebase.google.com/docs. [Accessed: 26-Jun-2019].

[33] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *Int. J. Forecast.*, vol. 14, no. 1, pp. 35–62, 1998.

[34] "Trepn Power Profiler - Qualcomm Developer Network." [Online]. Available: https://developer.qualcomm.com/software/trepn-power-profiler/. [Accessed: 27-Jan-2019].

[35] "Meetup Android Application." [Online]. Available: https://www.meetup.com/topics/android-developers/. [Accessed: 17-Jan-2019].

[36] "Nearify - Discover Events/Friends Near You." [Online]. Available: https://www.nearify.com/. [Accessed: 17-Jan-2019].

[37] "Nearby | Friends recommendation Application." [Online]. Available: https://blog.wnmlive.com/. [Accessed: 17-Jan-2019].

.