# Looking Back on the Current Day: Interruptibility Prediction Using Daily Behavioral Features

**Minsoo Choy**[†] **Daehoon Kim**[†] **Jae-Gil Lee**[†*] **Heeyoung Kim**[‡] **Hiroshi Motoda**[§]
[†]Graduate School of Knowledge Service Engineering, KAIST
[‡]Department of Industrial and Systems Engineering, KAIST
[§]Institute of Scientific and Industrial Research, Osaka University
{minsoo.choy, daehoon.kim, jaegil, heeyoungkim}@kaist.ac.kr, motoda@ar.sanken.osaka-u.ac.jp

## ABSTRACT

When a person seeks another person's attention, it is of prime importance to assess how *interruptible* the other person is. Since smartphones are ubiquitously used as communication media these days, *interruptibility prediction* on smartphones has started to attract great interest from both academia and industry. Previous studies, in general, attempted to model interruptibility using the behaviors at the current moment and in the immediate past (e.g., 5 minutes before). However, a person's interruptibility at a certain moment is indeed affected by his/her preceding behaviors for several reasons. Motivated by this long-term effect, in this paper we propose a novel methodology of extracting features based on past behaviors from smartphone sensor data. The primary difference from previous studies is that we systematically consider a *longer history of up to a day* in addition to the current point and the immediate past. To represent behaviors in a day accurately and compactly, our methodology divides a day into multiple timeslots and then, for each timeslot, derives relevant features such as the temporal shapes of the time series of the sensor data. In order to verify the advantage of our methodology, we collected a data set of smartphone usage from 25 participants for four weeks and obtained a license to a large-scale public data set constructed from 907 users over approximately nine months. The experimental results on the two data sets show that *looking back to the beginning of the current day* improves prediction accuracy by up to 16% and 7%, respectively, compared with the baseline and state-of-the-art methods.

### ACM Classification Keywords

H.1.2. Models and Principles: User/Machine Systems; H.5.2. Information Interfaces and Presentation (e.g., HCI): User Interfaces

### Author Keywords

Human Interruptibility; Availability; Prediction; Data Mining; Machine Learning; Mobile Phone; Sensor

*Jae-Gil Lee is the corresponding author.



*UbiComp '16,* September 12-16, 2016, Heidelberg, Germany
© 2016 ACM. ISBN 978-1-4503-4461-6/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2971648.2971649

## INTRODUCTION

*Human interruptibility*, simply *interruptibility*, in general is defined by the degree of how opportune it is to interrupt a person [15]. The probability of replying to an instant message or checking a notification at a particular moment is a typical example of interruptibility. Then, *interruptibility prediction* is to assess another person's interruptibility *prior to* interaction with him/her [8, 14]. With accurate prediction, we can expect a quick and high-quality response to an interruption, and the cognitive burden of the person interrupted is reduced significantly [11, 18] . Thus, the importance of interruptibility prediction is being widely recognized since it is beneficial for both those who interrupt and those who are interrupted [30].

Interruptibility prediction has been extensively studied in various scenarios: office environments [8, 14], desktop computers [12, 13], and mobile devices [19, 21, 22, 23, 24, 25, 28]. In particular, owing to the prevalence of mobile devices [4, 26, 28], huge amounts of research effort are currently being devoted to interruptibility prediction on mobile devices—smartphones. Previous studies have demonstrated that interruptibility can be predicted fairly well (with an accuracy of over 70%) using various types of context information.

One of the active topics in this direction is to determine what data to capture to represent the *current context* [20, 30], because the advances in ubiquitous sensing technology provide us with abundant contextual data. Regardless of data sources, one dominating assumption is that the current context can best be modeled by the observations obtained *at that exact point in time*. That is, previous studies attempted to represent interruptibility mostly using the "present-time" features captured at the specific moment. Examples of these features include the current ringer mode and the current screen on/off status. In addition to these features, more recent studies have started considering users' *past* behaviors, such as events occurring in the previous one or five minutes [8, 14, 25] and the time elapsed since the last event [21, 22, 23]. This is reasonable because the consequences of past behaviors and history are part of the current context [9]. Nonetheless, these studies still do not consider past behaviors extensively since they reflect only the *immediate* past.

### Methodology

In this paper, we tackle the problem of systematically incorporating *past behaviors* into interruptibility prediction. Differing from existing research that considers only the present time and the immediate past, our methodology considers a longer history of up to one day. Here, we contend that proper consideration of past behaviors plays a key role in accurate prediction. The intuition behind our methodology is two-fold as shown below (see “Study Design" for the details).

- **Self-regulation** or **conservation of energy**: People consciously manage or guide their own thoughts and behaviors [16]. In addition, the amount of human activity *per day* is in fact limited and conserved [33]. Thus, for example, if a person did not concentrate on work in the morning, the person would probably work harder in the afternoon to finish a planned task within the day.
- **Prolongation**: The effect of an event could last for a long time [1]. Hence, for example, if a phone call to someone makes a caller feel relieved, the caller is more willing to do a favor after the phone call during the entire day.

Improving interruptibility prediction based on past behaviors is challenging. First, it is important to determine how far back we need to look. We empirically verify that looking back on the current day is sufficient to achieve the best result. Furthermore, since a temporal window is relatively long (i.e., from several hours to a day), a novel approach to feature extraction from smartphone usage data is needed for effective prediction. We carefully derive relevant features that include the statistical measures, value distributions, and temporal shapes of the time series of smartphone sensor data.

Figure 1 shows the concept of our methodology. In addition to the features extracted from the present time and the immediate past, those extracted *from the current moment back to the beginning of a day* are provided to the feature selection module. Many more features are derived from the *today* window than from the current point and the immediate-past window because of its longer duration. Finally, only the discriminative features resulting from the feature selection module are used for training and prediction. Here, the interruptibility at the current moment is “predicted” using the sensor data that has been collected in the past. These training and prediction are performed individually for each user.

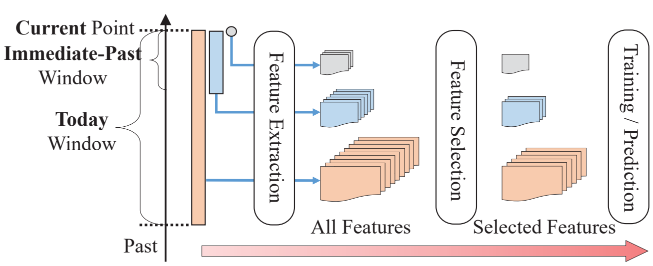



**Figure 1. The main concept of our methodology.**

We note that our work is orthogonal to existing work that explores predictive data sources (e.g., [17]). Our sophisticated design supports any time-series data of numeric, binary, and nominal variables. Thus, given a set of attributes, we attempt to maximize the benefits of those attributes by harnessing daily behaviors.

### Contribution

Overall, the contributions of this paper are summarized as follows. First, we propose a novel methodology of systematically extracting *daily* features in support of interruptibility prediction. Second, in order to verify the advantage of our methodology, we collect smartphone usage data from 25 participants for four weeks as a field study. Third, by extensive experiments, we confirm that our methodology of looking back on the current day achieves the highest prediction accuracy and improves accuracy by up to 16% compared with the baseline of using only the present-time features and by up to 7% compared with the state-of-the-art methods [8, 14, 25]. Interestingly, the features generated *a few hours previously* are shown to be often more influential than those generated at that moment.

## RELATED WORK

In this section, we briefly review the state-of-the-art related work. Pejovic et al. [20] provided a broad overview of mobile sensing and context prediction. The authors, without confining to interruptibility prediction, presented a survey of human activities that can be predicted using their mobile phone usages as well as machine learning techniques used for such predictions. Notably, Turner et al. [30] presented an extensive survey on interruptibility prediction in three dimensions: scenarios for interruptibility, data collection, and prediction. Thus, we present a rather *focused* survey on feature extraction for past behaviors that can be found in some recent studies, though they do not fully exploit past behaviors.

Hudson et al. [14] and Fogarty et al. [8] studied interruptibility prediction in office environments. The authors coded the data recorded by the cameras in the offices and logged 23 events or situations, e.g., speaking, writing, sitting, standing, or on the phone, to act as simulated sensors. The entire duration was divided to 15 second intervals. Then, the authors derived the features based on the intervals that belonged to the past 30 seconds, 1 minute, 2 minutes, and 5 minutes, respectively. Among these *derived* features, whether a talk event occurred in any interval during the past 30 seconds was turned out to be very predictive. This work achieved the accuracy of up to 82.4%. Sarker et al. [25] studied availability prediction for just-in-time interventions (JITI) in health monitoring applications. Various data were collected from both wearable wireless sensors and smartphones. As for derived features, the authors adopted the same convention [8, 14]. This work achieved the accuracy of up to 77.9%.

Pielot et al. [21, 22, 23] studied interruptibility prediction on mobile devices using various definitions of interruptibility: call availability [21], instant message viewing [22], and boredom [23]. These studies commonly include the features that represent the time since the last event (e.g., screen on and call). Many of these features, such as the time since ringer change, the time since last screen on, and the time since the last outgoing call, were ranked at the top among important features. The time since the last event, however, tends to be short since most events in consideration happen frequently. This series of work achieved the accuracy of up to 83.2% (call availability), 70.6% (instant message viewing), and 82.9% (boredom).

Pejovic and Musolesi [19] designed and implemented InterruptMe, which is an interruption management library for Android smartphones. In order to predict responses to notifications, the authors basically used the present-time features. The precision and recall were 0.64 and 0.41 respectively. The average response time was reduced from 22 minutes to 12 minutes when using InterruptMe compared with random notifications. Then, the authors attempted to find out the difference between interruptible and non-interruptible moments, in terms of the mean and standard deviation of an attribute for the entire duration of the experiment (two weeks), but they did not succeed. Though the previous days up to two weeks were considered, the mean and standard deviation of a very long duration are too simplified features to have predictive power.

Overall, to the best of our knowledge, there is no existing work that considers a *sufficiently long* duration of the past with keeping the *predictive power* for interruptibility prediction

## STUDY DESIGN

In this section, we explain the theory and rationale behind the design of our methodology and propose our research questions that will be explored in this paper.

### Why Looking Back on the Past?

The features based on the immediate past are important for interruptibility prediction, as witnessed by previous studies [8, 14, 21, 22, 23, 25]. We would like to reconfirm the benefit of the immediate past (**RQ1**). More importantly, a relatively distant past (i.e., the previous several hours) impacts greatly on interruptibility at the current moment. Though there could be many reasons, we believe that (i) *self-regulation* or *conservation of energy* and (ii) *prolongation* explain a significant proportion of the phenomenon.

#### *Self-Regulation or Conservation of Energy*

People, for goal setting and goal striving, plan and execute actions that promote goal attainment as well as shield those goals from distraction and disruption [16]. With the popularity of the books "Seven Habits of Highly Effective People" by Stephen Covey and "The Effective Executive" by Peter F. Drucker, people are aware of the importance of time management. In order to reduce wasted time, people keep examining time usage up to that moment and adjusting their approaches to goal setting and goal striving. Consequently, people consciously regulate their own thoughts and behaviors based on the past.

It turns out that, even *without conscious management*, the energies that humans consume per day are limited and conserved [33], just like the law of conservation of energy. To support this clam, Yano [33] measured the frequency of arm movements in a one-minute interval using wearable sensors. The distribution of the frequencies in a day followed a similar exponential distribution across all participants. Based on *unconscious prioritization*, for example, a person suppresses his/her activities in the morning and focuses on a business meeting, which requires more energies, in the afternoon. That is, if high frequency intervals did not appear yet, high frequency intervals will eventually appear afterward.

#### *Prolongation*

Mood, which is normally a reaction to a cumulative sequence of events, is prolonged, *not* instantaneous [1]. Mood could last all day or longer. It is apparent that mood affects interruptibility. Turner et al. [30] categorized the factors of interruptibility into physiological ability, cognitive affect, and user sentiment. In addition, *intentional* unavailability is wanting not to be interrupted because a person is not in the right mood. Salovaara et al. [24] reported that this type of unavailability took 35% of all unavailable moments. Thus, if an event affecting a person's mood happened, its effect could last for a long time.

### How Far Back to Look?

Many evidences that humans have a *daily* routine can be found in the literature. Circadian rhythm also supports the daily routineness. That is, humans' behaviors are highly periodic where the periodicity is a *day*. Song et al. [27] collected mobility patterns (trajectories) from 50,000 individuals for three months and found a 93% potential predictability in user mobility across the whole user base. The authors concluded that, despite humans' deep-rooted desire for change and spontaneity, humans' daily mobility is, in fact, characterized by a deep-rooted regularity. Also, Yano [33] showed that the frequencies of arm movements were very similar at the same time of day during the entire year.

Putting the pieces together, we expect that we need to dwell on the period from the current moment back to the beginning of a day since our routine starts with a day (**RQ2**). Then, since daily patterns are repetitive, it will *not* be very helpful to look further back beyond that day if the data on the day is available (**RQ3**). Last, for the same reason, the behavior done one or two days before the target day can be a good substitute for that done on the target day (**RQ4**). Here, both days should be altogether on the weekday or at the weekend.

### Research Questions

We now summarize four research questions as follows.

- **RQ1**: Interruptibility is affected by the immediate past behavior as well as the current status.
- **RQ2**: The accuracy of interruptibility prediction improves significantly when using the behavior of the current day.
- **RQ3**: Looking further back beyond the current day is not very helpful for interruptibility prediction if the data on the current day is available.
- **RQ4**: The behavior of the target day on weekdays can be replaced with that of a preceding day on weekdays without reducing much accuracy.

## INTERRUPTIBILITY DATA SETS

We used two real-world smartphone usage data sets: the KAIST data set and the Device Analyzer data set [31]. The former is our proprietary data set, and the latter is a public data set. Especially, the Device Analyzer data set is known to be the largest collection of smartphone usage data, and we extracted 907 users for use in our experiment in the order of the number of instances recorded. In both data sets, hour of day and day of week were attached to every recording. Interruptibility is modeled as a *binary* state as typically done by recent studies [19, 22, 23, 25]. Table 1 shows the general statistics of the two data sets. The first column represents the number of attributes which will be detailed in Tables 2 and 3 respectively. The total number of interruptibility labels (interruptible or non-interruptible) is reported in the last column.

| Data Set | # Attrs. | # Users | # Labels |
|---|---|---|---|
| KAIST | 24 | 25 | 4,103 |
| Device Analyzer | 26 | 907 | 1,646,066 |

**Table 1. Statistics of the two data sets.**

## KAIST Data Set

### *Participants*

We conducted a field study with 25 participants who installed our own data-collection application and reported their data for four weeks. The goal of this study was to obtain not only smartphone usage data but also the ground truth on the participants' interruptibility through experience sampling. Among 25 participants (20 men and 5 women), 5 were recruited from our department, and 20 were from an online community. For the former group, we personally asked them to join the experiment; for the latter group, we posted a wanted advertisement on the online community and selected 20 eager users from the applicants. The participants consisted of 15 undergraduate students, 5 master students, and 5 PhD students. All of them lived in a dormitory of KAIST. Then, we provided all participants with the detailed instructions and received explicit consent from them before the experiment. After the experiment was complete, we paid about US$100 to each participant. This study was approved by the KAIST institutional review board (IRB).

### *Data Collection*

All participants downloaded the data-collection application from Google Play and installed it on their Android smartphone. Our application supports Android 4.0 or higher. Table 2 shows the attributes that the application collected in the background. Each participant sent us his/her weekly data at the end of each week, and we verified the data to give him/her feedback on the quality of the data. We did *not* collect any personal information that can be used for inferring the data owner as per the recommendation of the IRB. This data collection was run for four weeks from February 2015 to March 2015.

### *Experience Sampling*

We collected the ground-truth information about the participants' state of interruptibility via experience sampling [5]. The *experience sampling method (ESM)* is a signal-contingent method of data collection from participants about their current experience or situation. In our case, we collected *in-situ* self-reports on the subjective state of interruptibility.

A notification in Figure 2 popped up—*five* times a day randomly on the hour between 9 a.m. and 10 p.m.—per trigger from our server. The notification asked the participants to answer the question with "Yes" or "No". All participants were explained about the meaning of the question: "you are interruptible if you are willing to do a simple task by spending less than ten minutes right now."

A participant's interruptibility was recorded together with temporal information (e.g., time and date) when he/she responded

| Attribute | Description | Type | # Instances |
|---|---|---|---|
| cpu | CPU usage | numeric | 67,590 |
| bat_lev | Battery level | numeric | 242,560 |
| bat_temp | Battery temperature | numeric | 242,560 |
| cell_strn | Cellular signal strength | numeric | 95,071 |
| wifi_strn | WiFi signal strength | numeric | 40,482 |
| ill | Ambient light level | numeric | 37,093 |
| accel_x | Acceleration force (X-axis) | numeric | 119,347 |
| accel_y | Acceleration force (Y-axis) | numeric | 119,347 |
| accel_z | Acceleration force (Z-axis) | numeric | 119,347 |
| accel_tot | Acceleration force (total) | numeric | 119,347 |
| airplane | Airplane mode on/off | binary | 67,590 |
| screen | Screen on/off | binary | 67,590 |
| headset | Headset mode on/off | binary | 67,590 |
| cell | Cellular mode on/off | binary | 95,071 |
| wifi | Wifi mode on/off | binary | 40,482 |
| charge | Charge mode on/off | binary | 242,560 |
| ringtone | Ringtone mode | nominal | 67,590 |
| charge_stat | Charge status | nominal | 242,560 |
| ssid | Connected Wifi SSID | nominal | 40,482 |
| app_pkg | Application package name | nominal | 264,520 |
| app_cat | Application category | nominal | 264,520 |
| location | Location name (district) | nominal | 52,744 |
| call | Phone call event | nominal | 3,530 |
| sms | Message event | nominal | 4,964 |

**Table 2. Attributes collected in the KAIST data set.**

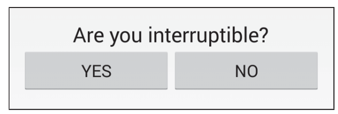



**Figure 2. Screen capture of the experience sampling probe.**

to a question. If a participant did not respond within ten minutes after receiving a question, we recorded his/her status as "not interruptible" at that time.

## Device Analyzer Data Set

### *Description*

The Device Analyzer project is being maintained by the University of Cambridge.[1] Its data set contains over 100 billion records of Android smartphone usage from over 17,000 devices across the world, which is known to be the largest collection of smartphone usage data. We officially obtained a license from the University of Cambridge and downloaded the snapshot of the data set as of November 2015. The size of the raw data reached around 7.5 terabytes. While the project collects more than 50 attributes, we selected the 26 attributes that correspond to those of the KAIST data set. Table 3 lists all the attributes used in this paper.

Among 9,641 users in total, we extracted the users who had sufficiently many records to achieve reliable results. 907 users were chosen in the order of the number of incoming calls to cover 70% of all incoming calls. In this set of selected users, the number of incoming calls for each user ranged from 743 to 13,635 and averaged out at 2,062. Their data were recorded for 274 days on the average.

### *Ground Truth*

Since the Device Analyzer data set does not contain experience sampling data, we treat *call availability* [21] as interruptibility.

[1] https://deviceanalyzer.cl.cam.ac.uk/

| Attribute | Description | Type | # Instances |
|---|---|---|---|
| bat_lev | Battery level | numeric | 187,406,246 |
| bat_temp | Battery temperature | numeric | 187,398,446 |
| vol_music | Media (music) volume | numeric | 63,501,119 |
| vol_alarm | Alarm sound volume | numeric | 63,501,119 |
| vol_voicecall | Voice call sound volume | numeric | 63,501,119 |
| vol_system | System sound volume | numeric | 63,501,119 |
| vol_ring | Ringtone sound volume | numeric | 63,501,119 |
| vol_noti | Notification sound volume | numeric | 63,501,119 |
| accel | Acceleration force | numeric | 31,949,667 |
| light | Ambient light level | numeric | 25,560,714 |
| sms_unread_cnt | Number of unread SMS | numeric | 4,719,150 |
| airplane | Airplane mode on/off | binary | 9,486,295 |
| screen | Screen on/off | binary | 35,960,762 |
| headset | Headset mode on/off | binary | 1,739,896 |
| wifi | Wifi mode on/off | binary | 2,241,508 |
| wifi_conn | Wifi connectivity | binary | 6,155,453 |
| mobile_conn | Mobile connectivity | binary | 7,460,751 |
| bluetooth | Bluetooth on/off | binary | 250,749 |
| charge | Charge mode on/off | binary | 10,937,637 |
| ringtone | Ringtone mode | nominal | 9,700,121 |
| charge_stat | Charge status | nominal | 10,937,637 |
| display_orient | Display orientation | nominal | 10,705,032 |
| app_pkg | Application package name | nominal | 101,442,234 |
| app_cat | Application category | nominal | 101,442,234 |
| location | Location (LAC, CID) | nominal | 64,222,074 |
| sms | Message event | nominal | 2,280,439 |

**Table 3. Attributes used in the Device Analyzer data set.**

Hence, interruptibility in this data set is defined differently from that in the KAIST data set. Alternatively to the ESM, this implicit labeling involves observing user actions and making deductions, just like Pielot et al. [22] did using notification dismissal. *Implicit* labeling is as widely used as *explicit* labeling (experience sampling) [30]. Our labeling is reasonable because users in non-interruptible circumstances can not or do not pick up incoming calls because of unavoidable, enforced, intentional, or negligent unavailability [24]. We excluded call-related attributes from prediction since they directly indicate interruptibility.

A user is regarded as being interruptible when he/she picks up an incoming call and continues the call for at least ten seconds. In contrast, a user is regarded as being *not* interruptible when he/she does not pick up an incoming call or quits the call within just ten seconds. Here, the minimum call duration means the time needed to say some short message, e.g., "Sorry, I am busy right now. Can I call you back later?" When we shortened the minimum call duration from ten seconds to two seconds, the proportion of interruptible moments increased by 2.9%, and prediction accuracy slightly improved by 2–3%.

### Detailed Statistics

In order to examine the data sets at fine granularity, we divide a day into six equi-width timeslots as in Figure 3.

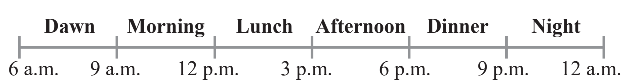
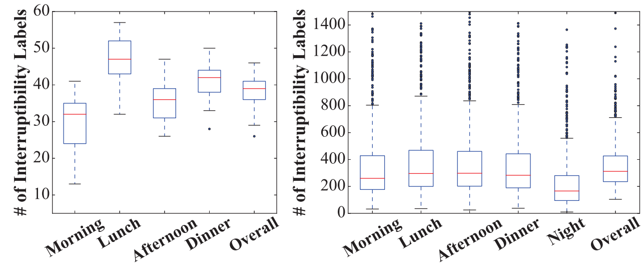



**Figure 3. Six timeslots in a day.**

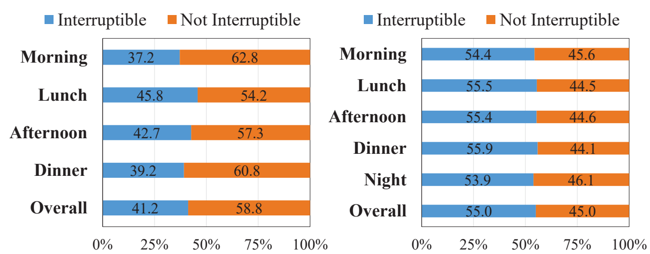
Figure 4 shows the number of interruptibility labels (i.e., interruptible or non-interruptible) *per user* in each timeslot through the experimental period. The median numbers range between 32 and 47 in Figure 4(a) and between 166 and 297 in Figure 4(b). While the KAIST data set has a sufficient number of labels, the Device Analyzer data set has a significantly larger number of labels.




(a) KAIST set. (b) Device Analyzer set.

**Figure 4. Number of interruptibility labels in each timeslot.**

Figure 5 shows what proportion of the labels are interruptible or not in each timeslot. In Figure 5(a), 41.2% of the labels indicate being interruptible, and 58.8% of the labels indicate being not interruptible. In Figure 5(b), the corresponding proportions are 55% and 45% respectively. The proportion of being interruptible is higher in Figure 5(b) than in Figure 5(a). We note that the two label values tend to be balanced across all timeslots in both data sets. In addition, in the Device Analyzer data set archived for a sufficiently long period, the proportion for *each day of week* turned out to be very similar to the proportion for all days in Figure 5(b).




(a) KAIST set. (b) Device Analyzer set.

**Figure 5. Proportion of interruptibility label values in each timeslot.**

## METHODOLOGY: DAILY FEATURE EXTRACTION

In this section, we propose our methodology of extracting daily features and modeling the interruptibility using the extracted features.

### Temporal Windows

First of all, in order to answer **RQ1–RQ4**, we define three types of temporal windows in Figure 6 and consider them together with the current point. In Definition 1, we clarify the source of a feature depending on whether it is extracted from the current point or a temporal window.

1. **Current point**: the current moment when interruptibility needs to be predicted
2. **Immediate-past window**: the interval from the current point back to 15 minutes before
3. **Today window**: the interval from the current point back to the beginning of the current day

4. **Yesterday window** (or **the-day-before-yesterday window**): the interval from the end of the latest previous day (or the second-latest previous day) back to the beginning of the latest previous day (or the second-latest previous day)

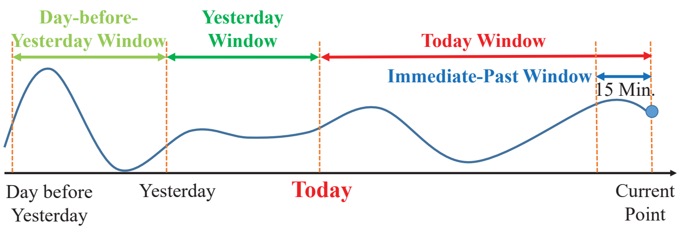
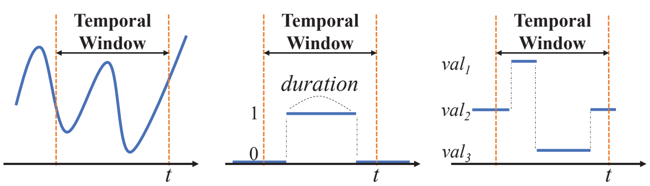



**Figure 6. Temporal windows used for feature extraction.**

**Definition** *1.* The *basic* features are those extracted from the current point in Figure 6. The *extended* features are those extracted from a temporal window in Figure 6. We specifically call the extended features from the today window the *daily* features. □

## Extended Features

To cover various data sources, we categorize attributes (variables) into three types: numeric, binary, and nominal attributes, as shown in Figure 7. Since each attribute type has distinct characteristics, we define the extended features separately for each type so that they best represent the attribute values of the type in a given temporal window.




(a) Numeric. (b) Binary. (c) Nominal.

**Figure 7. Three types of the attributes in the interruptibility data sets.**

### *Overview and Examples*

Table 4 shows the list of extended features for each attribute type, complementary to Figure 7. For numeric attributes, the mean and standard deviation are calculated to represent the central tendency and dispersion of the values in a temporal window; in addition, a *discrete wavelet transform (DWT)* is applied to capture the general trend (i.e., shape) in a given window, which will be discussed in detail. For binary attributes, since the semantics of "0" and "1" are opposed to each other, we keep the duration of "1" samples and the number of transitions from "0" to "1" in a temporal window. Since a nominal attribute is a generalization of a binary attribute, we keep such duration and number for each possible value.

While the current point and the immediate-past window are considered as *atomic* units, the today window, the yesterday window, and the day-before-yesterday window are *partitioned* into six *timeslots*—`dawn`, `morning`, `lunch`, `afternoon`, `dinner`, and `night`—according to Figure 3 before deriving extended features. For the today window, the timeslot to which the current point belongs is considered *partially* up to the present time. For example, if the present time is 8 p.m., the timeslot `dinner` spans from 6 p.m. to 8 p.m. (not 9 p.m.).

| **Measure** | **Description** |
|---|---|
| | **Numeric Attributes** (Figure 7(a)) |
| `mean` | the mean of the samples |
| `std` | the standard deviation of the samples |
| `dwt` | the 32 DWT coefficients of the samples |
| | **Binary Attributes** (Figure 7(b)) |
| `dur` | the sum of the duration of "1" samples |
| `num` | the total number of transitions to "1" |
| | **Nominal Attributes** (Figure 7(c)) |
| $\texttt{val}_i\texttt{\_dur}$ | the sum of the duration of "$val_i$" samples |
| $\texttt{val}_i\texttt{\_num}$ | the total number of transitions to "$val_i$" |

**Table 4. Extended features derived from a temporal window.**

The goal of this partition is to shorten the length of a temporal window such that each interval has coherent semantics, because considering a too long interval *as a whole* may loose important information [19].

We now summarize how an extended feature is constructed in Figure 8. If a temporal window is immediate-past, the measures except `dwt` in Table 4 are calculated for the window. Otherwise, a window is split into timeslots, and then the measures are calculated for each timeslot. A feature name is denoted by concatenating the names of an attribute, a temporal window, a timeslot, and a measure, e.g., `cpu_today_lunch_std`.

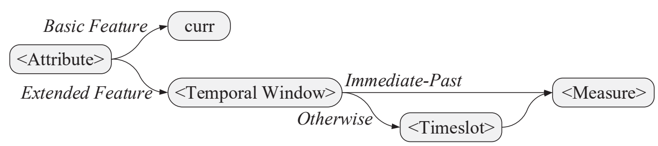



**Figure 8. Notation and composition of extended features.**

**Example** *1.* Let's consider the values of the two attributes in an immediate-past window as shown below. The attribute `accel_x` is numeric, and the attribute `ringtone` is nominal.

| | 9:00 | 9:03 | 9:06 | 9:09 | 9:12 | 9:15 |
|---|---|---|---|---|---|---|
| `accel_x` | 9.3 | 8.7 | 10.8 | 11.1 | 9.5 | 9.6 |
| `ringtone` | silent | silent | silent | silent | normal | normal |

Current Point ↑

- Basic features: 9.6 and "normal" for `accel_x` and `ringtone` respectively.
- Extended features: For `accel_x_imm-past`, `mean` = 9.8 and `std`= 0.9. For `ringtone_imm-past`, `normal_dur` = 3 minutes and `normal_num` = 1; `vibrating_dur` = 0 minute and `vibrating_num` = 0; `silent_dur` = 12 minutes and `silent_num` = 1. □

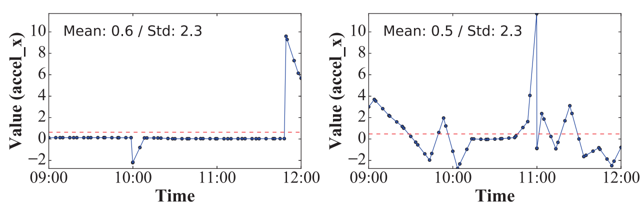
### *Discrete Wavelet Transform (DWT) Features*

The DWT has been widely used for compression and dimensionality reduction owing to its capability of capturing the major trends of underlying data. It decomposes an input sequence into a set of *wavelets* and produces a set of coefficients of the same size as the sequence.[2] The Haar wavelet [2], which is very simple yet effective, is adopted in this work. The coefficients, as going from the first to the last, indicate the frequency of a *finer* temporal domain. A key advantage over other transforms (e.g., Fourier transforms) is *temporal resolution* that

[2]Refer to the tutorial [29] for the details about the DWT.

captures location in time as well as frequency, which is essential for our problem since the time when an event happened should be preserved. Before going into the details, we present a motivating example for the DWT.

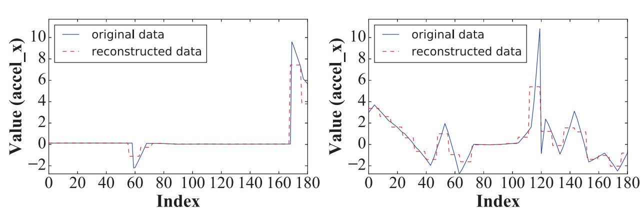
**Example** *2.* Figure 9 shows the values of `accel_x` in the KAIST data set. Between the two different timeslots of Figures 9(a) and 9(b), the mean, which is denoted by the red dashed line, and the standard deviation are almost the same, although the shapes are very different from each other. User 22 almost did not move on February 14 (Figure 9(a)), whereas User 22 frequently moved on February 24 (Figure 9(b)). In fact, the interruptibility at the *dinner* on February 14 was different from that at the *dinner* on February 24. Thus, the temporal shape is related to interruptibility, and the DWT is needed to capture the temporal shape that can be characterized neither by the mean nor by the standard deviation. □




(a) User 22 / 2015-02-14 / `accel_x`. (b) User 22 / 2015-02-24 / `accel_x`.

**Figure 9. A motivating example for using the DWT.**

We derive DWT coefficients for a sequence generated from each 3-hour timeslot covered by a temporal window. We do *not* apply the DWT to the immediate-past window since its duration is not long enough. First, a sequence of length 180 is constructed by the values at every minute. If a value does not exist, it is estimated by linear interpolation between two consecutive timestamps. Then, we pad zeros on the right side of the sequence to make its length 256 because the DWT is defined for the sequences with length of a power of 2. Last, after applying the DWT to the input sequence, only the *first* 32 *coefficients* are selected for dimensionality reduction, and such an approach is widely accepted when leveraging DWT coefficients as an attribute [3, 32]. Figure 10 shows the sequences obtained by restoring those sequences in Figure 9 with the 32 coefficients. For both sequences in Figures 10(a) and 10(b), the shape of the restored sequence in red is very close to that of the original sequence in blue.




(a) User 22 / 2015-02-14 / `accel_x`. (b) User 22 / 2015-02-24 / `accel_x`.

**Figure 10. Reconstruction using the first 32 Haar wavelet coefficients.**

### Feature Configurations

Table 5 illustrates the definitions of the seven feature configurations subsequently used in this paper. `CURR` takes account of the current point only. `IPAST` expands feature extraction to the immediate past. `DAY[]` takes advantage of the features constructed from a long duration in addition to those used by `IPAST`; `0` indicates the current day, `-1` one day before that day, and `-2` two days before that day; a colon denotes an inclusive range. For example, `DAY[0]` takes account of the current point, the immediate-past window, and the today window.

| Time / Conf. | D-b-Yesterday | Yesterday | Today | Imm-Past | Current |
|---|---|---|---|---|---|
| **CURR** | | | | | ✓ |
| **IPAST** | | | | ✓ | ✓ |
| **DAY[0]** | | | ✓ | ✓ | ✓ |
| **DAY[-1:0]** | | ✓ | ✓ | ✓ | ✓ |
| **DAY[-2:0]** | ✓ | ✓ | ✓ | ✓ | ✓ |
| **DAY[-1]** | | ✓ | | ✓ | ✓ |
| **DAY[-2]** | ✓ | | | ✓ | ✓ |

**Table 5. Feature configurations used in this paper.**

**Example** *3.* Let's consider a numeric attribute `cpu`. If the current point appears in the timeslot `night`, the configuration `DAY[0]` produces 21 features in total, as shown in Figure 11. Concatenation of the nodes by following arrows from the root to a leaf composes a feature. □

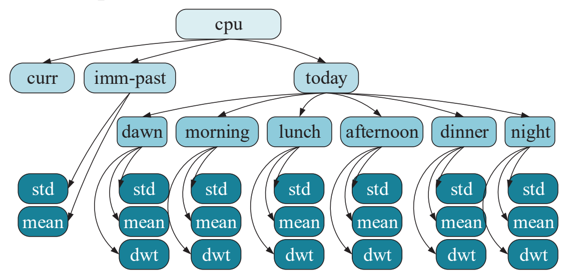



**Figure 11. List of all features for the attribute cpu in `DAY[0]`.**

## EVALUATION RESULTS

In this section, we report the results of a series of experiments designed to answer each research question.

### Experimental Setting

*Data Preprocessing*

We improved the quality of the raw data by preprocessing. First, numeric attribute values were normalized to between 0 and 1 by min-max normalization and then discretized by the MLDPC method [7] that determines the optimal cut points by supervised learning. Hence, small fluctuations in attribute values were smoothed out. Second, the instance timestamps were made the same across all attributes of a user. A missing value at a certain timestamp was estimated by linear interpolation for numeric attributes and by forward filling, which uses the immediately previous value, for binary and nominal attributes. Third, if there were too many possible values in a nominal attribute (e.g., location identifiers and application names), we selected the most frequent 10 values and grouped all other infrequent values into a single value.

*Feature Selection and Prediction*

Regarding feature selection, we used the *correlation-based feature selection* (CFS) [10] method implemented in Weka[3]. The CFS method selects a subset of features that are highly correlated with the class while having low intercorrelation.

[3] http://www.cs.waikato.ac.nz/ml/weka/

Regarding prediction, we used four classification methods: *naive Bayes classifier* (**NB**), *support vector machine* (**SVM**), *random forest* (**RF**), and *C4.5 decision tree* (**C4.5**). Because of space limitations, we present only the results of naive Bayes classifiers except **RQ1** and **RQ2**.

### *Compared Methods*

We compared the seven feature configurations in Table 5.

- **Baseline** (`CURR`): corresponding to earlier work (e.g., [15]) that uses only the present-time features
- **State-of-the-art** (`IPAST`): corresponding to recent work (e.g., [8, 14, 25, 26]) that uses the immediate-past features as well
- **Proposed methodology** (`DAY[0]`): using the *daily* features as well
- **Variation** (`DAY[-1:0]`, `DAY[-2:0]`, `DAY[-1]`, `DAY[-2]`): using the data of one or two days ago

### *Data Sets*

We used the data on *all* days for **RQ1** and **RQ2**, but we used the data only on Wednesday, Thursday, and Friday for **RQ3** and **RQ4**. When we address **RQ3** and **RQ4**, since the yesterday and the-day-before-yesterday windows are additionally considered, we want to make sure that all temporal windows span through weekdays in order to avoid a possible bias between weekdays and weekends. In addition, the timeslot `night` was not used for prediction in the KAIST data set owing to the lack of the ground truth, whereas it was used in the Device Analyzer data set.

Table 6 shows the total number of features extracted by each configuration when the current point belongs to `dinner`. Only the number of `DAY[0]` is affected by the current point since it includes the timeslots *up until that point*, whereas those of the other configurations are not. The number of features increase as the duration used for feature extraction gets longer.

| Data Set / Conf. | **KAIST** (`dinner`) | **Device Analyzer** (`dinner`) |
|---|---|---|
| **CURR** | 70 | 71 |
| **IPAST** | 195 | 199 |
| **DAY[0]** | 2,420 | 2,599 |
| **DAY[-1]**, **DAY[-2]** | 2,865 | 3,079 |
| **DAY[-1:0]** | 5,090 | 5,479 |
| **DAY[-2:0]** | 7,760 | 8,359 |

**Table 6. Number of features used for prediction in dinner.**

### *Measurement*

We built a *personalized* classification (prediction) model for each person using his/her own data only. It would be difficult to apply a global model to all users because the important features in each personalized model are typically different across persons. Then, we measured the accuracy and kappa by 5-fold cross validation for the relatively small KAIST data set and 10-fold cross validation for the Device Analyzer data set. The *accuracy* is the proportion of true results (both true positives and true negatives) among the total number of instances classified; the *kappa* measures the agreement between predicted labels and true labels. Last, we reported the averages of accuracy values and kappa values from all users. The significance of the difference between accuracy values was tested for all pairs of configurations using the two-tailed $t$-test. The $t$-test was performed on *overall* accuracy, i.e., the average of the accuracy values on all timeslots. The results are summarized in Tables 7 and 8, where the *p-value* and the number ($N$) of samples used for each test are presented. The number of asterisks denotes the statistical significance. The $t$-test results that correspond to **RQ1**, **RQ2**, **RQ3**, and **RQ4** are indicated in colors of orange, yellow, green, and blue, respectively.

| | | CURR | IPAST | DAY[0] | DAY[-1:0] | DAY[-2:0] | DAY[-1] | DAY[-2] |
|---|---|---|---|---|---|---|---|---|
| **CURR** | P-value | | | | | | | |
| | N | | | | | | | |
| **IPAST** | P-value | 0.0001*** | | | | | | |
| | N | RQ1 25 | | | | | | |
| **DAY[0]** | P-value | 0**** | 0**** | | | | | |
| | N | RQ2 25 | 25 | | | | | |
| **DAY[-1:0]** | P-value | 0**** | 0.0118* | NS | | | | |
| | N | 24 | 24 | 24 | | | | |
| **DAY[-2:0]** | P-value | 0**** | 0.0118* | NS | NS | | | |
| | N | 24 | 24 | RQ3 24 | 24 | | | |
| **DAY[-1]** | P-value | 0**** | NS | 0.0283* | 0.0283* | 0.0283* | | |
| | N | 24 | 24 | 24 | 24 | 24 | | |
| **DAY[-2]** | P-value | 0**** | NS | 0.0283* | 0.0283* | 0.0283* | NS | |
| | N | 24 | 24 | RQ4 24 | 24 | 24 | 24 | |

NS: $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$

**Table 7. T-test results for the KAIST data set.**

| | | CURR | IPAST | DAY[0] | DAY[-1:0] | DAY[-2:0] | DAY[-1] | DAY[-2] |
|---|---|---|---|---|---|---|---|---|
| **CURR** | P-value | | | | | | | |
| | N | | | | | | | |
| **IPAST** | P-value | 0**** | | | | | | |
| | N | RQ1 907 | | | | | | |
| **DAY[0]** | P-value | 0**** | 0**** | | | | | |
| | N | RQ2 907 | 907 | | | | | |
| **DAY[-1:0]** | P-value | 0**** | 0**** | 0.0487* | | | | |
| | N | 883 | 883 | 883 | | | | |
| **DAY[-2:0]** | P-value | 0**** | 0**** | 0.008** | NS | | | |
| | N | 883 | 883 | RQ3 883 | 883 | | | |
| **DAY[-1]** | P-value | 0**** | 0**** | 0.0128* | 0**** | 0**** | | |
| | N | 883 | 883 | 883 | 883 | 883 | | |
| **DAY[-2]** | P-value | 0**** | 0**** | 0.0212* | 0**** | 0*** | NS | |
| | N | 883 | 883 | RQ4 883 | 883 | 883 | 883 | |

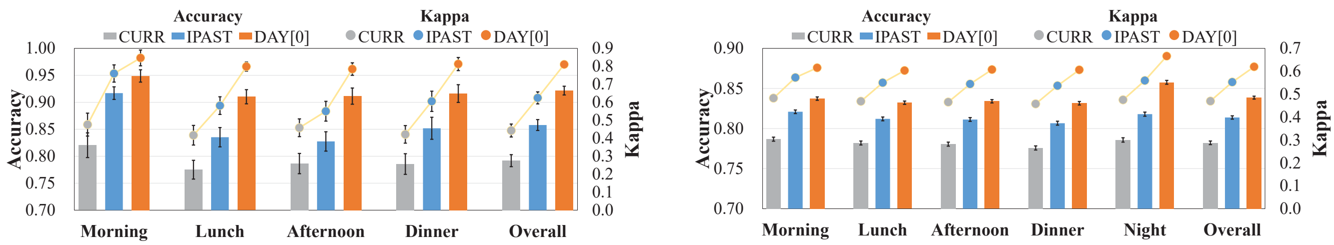
NS: $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$

**Table 8. T-test results for the Device Analyzer data set.**

### RQ1 and RQ2: Daily Features

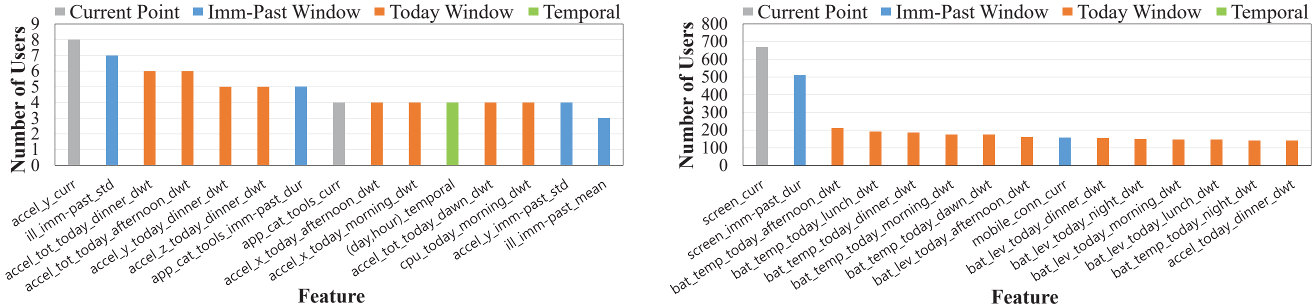
Figure 12 shows the accuracy and kappa calculated based on different features to address **RQ1** and **RQ2** for both data sets. Error bars indicate the standard error. In both data sets, `DAY[0]` achieved the highest accuracy and kappa, followed by `IPAST` and `CURR`, for all timeslots. In addition, as the orange and yellow areas in Tables 7 and 8 which correspond to **RQ1** and **RQ2** respectively show, there are *statistically significant* differences between `CURR` and `IPAST` and between `IPAST` and `DAY[0]`. Table 9 shows the results of all four classifiers for the experiment in Figure 12, where the colored cells correspond to the values in the plot. Here, there was no big difference among the classifiers.

Figure 13 shows the top-15 discriminative features for `DAY[0]`. Here, the importance of a feature is determined by the number of users whose model still contains it after feature selection. We show the results only for the last timeslot—`dinner` for the KAIST data set and `night` for the Device Analyzer data set—to avoid redundancy since we found a persistent consistency among all timeslots. As shown in Figure 13, the features




(a) KAIST data set. (b) Device Analyzer data set.

**Figure 12. Accuracy and kappa based on different features to address RQ1 and RQ2.**




(a) KAIST data set (for `dinner`). (b) Device Analyzer data set (for `night`).

**Figure 13. Top-15 discriminative features in `DAY[0]`.**

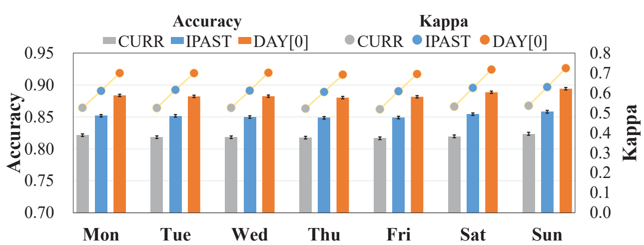



**Figure 14. Decomposition of Figure 12(b) into each day of week.**

from the current point and the immediate-past window were ranked the first and the second respectively. However, the other features were mostly extracted from the today window. Interestingly, even though we predicted the interruptibility for the last timeslot, many of these "today-window" features came from earlier timeslots (even including `dawn`): 5 out of 8 in the KAIST data set and 10 out of 12 in the Device Analyzer data set. This indeed confirms our claim that the behaviors in the previous several hours affect the current interruptibility.

In Figure 13, we observe that many of the discriminative features for the KAIST data set are accelerometer-related ones and those for the Device Analyzer data set are screen or battery-related ones, all of which are closely related to movement or usage of smartphones. This result on important sensor categories is consistent with Dey et al. [6]'s work.

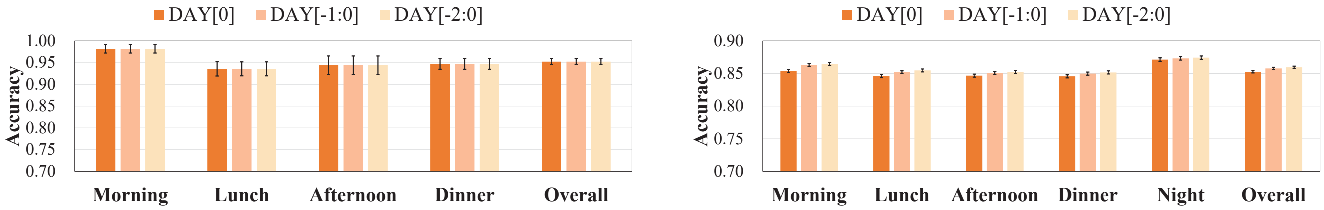
Then, we decomposed the results for all days in Figure 12(b) into those for *each day of week*, because human behavior is typically different on weekdays than on weekends. The KAIST data set was not examined since there were only four days on each day of week. In Figure 14 for the Device Analyzer data set, the results showed the tendency identical to Figure 12(b) across all *days of week*. Thus, our methodology is indeed beneficial irrespective of day of week. We conjecture that

| Conf. | Accuracy (KAIST) | | | | Accuracy (Device Analyzer) | | | |
|---|---|---|---|---|---|---|---|---|
| | NB | SVM | RF | C4.5 | NB | SVM | RF | C4.5 |
| **Morning** | | | | | | | | |
| CURR | 0.82 | 0.75 | 0.77 | 0.77 | 0.79 | 0.77 | 0.78 | 0.77 |
| IPAST | 0.92 | 0.86 | 0.87 | 0.84 | 0.82 | 0.81 | 0.81 | 0.81 |
| DAY[0] | **0.95** | 0.90 | 0.91 | 0.85 | **0.84** | 0.82 | 0.83 | 0.82 |
| **Lunch** | | | | | | | | |
| CURR | 0.78 | 0.71 | 0.71 | 0.74 | 0.78 | 0.77 | 0.77 | 0.77 |
| IPAST | 0.84 | 0.79 | 0.80 | 0.80 | 0.81 | 0.80 | 0.81 | 0.80 |
| DAY[0] | **0.91** | 0.88 | 0.87 | 0.83 | **0.83** | 0.82 | 0.82 | 0.82 |
| **Afternoon** | | | | | | | | |
| CURR | 0.79 | 0.73 | 0.73 | 0.74 | 0.78 | 0.77 | 0.77 | 0.77 |
| IPAST | 0.83 | 0.81 | 0.78 | 0.79 | 0.81 | 0.80 | 0.80 | 0.80 |
| DAY[0] | **0.91** | 0.88 | 0.88 | 0.84 | **0.83** | 0.82 | 0.82 | 0.82 |
| **Dinner** | | | | | | | | |
| CURR | 0.79 | 0.73 | 0.72 | 0.75 | 0.78 | 0.76 | 0.77 | 0.77 |
| IPAST | 0.85 | 0.80 | 0.81 | 0.79 | 0.81 | 0.79 | 0.80 | 0.80 |
| DAY[0] | **0.92** | 0.89 | 0.88 | 0.84 | **0.83** | 0.82 | 0.82 | 0.81 |
| **Night** | | | | | | | | |
| CURR | - | - | - | - | 0.79 | 0.76 | 0.77 | 0.76 |
| IPAST | - | - | - | - | 0.82 | 0.79 | 0.80 | 0.80 |
| DAY[0] | - | - | - | - | **0.86** | 0.84 | 0.84 | 0.82 |
| **Overall** | | | | | | | | |
| CURR | 0.79 | 0.73 | 0.73 | 0.75 | 0.78 | 0.77 | 0.77 | 0.77 |
| IPAST | 0.86 | 0.82 | 0.82 | 0.80 | 0.81 | 0.80 | 0.81 | 0.80 |
| DAY[0] | **0.92** | 0.89 | 0.89 | 0.84 | **0.84** | 0.82 | 0.83 | 0.82 |

**Table 9. Accuracy results for RQ1 and RQ2 with all four classifiers.**

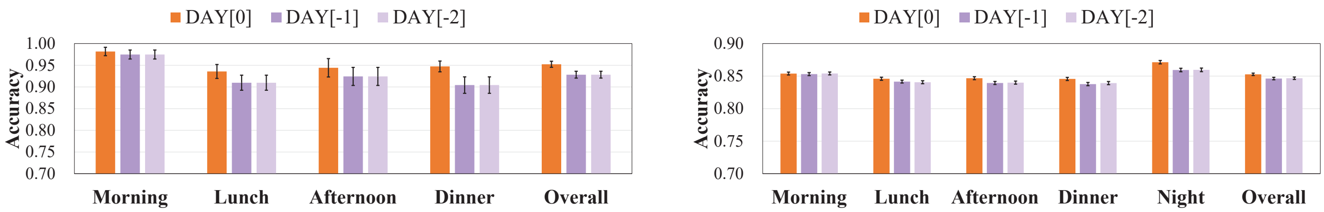
conservation of energy and prolongation are effective also at the weekend whereas self-regulation might not.

In conclusion, although the accuracy achieved by using only the basic features—79% (kappa = 0.44) in the KAIST data set and 78% (kappa = 0.47) in the Device Analyzer data set in overall—is also acceptable, we can even increase the accuracy by leveraging the daily features—up to 92% (kappa = 0.81) in the KAIST data set and 84% (kappa = 0.62) in the Device Analyzer data set in overall. The largest improvements of `DAY[0]`




(a) KAIST data set. (b) Device Analyzer data set.

**Figure 15. Accuracy based on different features to address RQ3.**




(a) KAIST data set. (b) Device Analyzer data set.

**Figure 16. Accuracy based on different features to address RQ4.**

compared with `CURR` and `IPAST` were shown to be 16% (from 73% to 89%) and 7% (from 82% to 89%) respectively when using the **SVM** or **RF** classifiers in the KAIST data set.

### RQ3: Temporal Window Length

Figure 15 shows the accuracy calculated based on different features to address **RQ3** for both data sets. Prior to feature selection, `DAY[-1:0]` and `DAY[-2:0]` produce more (about twice or three times) features than `DAY[0]` because additional timeslots are considered in the yesterday and the-day-before-yesterday windows, as shown in Table 6. Despite a larger number of features, however, we did not observe significant increases in accuracy for `DAY[-1:0]` and `DAY[-2:0]` compared with `DAY[0]`. In particular, there was almost no increase in accuracy in the KAIST data set on all timeslots. On the other hand, in the Device Analyzer data set, the accuracy for `DAY[-1:0]` or `DAY[-2:0]` was slightly higher than that for `DAY[0]`. However, the increase from `DAY[0]` to `DAY[-1:0]` is not statistically significant at the significance level of 0.01, and the increase from `DAY[0]` to `DAY[-2:0]` is also almost not, as shown in Table 8. In conclusion, looking further back beyond the current day is not very helpful for increasing the prediction accuracy of interruptibility when the data on the current day does exist.

### RQ4: Daily Routineness

Figure 16 shows the accuracy calculated based on different features to address **RQ4** for both data sets. It was observed that the accuracy values for both `DAY[-1]` and `DAY[-2]` were slightly lower than that for `DAY[0]`. This implies that the data from the current day is more helpful to predict interruptibility than the data from the latest (or second-latest) previous day in spite of the repetitive daily patterns. However, the decrease in accuracy from `DAY[0]` to `DAY[-1]` or `DAY[-2]` is not statistically significant at the significance level of 0.01, as shown in Table 7. In conclusion, the data from the latest (or second-latest) previous day can be a good substitute when the model suffers from the lack of the data from the current day.

## CONCLUSION AND IMPLICATION

In this paper, we proposed a feature extraction methodology for interruptibility prediction using smartphone usage data. We conducted a field study and performed extensive experiments on two real-world data sets. Our methodology of looking back on the current day achieved the accuracy of over 90%, being higher than the baseline and state-of-the art methods by up to 16% and 7% respectively. The improvement was attributed to the fact that *daily* behavioral features were included in the predictive features of many users. We also found out that looking further back beyond the current day did not improve accuracy owing to the daily routineness of human behaviors. We, thus, confirmed that a day's behavior is replaceable with another day's behavior for the same reason.

We believe that smartphone applications benefiting from our methodology will improve *communication efficiency* dramatically, based on a better understanding of when and how to engage with users. A potential application scenario that we envision is the real-time mobile Q&A service. When a user asks a question on such a smartphone application, the question is delivered to a set of expert users; when some of them answer the question, the answers are immediately delivered to the questioner. Thus, the success of this service depends upon selection of expert users who are interruptible at that moment.

As the future work, we plan to improve the usability of our methodology by coping with the cold start problem. Since our methodology is based on personalization, prediction for a user becomes reliable after the training data for the user has been collected for a sufficiently long period (at least three weeks). A hybrid approach using both personal and aggregated data could reduce the training requirements for new users.

## ACKNOWLEDGMENTS

This material is based upon work supported by the Air Force Office of Scientific Research, Asian Office of Aerospace Research and Development (AOARD) under award number FA2386-15-1-4060.

## REFERENCES


1. Christopher Beedie, Peter Terry, and Andrew Lane. 2005. Distinctions between Emotion and Mood. *Cognition and Emotion* 19, 6 (2005), 847–878.
2. C. Sidney Burrus, Ramesh A. Gopinath, and Haitao Guo. 1997. *Introduction to Wavelets and Wavelet Transforms*. Prentice Hall.
3. Pimwadee Chaovalit, Aryya Gangopadhyay, George Karabatis, and Zhiyuan Chen. 2011. Discrete Wavelet Transform-Based Time Series Analysis and Mining. *Comput. Surveys* 43, 2 (2011), 6:1–6:37.
4. Minsoo Choy, Jae-Gil Lee, Gahgene Gweon, and Daehoon Kim. 2014. Glaucus: Exploiting the Wisdom of Crowds for Location-Based Queries in Mobile Environments. In *Proc. 8th Int'l Conf. on Weblogs and Social Media*.
5. Thomas H. Davenport and John C. Beck. 2002. *The Attention Economy: Understanding the New Currency of Business*. Harvard Business Review Press.
6. Anind K. Dey, Katarzyna Wac, Denzil Ferreira, Kevin Tassini, Jin-Hyuk Hong, and Julian Ramos. 2011. Getting Closer: An Empirical Investigation of the Proximity of User to Their Smart Phones. In *Proc. 2011 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 163–172.
7. Usama M. Fayyad and Keki B. Irani. 1993. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *Proc. 13th Int'l Joint Conf. on Artificial Intelligence*. 1022–1027.
8. James Fogarty, Scott E. Hudson, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C. Lee, and Jie Yang. 2005. Predicting Human Interruptibility with Sensors. *ACM Trans. on Computer-Human Interaction* 12, 1 (2005), 119–146.
9. Bernard Guerin. 2016. *How to Rethink Psychology: New Metaphors for Understanding People and Their Behavior*. Routledge.
10. Mark A. Hall. 1999. *Correlation-based Feature Selection for Machine Learning*. Ph.D. Dissertation. The University of Waikato, Hamilton, New Zealand.
11. Joyce Ho and Stephen S. Intille. 2004. Using Context-Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In *Proc. 19th ACM Int'l Conf. on Computer-Supported Cooperative Work and Social Computing*. 507–510.
12. Eric Horvitz and Johnson Apacible. 2003. Learning and Reasoning about Interruption. In *Proc. 5th Int'l Conf. on Multimodal Interfaces*. 20–27.
13. Eric Horvitz, Paul Koch, and Johnson Apacible. 2004. BusyBody: Creating and Fielding Personalized Models of the Cost of Interruption. In *Proc. 19th ACM Int'l Conf. on Computer-Supported Cooperative Work and Social Computing*. 507–510.
14. Scott Hudson, James Fogarty, Christopher Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny Lee, and Jie Yang. 2003. Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study. In *Proc. 2003 SIGCHI Conf. on Human Factors in Computing Systems*. 257–264.
15. Nicky Kern, Stavros Antifakos, Bernt Schiele, and Adrian Schwaninger. 2004. A Model for Human Interruptability: Experimental Evaluation and Automatic Estimation from Wearable Sensors. In *Proc. 8th IEEE Int'l Sympo. on Wearable Computers*. 158–165.
16. Traci Mann, Denise de Ridder, and Kentaro Fujita. 2013. Self-Regulation of Health Behavior: Social Psychological Approaches to Goal Setting and Goal Striving. *Health Psychology* 32, 5 (2013), 487–498.
17. Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications. In *Proc. 2015 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 813–824.
18. Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K. Dey, and Hideyuki Tokuda. 2015. Reducing Users' Perceived Mental Effort due to Interruptive Notifications in Multi-Device Mobile Environments. In *Proc. 2015 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 475–486.
19. Veljko Pejovic and Mirco Musolesi. 2014. InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. In *Proc. 2014 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 897–908.
20. Veljko Pejovic and Mirco Musolesi. 2015. Anticipatory Mobile Computing: A Survey of the State of the Art and Research Challenges. *Comput. Surveys* 47, 3 (2015), 47:1–47:29.
21. Martin Pielot. 2014. Large-Scale Evaluation of Call-Availability Prediction. In *Proc. 2014 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 933–937.
22. Martin Pielot, Rodrigo de Oliveira, Haewoon Kwak, and Nuria Oliver. 2014. Didn't You See My Message?: Predicting Attentiveness to Mobile Instant Messages. In *Proc. 2014 SIGCHI Conf. on Human Factors in Computing Systems*. 3319–3328.
23. Martin Pielot, Tilman Dingler, Jose San Pedro, and Nuria Oliver. 2015. When Attention is not Scarce - Detecting Boredom from Mobile Phone Usage. In *Proc. 2015 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 825–836.
24. Antti Salovaara, Antti Lindqvist, Tero Hasu, and Jonna Hakkila. 2011. The Phone Rings But the User Doesn't Answer: Unavailability in Mobile Communication. In *Proc. 13th Int'l Conf. on Human Computer Interaction with Mobile Devices and Services*. 503–512.

25. Hillol Sarker, Moushumi Sharmin, Amin Ahsan Ali, Md. Mahbubur Rahman, Rummana Bari, Syed Monowar Hossain, and Santosh Kumar. 2014. Assessing the Availability of Users to Engage in Just-in-Time Intervention in the Natural Environment. In *Proc. 2014 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 909–920.

26. Choonsung Shin, Jin-Hyuk Hong, and Anind K. Dey. 2012. Understanding and Prediction of Mobile Application Usage for Smart Phones. In *Proc. 2012 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 173–182.

27. Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. 2010. Limits of Predictability in Human Mobility. *Science* 327, 5968 (2010), 1018–1021.

28. Vijay Srinivasan, Saeed Moghaddam, Abhishek Mukherji, Kiran K. Rachuri, Chenren Xu, and Emmanuel Munguia Tapia. 2014. MobileMiner: Mining Your Frequent Patterns on Your Phone. In *Proc. 2014 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 389–400.

29. Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. 1995. Wavelets for Computer Graphics: A Primer, Part 1. *IEEE Computer Graphics and Applications* 15, 3 (1995), 76–84.

30. Liam D. Turner, Stuart M. Allen, and Roger M. Whitaker. 2015. Interruptibility Prediction for Ubiquitous Systems: Conventions and New Directions from a Growing Field. In *Proc. 2015 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing*. 801–812.

31. Daniel Wagner, Andrew Rice, and Alastair Beresford. 2013. Device Analyzer: Understanding Smartphone Usage. In *Proc. 10th Int'l Conf. on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 195–208.

32. Yi-Leh Wu, Divyakant Agrawal, and Amr El Abbadi. 2000. A Comparison of DFT and DWT Based Similarity Search in Time-Series Databases. In *Proc. 9th Int'l Conf. on Information and Knowledge Management*. 488–495.

33. Kazuo Yano. 2014. *Invisible Hand of Data*. Soshisha.