Regular Paper

Annotation Method for Human Activity and Device State Recognition Based on Smartphone Notification Removals

Ryota Sawano[1,a)] Kazuya Murao[1,2,b)]

Received: December 23, 2019, Accepted: July 7, 2020

Abstract: With the increasing spread of smartphones and wearable devices equipped with various sensors, human activities, biometric information, and surrounding situations can be recognized. The process of human activity recognition must construct a model that has learned annotated sensor data, i.e., ground truth, labels, or answer activity, in advance. Therefore, a large and diverse set of annotated data is required to improve and evaluate model performance. It is difficult to judge a user's situation even after observing acceleration data; thus, it is necessary to annotate the collected acceleration data. In this paper, we propose a method to estimate user and device situations from the user's response to a notification generated by a device, e.g., a smartphone. The user and device situations are estimated from the user's response time to the notification and the device's acceleration values. An estimation result with high confidence is given to the sensor data as an annotation. Increasing the frequency of notifications, response to the notifications can be used as a sensor. We assume that acceleration values are affected by a user and device situation when the device notifications are taken instantly after its generation. The system pursues a high precision of estimation by selecting input acceleration data based on the interaction to the notification so that the estimations can be used as annotations. Through an evaluation experiment, for seven types of annotation classes, an average precision of 0.769 and 0.963 for user-independent experiments and user-dependent experiments were achieved, respectively. We also tested the proposed method in a natural environment, where 25 correct annotations were given for 45 responses to notifications, no annotations were given for 19 responses, and only one incorrect notification was observed.

Keywords: human activity recognition (HAR), accelerometer, smartphone, notification, labeling, annotation

1. Introduction

With the increasing spread of smartphones and wearable devices equipped with various sensors, human activities, biometric information, and surrounding situations can be recognized anytime and anywhere through sensor data, e.g., such as acceleration [11], angular velocity, light, pulse, position, radio wave status, electromyogram [21], electrocardiogram [7], galvanic skin reflexes [15], and manually configured devices [17]. The obtained information is applied to many services, e.g., a health management system [15] that automatically extracts life patterns and warns of lack of exercise and overwork, support during assembly and maintenance tasks [31] that presents manuals and required tools by predicting the next task from the current operation, medical support systems that record time of medication and blood sugar level measurement outside hospital environments, sports support systems to acquire the number of times of tackle and sprint and strength [6], entertainment whose effect changes according to audience behavior [25], personal authentication based on gait, input interfaces, and games.

For human activity recognition (HAR), a model that has learned annotated sensor data, i.e., ground truth, labels, or answer activities, must be constructed in advance. Therefore, a large and diverse annotated dataset is required to improve and evaluate model performance. Sensor data can be stored on servers and cloud storage via Wi-Fi and cellular networks at low cost once the application is released; however, annotations must be applied manually in a separate process.

In addition, annotation must be accurate. Accurate annotations can be collected by video recording the test subject; however, it is difficult to employ cameras when the application is widely distributed to general users. Annotations by a subject not under surveillance are not reliable. Images, sounds, and texts can be annotated after data collection because such data can be understood by humans; however, it is difficult to infer the situation of acceleration data by seeing it. In short, there is no accurate and scalable annotation method for human activity recognition.

Recently, due to the diverse and large amounts of high-quality data handled by smart devices, notifications occur continuously, e.g., scheduler reminders, messages from friends on social media, weather forecasts, and news notifications. Some notifications may be taken by the user instantly; however, most are not taken because the users cannot be aware of them (e.g., while sleeping, being away from the smartphone, during a presentation, or in a crowded train). The notification reaction rate generated from online shopping and news apps affects sales volumes and advertisement expenses, and research into increasing the notification opening rate has attracted a great deal of attention.

In this paper, we propose a method to estimates user and de-

[1] Graduate School of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525–8577, Japan
[2] PRESTO, Japan Science and Technology Agency, Kawaguchi, Saitama 332–0012, Japan
[a)] ryota.sawano@iis.ise.ritsumei.ac.jp
[b)] murao@cs.ritsumei.ac.jp

vice situations from user responses to notifications generated by a device, e.g., a smartphone. User and device situations are estimated from the user's response time to a notification and the device's acceleration values. Increasing the frequency of notifications, response to the notifications can be used as a sensor. We assume that acceleration values are affected by a user and device situation when the device notifications are taken instantly after its generation. The system pursues a high precision of estimation by selecting input acceleration data based on the interaction to the notification so that the estimations can be used as annotations. Some may think that human activity recognition requires recognition results continuously or at an arbitrary timing of the application, while the occurrence of notifications is sporadic. It is not necessary to annotate all the data collected since unlabeled data can be discarded in the training phase. Only the confidence samples are annotated in this study. If even a small amount of the collected data can be annotated automatically, we can achieve the automatic construction of a large annotated dataset. For these reasons, we would say that notification is compatible with annotation.

The contributions of the proposed method are as follows.

- It is highly scalable and sustainable because it uses the user's interaction in response to a notification as part of normal smartphone use, rather than annotations based on the user's spontaneous recording of images, videos, notes, etc., as is the case with existing methods.
- Compared to spontaneous annotation, the annotations are not affected by the user's error, imprecise response, or non-response, and the quality of the annotations is stable.
- A high precision is achieved by annotating only when a user responds immediately to a notification. If even a small amount of the collected data can be annotated automatically, we can achieve the automatic construction of a large annotated dataset, although the rate at which annotations can be given to the acquired acceleration data, i.e., recall, is reduced.

2. Related Work

2.1 Annotation Method

Annotated data is required for human activity recognition with an accelerometer as well as computer vision, voice recognition, and natural language processing. Images, sounds, and texts can be annotated after the data collection as these data can be understood by a human, however it is difficult to infer the situation of acceleration data by seeing it.

In many studies into HAR using an accelerometer, annotations were collected by video-recording subjects or taking a memo of activities. Annotating sensor data from video is a manual task that requires more time than the duration of the original sensor data. With the recent development of deep learning, caption generation for images [24] and videos [26] has also been actively studied. Since detailed descriptions can be generated as sentences rather than words, it is possible to annotate sensor data automatically if the user's video is recorded. However, it is difficult to use a camera and record a user's image or video when the application is distributed widely to general users.

For annotation using memo, noise may be included in the data since memo is taken during activities, i.e., movement of taking a memo is included in sensor data. Moreover, it is bothersome to record activities all the time the user's activity changes. In contrast to the voluntary recording of the user's notes, another method is called experience sampling [18], in which the system asks the user about the situation and the user responds to it. We can collect annotations by sending a notification to the user's terminal and letting the user choose a choice or describe freely. The authors previously proposed a labeling method for activity recognition using an execution sequence of activities [16]. This method partitions and classifies unlabeled data into segments, and then clusters and assigns a cluster to each segment. Then, labels are assigned according to the best-matching assignment of clusters with the user-recorded activities. This method obtained a precision of 0.812 for data about seven types of activities. It was also confirmed that recognition accuracy with training data labeled by their method gave a recall of 0.871, which was equivalent to that of the ground truth.

These memo-based annotations would work well under a laboratory setting where researchers can accompany the users. When scaling sensor data collection beyond the internet, a variety of people will be users, some of whom might not be hardworking. Annotations by the users not under surveillance is unreliable. In short, annotations recorded by the user are unreliable and unscalable.

The task of annotating for human activity recognition still requires a large amount of time and labor, which is a barrier to construct an activity recognition system with ease. Active learning [8], [13] has been proposed as a method to reduce annotation tasks in the field of machine learning. When there is a labeled dataset and an unlabeled dataset, it effectively selects a sample from the unlabeled dataset and assigns a label to that sample by human power or other means, and trains it. In this way, when there is a small amount of labeled data and a large amount of unlabeled data, only a portion of the unlabeled data can be labeled instead of all the unlabeled data, thus saving time and labor. However, since labeling is done by human power, it is effective for image, sound, and text because a human can annotate it, but in the case of acceleration-based behavioral recognition, a recording such as a video is required.

2.2 Semi-supervised Learning

One method to learn fully labeled data is called supervised learning, and learning partially labeled data is referred to as semi-supervised learning. Both supervised and semi-supervised learning techniques use a classifier to obtain recognition results. These classifiers require fully labeled data, and unlabeled data are discarded. Semi-supervised learning spreads the labels of labeled data to unlabeled data, and then the learning process is performed as supervised learning. However, the spread labels only represent inference and may be incorrect.

In a previous study [20], several semi-supervised learning methods were compared, and a simple self-training algorithm was introduced [23]. Assume a small amount of labeled data and a large amount of unlabeled data. The self-training algorithm constructs models from labeled data, and then classifies unlabeled data using the models. The classification results are then fed back to the unlabeled data as labels, and all data are labeled. Then, the recognition models are reconstructed using all available data.

Maja et al. [19] proposed a method to spread the labels of labeled data to unlabeled data by focusing on the fact that the labels for similar data in the feature and time domains are likely to be the same. This method constructs graphs, where the data are nodes and vertices represent similarity values. Then, this method calculates the similarity among data from the distance of feature values and time difference. The labels of the labeled data are spread to unlabeled data with a high similarity. The amount of labels for each activity is controlled to follow the prior distribution of activities. In the evaluation, labeling accuracy is measured for labeling intervals of 10 to 180 minutes, and this method obtained 90% accuracy for the 10-minute interval and 55% accuracy for the 180-minute interval. Labeling in 10 minutes equals 2.5% of all data, and the 180-minute interval equals 0.1% of all data.

In addition, a previous study that investigated the eigenspace has been reported [30]. Here, a single eigenspace is obtained using principal component analysis, which is primarily used to reduce the dimensionality of multidimensional data; however, multiple eigenspaces can be found using a multiple eigenspace algorithm, and, by applying it to acceleration data, each sample belongs to one of the eigenspaces. This study focused on the fact that there is a relationship between the eigenspace and the activity, and uses the indices of eigenspace as labels to train a support vector machine. Then, this method consolidates the eigenspaces on error ratio. By giving a small amount of labeled data to the eigenspace, the indices of the eigenspace and activities are associated. In an evaluation of eight types of activities, 88.3% recognition accuracy with 80% labeled data and 80.3% accuracy with 20% labeled data were obtained, and these results are higher than the results obtained using only labeled data. However, their evaluations were conducted in an environment where the amount of labels decreased evenly over time, which means that the frequency of labeling lessens but labeling is still needed. Therefore, this approach cannot be considered a fundamental solution.

Evaluations in these studies are conducted in the environment where the amount of label is evenly decreased over the time, which means that the frequency of labeling lessens but labeling is still needed. Semi-supervised learning works effectively if a certain amount of annotated data can be obtained. The proposed method also provides annotations based on the data collected in advance, but it also uses some novel information, the notification response time, which can be collected automatically, to limit the number of samples that can accurately be annotated.

2.3 Effect, Control, and Management of Notifications

It has been reported that notifications on mobile devices affect the user's performance. The notification can be considered an interrupt, and it has been reported that untimely interruptions can increase stress and reduce productivity [9], [12].

Many studies have investigated detecting notification timing that does not disturb the user. For example, Okoshi et al. [27], [28] focused on the physical activity breakpoint, which is the boundary of the user's action, as appropriate timing to open push notifications. When a smartphone receives a notification, it is suspended. Then, if a break point is detected based on the result of activity recognition obtained using the device's accelerometer, the system shows the notification to the user, which improves the notification opening rate. In addition, Okoshi et al. [29] also conducted experiments with real services. They integrated estimation logic using mobile sensing and machine learning into Yahoo! Japan's Android application and experimented with 680,000 users. As a result, the notification response time of the user was reduced, and it successfully investigated how many times the user used the app per week and confirmed user engagement improvement. Ho et al. [4] proposed a method to optimize notification timing and improved the notifications response time using a reinforcement learning-based personalization method called Nurture. These studies have improved the notification response time and opening rate by controlling notification timing. In addition, studies into content management based on user preference have been conducted. It has been reported that uninteresting notices, e.g., promotional emails and invitations to games on social networking service (SNS), tend to be removed without opening [3], [14]. In addition, users can become annoyed by unnecessary notifications from unrelated applications or applications to be uninstalled [2], [3]. Mehrotra et al. [1] proposed a method that detects user preference from a combination of notification title and activity, location, place, etc., and only allows notifications that are meaningful to the user.

3. Proposed Method

We propose a method to annotate sensor data by estimating user and device situations from the user responses to notifications generated by a device, e.g., a smartphone. This section describes the proposed method.

3.1 Assumed Environment

We assume that the users of the proposed method are people who want to collect annotated data, e.g., researchers evaluating the performance of a new recognition algorithm and application engineers developing new general purpose applications using the HAR technique.

As shown in **Fig. 1**, a researcher looks for workers using a crowdsourcing service, e.g., Amazon Mechanical Turk, or looks for volunteers from among their acquaintances. Then, the workers and volunteers install our application, which logs sensor data and responses to notifications. While the smartphone is in use, when an installed application (not our app) generates a notification and the notification is removed by the user, our application logs information about the notification, e.g., application name, posted time, and removed time. The detailed implementation of the application is explained in a later section. Our application uploads the sensor data and response to notifications to the server or cloud storage at fixed intervals. On the server, the proposed method annotates the data using the response to notifications and acceleration values. Finally, the researcher can obtain annotated data. In addition, our system can generate dummy notifications from the researcher to arbitrary users grouped by user attributes,

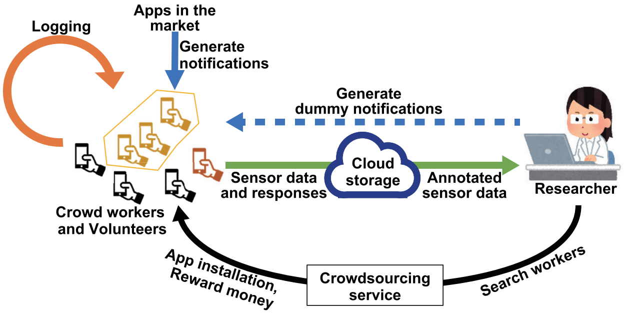


Fig. 1 Assumed environment.

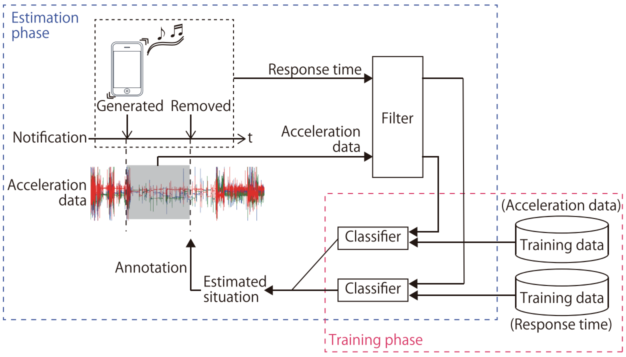


Fig. 2 Flow of the proposed method.

e.g., gender and age, which enables us to collect responses to notifications under equalized conditions. Therefore, once users have installed our application, annotated data are collected automatically.

3.2 Overview of Proposed Method

The flow of the proposed method is shown in **Fig. 2**. Before using the system, as a training phase, the response time from notification generation to notification removal and the smartphone's acceleration data from notification generation to notification removal are collected in all situations to be annotated. In this paper, the following seven situations are assumed classes to be annotated: (1) standing while holding the smartphone unused in a hand; (2) using the smartphone and placing it on a table; (3) leaving the smartphone unused on a table; (4) holding the smartphone and using it; (5) standing with the smartphone in a pocket; (6) walking while holding the smartphone unused in a hand; (7) walking while using the smartphone. Then, we construct models of two classifiers using the acceleration data and response time. One classifier estimates the seven situations from the response time and the other estimates the seven situations from the acceleration data.

After the training phase, as an estimation phase, when a notification is generated at the user device and the user removes it, the response time and acceleration data from notification generation to notification removal are obtained. In order to validate only the data when the notification has been taken instantly after its generation, the proposed system discards the obtained data with a long response time at filter. The validated response time and acceleration data are fed into the classifiers separately, and each classifier outputs more likely situation classes. At last, if the situation classes from the classifiers include common class, the class is fed back to the input acceleration data as an annotation. If there is no common class, annotation is not given to the input data.

3.3 Response Time Filtering

When a notification is generated, i.e., when an application posts a notification to the operating system, the smartphone informs the user of the notification using the LED, sound, vibration, and on-screen visual effects, the system records the timestamp of notification generation T_{gen}. Then, when the notification is removed from the notification area by swiping or tapping the notification bar, the system records the timestamp of notification removal T_{rm}. The time difference T_{diff} between T_{gen} and T_{rm} is calculated as follows (referred to as the notification response time).

$$T_{diff} = T_{rm} - T_{gen} \tag{1}$$

When the user was not aware of the notification instantly after its generation and removed it later, the response time T_{diff} becomes long. In this case, it is difficult to estimate the user and device situation since the user may have performed several actions before deleting the smartphone notification. Therefore, the system discards acceleration data whose T_{diff} is longer than T_{filter}. Regarding T_{filter}, if T_{filter} is too long, precision of annotation would deteriorate. On the other hand, if T_{filter} is too short, annotations can only be given to the motions that can be completed in a short time, such as when the smartphone is in the hand, resulting in the limited types of annotations. Therefore, notifications with a response time longer than T_{filter} are discarded. T_{filter} is the longest of the average response times for all annotations (set to 10 s in this paper).

When the user did not respond to the notification intentionally, e.g., while in a meeting, response time cannot be calculated as the removal time T_{rm} is not obtained. In this case, the system is not executed.

Then, if $T_{diff} \leq T_{filter}$ is satisfied, acceleration data $[[x(T_{gen}), \ldots, x(T_{rm})], [y(T_{gen}), \ldots, y(T_{rm})], [z(T_{gen}), \ldots, z(T_{rm})]]$ between T_{gen} and T_{rm} is extracted, where $x(t)$, $y(t)$, and $z(t)$ are the acceleration values of three axes x, y, and z at time t, respectively. In other words, acceleration data is extracted over a window whose size is T_{diff} starting at T_{gen}. This segmentation is applied to training data preparation as well.

3.4 Situation Estimation Using Response Time

Histograms of response time of training data collected in advance ranging from 0 to 10 s are created for each class. Here, the histogram bin width is 0.5 s, and the number of bins is 20. The frequency of each bin is denoted $h(k)$ ($k = 1, \ldots, 20$), and the likelihood of the response time T_{diff} is obtained as follows:

$$L(T_{diff}) = \frac{h(ceil(T_{diff}/0.5))}{\sum_{k=1}^{20} h(k)}, \tag{2}$$

Here, $ceil(x)$ is the ceiling function, which maps x to the smallest integer greater than or equal to x, i.e., $ceil(1.25) = 2$. Specifically, given a 100-sample dataset, if the number of samples whose response times are in the range $1 < t \leq 1.5$ is five,

$L(1.2) = 5/100 = 0.05$.

The proposed system calculates the likelihood of T_{diff} for all annotation classes. If the highest likelihood is below the threshold L_{th}, the situation estimation result obtained using response time is empty Φ. If, the likelihood of multiple classes exceeds L_{th}, a set of these classes becomes the result. We consider that the distributions of response time in different classes may overlap, and the distribution of response time in a class may be wider; therefore L_{th} is set to a low value to avoid outputting incorrect results. Note that we set the L_{th} to 0.05 to avoid rejective the correct results.

3.5 Situation Estimation Using Acceleration Data

Calculating the similarity between time-series data is required to do data mining in these fields. A simple method to measure similarity is Euclidean distance; however this approach has several drawbacks, e.g., it is susceptible to temporal distortion and the number of samples in two data sequences must be equal. The dynamic time warping (DTW) [5] algorithm measures the similarity of two time-series data, which mitigates the drawbacks of the Euclidean distance approach. The DTW algorithm calculates temporal nonlinear elastic distance, and the similarity between two sequences that may vary in time or speed can be measured. In addition, with the DTW algorithm, the number of both samples does not need to be equal. The proposed method calculates DTW distance between the input acceleration data and training acceleration data.

Detailed algorithm is as follows. When two time-series gesture data $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_n)$ are compared, whose length are m and n, respectively, an $m \times n$ matrix d is defined by $d(i, j) = (x_i - y_j)^2$. Subsequently, a warping path $W = (w_1, \ldots, w_k)$ is found, which is a path of pairs of indices of X and Y. At that time, the pass W is meeting the following three conditions.

- Boundary condition
  $w_1 = (1, 1), w_k = (m, n)$
- Seriality
  $w_k = (a, b), w_{k-1} = (a', b') \Rightarrow a - a' \leq 1 \wedge b - b' \leq 1$
- Monotony
  $w_k = (a, b), w_{k-1} = (a', b') \Rightarrow a - a' \geq 0 \wedge b - b' \geq 0$

So as to find the path with the lowest cost with meeting the above conditions, the following steps are applied.

```
1. $\mathrm{DTW}(0, 0) = 0, \mathrm{DTW}(i, 0) = \mathrm{DTW}(0, j) = \infty$
   $(1 \leq i \leq m, 1 \leq j \leq n)$
2. for $i = 1$ to $m$
     for $j = 1$ to $n$
       $\mathrm{DTW}(i, j) = d(i, j) + \min\{\mathrm{DTW}(i-1, j-1),$
                  $\mathrm{DTW}(i-1, j), \mathrm{DTW}(i, j-1)\}$
3. return $\mathrm{DTW}(m, n)/(m + n)$
```

The obtained cost $\mathrm{DTW}(m, n)$ becomes a distance between X and Y. The returned $\mathrm{DTW}(m, n)$ is divided by $m + n$ since DTW distance increases with the length of the training data and test data.

The proposed method calculates the distance between the input acceleration data and the acceleration data of all annotation classes collected in advance. If the shortest distance is greater than the threshold D_{th}, the situation estimation result obtained using acceleration data is empty; otherwise, the class whose distance is the shortest is considered the result.

Actually, notifications are often taken in situations other than the assumed annotation classes. If the untrained acceleration data are fed to the system, the DTW distance of all annotation classes become large, and the result is erroneously output from among the classes because the system determines that the class with the shortest DTW distance is the result. D_{th} is employed to avoid outputting the result in unexpected situations. Here D_{th} is set to $\mu + 2\sigma$, where μ and σ are the mean and standard deviation. For example, if there are three types of annotations and five samples in the training data for each annotation, you can calculate the DTW distances of one sample and 14 samples excluding yourself. Among them, if both samples with the minimum DTW distance are the same annotation, the average and standard deviation are calculated, and this is performed for all 15 samples.

Table 1 Examples of annotation based on situation estimation results (C1, C2, and C3 are annotation classes; Φ is empty set; – represents no annotation given).

Cases	Response time	Acceleration	Annotation
#1	C1	C1	C1
#2	C1, C2	C1	C1
#3	C2	C1	–
#4	C1, C2	C3	–
#5	Φ	C1	–
#6	C1	Φ	–
#7	Φ	Φ	–

3.6 Annotation

Using the situation estimation results obtained using the response time and acceleration data, the proposed method determines whether an annotation is given. The judgment cases are listed in **Table 1**. If the response time and acceleration data results have a common class, the class is given to the acceleration data as an annotation (refer to cases #1 and #2 in Table 1). If the response time and acceleration results do not have a common class, no annotation is not given to the acceleration data (refer to cases #3 to #7 in Table 1).

4. Implementation

The Android OS includes the Notification Listener Service API [22], which can obtain information about notifications. Our android application obtains the notification type, content, notification generation time, notification removal time, sound volume at notification generarion, and vibration on/off setting at notification generation using the Notification Listener Service API. When installing our application, the user is required to turn on the notification access to the application, which differs from general permission confirmation, e.g., "Allow [app name] to access photos, media, and files on your device?"

Note that our application runs in the background. In addition, the user interface and data logging are separated in our app. By terminating the app using the task list, only the user interface stops (data logging continues to run in the background). By launching the app again, the user interface starts and data logging is unaffected. However, by selecting "DISABLE," "FORCE STOP," or "UNINSTALL" on the app info screen, all the application functions will stop (including data logging). The application collects acceleration and notification data. The application can receive Firebase Cloud Messaging (FCM) by Google [10]. FCM is a reliable cross-platform messaging solution that makes it possible to send and receive messages and notifications on iOS, Android, and the web. Data for the acquired notification information and acceleration data are stored in Firebase Cloud Storage.

5. Evaluation

We evaluate the accuracy of annotations obtained using the proposed method to verify its effectiveness. Two kinds of evaluation experiments were conducted. The first one was a laboratory environment where data were collected for the situations of the required annotation classes, and the annotation accuracy was evaluated. The second experiment was a natural environment where data were collected for two days (outside of the laboratory). In the natural environment, test subjects lived as usual and received notifications without considering the annotation classes; therefore, notifications were often received in unexpected situations. The purpose of this experiment was to observe how accurately annotations were given in expected situations and not given in unexpected situations.

5.1 Laboratory Experimental Environment

5.1.1 Setup

Five male subjects in their twenties participated in this experiment. The experiment was conducted in the laboratory at noon in November 2019. The subjects took annotations in seven situations of annotation classes: (1) standing while holding the smartphone unused in a hand; (2) using the smartphone and placing it on a table; (3) leaving the smartphone unused on a table; (4) holding the smartphone and using it; (5) standing with the smartphone in a pocket; (6) walking while holding the smartphone unused in a hand; (7) walking while using the smartphone. These data were used for training and testing.

In addition, for unexpected situations, the subjects took annotations for three additional situations: (8) smartphone in chest pocket; (9) leaving the smartphone unused on a bed; (10) standing with the smartphone in a bag. These data were only used for testing to determine whether the proposed method did not annotate sensor data in unexpected situations.

The subjects used a smartphone (ASUS ZenFone3, ZS570KL, Android 8.0.0, acceleration sampling rate of 400 Hz). One of the authors intentionally generated dummy notifications approximately 100 times for each situation using FCM from a laptop (Lenovo ThinkPad X1 Carbon, Windows 10) with 15-second intervals. A total of (10 situations) × (100 times) × (5 subjects) = 5,000 notifications were sent, of which 4,868 notifications were taken within 10 s and used in the evaluation.

Subjects were asked to swipe out the notifications on the smartphone's screen. Specifically, our instructions were *"swipe out the notification as soon as possible when you are aware of it"*. Note that the notifications were provided using sound and vibration; therefore, all notifications were basically swiped out. The user was then asked to return to the initial situation. This procedure was iterated 100 times with 15-second intervals. Then, we took a 30-minute break between different situations. Data in classes (1) to (5) were collected on one day, and data in classes (6) to (10) were collected on another day.

Each subject used the same smartphone. The smartphone was not locked. The subjects were instructed not to change the smartphone settings. The subjects were instructed to browse the internet while using the smartphone, and not to use any other applications. Regarding a pocket for class (5), the subjects wore the trousers they were wearing on that day. The subjects wore trousers with front side pockets and none of the subjects wore irregular trousers. In addition, regarding a chest pocket for class (8), the subjects were asked to wear a collared shirt that the authors provided. There was one chest pocket on the left chest. Regarding a bed for class (9), the users were asked to lie down on the bed that the authors provided and to place the smartphone next to the pillow. Then, when the notifications were generated, the subjects picked up the smartphone and swiped out the notifications with remaining lying down. Regarding a bag for class (10), the subjects were asked to use the same bag that the authors provided and put the smartphone in the bag. There was nothing in the bag, the bag had no zippers, and the bag was open during the experiment.

The data for situations (1) through (7) were used for training, and the data for situations (1) through (10) were used for testing. We applied two experimental designs, i.e., cross-validation across subjects (user-independent), and cross-validation across the samples per subject (user-dependent). To compare the proposed method, we tested a method that uses response only response time and a method that uses only acceleration data. The former method finds the class of maximum likelihood upon response time and outputs the class as an annotation if the likelihood exceeds the L_{th} (0.05; the same as the proposed method). The latter method finds the class of the shortest DTW distance from the acceleration data. Here the output is as an annotation if the distance is less than the D_th ($\mu + 2\sigma$ over the training data; the same as the proposed method).

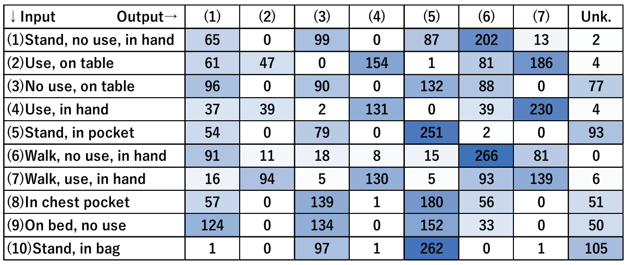
5.1.2 Results of User-independent Experiment

The precision, recall, and F-measure of the compared and proposed methods for the seven annotation classes in the user-independent design are shown in **Table 2**. In addition, to facilitate a thorough investigation, confusion matrices for the three methods in the user-independent design are shown in **Fig. 3**, **Fig. 4**, and **Fig. 5**. Here, "Unk." shows that the number of rejections, i.e., the number of inputs for which our system did not output annotations. Note that precision is the most important metric because the purpose of this work is annotation. Annotations do not have to be given to all the data; therefore, recall, i.e., coverage, is not critical. However, annotations given to sensor data must be very accurate; therefore, high precision, i.e., the accuracy of the output, is required.

The compared method using response time achieved the highest precision of 0.309 (average precision: 0.222). The compared method using acceleration achieved the best precision of 0.936 (average precision: 0.743). The proposed method achieved the best precision of 0.990 (average precision: 0.769). Among the three methods, the proposed method demonstrated the best pre-

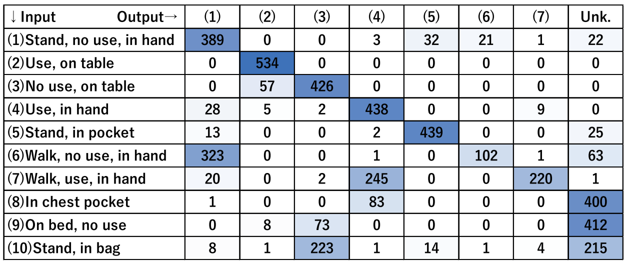
Table 2 Annotation accuracy of compared and proposed methods (user-independent design).

Annotation class	Response time			Acceleration			Proposal (Response+Acc)		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
(1) Stand, no use, in hand	0.108	0.139	0.121	0.497	0.831	0.622	0.509	0.632	0.564
(2) Use, on table	0.246	0.088	0.130	0.883	1.00	0.938	0.891	0.919	0.905
(3) No use, on table	0.136	0.186	0.157	0.587	0.882	0.705	0.534	0.524	0.529
(4) Use, in hand	0.308	0.272	0.289	0.567	0.909	0.698	0.618	0.780	0.690
(5) Stand, in pocket	0.231	0.524	0.321	0.905	0.916	0.911	0.990	0.628	0.769
(6) Walk, no use, in hand	0.309	0.543	0.394	0.823	0.208	0.332	0.879	0.178	0.295
(7) Walk, use, in hand	0.214	0.285	0.244	0.936	0.451	0.609	0.960	0.398	0.562
Average	0.222	0.291	0.237	0.743	0.742	0.688	0.769	0.580	0.616



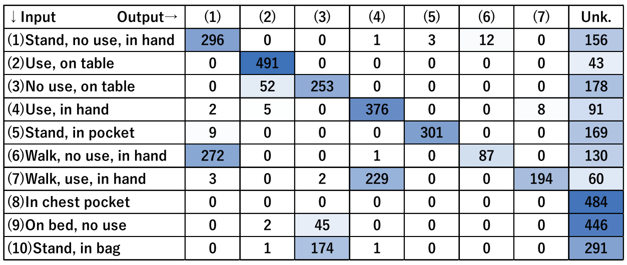
↓Input Output→	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Unk.
(1)Stand, no use, in hand	65	0	99	0	87	202	13	2
(2)Use, on table	61	47	0	154	1	81	186	4
(3)No use, on table	96	0	90	0	132	88	0	77
(4)Use, in hand	37	39	2	131	0	39	230	4
(5)Stand, in pocket	54	0	79	0	251	2	0	93
(6)Walk, no use, in hand	91	11	18	8	15	266	81	0
(7)Walk, use, in hand	16	94	5	130	5	93	139	6
(8)In chest pocket	57	0	139	1	180	56	0	51
(9)On bed, no use	124	0	134	0	152	33	0	50
(10)Stand, in bag	1	0	97	1	262	0	1	105

Fig. 3 Confusion matrix of annotations for compared method using response time in user-independent design.



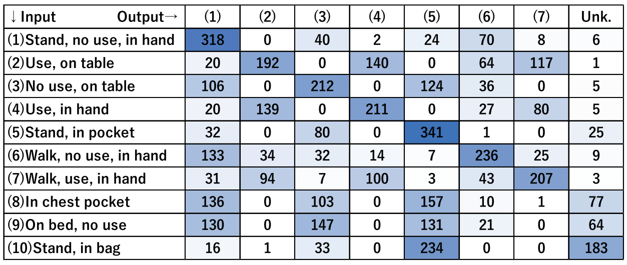
↓Input Output→	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Unk.
(1)Stand, no use, in hand	389	0	0	3	32	21	1	22
(2)Use, on table	0	534	0	0	0	0	0	0
(3)No use, on table	0	57	426	0	0	0	0	0
(4)Use, in hand	28	5	2	438	0	0	9	0
(5)Stand, in pocket	13	0	0	2	439	0	0	25
(6)Walk, no use, in hand	323	0	0	1	0	102	1	63
(7)Walk, use, in hand	20	0	2	245	0	0	220	1
(8)In chest pocket	1	0	0	83	0	0	0	400
(9)On bed, no use	0	8	73	0	0	0	0	412
(10)Stand, in bag	8	1	223	1	14	1	4	215

Fig. 4 Confusion matrix of annotations for compared method using acceleration data in user-independent design.



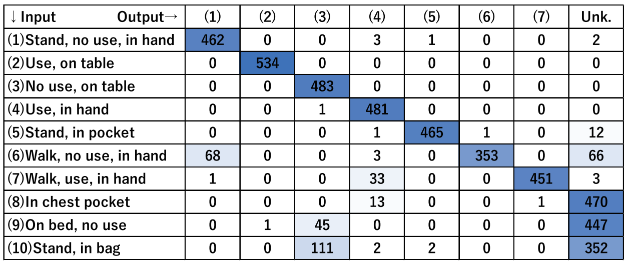
↓Input Output→	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Unk.
(1)Stand, no use, in hand	296	0	0	1	3	12	0	156
(2)Use, on table	0	491	0	0	0	0	0	43
(3)No use, on table	0	52	253	0	0	0	0	178
(4)Use, in hand	2	5	0	376	0	0	8	91
(5)Stand, in pocket	9	0	0	0	301	0	0	169
(6)Walk, no use, in hand	272	0	0	1	0	87	0	130
(7)Walk, use, in hand	3	0	2	229	0	0	194	60
(8)In chest pocket	0	0	0	0	0	0	0	484
(9)On bed, no use	0	2	45	0	0	0	0	446
(10)Stand, in bag	0	1	174	1	0	0	0	291

Fig. 5 Confusion matrix of annotations for the proposed method in user-independent design.



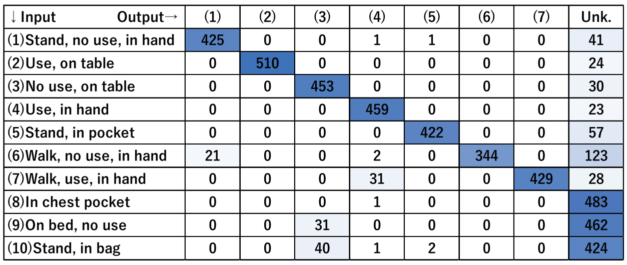
↓Input Output→	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Unk.
(1)Stand, no use, in hand	318	0	40	2	24	70	8	6
(2)Use, on table	20	192	0	140	0	64	117	1
(3)No use, on table	106	0	212	0	124	36	0	5
(4)Use, in hand	20	139	0	211	0	27	80	5
(5)Stand, in pocket	32	0	80	0	341	1	0	25
(6)Walk, no use, in hand	133	34	32	14	7	236	25	9
(7)Walk, use, in hand	31	94	7	100	3	43	207	3
(8)In chest pocket	136	0	103	0	157	10	1	77
(9)On bed, no use	130	0	147	0	131	21	0	64
(10)Stand, in bag	16	1	33	0	234	0	0	183

Fig. 6 Confusion matrix of annotations for compared method using response time in user-dependent design.



↓Input Output→	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Unk.
(1)Stand, no use, in hand	462	0	0	3	1	0	0	2
(2)Use, on table	0	534	0	0	0	0	0	0
(3)No use, on table	0	0	483	0	0	0	0	0
(4)Use, in hand	0	0	1	481	0	0	0	0
(5)Stand, in pocket	0	0	0	1	465	1	0	12
(6)Walk, no use, in hand	68	0	0	3	0	353	0	66
(7)Walk, use, in hand	1	0	0	33	0	0	451	3
(8)In chest pocket	0	0	0	13	0	0	1	470
(9)On bed, no use	0	1	45	0	0	0	0	447
(10)Stand, in bag	0	0	111	2	2	0	0	352

Fig. 7 Confusion matrix of annotations for compared method using acceleration data in user-dependent design.



↓Input Output→	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Unk.
(1)Stand, no use, in hand	425	0	0	1	1	0	0	41
(2)Use, on table	0	510	0	0	0	0	0	24
(3)No use, on table	0	0	453	0	0	0	0	30
(4)Use, in hand	0	0	0	459	0	0	0	23
(5)Stand, in pocket	0	0	0	0	422	0	0	57
(6)Walk, no use, in hand	21	0	0	2	0	344	0	123
(7)Walk, use, in hand	0	0	0	31	0	0	429	28
(8)In chest pocket	0	0	0	1	0	0	0	483
(9)On bed, no use	0	0	31	0	0	0	0	462
(10)Stand, in bag	0	0	40	1	2	0	0	424

Fig. 8 Confusion matrix of annotations for the proposed method in user-dependent design.

cision; however, 0.769 precision is insufficient for annotation. These results were due to the fact that the distribution of response times and the accelerations differed for each person.

5.1.3 Results of User-dependent Experiment

As with the user-independent design, here, the precision, recall, and F-measure of the compared and proposed methods for seven annotation classes in the user-dependent design are shown in **Table 3**. In addition, confusion matrices for the three methods in the user-dependent design are shown in **Fig. 6**, **Fig. 7**, and **Fig. 8**.

The compared method using response time achieved the best precision of 0.473 (average precision: 0.400), and the compared method using acceleration data achieved the best precision of 0.998 (average precision: 0.930). However, the proposed method achieved the best precision of 1.00 (average precision: 0.963). Therefore, among the three methods, the proposed method demonstrated the best performance.

5.1.4 Discussion

From the results obtained by the compared method using response time (Fig. 3 and Fig. 6), the output annotations were distributed on the nondiagonal cells, which indicate that most annotations were incorrect. For unexpected situations (8), (9), and (10), only 50 to 183 (of approximately 500) trials were identified correctly as unknown. It can be said that the distributions of response time for several annotation classes overlapped, and it was difficult to identify the situation by considering only the likelihood of response time.

From the results for the compared method using acceleration data in the user-dependent design (Fig. 7), we found that the output annotations for inputs (1) to (7) were almost correct. The in-

Table 3 Annotation accuracy of compared and proposed methods (user-dependent design).

Annotation class	Response time			Acceleration			Proposal (Response+Acc)		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
(1) Stand, no use, in hand	0.338	0.679	0.451	0.870	0.987	0.925	0.953	0.908	0.930
(2) Use, on table	0.417	0.360	0.387	0.998	1.00	0.999	1.00	0.955	0.977
(3) No use, on table	0.324	0.439	0.373	0.755	1.00	0.860	0.865	0.938	0.900
(4) Use, in hand	0.452	0.438	0.445	0.897	0.998	0.945	0.927	0.952	0.940
(5) Stand, in pocket	0.334	0.712	0.455	0.994	0.971	0.982	0.993	0.881	0.934
(6) Walk, no use, in hand	0.465	0.482	0.473	0.997	0.720	0.836	1.00	0.702	0.825
(7) Walk, use, in hand	0.473	0.424	0.447	0.998	0.924	0.960	1.00	0.879	0.936
Average	0.400	0.505	0.433	0.930	0.943	0.930	0.963	0.888	0.920

puts of an unexpected situation (10) were identified correctly as unknown (75.4%), which means the annotations were not given to the data in unexpected situations. With the compared method using acceleration data in the user-independent design (Fig. 4), the situations (6) and (7) produced many misclassifications, which may have been caused by the difference between individuals while walking.

Finally, the results for the proposed method in user-dependent design (Fig. 8) demonstrate that annotations incorrectly given by the compared method were omitted, as indicated by the results classified as unknown. For example, as shown in Fig. 7, class (6) was estimated as class (1), and class (10) was estimated as class (3). There were many incorrect estimation results; however, the distribution of response time for (1) and (3) is fast-sided because the user was stationary while holding the smartphone in their hand. Even if the acceleration values of the input data were close to the acceleration values of class (1) or (3), the response time was slower, which resulted in an unknown result by the proposed method (Fig. 8). The state that could not be omitted only by acceleration can be omitted by using response time together with acceleration values. Note that the precision increased by 0.033 over that of the comparison method.

5.2 Experiment in Natural Environment

In an actual use of our system, users will receive notifications in any situations other than our expectations. In order to observe how accurately the system filters out input data obtained in the unexpected situations and how correctly the system annotates input data obtained in the expected situations, we conducted an experiment where users receive notifications in daily life out of the laboratory.

5.2.1 Experimental Setup

Three of the five male subjects from the laboratory experiment (referred to as subjects A, B, and C) participated in this experiment. The subjects were asked to use the same smartphone as in the laboratory setting (ASUS Zen-Fone3, ZS570KL, Android 8.0.0, acceleration sampling rate of 400 Hz) for two days in their daily lives. In order to collect samples efficiently, dummy notifications were generated automatically approximately once every 30 minutes using the same laptop as in the laboratory setting (Lenovo ThinkPad X1 Carbon, OS Windows 10) via FCM using a Python program. We gave the subjects a smartphone in which our app was installed and asked them to use it freely. We also asked them to remove notifications only when they can do. The subjects recorded the groundtruth on the smarphone. When the user situation was one of classes (1) to (7), the corresponding number was input; otherwise, "other" was input. Here, the data collected in the laboratory experiment were used for training, and we performed two experiments, i.e., using the data of the other subjects for training (user-independent), and mixing all data for training (user-dependent).

Table 4 Number of notifications generated, removed, and removed within 10 s in natural environment.

Subject	User response			System output in user-dependent				System output in user-independent			
	Gen	Rm	Rm in 10 s	TP	TN	FN	FP	TP	TN	FN	FP
A	67	24	17	12	2	3	0	1	2	14	0
B	84	32	23	11	4	7	1	14	4	4	1
C	11	8	5	2	2	1	0	2	2	1	0

5.2.2 Results

The number of notifications generated, removed, and removed within 10 s, as well as the outputs of the proposed method in the user-dependent and user-independent designs for the subjects are shown in **Table 4**. For subject C, the number of notifications generated is small. Data for only five hours were recorded because network conditions were poor. In the table, TP stands for true positive and indicates the number of annotations correctly given for notifications removed within 10 s in classes (1) to (7); TN stands for true negative and indicates the number of "no annotation" correctly given for notifications removed within 10 s in other situations; FN stands for false negative and indicates the number of "no annotation" incorrectly given for notifications removed within 10 s in classes (1) to (7); FP stands for false positive and indicates the number of annotations incorrectly given for notifications removed within 10 s in any situations. Specifically, when the user removed a notification within 10 s in class (1) and our system gave no annotation, it is counted as FN. When the user removed a notification within 10 s in class (8) which should not be annotated and our system incorrectly gave annotation class (5), it is counted as FP.

The following part of this subsection explains the system output in detail. For subject A, 67 notifications were generated during the experiment, 24 of which were removed regardless of response time, and 17 of which were removed within 10 s. The detailed annotations given for the 17 notifications are shown in **Table 5**. For the table, 2, 12, 1, and 2 notifications were removed in classes (2), (3), (4), and other class, respectively. Of these, in the user-dependent design, 2 of 2 (hereafter, denoted as 2/2) for class (2) were correctly annotated, 9/12 for class (3) were correctly annotated, 1/1 for class (4) was correctly annotated, and 2/2 for other class were correctly annotated as no annotation; however, 3/12 for class (3) were incorrectly annotated as no annotation.

Table 5 Annotations given for notifications removed within 10 s for subject A in natural environment.

True class	Response in 10 s	Annotation user-dependent	Annotation user-independent
(1)	0		
(2)	2	(2):2	(2):1, no annotation:1
(3)	12	(3):9, no annotation:3	no annotation:12
(4)	1	(4):1	no annotation:1
(5)	0		
(6)	0		
(7)	0		
other	2	no annotation:2	no annotation:2
total	17		

Table 6 Annotations given for notifications removed within 10 s for subject B in natural environment.

True class	Response in 10 s	Annotation user-dependent	Annotation user-independent
(1)	0		
(2)	0		
(3)	15	(3):11, no annotation:4	(3):14, no annotation:1
(4)	3	(1):1, no annotation:2	(1):1, no annotation:2
(5)	1	no annotation:1	no annotation:1
(6)	0		
(7)	0		
other	4	no annotation:4	no annotation:4
total	23		

Therefore, TP is 12, TN is 2, FN is 3, and FP is 0. Here, the annotation precision was 1.00 (12/12), and recall was 0.80 (12/15). In the user-independent design, 1/2 for class (2) and 2/2 for other class were correctly annotated; however, 1/2 for class (2), 12/12 for class (3), and 1/1 for class (4) were incorrectly annotated as no annotation. Therefore, TP is 1, TN is 2, FN is 14, and FP is 0. Here, the annotation precision was 1.00 (1/1), and recall was 0.07 (1/15).

For subject B, 84 notifications were generated during the experiment, 32 of which were removed regardless of response time, and 23 of which were removed within 10 s. The detailed annotations given for the 23 notifications are shown in **Table 6**. For the table, 15, 3, 1, and 4 notifications were removed in classes (3), (4), (5), and the other class, respectively. Of these, in the user-dependent design, 11/15 for class (3) and 4/4 for other class were correctly annotated; however, 4/15 for class (3), 2/3 for class (4), 1/1 for class (5) were incorrectly annotated as no annotation, and 1/3 for class (3) was incorrectly annotated as class (1). Therefore, TP is 11, TN is 4, FN is 7, and FP is 1. Here, the annotation precision was 0.92 (11/12), and the recall was 0.61 (11/19). In the user-independent design, 14/15 for class (3) and 4/4 for other class were correctly annotated. 1/15 for class (3), 2/3 for class (4), and 1/1 for class (5) were incorrectly annotated as no annotation, and 1/3 for class (4) was incorrectly annotated as class (1). Therefore, TP is 14, TN is 4, FN is 3, and FP is 1. Here, the annotation precision was 0.93 (14/15), and recall was 0.74 (14/19).

For subject C, 11 notifications were generated during the experiment, 8 of which were removed regardless of response time, and 5 of which were removed within 10 s. The detailed annotations given for the five notifications are shown in **Table 7**. For the table, 1, 1, 1, and 2 notifications were removed in classes (1), (3), (5), and other class, respectively. Of these, in user-dependent design, 1/1 for class (1), 1/1 for class at (3), and 2/2 for other class were correctly annotated; however, 1/1 for class (5) was incorrectly annotated as no annotation. Therefore, TP is 2, TN is 2, FN is 1, and FP is 0. Here, the annotation precision was 1.00 (2/2), and recall was 0.67 (2/3). In the user-independent design, 1/1 for class (1), 1/1 for class (3), and 2/2 for other class were correctly annotated; however, 1/1 for class (5) was incorrectly annotated as no annotation. Therefore, TP is 2, TN is 2, FN is 1, and FP is 0. Here, the annotation precision was 1.00 (2/2), and recall was 0.67 (2/3).

Table 7 Annotations given for notifications removed within 10 s for subject C in natural environment.

True class	Response in 10 s	Annotation user-dependent	Annotation user-independent
(1)	1	(1):1	(1):1
(2)	0		
(3)	1	(3):1	(3):1
(4)	0		
(5)	1	no annotation:1	no annotation:1
(6)	0		
(7)	0		
other	2	no annotation:2	no annotation:2
total	5		

5.2.3 Discussion

From the results, we found that subject A obtained more annotations in the user-dependent design than the user-independent design because the response of subject B was quick in the laboratory environment, which did not fit the distribution of subject A's response time in the natural environment. However, further investigations are required to evaluate the performance of the proposed method using training data from many people in a user-independent design.

For the three subjects, the annotation precision was 0.92+ but recall was low, which means accurate annotation can be given to the limited amount of data; However, since the proposed method has an automatic annotation collection mechanism, even if the recall is small, it can collect a large amount of annotated data for a large number of people for a long period of time.

6. Limitations

There are innumerable situations in real life. This paper employed only a few situations. As an example, class (5) smartphone in the pocket was assumed standing still, and our method may incorrectly annotate data as class (5) when the user takes a smartphone from pocket and removes a notification while *sitting*, or may not annotate data.

In addition, test subjects were all male and wore similar clothes. Clothes for females have more diverse design, and the position of pockets is different from clothes to clothes. The proposed system has to learn the data with target clothes to annotate the data. However, even if the proposed system has not learned the data with several users' clothes, our system would not give annotations to the data rather than give incorrect annotation. If a researcher wants to add a new kind of annotation, its training data is needed, which may interfere with other annotation classes. A scaled investigation with many users recruited through crowdsourcing for the long term is our future study.

7. Conclusion

In this paper, we have proposed a method to estimate the user and device situations from the user responses to the notifications generated by a smartphone. In an evaluation experiment, an average precision of 0.769 and 0.963 for user-independent and user-dependent experiments was obtained by the proposed method, respectively. We also tested the proposed method in a natural environment, where 25 correct annotations were given for 45 responses to notifications with only a single incorrect notification. In future, we will conduct long-term experiments to determine whether the proposed method is effective in various unexpected situations. We will also evaluate the improvement in the performance of classifier by using data annotated with the proposed method. In addition, we will verify the method in other user-independent situations.

Acknowledgments This research was supported in part by ACT-I (JPMJPR17UR), the Japan Science and Technology Agency.

References

[1] Mehrotra, A., Hendley, R. and Musolesi, M.: PrefMiner: Mining User's Preferences for Intelligent Mobile Notification Management, *UbiComp 2016*, pp.1223–1234 (2016).

[2] Felt, A.P., Egelman, S. and Wagner, D.: I've got 99 problems, but vibration ain't one: A survey of smartphone users' concerns, *SPSM 2012*, pp.33–44 (2012).

[3] Shirazi, A.S., Henze, N., Dingler, T., Pielot, M., Weber, D. and Schmidt, A.: Large-scale assessment of mobile notifications, *CHI 2014*, pp.3055–3064 (2014).

[4] Ho, B.-J., Balaji, B., Koseoglu, M. and Srivastave, L.M.: Nurture: Notifying Users at the Right Time Using Reinforcement Learning, *UbiComp 2018*, pp.1194–1201 (2018).

[5] Myers, C., Rabiner, L. and Rosenberg, A.: Performance tradeoffs in dynamic time warping algorithms for isolated word recognition, *IEEE Trans. Acoustics, Speech, and Signal Processing*,Vol.ASSP-28, No.6, pp.623–635 (1980).

[6] CATAPULT, available from ⟨https://www.catapultsports.com/⟩.

[7] Shen, C.-L., Kao, T., Huang, C.-T. and Lee, J.-H.: Wearable band using a fabric-based sensor for exercise ECG monitoring, *ISWC 2006*, pp.143–144 (2006).

[8] Lewis, D.D. and Gale, W.A.: A sequential algorithm for training text classifiers, *17th ACM International Conference on Research and Development in Information Retrieval*, pp.3–12 (1994).

[9] Cutell, E., Crezwinski, M. and Horvitz, E.: Notification, Disruption, and Memory: Effects of messaging interruptions on memory and performance, *INTERACT 2001*, pp.236–239 (2001).

[10] Firebase Cloud Messaging, available from ⟨https://firebase.google.com/⟩.

[11] Naya, F., Ohmura, R., Takayanagi, F., Noma, H. and Kogure, K.: Workers' Routine Activity Recognition using Body Movements and Location Information, *ISWC 2006*, pp.105–108 (2006).

[12] Mark, G., Iqbal, S.T., Czerwinski, M., Johns, P. and Sano, A.: Email Duration, Batching and Self-interruption: Patterns of Email Use on Productivity and Stress, *CHI 2016*, pp.1717–1728 (2016).

[13] Zhu, J., Wang, H., Hovy, E. and Ma, M.: Confidence-based stopping criteria for active learning for data annotation, *ACM Trans. Speech Lang. Process.*, Vol.6, No.3 (2010).

[14] Fischer, J.E., Yee, N., Bellotti, V., Good, N., Benford, S. and Greenhalgh, C.: Effects of content and time of delivery on receptivity to mobile interruptions, *MobileHCI 2010*, pp.103–112 (2010).

[15] Ouchi, K., Suzuki, T. and Doi, M.: Lifeminder: A wearable healthcare support system using user's context, *IWSAWC 2002*, pp.791–792 (2002).

[16] Murao, K. and Terada, T.: Labeling Method for Acceleration Data using an Execution Sequence of Activities, *HASCA 2013, UbiComp Adjunct*, pp. 611–622 (2013).

[17] Van Laerhoven, K. and Gellersen, H.W.: Spine versus porcupine: A study in distributed wearable activity recognition, *ISWC 2004*, pp.142–149 (2004).

[18] Larson, R. and Csikszentmihalyi, M.: The experience sampling method, *New Directions for Methodology of Social and Behavioral Science* (1983).

[19] Stikic, M., Larlus, D. and Schiele, B.: Multi-graph based semi-supervised learning for activity recognition, *ISWC 2009*, pp.85–92 (2009).

[20] Stikic, M., Van Laerhoven, K. and Schiele, B.: Exploring semi-supervised and active learning for activity recognition, *ISWC 2008*, pp.81–88 (2008).

[21] Toda, M., Akita, J., Sakurazawa, S., Yanagihara, K., Kunita, M. and Iwata, K.: Wearable Biomedical Monitoring System Using TextileNet, *ISWC 2006*, pp.119–120 (2006).

[22] Notification Listener Service, available from ⟨https://developer.android.com/reference/android/service/notification/NotificationListenerService⟩.

[23] Chapelle, O., Scholkopf, B. and Zien, A.: Adaptive computation and machine learning, *MIT Press* (2006).

[24] Vinyals, O., Toshev, A., Bengio, S. and Erhan, D.: Show and Tell: A Neural Image Caption Generator, *2015 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR2015*), pp.3156–3164 (2015).

[25] Izuta, R., Murao, K., Terada, T. and Tsukamoto, M.: Early Gesture Recognition Method with an Accelerometer, *International Journal of Pervasive Computing and Communications*, Vol.11, pp.270–287 (2015).

[26] Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T. and Saenko, K.: Sequence to Sequence – Video to Text, *2015 IEEE International Conference on Computer Vision* (*ICCV 2015*), pp.4534–4542 (2015).

[27] Okoshi, T., Ramos, J., Nozaki, H., Nakazawa, J., Dey, A.K. and Tokuda, H.: Attelia: Reducing User's Cognitive Load due to Interruptive Notifications on Smartphones, *PerCom 2015*, pp.96–104 (2015).

[28] Okoshi, T., Ramos, J., Nozaki, H., Nakazawa, J., Dey, A.K. and Tokuda, H.: Reducing users' perceived mental effort due to interruptive notifications in multi-device mobile environments, *UbiComp 2015*, pp.475–486 (2015).

[29] Okoshi, T., Tsubouchi, K., Taji, M., Ichikawa, T. and Tokuda, H.: Attention and Engagement-Awareness in the Wild: A Large-Scale Study with Adaptive Notifications, *PerCom 2017*, pp.100–110 (2017).

[30] Huynh, T. and Schiele, B.: Towards less supervision in activity recognition from wearable sensors, *ISWC 2006*, pp.3–10 (2006).

[31] Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P. and Troster, G.: Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario, *ISWC 2006*, pp.97–104 (2006).

Ryota Sawano received a B.Eng. degree from Ritsumeikan University in 2020. He is currently a student in the master course at Ritsumeikan University, Japan. He is interested in wearable computing and mobile computing.

Kazuya Murao is an Associate Professor at the College of Information Science and Engineering, Ritsumeikan University, Japan. He is also a PRESTO researcher at the Japan Science and Technology Agency, Japan. He received his B.Eng., M.Info.Sci., and Ph.D. degrees from Osaka University in 2006, 2008, and 2010, respectively. From 2011 to 2014, he was an Assistant Professor at Kobe University, Japan. From 2014 to 2017, he was an Assistant Professor at Ritsumeikan University, Japan. He is currently investigating wearable computing, ubiquitous computing, and human activity recognition. He is a member of IEEE and ACM.