Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7619102, 11 pages
https://doi.org/10.1155/2018/7619102

Research Article

BTP: A Bedtime Predicting Algorithm via Smartphone Screen Status

Kun Niu,[1] **Shubo Zhang**,[2] **Haizhen Jiao**,[1] **Cheng Cheng**,[1] **and Chao Wang**[1]

[1]*School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Software Engineering Architecture Design Department of Consumer Business Group, Huawei Technologies Co., Ltd., Beijing 100095, China*

Correspondence should be addressed to Kun Niu; niukun@bupt.edu.cn

Received 19 April 2018; Revised 17 August 2018; Accepted 8 October 2018; Published 22 October 2018

Guest Editor: Kok-Seng Wong

For smartphone service providers, it is of vital importance to recognize characteristics of customers. The process of recognizing these characteristics is generally referred to as user profile, which provides knowledge basis for business decisions, enables intelligent services, and brings unique competitiveness. As a basic component of user profile, bedtime could reflect lifestyle, health condition, and occupation of people. This paper presents a flexible algorithm named BTP (Bedtime Prediction), which is designed for predicting wake time and bedtime by analysing screen status of smartphone. BTP first collects screen status log data of user's smartphone and conducts preprocessing with a series of auxiliary user profiles. Then, it detects and records users' wake time and bedtime of one day by searching and combining major screen extinguish periods in the past 24 hours. Finally, BTP predicts future bedtime by matching current screen status sequence with all historical records. By applying BTP, most of night and morning scenario-based applications could provide more considerate services, rather than following fixed execution time like alarm clock. Experiments on practical applications prove that BTP can effectively predict wake time and bedtime without applying complicated machine learning algorithms or uploading data to server.

1. Introduction

From eras of keeping records by tying knots to big data, data analysis technique has been developing throughout the history of human civilization. With tide of mobile Internet, enterprises, and governments, even individuals have begun to exploit their own business to portable devices especially smartphones. At present, mobile Internet has already exceeded personal computer on number of users. To serve customers better, it is very important to identify their preferences and characteristics. Because of different life background, Internet business must take care of personalized needs. This way of identifying user characteristics is generally called user profile.

Accurate user behaviour prediction could help mobile Internet service providers optimize their business on extensive scenarios. For instance, system may preload application which has the highest probability of being used next moment to shorten users' waiting time. Getting up and going to bed are two critical incidents that normally indicate the start and end of human schedule and intelligent service as well. Besides, sleep pattern can reflect the health condition of a person [1, 2]. It is possible to unite sleep pattern with other characteristics such as age, gender, and working time to conclude subhealth. Furthermore, applications could give customized suggestions based on these conclusions. "Pzizz" [3] helps people fall asleep and "Sleep Cycle" [4] offers waking up service. Both these "lullaby" and alarm clock apps strictly depend on manual activation and detailed settings. It is meaningful to liberate users from repeated mechanical operations and make services more natural and flexible.

Smartphone, integrated with multiple sensors and chips, becomes a primary user data acquisition platform owing to tight connection with people. Many researchers have made researches and applications based on usage records from smartphones for modelling user profile. Reference [5] applied supervised learning methods to infer user profiles, such as religion, relationship status, spoken languages, countries of

interest, and whether the user is a parent of small children, by observing only a single snapshot of installed apps. Experiments on 200 smartphones show that for most traits the model can achieve over 90% precision. Reference [6] focused on analysing user profiles by mining usage information from multiple NFC-based applications, both on smartphone and server. Reference [7] firstly attempted to understand users through Service Set IDentifier (SSID) information. It also proposed a data cleaning and information enrichment framework for enabling user preference understanding based on the collected Wi-Fi logs. Reference [8] developed a method to predict the next app which improves home screen apps' usage experience.

This paper aims at solving two tasks. The first one is to detect getting up and going to bed incidents. The second one is to predict the next wake time and bedtime.

The issue of sleep study is put forward by medical care institutes at first. Hospital apply Polysomnography (PSG) [9, 10] to diagnose sleep disorders by records brains waves, oxygen level in blood, heart rate, and breathing, as well as eye and leg movements of participants. However, this method requires professional devices and specialized sleep environment, which is infeasible for daily use.

Wearable devices are the second choice. Devices are worn on the wrist, used to record movements and estimate sleep parameters which are called wrist actigraph (ACT) [11]. Although actigraph has been doubt the validity of identifying wake or sleep from a medical point of view, wearable device with fine identification algorithm is still fit for daily use owing to its low cost and portability. Reference [12] proposed using a wrist actigraph to estimate sleep time at an early stage. Reference [13] later developed a monitor system based on it and replacing manual calculation by scoring algorithms. And [14] further improved it, making the algorithms distinguish sleep from wakefulness in approximately 88% of minutes scored. In recent years, development of integrated circuit (IC) and Internet of things (IoT) enables miniaturization of sensors. Activity trackers like Fitbit wristband and smartwatches like Apple Watch make it possible to collect and analyse wrist movement data of users. Reference [15] employed an artificial neural network to classify sleep and wakefulness based on night wrist actigraph data. Reference [16] integrated unsupervised machine learning algorithms and domain knowledge heuristics to detect sleep or wake status.

Without extra device, there are also methods that utilize smartphone as a monitor and upload data to calculate sleep durations. Reference [17] presented a model named BES inferring sleep duration by analysing smartphone usage patterns. The features include external brightness, phone usage, motion status, and background noise. The collection of these attributes requires a resident monitoring application on smartphones. Reference [18] designed and implemented an application named Toss "N" Turn that can infer sleep and sleep quality by captured sensor data including sound amplitude, acceleration, light intensity and screen proximity, running apps, battery, and screen status. This paper verifies its validity by notify users to enter ground truth every morning.

However, there are three limitations of these methods listed above:

(1) Although wearable devices have developed rapidly in recent years, the market penetration rate is still limited. Worldwide shipments of wearable devices are 25.1 million, while there are 344.4 million smartphones (source: International Data Corporation, IDC). In addition, different manufactures would apply different types of sensors, and there is no unified API for developing software for all of them. This solution evidently limits intelligent services' coverage rate.

(2) Most of current methods demand transmitting data to web server and executing algorithms on server cluster. That would be involved in privacy protection issue. Uploading privacy data without users' explicit permission would cause severe punishment by law in many countries. General Data Protection Regulation (GDPR) regulated that data transfer requires user's consent in European Union. Besides, claiming various large amount data collection would cause disgust of users.

(3) The computational resources of smartphone are strictly limited. Massive and heavy computation would cause rapid loss of power, high occupancy of memory and kernel. Each of these factors would burden the running time of device and lead to decrease of user experience. Furthermore, computing environment on smartphone does not support machine learning especially deep learning algorithms very well.

For the second task, to the best of our knowledge, previous research is rare to be find. Literatures retrieved by keywords like "sleep time prediction" always focused on how to detect user's wake/sleep status or calculate sleep duration. In this paper, we classify them as solutions to task one, detection. Since intelligent services based on user profile and usage scenario just become popular recently, lack of study on predicting wake time and bedtime is reasonable, for example, Google released Awareness API on I/O developer conference, 2016 [19]. Awareness API focuses on enabling context-aware services by mining user habits. Alarm clock would not disturb your sleep if it learned when you fall asleep last night and next meeting. BTP takes a step forward, attempt to predict user's future action which is a valuable and novel work.

The contributions of our work are as follows:

(1) A lightweight predicting algorithm: BTP simply utilize data collected from smartphone without additional wearable devices. Screen status log data could be retrieved from smartphone operating system because screen light up and extinguish event are standardized system broadcasts. Besides, thanks to the ubiquity of smartphone, BTP can cover the widest potential user.

(2) Avoid the dilemma of collecting data without breaking privacy protection regulations. BTP is also not involved with web service or cloud computing. In other words, BTP plays a role of data processor rather than controller. Users would not worry about the risk of personal information disclosure.

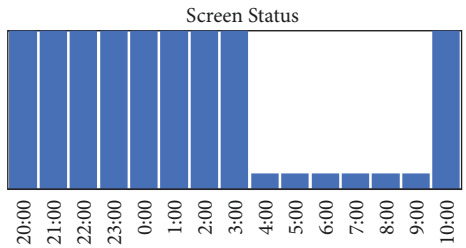

Figure 1: Back home late case (the user always goes to bed around 22:00).

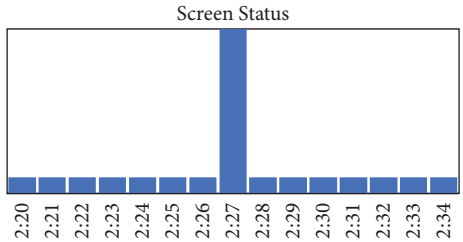

Figure 2: Unexpected interruption case (there is an incoming call at 2:27).

(3) BTP consume limited resource on smartphone. It does not need special framework or extra library. The calculation would not cause any appreciable delay. By our experiments, BTP performs acceptable complexity.

2. Ideas

2.1. Difficulties. Detection and prediction process of BTP may be crushed by many incidents. It is not enough to use screen status log data only. In actual projects and researches, we met the following difficulties.

The first is abnormal behaviour. Someone lives regularly, but back home very late or stay out all night occasionally. This case should be determined as outlier in data analysis theory. BTP contains models based on pattern knowledge, so these outliers must be detected and abandoned. Figure 1 demonstrates a typical outlier.

The second is unexpected interruption. When people are sleeping, incoming calls, receiving messages, and notifications from applications may light up screen, ring, or vibrate and then cause waking up. It is also possible that people fall into deep sleep and ignore these events. But for screen status log data, a continues screen extinguish period is interrupted by a strange event. The screen states of an unexpected interruption case is shown in Figure 2.

The third is shutting down of smartphone. If smartphone is shut down, both screen status and auxiliary events are missing. Some people do not like flight mode, they shut down smartphone at bedtime and start up it when getting up. But a worse case is, user may forget to charge the battery and then causing smartphone power off automatically. Losing log data is the biggest threaten and risk of BTP. Figure 3 shows the screen states of a shutting down case.

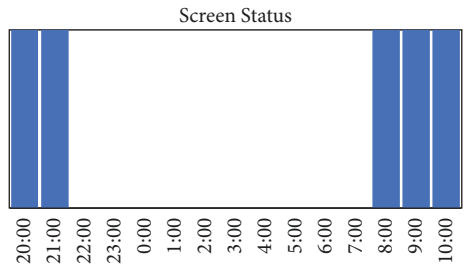

Figure 3: Shutting down case (the user shut down smartphone before 22:00 and starts up it after 8:00).

The above three difficulties are mentioned with detection, and the fourth is about prediction. Because of different life experience, regulations of wake time and bedtime between people are customized.

Most of people works at office or factory, that is, a stable target location. They show an identifiable wake-up time and bedtime distribution during weekdays. But many of them are freestyle on weekends. Others do not work from Monday to Friday. Because of their occupational characteristics, they exhibit strange distributions, such as going to work every two days or three days' work and one day's rest. That leads to big error in predicting with simple time series models or trying to describe these patterns by weekday and weekend frame. Figure 4 shows the patterns of weekdays and weedends.

2.2. Solutions. Besides difficulties, we can utilize two advantages to fix and improve our model. One is alarm clock event. For most people, they go to work at weekdays. Alarm clock ringing moment is also wake time. This rule has very high confidence and we can check accelerator activated event to confirm that. Figure 5 shows screen states when alarm clock event occures.

The other is accelerator activated event. When accelerator is activated, we know that smartphone is moved and the user is awakened. Figure 6 shows screen states when accelerator activated.

We overcome and reduce the effect of the four difficulties mentioned above by the following solutions.

(1) Abnormal Behaviour. BTP supposes that people sleep at home. It abandoned sequences not at home to avoid the impact of abnormal behaviour. Figure 7 shows screen states after abandoning abnormal behaviours.

(2) Unexpected Interruption. Sleep disturbances are resolved by merging adjacent periods of screen extinguish status. Here we define central sleep point, about 3 o'clock, the sleepiest time of the day. When people are waked by interruption event near central sleep point, they are likely to remain sleep in a few minutes. But when they are less sleepy, such as 5:00, people may get up directly. Figure 8 shows the case about connections two adjacant periods.

(3) Shutting Down of Smartphone. For the users who switch on and off regularly and reasonably, BTP regard the power on and off events as wake time and bedtime. But we treat

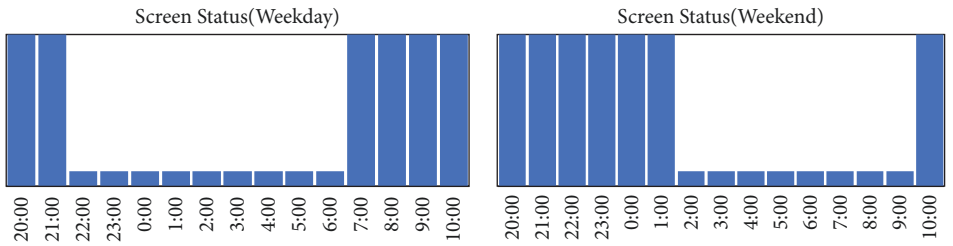

Figure 4: Weekday and weekend patterns (in weekend, go to bed and get up later than weekday).

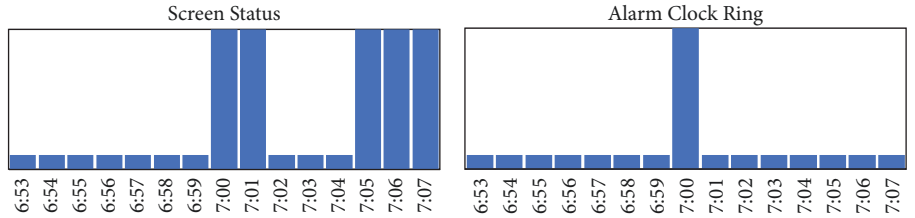

Figure 5: Alarm clock event (alarm rings at 7:00 and wakes the user).

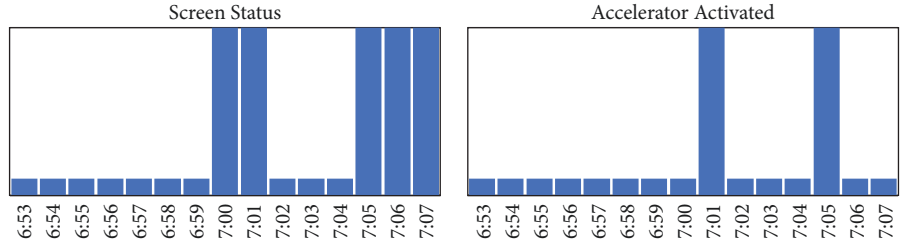

Figure 6: Accelerator activated event (the user touches screen at 7:01 to stops ringing and gets up at 7:05).

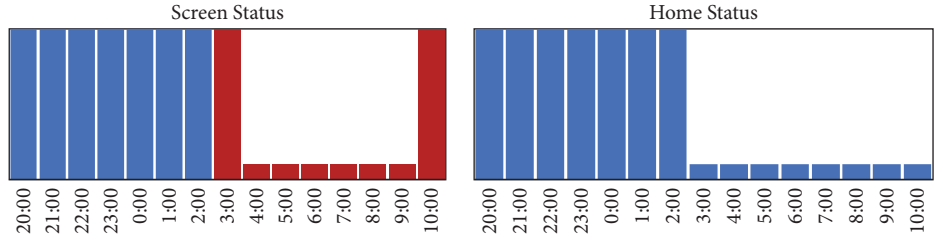

Figure 7: Abandon abnormal behaviour (the sequence after 3:00 is abandoned for going out home).

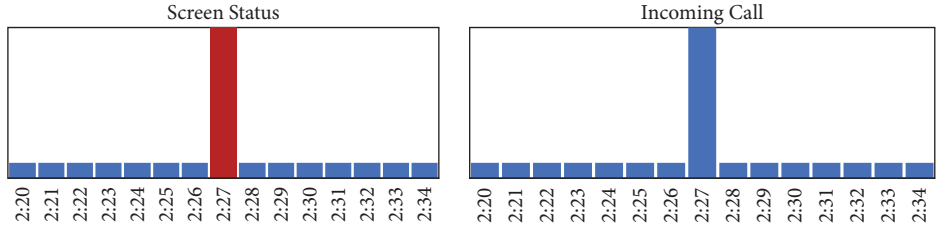

Figure 8: Connection of adjacent period (screen light up at 2:27 is ignored by BTP).

unregular power off event as outlier and abandon it. Figure 9 shows screen states when power off event occured.

(4) Customized Regulations between People. We used to think that user behaviour prediction should be based on weekday and weekend frame, but it is confused by customized weekday regulations, resulting in greater error. Then we changed our strategy and matched the current screen status sequence with other sequences of each past day. The most similar sequences represent the same behaviour pattern. Since waking up and

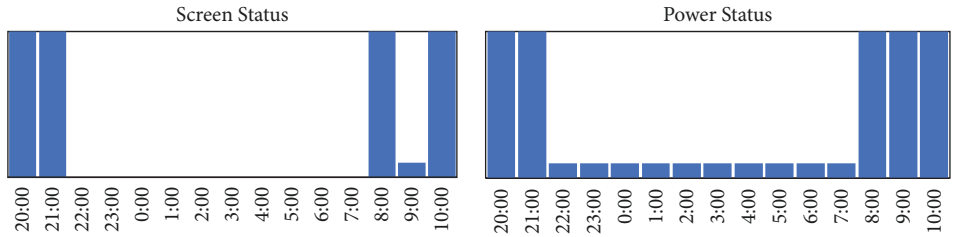

Figure 9: Reasonable power off event (when powering off, screen status is "null" but not "0").

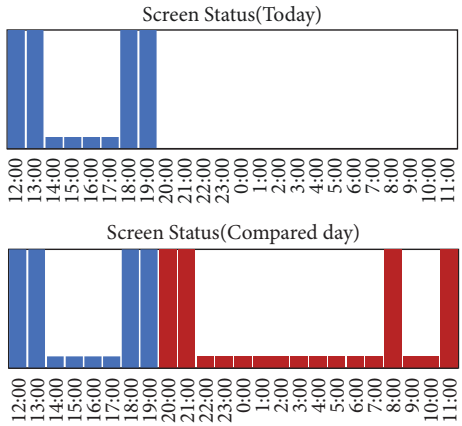

Figure 10: Similar behaviour pattern (next bedtime and wake time are predicted as 22:00 and 8:00).

going to bed are beginning and end of whole day activities, they are naturally reflected in this similar pattern system. Figure 10 shows two similar behaviour patterns.

To sum up, we effectively improving the accuracy and availability of screen status sequence in detecting and predicting wake-up time and bedtime by introducing a series of auxiliary events. The main process of BTP is shown in Figure 11. BTP firstly collects the user's screen status log data and other auxiliary events and then searches and connect main screen extinguish periods to get daily detection result. Then prediction is operated by matching current screen status sequence with the historical sequences like K-Nearest Neighbour.

3. Methods

3.1. Data Structure. Firstly, we define data structure. Log data contains information about both screen status and auxiliary factor. An event consists of date, moment, and event. Here is an example of JSON string.

```
{
 "Date": "2018-09-01",
 "Moment": "22:39:02",
 "Event": "SON"
}.
```

Table 1: Event code.

Target	Event Code	Meaning
Record screen status	SON	Screen light up
	SOFF	Screen extinguish
Filter period not at home	AH	Arrive at home
	LH	Leave home
Filter unexpected interrupt	IC	Incoming call
	RM	Receiving message
	AN	App notifications
Record alarm clock status	ACR	Alarm clock ring
Record power status	PON	Power on
	POFF	Power off
Decide awakened or not	AA	Accelerometer activated

Different events are distinguished by event codes. Here the code SON represents screen light up. All the event codes and corresponding meanings are shown in Table 1.

For task one, detection of wake time and bedtime, we input log dataset with all types of events from a smartphone throughout many days. Here is part of it.

```
{...
 {"Date": "2018-09-01", "Moment": "22:39:02", "Event":
  "SON"}, //Screen light on;
 {"Date": "2018-09-01", "Moment": "22:45:02","Event":
  "SOFF"}, //Scree extinguish;
 {"Date": "2018-09-01", "Moment": "22:49:50", "Event":
  "AA"}, //Pick out phone;
 {"Date": "2018-09-01", "Moment": "22:49:55", "Event":
  "SON"}, //Unlock phone;
 {"Date": "2018-09-01", "Moment": "22:50:01", "Event":
  "SOFF"}, //Screen extinguish;
 {"Date": "2018-09-01", "Moment":"22:50:02", "Event":
  "POFF"} //Phone shutdown.
...}.
```

The output of detection is as follows:

```
{...
 {"Date": "2018-09-01", "Bedtime": "22:39:02", "Wake
  time": "6:29:59"},
 {"Date": "2018-09-02", "Bedtime": "23:12:45", "Wake
  time": "6:40:04"},
```

Figure 11: Processing flow of BTP.

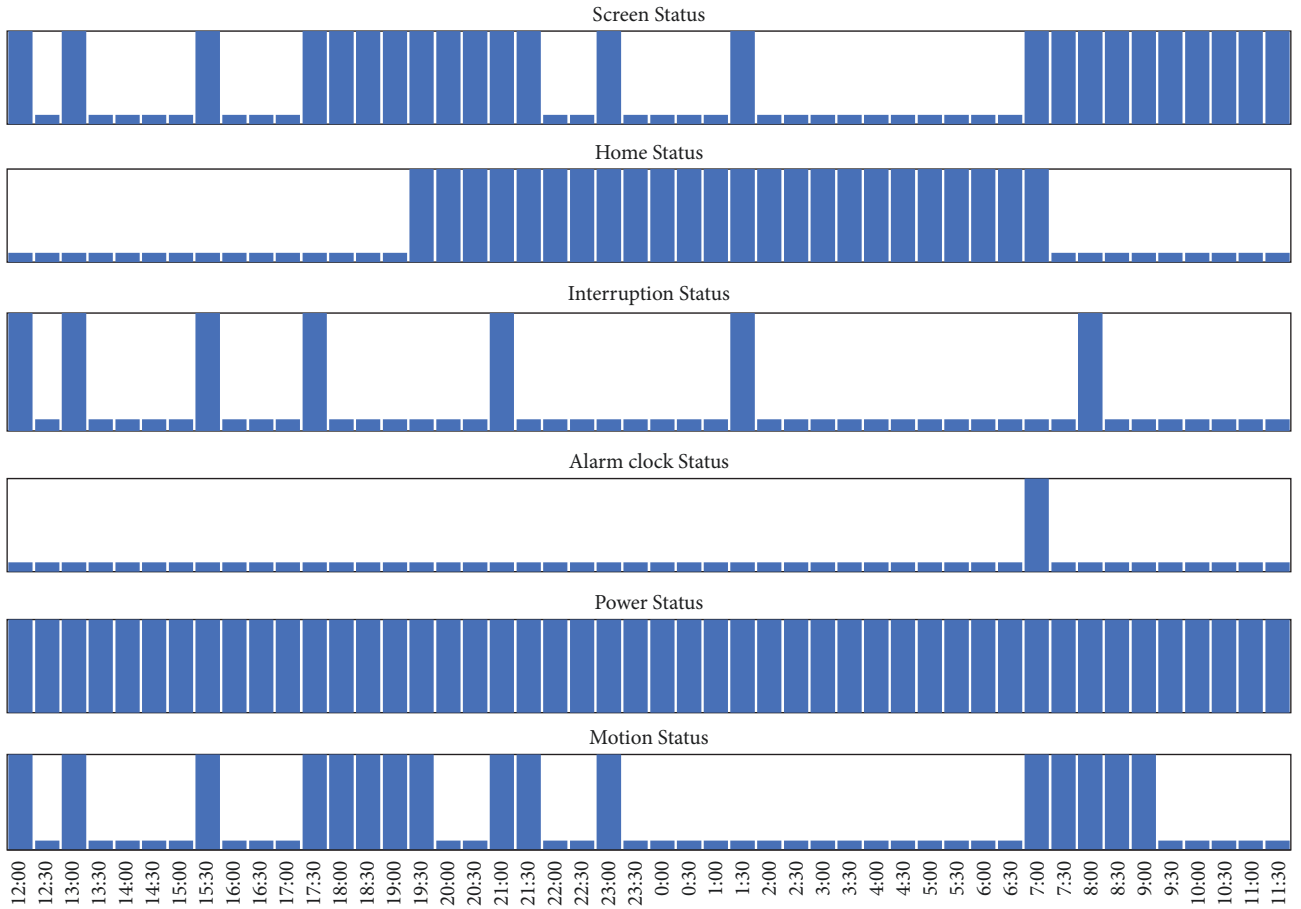

Figure 12: Whole day time-event sequences.

{"Date": "2018-09-03", "Bedtime": "22:59:12", "Wake time": "6:30:09"},
...}.

The human sleep pattern follows cycles of days generally. When discussing a single day, we must define the beginning of it. We set border point between two days at 12:00 instead of 0:00. Then we avoid interrupting continues sleeping that is also a continues screen extinguish period in our model.

For task two, prediction of wake time and bedtime, we also input log dataset as above. The output of prediction is as follows, if current time is "2018-09-10 18:00:00",

{"Date": "2018-09-10", "Bedtime": "22:40:15", "Wake time": "6:30:00"}.

Here wake time is referring to "2018-09-11 6:30:30" actually.

3.2. Data Collection and Preprocessing. Before running detection and prediction process, we must transform log data into several binary sequences. Considering that there are 1440 minutes in a day, we separate different events into six 1440-length sequences. For screen status, 0 represents screen extinguish and 1 represents screen light up. The sequences are shown in Table 2. For screen status and power status, we sampled the first second status of a minute. Yet for the other sequence, if there is any event occurred in the minute, the status will be 1.

Figure 12 demonstrates a timeline of a volunteer.

3.3. Method Process. BTP generally has two main phases, detection and prediction. Figure 13 shows the floatchart of detection.

In detection part, BTP follow four steps for each day to be detected.

Table 2: Sequence list.

Sequence	Coding Pattern	
	1/Event Code	0/Event Code
Screen Status	Screen light up/SON	Screen Extinguish/SOFF
Home Status	Arriving home/AH	Leave home/LH
Interruption Status	Incoming call/IC Receiving message/RM App notifications/AN	No interruption event
Alarm clock Status	Alarm clock ring/ACR	No ringing event
Power Status	Power on/PON	Power off/POFF
Motion Status	Accelerometer activated/AA	No activated

Table 3: Pseudocode of detection.

Series	Procedure	Description
1	for d_i in UndetectedDays = $\{d_1, \cdots, d_n\}$ do	
2	d_i.SSS = Filter(d_i.SSS)	Filter out screen status sequence that user is not at home
3	OffList = SelectLong (d_i.SSS)	Select continuous screen off sequences that last over 30 minutes
4	<sleep_time, wake_time> = JoinSeries(OffList)	Merge periods and get wake time and bedtime
5	SWTimeList.add(<sleep_time, wake_time>)	Record the results of consecutive days
6	end for	
7	end	

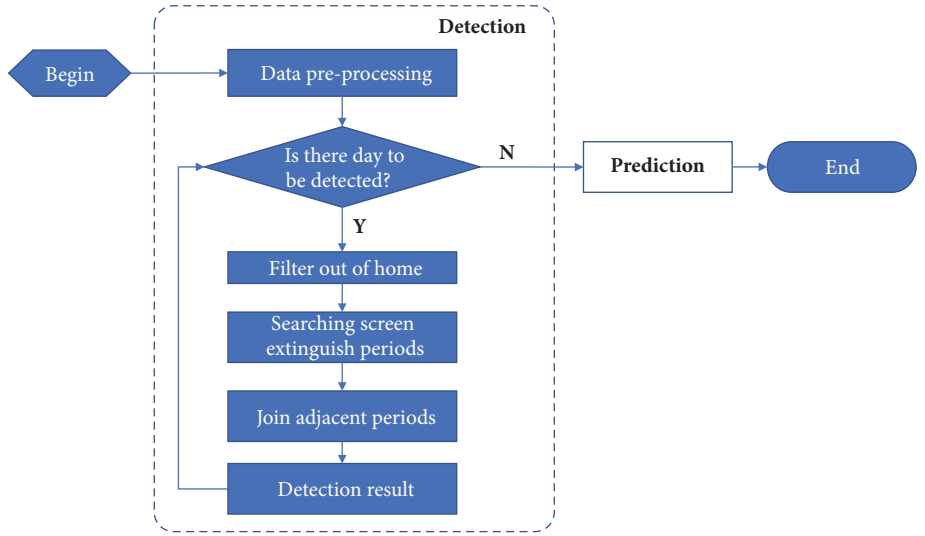

Figure 13: Flowchart of detection part.

(1) Filter out period of screen extinguish status when user is not home by searching home status sequence.

(2) Search screen status sequence and find out continues screen extinguish period which last over 30 minutes, then sort them in timeline.

(3) For each period selected by step 2, BTP uses (1) to determine whether this period can be united with its adjacent periods; that is, if time interval between them is shorter than T, we will conjunct the two adjacent sequences as one. In formula 2, T is a threshold that determine whether to join two sequences, l is used to adjust absolute value, t is minutes from 12:00 to middle of period, c is minutes from 12:00 to the time that people have deepest sleep, and off is an offset value.

$$T = l \times \frac{1}{1 + e^{|t-c|/60-off}} \quad (1)$$

(4) Adjust result with sequences of interruption status, alarm clock status, power status, and motion status.

Table 3 shows the pseudocode of detection.

After getting wake time and bedtime, as shown in Figure 14, predicting part has following steps:

Table 4: Pseudocode of prediction.

Series	Procedure	Description
1	for d_i in HistoryDays = $\{d_1, \cdots, d_n\}$ do	
2	SimList.add(CalSim(d, d_i))	Calculate similarities between today and history days
3	end for	
4	SelList = GetK(SimilarityList, k)	Select the K most similar sequences
5	Result = WA(SelList)	Calculate result Using weighted average
6	end	

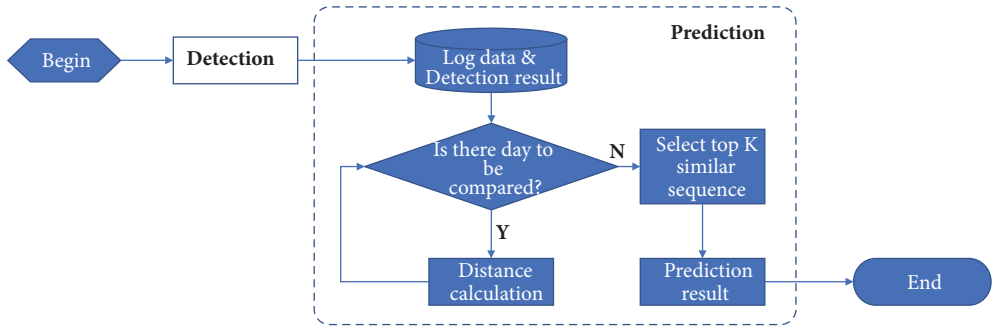

Figure 14: Flowchart of prediction part.

(1) Collect screen status sequence of today (from 0:00 to current time).

(2) Calculate similarities between today's screen sequence and each historical sequence using (2). Here, s is screen status sequence from 0:00 to current time, t is the sequence to be compared, and m is minutes from 0:00 to current time.

$$dist\,(s,t) = \sum_{i=1}^{m} |s_i - t_i| \tag{2}$$

(3) Select detection result according to the k most similar sequences and use (3) to generate result. *result* is prediction result, s_i is the i_{th} sequence of the k most similar sequences, and D_i is previous i_{th} detection result.

$$result = \sum_{i=1}^{k} \left[\frac{\sum_{j=1}^{k} \left(dist\left(s, s_j\right) - dist\left(s, s_i\right) \right)}{(\mathrm{k}-1) * \sum_{j=1}^{k} dist\left(s, s_j\right)} * D_i \right] \tag{3}$$

Table 4 shows the pseudocode of prediction.

4. Experimental Result

4.1. Experimental Background. The measurement criteria for detection is Mean Absolute Error (MAE), shown in (4). Here, n is number of people involved in this experiment, y_i is true value, and $f(x_i)$ is the detected value of methods to be tested.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - f\,(x_i)|}{n} \tag{4}$$

The measurement criteria for prediction are RMSE, which are shown in (5). Here N is number of samples. x_i represents true value, and x_i' is the predicted value.

$$RMSE = \sqrt{\frac{\sum_{1}^{N} \left(x_i - x_i'\right)^2}{N}} \tag{5}$$

Our dataset contains 30 volunteers' data. They installed the same data acquisition application we distributed on their smartphones. The application has four missions: collecting log data, detecting both wake time and bedtime, uploading original data and result to server, and fetching feedback. Here we upload data for debug and sign agreement with volunteers. The client app is resident in android and collects log data of system day and night. At the end of a day, usually around 0:00, it starts our detection method to detect when the user wakes up and goes to bed. It also regularly displays questionnaires and returns feedback to server. These questionnaires only have a simple question: When did you fall asleep last night and wake up today? To ensure that the volunteers answer questions carefully, some feedback which is always confused with prediction value is filtered after checking their original log data. After completing 13 days' continuous test, 39 volunteers accomplished experiments. But nine of them were filtered out for careless answers. During the experiment, we ran the phase one of BTP to detect wake time and bedtime and ran BES, proposed by Chen et.al. [17], to detect sleep duration. Then, we ran the phase two of BTP, moving average (MA) [20], and exponential smoothing (ES) [20] to predict future wake time and bedtime on the left 30 samples with 13 days records and then calculate RMSE.

All experiments were performed on volunteers' Huawei P10 smartphones with EMUI 8.0 (Android 8.0), Kirin 960

Table 5: Detection of sleep duration.

Task	Sleep Duration Detection	
Algorithm	BTP	BES
MAE	25.84	48.19

Table 6: Detection result of BTP.

Criteria	Wake Time	Bedtime	Sleep Duration
RMSE	0.41	0.75	0.81
E_{max}	170	286	282
E_{min}	0	0	0
MAE	10.56	19.09	25.84

Table 7: MAE of different c ($off = 3$, $l = 17$).

c	Wake Time	Bedtime	Sleep Duration
840/2:00	12.99	21.86	29.57
900/3:00	10.56	19.09	25.84
960/4:00	14.61	19.32	27.13

(4×2.4GHz+4×1.8GHz), 4GB RAM, and 64GB ROM. We used Android Studio 2.3.3 with Android SDK 7.0 and Java 1.8 to develop the data acquisition application, which collects and uploads data to server regularly. The server for receiving data runs a CentOS 6.5 environment coupled with Intel(R) Xeon(R) CPU 4×2.5GHz, 64GB RAM, 1200GB HD@15000rpm. To compare BTP and other algorithms, they were both executed on a workstation of Think Station P300 with Windows10 64bit, Intel i7@4×3.6GHz, 16GB RAM@1600MHz, SATA3 hard disc 2TB@7200rpm.

4.2. Result of Detection

4.2.1. Comparison. The BES algorithm we implemented has a MAE of 48.19 rather than 42.5 in Chen's paper because of different datasets. The comparison result is shown in Table 5. BTP has a much lower MAE than BES on the same practical dataset.

Table 6 further presents RMSE, Maximum Absolute Error (E_{max}), Minimum Absolute Error (E_{min}), and MAE of the detection part of BTP for showing its reliability. Since alarm clock rings regularly in weekday, wake time is more accurate than bedtime.

4.2.2. Parameter Discussion. There are three parameters in Eq. (1), c, off, and l. We use control variate method to carry out parameter tuning by using MAE as evolution metric. Table 7 shows that the best c is around 900. That means 3:00 is really the central point of people's sleeping.

In Tables 8 and 9, we found that the best choice is $off = 3$ and $l = 17$. The two parameters together decide how far we should join two adjacent screens extinguish period. A too small length will bring irrelevant periods and too big length will also make sleep duration over splitting.

Table 8: MAE of different off ($c = 900$, $l = 17$).

off	Wake Time	Bedtime	Sleep Duration
2	12.86	18.86	26.98
2.5	10.56	19.09	25.84
3	10.23	18.78	24.79

Table 9: MAE of different l ($off = 3$, $c = 900$).

l	Wake Time	Bedtime	Sleep Duration
13	15.60	24.39	34.35
15	12.15	20.32	28.24
17	10.56	19.09	25.84
19	12.00	9.06	27.67
21	18.03	29.35	38.54

Table 10: Time complexities of BTP and BES.

Execute Times	BTP (s)	BES (s)
10	6.03	6.61
20	12.70	13.52
40	25.51	26.90
80	50.00	53.87
160	101.01	107.25

4.2.3. Time Complexity. From Table 10, we prove that BTP and BES are both following a linear complexity. They are fast so we repeated them to record their time consuming. BTP does not need regression as BES but must endure heavier I/O consuming.

4.3. Result of Prediction

4.3.1. Comparison. We compare BTP with Moving Average (MA) and Exponential Smoothing (ES). MA and ES are two classic algorithms for time series prediction. The result is shown in Table 11. BTP outperforms MA and ES in all metrics because wake time and bedtime have periodicity and the cycles vary from person to person.

4.3.2. Parameter Discussion. We applied control variate method to identify the influences brought by different parameters in m in (2) and k in (3). Table 12 illustrates that, with the incensement of k, the result is better. It proves that assembly learning takes advantage of robustness. But we only have records of 13 days; k is limited.

Table 13 shows that accuracy increases with longer compared sequence. Here, $m = 1080$; the 75% length of a day strikes balance between accuracy and usability.

D represents the number of historical days we use for prediction. If we have more historical samples, the model will be more robust. Here $D = 6$ may meet an overfitting condition. The result of experiments with different D are shown in Table 14.

4.3.3. Time Complexity. The time complexities of BTP, MA, and ES are $O(n)$. In practical, MA and ES would execute faster

Table 11: Prediction results.

Status	Wake Time			Bedtime		
Algorithm	BTP	Moving average	Exponential smoothing	BTP	Moving average	Exponential smoothing
RMSE	0.34	0.47	0.70	0.52	0.97	0.70
E_{max}	61.03	84.25	141.40	72.20	149.75	86.43
E_{min}	0.76	1.16	0.79	0.36	1.67	1.13
MAE	9.26	21.86	34.23	24.15	47.04	34.79

Table 12: MAE of different k ($m = 1080$, $D = 12$).

k	Wake Time	Bedtime
1	6.23	24.65
3	9.26	24.15
5	6.66	23.55
7	5.47	21.81

Table 13: MAE of different m with ($k = 3$, $D = 12$).

m	Wake Time	Bedtime
360	13.17	27.89
720	12.89	26.20
1080	9.26	24.15
1440	9.25	23.63

Table 14: MAE of different D ($k = 3$, $m = 1080$).

D	Wake Time	Bedtime
6	11.64	23.58
9	17.86	29.73
12	9.72	24.16

because the size of data is rather smaller. BTP uses historical screen status sequence for matching similar pattern. Each prediction requires traversing $1440 \times T$, and T represents number of compared days. However, MA and ES only use historical wake time and bedtime. This difference is negligible when running in a smartphone instead of centralized calculation in server.

4.4. Survey. To collect feedback of volunteers, we design a survey for them. The questionnaire only has two questions:

(1) Did you fell asleep in ["detected bedtime"] last night?

   (a) Exactly or deviation is under 15mins (get score 1)
   (b) Deviation is in 15-30mins (get score 0.5)
   (c) Deviation is more than 30mins (get score 0)

(2) Did you get up in ["detected wake time"] this morning?

   (a) Exactly or deviation is under 15mins (get score 1)
   (b) Deviation is in 15-30mins (get score 0.5)
   (c) Deviation is more than 30mins (get score 0)

Table 15: Results of the survey.

Answers	Wake Time	Bedtime
a	25	9
b	3	14
c	2	7
Total Score	26.5	16

Results of our survey are shown in Table 15. It is quite relevant to MAE of algorithms.

5. Conclusions

This paper presented a bedtime predicting algorithm named BTP. The goal of BTP is to precisely detect and predict wake time and bedtime of a person by mining screen status log of smartphone. The implement of BTP is independent of any actigraph or cloud infrastructure, which makes it fit for popular application and satisfy privacy protection. Experiments on 13 days screen log data of 30 persons indicated that BTP can effectively accomplish detection and prediction tasks without introducing extra devices or uploading data. We also compared it with BES, another sleep detection algorithm, and BTP exceed BES in accuracy and time consuming. Furthermore, our work shows that BTP could be applied as a critical context-aware component to improve intelligent service on smartphone.

Data Availability

Experiments in this paper involve practical datasets. The practical dataset belongs to certain enterprise and is limited to science research and protected by confidentiality agreement. But researchers can develop their own app to record people's behaviour to test their work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Projects on International Scientific and Technological Cooperation under the National Key R&D Program (NKP) no. 2016YFE0204500, the National Natural Science Foundation of China under Grants no. 61671081 and no. 61720106007, the Beijing Natural Science Foundation under Grant no. 4172042, and the Fundamental

Research Funds for the Central Universities, Beijing University of Posts and Telecommunications 2017 Education and Teaching Reform Project no. 2017JY31.

References

[1] F. P. Cappuccio, F. M. Taggart, N.-B. Kandala et al., "Meta-analysis of short sleep duration and obesity in children and adults," *SLEEP*, vol. 31, no. 5, pp. 619–626, 2008.

[2] M. Kuppermann, D. P. Lubeck, P. D. Mazonson et al., "Sleep problems and their correlates in a working population," *Journal of General Internal Medicine*, vol. 10, no. 1, pp. 25–32, 1995.

[3] https://pzizz.com/.

[4] https://www.sleepcycle.com/.

[5] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti, "Predicting user traits from a snapshot of apps installed on a smartphone," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 2, pp. 1–8, 2014.

[6] A. Andersen and R. Karlsen, "User profiling through NFC interactions: Mining NFC-based user information from mobile devices and back-end systems," in *Proceedings of the 14th ACM International Symposium on Mobility Management and Wireless Access, MobiWac 2016*, pp. 173–176, Malta, November 2016.

[7] Y. Fan, Y. Chen, K. Tung, K. Wu, and A. L. Chen, "A framework for enabling user preference profiling through Wi-Fi logs," in *Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 1550-1551, Helsinki, Finland, May 2016.

[8] R. Baeza-Yates, F. Silvestri, D. Jiang, and B. Harrison, "Predicting the next app that you are going to use," in *Proceedings of the 8th ACM International Conference on Web Search and Data Mining, WSDM 2015*, pp. 285–294, China, February 2015.

[9] C. A. Kushida, M. R. Littner, T. Morgenthaler et al., "Practice parameters for the indications for polysomnography and related procedures: an update for 2005," *SLEEP*, vol. 28, no. 4, pp. 499–521, 2005.

[10] N. J. Douglas, S. Thomas, and M. A. Jan, "Clinical value of polysomnography," *The Lancet*, vol. 339, no. 8789, pp. 347–350, 1992.

[11] C. P. Pollak, W. W. Tryon, H. Nagaraja, and R. Dzwonczyk, "How accurately does wrist actigraphy identify the states of sleep and wakefulness?" *SLEEP*, vol. 24, no. 8, pp. 957–965, 2001.

[12] D. J. Mullaney, D. F. Kripke, and S. Messin, "Wrist-actigraphic estimation of sleep time," *SLEEP*, vol. 3, no. 1, pp. 83–92, 1980.

[13] J. B. Webster, D. F. Kripke, S. Messin, D. J. Mullaney, and G. Wyborney, "An activity-based sleep monitor system for ambulatory use," *SLEEP*, vol. 5, no. 4, pp. 389–399, 1982.

[14] R. J. Cole, D. F. Kripke, W. Gruen, D. J. Mullaney, and J. C. Gillin, "Automatic sleep/wake identification from wrist activity," *SLEEP*, vol. 15, no. 5, pp. 461–469, 1992.

[15] G. Orellana, C. M. Held, P. A. Estevez et al., "A balanced sleep/wakefulness classification method based on actigraphic data in adolescents," in *Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014*, pp. 4188–4191, USA, August 2014.

[16] A. Domingues, T. Paiva, and J. M. Sanches, "Sleep and wakefulness state detection in nocturnal actigraphy based on movement information," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 2, pp. 426–434, 2014.

[17] G. Yang, J. Chen, H. Tenhunen, and L. Zheng, "Intelligent electrode design for long-term ECG monitoring at home: Prototype design using FPAA and FPGA," in *Proceedings of the 3d International ICST Conference on Pervasive Computing Technologies for Healthcare*, London, UK, August 2009.

[18] J.-K. Min, A. Doryab, J. Wiese, S. Amini, J. Zimmerman, and J. I. Hong, "Toss 'N' turn: Smartphone as sleep and sleep quality detector," in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems, CHI 2014*, pp. 477–486, Canada, May 2014.

[19] https://developers.google.cn/awareness/.

[20] V. Kotu and B. Deshpande, *Predictive Analytics and Data Mining*, 2015.