Sensors and Materials, Vol. 36, No. 10 (2024) 4501–4518
MYU Tokyo

S & M 3814

User Identification via Touch-screen Button Operation for Smart Home

Shigemi Ishida,* Kyohei Suda, and Hiroshi Inamura

Graduate School / School of Systems Information Science, Future University Hakodate,
116-2 Kamedanakanocho, Hakodate-shi, Hokkaido 041-8655, Japan

(Received July 1, 2024; accepted August 28 2024)

Keywords: user-aware device usage detection, user identification, touch-screen operation, machine learning

In smart homes, user-aware device usage detection is one of the fundamental tasks. User identification methods with no burden to users have been proposed. However, these methods rely on camera images, which have privacy issues for in-home scenarios. In this paper, we present a user identification method via a touch-screen button operation. The key idea is to utilize users' habits of button operations to identify users. We extract features from a time series of touch-screen operation data and identify users using supervised learning. Our experimental evaluations demonstrated that our user identification method identified users with an accuracy of 94.4%. With the limited amount of training data obtained in 10 trials, the accuracy was 92.8% when we used the latest training data, confirming the feasibility of our user identification method.

1. Introduction

User-aware device usage detection is one of the fundamental tasks in smart homes. User-aware device usage data are used for automatic or semi-automatic home appliance control in smart homes. For example, we might want to know who turned on a heater or a TV, and who pressed a brew button on a coffee machine to automatically control or suggest controlling these home appliances. User-aware device usage data, which can be considered an individual's activities of daily living data, are also useful for evaluating the health and ability of independence of elderly people in a multi-resident environment.

User authentication is a simple and promising approach to identifying a user. In smart homes, authentication is impractical owing to its high cost and intrusive characteristics. Non-intrusive gesture-based authentication has also been proposed,[1,2] but it hardly motivates people to perform specific gestures in a home.

Low-effort user identification methods have also been presented for multi-touch devices.[3–5] In these methods, users were identified on the basis of the features obtained from camera images, which are impractical for in-home user identification owing to privacy issues.

*Corresponding author: e-mail: ish@fun.ac.jp
Conference paper[36] containing preliminary results of this paper appeared in ICMU 2023.
https://doi.org/10.18494/SAM5211

In this paper, we present a user identification method based on a touch-screen button operation. Pohl *et al.* proposed a user identification method using a physical button with pressure and distance sensors.[6] Our method is on the same line and is designed for touch-screen devices using a smaller number of sensors. The key idea is to employ users' habits of button operations to identify users. We extract features from a time series of sensor data derived from a touch-screen device and identify users using supervised learning. This simple idea is based on the fact that most people have never learned how to press a button from other people. We have never been taught how long and strong we need to press a button on each home appliance. The button-press behavior is highly dependent on individuals.

A typical use-case scenario of the proposed user identification method is a user-specific automatic configuration. When a user presses the power button on a coffee machine, the coffee machine recognizes the user and completes the user-specific coffee configuration. The user only needs to press a brew button without any configuration. In this scenario, we want a low-effort and low-cost user identification method.

Specifically, our main contributions are twofold:

- We present a user identification method via button-press operations on touch-screen devices. We use a time series of press pressure data to distinguish users. Unlike the existing physical button-based user identification methods, we require no additional sensor and rely only on a touch-screen device.
- Experimental evaluations were conducted for touch-screen buttons with various types of button feedback to demonstrate the user identification performance of the proposed method. The experimental evaluations revealed that the proposed method identified users with an accuracy of more than 90% even with the limited amount of training data of 10 trials.

The remainder of this paper is organized as follows. In Sect. 2, we look through related work. In Sect. 3, we present the proposed user identification method, followed by experimental evaluations in Sect. 4. Finally, we conclude the paper in Sect. 5.

2. Related Work

User-aware device usage detection can be considered a simple activity recognition in smart-home scenarios. This study is related to in-home activity recognition, multi-person activity recognition, and user identification. In this section, we briefly look through related work in these fields.

2.1 In-home activity recognition

Human activity recognition in smart homes has been widely studied to realize smart services such as elderly monitoring, energy-efficient home control, and healthcare. Human activity recognition is divided into intrusive and non-intrusive approaches.

Intrusive approaches rely on wearable sensors attached to a human body to recognize an activity. In the literature, smartwatch-based activity recognition has been reported, which estimates activities mainly relying on wearable sensors such as a smartwatch.[7–12] Intrusive

approaches enable us to know who is doing what, although all users are required to wear a smartwatch, which is sometimes impractical owing to the high cost of wearable sensors.

Non-intrusive in-home activity recognition uses sensors embedded in a home. Activity recognition using electric current sensors,[13] passive infrared ray (PIR) and door sensors,[14] or wireless sensor networks[15] has been reported. To support complex activities such as concurrent or intermittent activities, deep-learning approaches have also been proposed.[16–18] Although these approaches show high activity recognition performance, no considerations have been taken on user identification.

Camera-based non-intrusive activity recognition has also been proposed.[19–24] This method uses an RGB or depth camera to estimate activities. Camera-based approaches are impractical for in-home scenarios owing to their privacy issues.

2.2 Multi-person activity recognition

In-home multi-person activity recognition is a challenging task. Several researchers have studied simultaneous user identification and activity recognition. The pioneering work attempts to map an event captured by sensors to a person to realize multi-person activity recognition using a naïve Bayesian classifier.[25] Multi-label classification is also a popular approach to multi-person activity recognition,[26,27] where activities by different people are labeled as a unique activity. These naive approaches have difficulties in practical environments. The changes of people and room environments highly affect the recognition performance.

Recent papers, such as the fuzzy c-means change point detection (CPD)-based approach,[28] sMRT,[29] GAMUT,[30] and MICAR,[31] have presented simultaneous user identification and activity recognition. Chen *et al.* presented Seq2Res and BiGRU+Q2L for multi-person in-home activity recognition.[32] The Seq2Res assigns sensor events to specific residents using a sequence-to-sequence architecture. The BiGRU+Q2L applies an attention mechanism to the segmented sensor data sequence, estimating the activity of each person.

Although these methods successfully realize multi-person activity recognition, a sufficient amount of data from many sensors is required. Our user identification method relies on the data derived only from a touch-screen device.

2.3 User identification

User identification can be easily realized by authentication. Biometric fingerprint authentication is prevalent nowadays and is popularly used on smartphones; however, authentication is unusual on home appliances owing to their high cost.

Gesture-based authentication, which uses body movements to identify users, is a popular non-intrusive authentication method. Mare *et al.* presented a user authentication method based on a smartphone lifting gesture determined by a wristband sensor.[1] Zhao and Tanaka reported a gesture authentication method using a Leap Motion hand-tracking device.[2] These methods require a user to repeat the same action for more than 20 trials or for 30 to 40 min to collect authentication data, which is impractical for an in-home scenario.

As wearable devices such as smartwatches become prevalent, touch-screen-based authentication for small screen devices has been studied. Zhao *et al.* raised an awareness of the unstudied nature of smartphone touch-screen authentication ported to smartwatches.[33] Owing to the limited size of a smartwatch display, the number of buttons on a screen is limited. In addition to buttons, the data derived from button-press operations are employed to achieve a higher authentication accuracy with the limited number of buttons. Song and Oakley presented an authentication method using pressure data on four buttons and hand movement acceleration data.[34,35] They used a wrist-worn touch-screen device capable of five-force-level input on a four-key interface. They collected press pressure and press position data as well as wrist motion data obtained as acceleration and gyroscope data in the *x*-, *y*-, and *z*-axes while a user sequentially pressed a four-digit passcode. Users were identified using a classifier on the basis of features extracted from the press pressure, position, and wrist motion data. This method requires a motion sensor, while our method requires the data derived only from a touch screen.

For single-button-operation user identification, Pohl *et al.* proposed a user identification method based on home appliance button operations.[6] Pohl *et al.* used a pressure-sensitive button containing a pressure sensor and an infrared (IR) distance sensor to capture an individual's button-press behavior and identified the features extracted from the pressure and IR distance data. However, this method requires the installation of distance and pressure sensors into press buttons, which is impractical for in-home scenarios owing to the installation cost.

In this paper, we only rely on button pressure data that can be derived from button switches. Our aim is to identify users, not authenticate them. We perform user identification with pressure data of a single button operation directly derived from button switches without additional sensors by sacrificing user identification performance.

In this paper, we present the basics of a user identification method based on touch-screen button operations in our preliminary work.[36] We extend our preliminary work to study more the user identification performance. Touch-screen devices provide no physical-move feedback, but provide visual, acoustic, and haptic feedback. We determine the effect of feedback on user identification performance in this study.

3. User Identifier Based on Touch-screen Operation

3.1 System overview

Our key idea is to identify users using their habits of home appliance operations. We have never been taught how long and strong to press a button on home appliances. We learn press operation by ourselves. The way users press buttons is therefore dependent on them. We extract features of button-press operations and identify users using supervised learning.

Figure 1 depicts an overview of our user identifier. The user identifier consists of data collection, feature extraction, and user identification blocks. When a user presses a touch-screen button, the data collection block collects the touch position and pressure data as time series sequences. The feature extraction block extracts features that represent users' specific behavior in button-press operations. The user identification block finally estimates who the user is using a

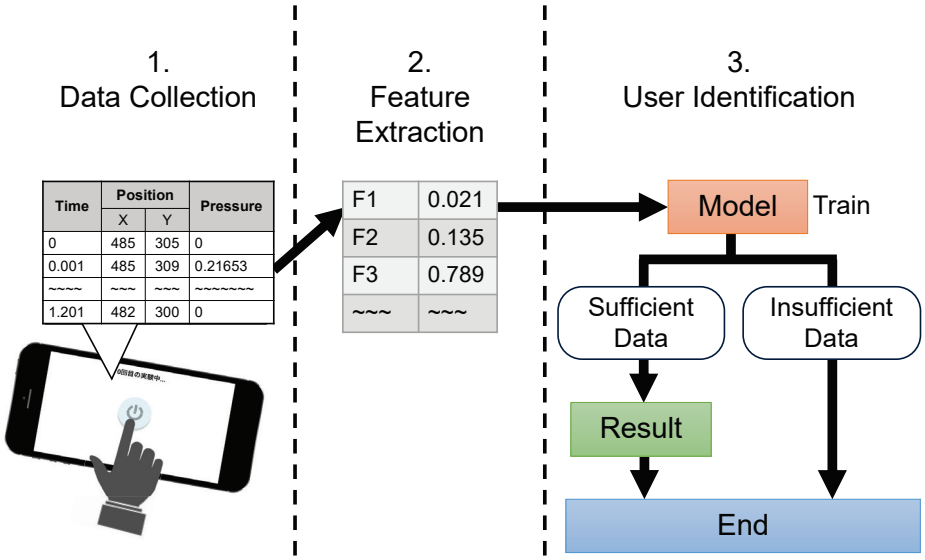

Fig. 1. (Color online) System overview.

supervised machine learning model. The machine learning model is trained with features passed from the feature extraction block until a sufficient amount of data is collected. During the model training phase, no user identification is performed.

To train the machine learning model, training data, i.e., data with a user label, are required. We assume that we have training data and focus on user identification in this paper. We believe that training data can be easily collected with semi-supervised and self-supervised machine learning during the daily use of home appliances.

In the following subsections, we present design details of each block.

3.2 Data collection

The data collection block collects data from a touch-screen device when a user presses a button. From a touch-screen device, press position in coordinates and press pressure values, which are real numbers in the range of 0–1, are collected. In the data collection block, these press data are stored as a time-series sequence with timestamps.

Let $\boldsymbol{s}_i$ be a single data sample, where is an index number. $\boldsymbol{s}_i$ is a four-dimensional vector described as

$$\boldsymbol{s}_i = \begin{bmatrix} t_i & x_i & y_i & p_i \end{bmatrix}^T, \tag{1}$$

where t_i, x_i, y_i, and p_i are the timestamp, x, y coordinates, and pressure, respectively, and T denotes the transpose operation. The single operation data D is a sequence of $\boldsymbol{s}_i$:

$$D = \{\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_{n-1}\}, \tag{2}$$

where $i = 0$ and $i = n - 1$ indicate the start and end of a single operation, respectively.

Note that the sampling rate depends on a touch-screen device and the implementation of the data collection block. The time difference $t_i - t_{i-1}$ between successive samples might not be constant. The implementation of a data collector for touch-screen devices might rely on an event-triggering function. In such an implementation, the sampling rate is inconstant because events such as touch-position and pressure changes occur at a non-uniform rate.

We consider that users have completed a press operation when a specific time had passed after the time the last data sample was obtained. If the implementation provides data samples at a low rate, it means that our proposed method cannot be applied because it cannot distinguish a large time difference of samples and the end of operation. We believe that this is a small limitation of our method because data sampling at less than 100 Hz is normally acceptable in recent devices.

3.3 Feature extraction

The feature extraction block extracts 28 dimensional features below, which are passed to the user identification block for supervised learning.

- Operation time length: $t_{n-1} - t_0$
  The time length of operation is dependent on individuals because button-press recognition is dependent on individuals.
- Number of data samples: n
  Here, we assume that the sampling rate is not constant. n is not proportional to operation time length. The number of data samples, which corresponds to the time length of button-press position and pressure changes, is dependent on the speed of a button operation.
- First coordinates obtained: x_0, y_0
  We assume that the button-press position is dependent on individuals.
- Range of x, y-axis position trajectory: $\max_{\boldsymbol{s}_i \in D} x_i - \min_{\boldsymbol{s}_i \in D} x_i, \max_{\boldsymbol{s}_i \in D} y_i - \min_{\boldsymbol{s}_i \in D} y_i$
  Normally, we do not slide our finger while pressing a button. The range of x, y-axis position trajectory indicates the size of the finger used for pressing the button.
- Mean, median, and standard deviation of the speed and acceleration of the position trajectory:

$$V = \left\{ \left. \frac{\sqrt{\left(x_i - x_{i-1}\right)^2 + \left(y_i - y_{i-1}\right)^2}}{t_i - t_{i-1}} \right| 1 < i < n-1 \right\}, \tag{3}$$

$$A = \left\{ \left. \frac{v_i - v_{i-1}}{t_i - t_{i-1}} \right| 2 \le i \le n-1 \right\}. \tag{4}$$

The mean, median, and standard deviation of sequences V and A are then calculated. While pressing a button, a small movement of the hand and finger results in the change in press position. The mean, median, and standard deviation of the speed and acceleration of the

position trajectory therefore include the effect of how individuals move their hand and finger while pressing a button.

- Total length of the position trajectory: $\sum_{i=1}^{n-1} v_i \left(t_i - t_{i-1}\right)$
  The total length of the position trajectory indicates how much individuals move their hand and finger while pressing a button.
- Time required for the deep press:
  We first create a set D_{deep} of data samples under a deep-pressed state, i.e., data samples whose pressure is 1:

$$D_{deep} = \left\{ \boldsymbol{s}_i \in D \mid p_i = 1 \right\}. \tag{5}$$

- We then find the first element to calculate the time required for the deep press:

$$\left(\min_{\boldsymbol{s}_i \in D_{deep}} t_i \right) - t_0. \tag{6}$$

  The time required for the deep press is the time required to reach the *completely pressed* state, which corresponds to the speed of button press.
- Time ratio of deep press to the total operation time:
  We first create a set T_{deep} of the time required for the deep press:

$$T_{deep} = \left\{ t_i - t_{i-1} \middle| \, p_i = 1, 1 \le i \le n-1 \right\}. \tag{7}$$

  The time ratio of the deep press to the operation time is then calculated as

$$\frac{1}{t_{n-1} - t_0} \sum\nolimits_{\tau \in T_{deep}} \tau. \tag{8}$$

  The time ratio of the deep press to the total operation time corresponds to the time required to reach the *completely pressed* state.
- Time required for the pressure to reach the button-press decision threshold:
  Touch-screen buttons usually have a threshold such that each button is considered to have been pressed when the pressure exceeds the threshold. Let the button-press decision threshold be pth. We create a set D_{press} of data samples under a button-pressed state:

$$D_{press} = \left\{ \boldsymbol{s}_i \in D \ \middle| p_i \ge p_{\mathrm{th}} \right\}. \tag{9}$$

  The time required for the pressure to reach the button-press decision threshold is then calculated as

$$\left(\min_{\boldsymbol{s}_i \in D_{press}} t_i \right) - t_0. \tag{10}$$

The time required for the pressure to reach the button-press decision threshold indicates the pressing speed until the button-press decision, which corresponds to the speed of pressing the button.

- Final pressure obtained: p_{n-1}
  The final pressure obtained corresponds to the speed of button release. Owing to the limitation of hardware response, the final pressure obtained becomes higher when a user rapidly releases the button.
- Distribution of pressure values in a single operation:
  The pressure range is [0, 1]. We divide the pressure range into 11 bins and define a set D_i of data samples whose pressure is in bin i:

$$D_i = \{\boldsymbol{s}_i \in D \,|\, 0.1i \le p_i < 0.1(i+1)\},\ 0 \le i \le 10. \tag{11}$$

The pressure distribution is derived by calculating the ratio of the size of D_i to that of D:

$$\frac{|D_i|}{|D|} = \frac{1}{n}|D_i|,\ 0 \le i \le 10. \tag{12}$$

The pressure distribution in a single operation indicates how the press pressure changes excluding time information.

Table 1 shows the list of 28 features that we extracted. These features are defined referring to our previous work.[37] We discuss the effectiveness of these features in Sect. 4.3.

3.4 User identification

The user identification block performs classification using a supervised machine learning algorithm to identify users. We do not limit the classification algorithm. A classification algorithm that supports multi-class classification is required because we assume more than one user. We first use a support vector machine (SVM) classifier with a linear kernel in our evaluation, referring to our previous work.[37] The user identification performance characteristics of various machine learning algorithms are compared in Sect. 4.4.

Prior to performing classification, each feature is standardized, i.e., Z-score normalized, to have a mean of 0 and a standard deviation of 1. Let Z_{train} and Z_{test} denote sets of a single feature of training and test data, respectively. The standardized test data set $\widetilde{Z}_{test}$ of the feature is derived as

Table 1
Features.

Features (Dimension)	Position	Pressure
Operation time length (1)	✓	✓
Number of data samples (1)	✓	✓
First x, y coordinates obtained (2)	✓	
Range of x, y-axis position trajectory (2)	✓	
Mean, median, and standard deviation of the speed and acceleration of the position trajectory (6)	✓	
Total length of the position trajectory (1)	✓	
Time required for the deep press (1)		✓
Time ratio of the deep press to the total operation time		✓
Time required for the pressure to reach the button-press decision threshold (1)		✓
Final pressure obtained (1)		✓
Pressure distribution in a single operation (11)		✓

$$\tilde{Z}_{test} = \left\{ \frac{z-\mu}{\sigma} \middle| z \in Z_{test} \right\}, \tag{13}$$

where μ and σ are the mean and standard deviation of Z_{train}, respectively.

4. Experimental Evaluation

To confirm the effectiveness of the user identification method presented in Sect. 3, we conducted experimental evaluations. We first evaluated the effect of the features presented in Sect. 3.3 on user identification accuracy. We then evaluated the effect of a machine learning algorithm on user identification accuracy. The impact of various types of feedback on user identification is studied with the effective features and machine learning algorithm. We also evaluated the effects of the amount of training data and short-term button-operation experience on user identification accuracy.

4.1 Implementation

Figure 2 shows the screen images of the data collector we implemented. The data collector is a Web application written in JavaScript deployed on Google Firebase Hosting. The button imitates a power button because almost all home appliances have a power button.

We derived press position and pressure data using a JavaScript event and Pressure.js library (https://pressurejs.com/), respectively. Note that the value provided by the Pressure.js library is a pressure such that the min–max range defined by sensor hardware is mapped to the range of 0–1. Pressure data can be derived on specific platforms with the Pressure.js library. We used an iPhone7 smartphone in our experiments.

In our experiment, the sampling rate of the data collection was not constant because we used an event-triggered function provided by the Pressure.js library in our implementation. As a

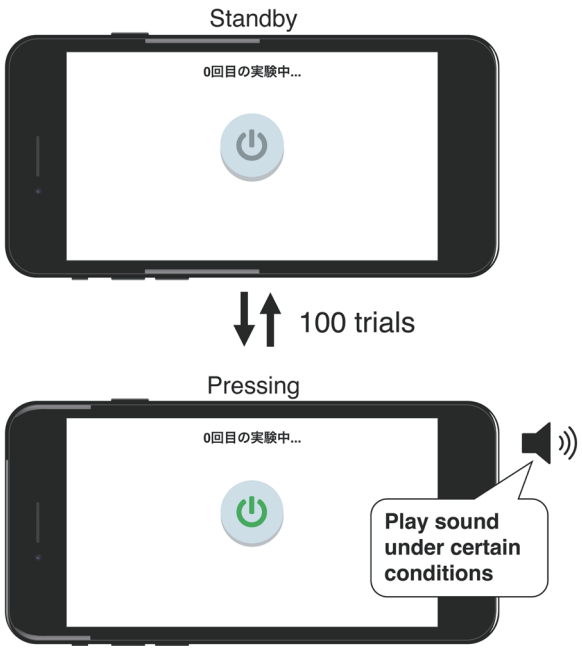

Fig. 2. (Color online) Screen images of the implemented button-press operation data collector.

reference, Fig. 3 shows an empirical cumulative probability of time differences between successive samples. Although the sampling rate was not consistent, we can confirm that almost all the samples were collected in less than 0.05 s.

4.2 Experimental setup

Figure 4 shows the experimental setup. The iPhone7 smartphone was almost vertically fixed firmly at the corner of the desk. We opened the data collector Web on the iPhone7 on a full-screen Web browser. We asked subjects to sit on a chair and repeat 100 trials of pressing the button on the iPhone7 smartphone screen. To make each trial an independent button operation, subjects were instructed to put their hands down each time the button was pressed. Note that the subjects had sufficient time to practice pressing the button before the experiment. This experiment was approved by the Ethics Committee of Future University Hakodate (permission #2021016).

Our user identification method aims to be used for sensing in smart homes. We assume that there are a few users living in a smart home. According to the Population Census in Japan 2020, 94.41% of households consist of 1–4 household members (Basic Complete Tabulation on Population and Households, 2020 Population Census, Population Census, https://www.e-stat.go.jp/en/stat-search/files?stat_infid=000032142481). We evaluated the user identification accuracy with four users in this paper.

The number of subjects is more than four and depends on the evaluation. We calculated user identification accuracy for each combination of four subjects among all the subjects. For each combination of four subjects, we extracted features from the data corresponding to the four

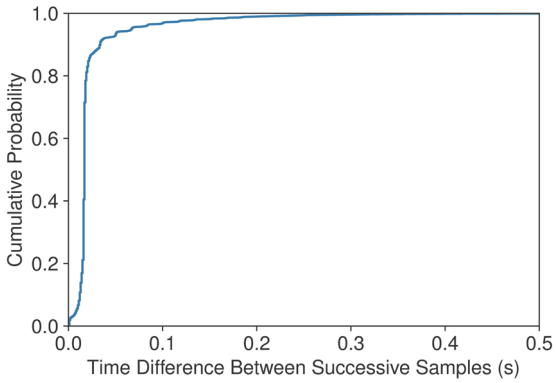

Fig. 3. (Color online) Empirical cumulative probability of time differences between successive samples.

Fig. 4. (Color online) Experimental setup.

subjects, deriving feature vectors. The derived feature vectors are split into two sets, i.e., training and test sets. We trained a user identification machine learning model with the training set and evaluated the user identification accuracy with the test set.

4.3 Feature selection

To determine effective features, we compared the user identification accuracies of the following three feature sets. The feature sets are defined over the features in Table 1.

- Position features:
  Thirteen features with the ✓ mark on the *Position* column in Table 1, which is related to the press position.

- Pressure features:
  Seventeen features with the ✓ mark on the *Pressure* column in Table 1, which is related to press pressure.
- All features:
  All the 28 features in Table 1.

We had eight subjects in their 20s in this evaluation. For each combination of four subjects, we calculated the mean accuracy of user identification in a 10-fold cross-validation. We then calculated the average of the mean accuracies for all the combinations of four subjects. We used an SVM classifier with a linear kernel in this evaluation.

Table 2 shows the user identification accuracy obtained with each feature set. From Table 2, we confirmed the following:

- The pressure and all feature sets showed an accuracy of more than 90%. We cannot judge whether this accuracy is sufficient because user identification relying only on a single operation of a touch-screen device has not been studied. As a reference, the accuracy of the single-button user identifier using a pressure sensor as well as a distance sensor on a physical button in the existing work with four subjects was approximately 90%.[6] We can roughly consider that our proposed user identification method achieved an accuracy comparable to that of the existing work.
- The pressure and all feature sets showed almost the same accuracy, while the position feature set showed an accuracy lower than the pressure and all feature sets by approximately 8%. Pressure-related features were more important for identifying users than position-related features.

We also evaluated the Gini feature importance for each feature set. We performed user identification using a Random Forest classifier and derived the Gini importance of each feature. The Gini importance of the feature set is calculated as the sum of the feature importance corresponding to features in the feature set. The total importance of all features is 1. Some features are included in the position and pressure feature sets. The sum of the feature importance of the position and pressure features exceeds 1.

Table 3 shows the Gini feature importance of each feature set. The all-feature set is excluded in Table 1 because the Gini importance equals 1 according to the definition. Table 3 indicates the following:

Table 2
User identification accuracy obtained with each feature set.

Feature set	Accuracy (%)
Position	84.1
Pressure	92.5
All	92.6

Table 3
Feature importance of each feature set.

Feature set	Importance
Position	0.403
Pressure	0.863

- The Gini importance of pressure features was more than double the importance of the position features. The pressure features contributed more to identifying users than did the position features.
- From Table 3, we can calculate the Gini importance of common features, i.e., features with marks both on position and pressure columns in Table 1, to be 0.403 + 0.863 − 1 = 0.266. The common-feature importance is 66.0% of the position-feature importance, while the pressure-feature importance 30.1%. Note that the number of common features is 2, while that of position features is 13. In the pressure features, the contribution of the common features was notable.

From the above results, we can conclude that the pressure features are effective for identifying users. We use pressure features in the following evaluations.

4.4 Machine learning algorithm

We used an SVM with a linear kernel as a supervised learning algorithm in Sect. 4.3. We compared the user identification performance characteristics of various supervised learning algorithms that support multi-class classification.

We changed the machine learning algorithm used in the user identification block and evaluated the mean accuracy in the same manner as in Sect. 4.3. We compared an SVM with a linear kernel, Random Forest, logistic regression, *k*-nearest neighbor with $k = 1$, Gaussian naive Bayes, stochastic gradient descent, decision tree, and LightGBM. For evaluations of an SVM, logistic regression, *k*-nearest neighbor, and stochastic gradient descent, the features were normalized before both training and user identification.

Table 4 shows the user identification accuracy obtained with each machine learning algorithm. A Random Forest shows the highest user identification accuracy. The number of features is relatively higher than the size of the dataset that we used, which might result in a high accuracy with a Random Forest. We used a Random Forest classifier in the following evaluations.

4.5 Short and long press

On many home appliances, there are two types of button operation, namely, short- and long-press operations. We compared the user identification performance obtained with the data

Table 4
User identification accuracies obtained with various machine learning algorithms.

ML algorithm	Accuracy (%)
Random forest	94.1
SVM (w/ linear kernel)	88.2
Logistic regression	82.4
k-Nearest neighbor (*k*=1)	20.2
Gaussian naive bayes	86.0
Stochastic gradient descent	34.8
Decision tree	89.7
LightGBM	93.7

derived during these two types of button operation.

In this evaluation, we had eight subjects in their 20s. The mean user identification accuracy was calculated for the data of each button operation in the same manner as in Sect. 4.3.

Table 5 shows the mean user identification accuracies obtained with short- and long-press operations. The mean user identification accuracy obtained with the short-press operation was slightly higher than that with the long-press operation.

4.6 Feedback

On some home appliances, touch-screen buttons give feedback. We evaluated the effect of audio and haptic feedback on user identification accuracy. We ignored visual feedback because a button on touch screens is often hidden by a user's finger when the button is pressed.

Audio or haptic feedback was made once when a button was pressed, i.e., the pressure exceeded the button-press threshold p_{th}. In this paper, we set $p_{th} = 0.5$, which is the middle of the pressure range. For audio feedback, we used the On-jin (Free sound effects On-Jin (https://on-jin.com/) wall-switch sound. For haptic feedback, we implemented an iOS app with the haptic feedback and with the same functions as the Web application presented in Sect. 4.1 because the haptic feedback function is unavailable in the Web application.

In this evaluation, we had five subjects in their 20s for each feedback experiment. For each subject and each type of feedback, pressure data were collected 100 times. We calculated user identification accuracy for each combination of four subjects among all the subjects. We calculated the user identification accuracy in a 10-fold cross-validation for each combination of the four subjects and derived the mean accuracy of all the combinations. Note that the five subjects for audio and haptic feedback were different.

Table 6 shows the user identification accuracy with button-press feedback. For both feedback types, the user identification accuracies with and without feedback were almost the same. We performed Welch's two-sample t-test on each of the sound and haptic feedback results. The values of sound and haptic feedback results were 0.558 and 0.581, respectively. At a significance level of 0.05, we can confirm that both feedback types have almost no effect on user identification accuracy.

Table 5
Mean user identification accuracies obtained with short- and long-press operations.

Press operation	Accuracy (%)
Short	94.1
Long	93.5

Table 6
User identification accuracy with button-press feedback.

Feedback	Accuracy (%)	
	w/ Feedback	w/o Feedback
Audio	89.6	87.2
Haptic	88.6	87.4

4.7 Amount of training data

Reducing the amount of training data with ground-truth labels is crucial for realizing a practical service. We evaluated the effect of the amount of training data on user identification accuracy.

In this evaluation, we used the same data as in the feature selection evaluation presented in Sect. 4.3. For each subject, the pressure data taken from the last 10 trials were used for accuracy evaluation. The user identification model was trained with the data taken from the trials immediately before the evaluation trials. We changed the number of trials of training data and evaluated the user identification accuracy. For example, when we evaluated the training data of 20 trials, the data from the 71st to 90th trials were extracted to train the model, and the user identification accuracy was calculated with the 91st to 100th trials. We calculated the user identification accuracies for all the combinations of four subjects and averaged them.

Figure 5 shows the user identification accuracy as a function of the amount of training data. From Fig. 5, we can confirm that the user identification accuracy exceeded 90% for all the amounts of training data. Even when we used 10 trials in the training, the accuracy was 93.5%. This indicates that we can achieve an accuracy of 90% with the labeled pressure data of button press obtained 10 times.

4.8 Short-term experience

The button-press behavior might change while the users gain experience. We evaluated the impact of subjects' short-term button-press experience on user identification accuracy.

We used the same data as in the feature selection evaluation presented in Sect. 4.3. For each subject, the pressure data taken from the last 10 trials were used as test data for accuracy evaluation. The user identification model was trained with the data taken from 10 successive trials except the test data. We changed the start of the training data trials from 1st, 11th, ..., to 81st and calculated the user identification accuracy. We calculated the user identification accuracies for all the combinations of four subjects and averaged them.

Figure 6 shows the user identification accuracy as a function of the number of training-data trials and indicates the following:

- The user identification accuracy was highly dependent on the training-data trial. The highest accuracy was 92.8% when the 81st–90th training data, i.e., the training data immediately before the evaluation trials, were used. Short-term experience had a significant impact on the user identification accuracy.
- The lowest accuracy was 75.8% when the 21st–30th training data were used. This was mainly caused by the insufficient practice of subjects. In our experiments, most subjects performed button-press practice only a couple of times up to 10 times. About 30 trials might have been required for the subjects to make the button press consistent.

The above results indicated that the time interval between the training and test data affected user identification accuracy. Newer training data seemed to show high accuracy. The user identification model should be retrained with the latest data.

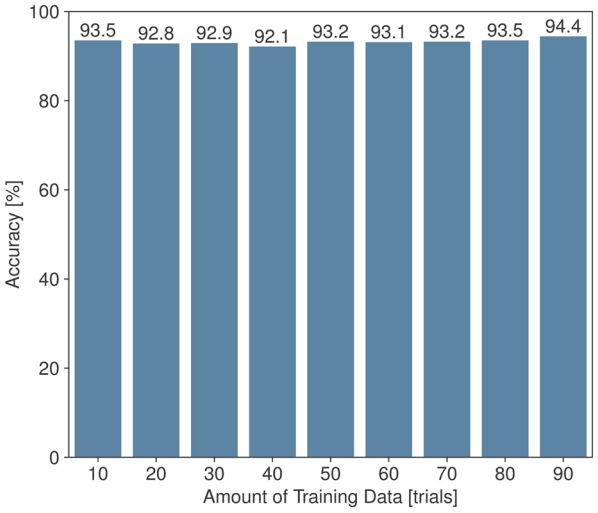

Fig. 5. User identification accuracy as a function of amount of training data.

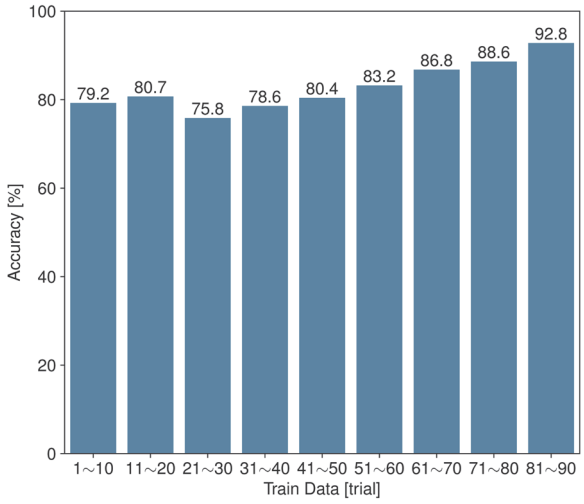

Fig. 6. User identification accuracy as a function of the number of training-data trials.

5. Conclusion

In this paper, we proposed a user identifier using home appliance operations. We focused on touch-screen button operations. Our user identifier extracts features from a time series of pressure data derived from a touch-screen device and performs supervised multi-class classification to identify users. We implemented our user identifier and evaluated the user identification accuracy using button pressure collected on an iPhone7 smartphone. The evaluation results revealed that the user identification accuracy was 94.4% when a sufficient amount of data was used for training the classifier model. Even with the limited amount of

training data obtained in 10 trials, the user identification accuracy was still 92.8% when we used the latest training data.

We are planning to evaluate the long-term performance of user identification. In a real-world environment, additional features such as the time of home appliance use might be effective in identifying users. A long-term real-world experiment leads us to study more features effectively identifying users in the real world. Extension to more types of home appliance controller, including a physical button, is another future work. For a physical button, we might also apply the findings in this paper. For example, press-pressure features might also have a greater impact than press-position features on a physical button. In that sense, we need to study the effects of pressure-sensor resolution and the type of button on user identification performance for a physical button.

Acknowledgments

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number JP21K11847 as well as the Cooperative Research Project of RIEC, Tohoku University.

References

1 S. Mare, R. Rawassizadeh, R. Peterson, and D. Kotz: Proc. Workshop on Usable Security (Internet Society, 2019) 1–12.
2 J. Zhao and J. Tanaka: Hand gesture authentication using depth camera, Advances in Information and Communication Networks, ser. Advances in Intelligent Systems and Computing, K. Arai, S. Kapoor, and R. Bhatia, Eds. (Springer, 2019) 641–654.
3 R. Ramakers, D. Vanacken, K. Luyten, K. Coninx, and J. Schöning: Proc. 25th Annual ACM Symp. User Interface Software and Technology (UIST) (ACM, 2012) 35–44. https://doi.org/10.1145/2380116.2380123
4 U. von Zadow, P. Reipschläger, D. Bösel, A. Sellent, and R. Dachselt: Proc. Int. Working Conf. Advanced Visual Interfaces (AVI) (ACM, 2016) 144–151. https://doi.org/10.1145/2909132.2909258
5 A. C. Evans, K. Davis, J. Fogarty, and J. O. Wobbrock: Proc. 2017 CHI Conf. Human Factors in Computing Systems (ACM, 2017) 35–47. https://doi.org/10.1145/3025453.3025793
6 H. Pohl, M. Krause, and M. Rohs: Proc. 2015 ACM Int. Joint Conf. Pervasive and Ubiquitous Computing, ser. UbiComp '15 (ACM, 2015) 403–407. https://doi.org/10.1145/2750858.2804270
7 N. F. Ince, C.-H. Min, A. Tewfik, and D. Vanderpool: EURASIP J. Adv. Signal Process. **2008** (2007) 273130. https://doi.org/10.1155/2008/273130
8 S. C. Mukhopadhyay: IEEE Sens. J. **15** (2015) 1321. https://doi.org/10.1109/JSEN.2014.2370945
9 F. Shahmohammadi, A. Hosseini, C. E. King, and M. Sarrafzadeh: Proc. 2017 IEEE/ACM Int. Conf. Connected Health: Applications, Systems and Engineering Technologies (CHASE) (IEEE, 2017) 321–329. https://doi.org/10.1109/CHASE.2017.115
10 A. Nandy, J. Saha, C. Chowdhury, and K. P. D. Singh: Proc. 2019 Int. Conf. Opto-Electronics and Applied Optics (Optronix) (IEEE, 2019) 1–6. https://doi.org/10.1109/OPTRONIX.2019.8862427
11 S. Paraschiakos, R. E. Cachucho, M. Moed, D. van Heemst, and S. P. Mooijaart: User Model. User-Adap. **30** (2020) 567–605. https://doi.org/10.1007/s11257-020-09268-2
12 C. Pham, S. Nguyen-Thai, H. Tran-Quang, S. Tran, H. Vu, T.-H. Tran, and T.-L. Le: IEEE Access **8** (2020) 86934. https://doi.org/10.1109/ACCESS.2020.2991731
13 E. Nakagawa, K. Moriya, H. Suwa, M. Fujimoto, Y. Arakawa, and K. Yasumoto: Proc. 2017 IEEE Int. Conf. Pervasive Computing and Communications Workshops (PerCom Workshops) (IEEE, 2017) 539–544. https://doi.org/10.1109/PERCOMW.2017.7917620
14 Y. Kashimoto, K. Hata, H. Suwa, M. Fujimoto, Y. Arakawa, T. Shigezumi, K. Komiya, K. Konishi, and K. Yasumoto: Adj. Proc. Int. Conf. Mobile and Ubiquitous Systems: Computing Networking and Services, ser. MOBIQUITOUS 2016 (Springer, 2016) 6–11. https://doi.org/10.1145/3004010.3006378

15 T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse: Pers. Ubiquit. Comput. **14** (2010) 489. https://doi.org/10.1007/s00779-009-0277-9
16 K. Thapa, Z. M. Abdullah Al, B. Lamichhane, and S.-H. Yang: Sensors **20** (2020) 5770. https://doi.org/10.3390/s20205770
17 U. Verma, P. Tyagi, and M. Kaur: Int. J. Sens. Netw. **41** (2023) 1. https://doi.org/10.1504/IJSNET.2023.128503
18 S. Ahmed, S. Irfan, N. Kiran, N. Masood, N. Anjum, and N. Ramzan: Sensors **23** (2023) 7095. https://doi.org/10.3390/s23167095
19 J. Sung, C. Ponce, B. Selman, and A. Saxena: Proc. 16th AAAI Conf. Plan, Activity, and Intent Recognition, ser. AAAIWS'11-16 (AAAI, 2011) 47–55.
20 E. Stone and M. Skubic: J. Ambient Intell. Smart Environ. **3** (2011) 349. https://doi.org/10.3233/AIS-2011-0124
21 H. Pirsiavash and D. Ramanan: Proc. 2012 IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (IEEE, 2012) 2847–2854. https://doi.org/10.1109/CVPR.2012.6248010
22 L. Wang, D. Q. Huynh, and P. Koniusz: IEEE Trans. Image Process. **29** (2019) 15. https://doi.org/10.1109/TIP.2019.2925285
23 J. T. S. Phang and K. H. Lim: Proc. 3rd Int. Conf. Machine Learning and Soft Computing, ser. ICMLSC '19 (ACM, 2019) 175–180. https://doi.org/10.1145/3310986.3311006
24 T. T. Zin, Y. Htet, Y. Akagi, H. Tamura, K. Kondo, S. Araki, and E. Chosa: Sensors **21** (2021) 5895. https://doi.org/10.3390/s21175895
25 A. S. Crandall and D. Cook: Proc. 2008 IET 4th Int. Conf. Intelligent Environments (IET, 2008) 1–8. https://doi.org/10.1049/cp:20081164
26 A. Alhamoud, V. Muradi, D. Böhnstedt, and R. Steinmetz: Proc. 6th Int. Conf. Internet of Things, ser. IoT'16 (ACM, 2016) 15–23. https://doi.org/10.1145/2991561.2991563
27 M. Jethanandani, A. Sharma, T. Perumal, and J.-R. Chang: Internet of Things **12** (2020) 100324. https://doi.org/10.1016/j.iot.2020.100324
28 D. Chen, S. Yongchareon, E. M.-K. Lai, J. Yu, and Q. Z. Sheng: IEEE Internet Things J. **8** (2021) 11193. https://doi.org/10.1109/JIOT.2021.3051574
29 T. Wang and D. J. Cook: IEEE Trans. Pattern Anal. Mach. Intell. **43** (2021) 2809. https://doi.org/10.1109/TPAMI.2020.2973571
30 T. Wang and D. J. Cook: IEEE Trans. Emerg. Topics Comput. **10** (2022) 1130. https://doi.org/10.1109/TETC.2021.3072980
31 L. Arrotta, C. Bettini, and G. Civitarese: Distrib. Parallel Databases **41** (2023) 571. https://doi.org/10.1007/s10619-022-07403-z
32 X. Chen, J. Cumin, F. Ramparany, and D. Vaufreydaz: Proc. 2024 IEEE Int. Conf. Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops) (IEEE, 2024) 1–6. https://doi.org/10.1109/PerComWorkshops59983.2024.10502415
33 Y. Zhao, Z. Qiu, Y. Yang, W. Li, and M. Fan: Proc. 2017 ACM Int. Symp. Wearable Computers, ser. ISWC '17 (ACM, 2017) 122–125. https://doi.org/10.1145/3123021.3123049
34 Y. Song and I. Oakley: Int. J. Hum.-Comput. Interact. **39** (2023) 893. https://doi.org/10.1080/10447318.2022.2049144
35 Y. Song and I. Oakley: PushID: Proc. HCI for Cybersecurity, Privacy and Trust (HCI-CPT) (Springer, 2022) 255–267. https://doi.org/10.1007/978-3-031-05563-8_17
36 K. Suda, S. Ishida, and H. Inamura: Proc. 2023 14th Int. Conf. Mobile Computing and Ubiquitous Network (ICMU) (IPSJ, 2023) 1–6. https://doi.org/10.23919/ICMU58504.2023.10412226
37 K. Suda, S. Ishida, and H. Inamura: Proc. 2022 IEEE 11th Global Conf. Consumer Electronics (GCCE) (IEEE, 2022) 714–715. https://doi.org/10.1109/GCCE56475.2022.10014053