

加速度による個人推定における加速度取得間隔と推定精度に関する一考察

高橋良颯¹ 園山敦也¹ 神山剛² 小口正人³ 山口実靖⁴

概要：スマートフォンの加速度センサから得られた加速度データを深層学習(Deep Neural Network)を用いて解析し、スマートフォンの保持者を推定する手法が提案されている。これまでの我々の研究では Android スマートフォンの加速度センサより歩行時の加速度を取得し、その加速度を LSTM を用いて解析し所有者を推定する手法を提案してきた。ただし、Android スマートフォンの加速度センサは加速度取得間隔を変更可能であり、取得間隔の改善による推定精度の向上などの考察はなされていない。

本稿では、加速度取得間隔の変更により推定精度がどのように変化するのかについて考察する。具体的には、SENSOR_DELAY_FASTEST と SENSOR_DELAY_GAME という二つの加速度取得間隔に着目し、これらの取得間隔で得られる加速度データの特徴の比較と、各取得間隔で推定を行ったときの推定精度の比較を行う。そして、より高い頻度で加速度を取得する SENSOR_DELAY_FASTEST で取得した方が、低い頻度で取得する SENSOR_DELAY_GAME で取得するよりも高い精度で推定できることを示す。

キーワード：加速度センサ，機械学習，LSTM，個人推定，加速度取得間隔，Android スマートフォン

1. はじめに

近年センサ情報を機械学習や人工知能を用いて解析する研究が活発におこなわれており、一例としてセンサデバイスから得た情報を用いて人間の動きを解析する研究[1][2]などが行われており、大きな成果を上げている。

過去の我々の研究[3]では、歩きながら Android スマートフォンを操作する状況でスマートフォンの加速度センサから得られた 3 軸加速度情報からそのスマートフォンを保持している個人を推定する手法を提案した。しかし、現在の Android スマートフォンは加速度の取得間隔を変更可能であり、その取得間隔の最適化による個人推定精度の向上などの考察はなされていない。本稿では、スマートフォンに搭載された加速度センサの加速度取得間隔を変更し(短くし)、同研究の推定手法の精度向上を実現することについて考察を行う。

具体的には、2 種類の取得間隔で得られた加速度の統計的特徴を比較し、2 種類の取得間隔で得られる加速度値の違いや、センサが提供する加速度データに関する調査や考察を行う。また、加速度取得間隔の違う 2 台の Android スマートフォンを保持しながら被験者が歩き、スマートフォンの加速度計からの加速度を取得する。そして、2 種類の取得間隔で得られた加速度を用いて推定を行い、加速度取得間隔と推定精度の関係を示す。

Android スマートフォンの加速度取得間隔は複数のものがあるが本稿では、我々の過去の研究[3]で用いた SENSOR_DELAY_GAME と、最も加速度取得間隔の短い SENSOR_DELAY_FASTEST の二つの取得間隔を用いて評価と考察を行う。本稿の考察や推定精度向上により、我々

が過去の研究[3]で応用例として示した、使用ユーザの推定に基づく推薦やサポートによるユーザエクスペリエンスの向上や、歩行によるユーザ認証や入館管理の精度向上などが達成できると考えられる。

2. 関連研究

本章にて、スマートフォンのセンサから得られた加速度データを解析する研究で、本研究と関連の強いものを示す。

最初にスマートフォンやウェアラブル端末に搭載されているセンサから得られたデータを機械学習などにより解析する研究を示す。Kobayashi らはスマートフォンの加速度計から得られる加速度や位置情報からデータを解析することでユーザの動作を推定する手法を提案している[1]。Eren らはスマートフォンに搭載された様々なセンサを用いて運転者のふるまいを推定し、その運転者の安全性を判断する手法を提案している[2]。Becker らはスマートウォッチから得られたセンサ情報を CNN(Convolutional Neural Network)によって解析し、音を発するジェスチャー(ノックや拍手など)を認識するシステム GestEar を提案している[4]。Brock らはスキーのジャンパーに装着したセンサ情報からスキージャンプ時のエラーを分類する手法を提案している[5]。同研究ではセンサとしては加速度計やジャイロスコープなどを使用しており、これらの情報を CNN を用いて解析することでエラー分類を行っている。Brunner らはスマートウォッチに搭載されたセンサからスイマーのスタイル(クロールや平泳ぎ)などを推定する手法を提案している[6]。同研究でも推定には CNN を用いており、推定精度は 97% となっている。これらの研究により、ユーザの動作をセンサ情報から高い精度で推定することが可能であることが示

1 工学院大学大学院
Kogakuin, University graduate school

2 長崎大学
Nagasaki University

3 お茶の水女子大学
Ochanomizu University

4 工学院大学
Kogakuin, University

されている。しかしこれらの研究ではスマートフォンを保持しているユーザの推定は行われていない。

次にセンサデータの解析により個人を推定する研究を示す。Shiらはスマートフォンのセンサから得た情報を解析しユーザ識別を行うユーザ認証フレームワーク SenGuard を提案している[7]。そして、音声、位置情報、マルチタッチ情報、移動情報の4つの情報を用いた認証が最も認証システムに適していることを示している。Shiらの研究では個人推定を行っているが、単純な統計値を用いた推定に留まっており、機械学習のような解析手法は使用されていない。また評価も2クラス分類にとどまっており、さらに多いクラスの分類に関する考察や多クラス分類における高い推定精度の実現は達成されていない。Muhammadらはスマートフォンから得た加速度データを解析し、所有者が正当なユーザであるか、またはそれ以外の者であるかを特定する手法を提案している[8]。Muhammadらの研究ではスマートフォンをポケットに入れた状態で加速度を集計し、そのデータを SVM(Support Vector Machine) や ANN(Artificial Neural Network)などを用いて解析している。著者らは ANN を用いる手法が最も正解率が高いことを示し、正解率は約95%となっている。佐藤らは歩きながらスマートフォンを操作する動作をする際のスマートフォンの加速度計センサから得た加速度データを機械学習を用いて解析し、ユーザを推定する手法を提案している[9][10]。これらの研究[9][10]ではスマートフォンの加速度計から得た加速度が、既知の2人の対象ユーザのうちどちらのものであるのかという推定を行っている。機械学習には DNN(Deep Neural Network)を用いている。また、加速度データを、X軸、Y軸、Z軸ごとに使用して推定する手法と絶対値を使用する手法の精度の比較を行っており、X軸のみを用いる手法の推定精度が最も高くなることを示している。その推定精度は87%となっている。Muhammadらの研究[8]や佐藤らの研究[9][10]では、加速度から保持しているユーザを推定することに取り組んでいるが、加速度を SVM や ANN や DNN を用いて解析しており、加速度を時系列データと学習していない。

過去の我々の研究[3]では、歩きながらスマートフォンを操作する際のスマートフォンの加速度計から得た加速度を LSTM(Long Short Term Memory)を用いて解析する手法が提案されている。性能評価では3軸すべての加速度を入力データとした2~5クラス分類を行い、5クラス分類にて96%の推定精度を実現している。ただし、加速度取得間隔と推定精度の関係の考察などはされておらず、加速度取得間隔の最適化による推定精度のさらなる向上などの余地が残されている。

3. 加速度取得間隔と取得加速度の関係

本章では、スマートフォンの加速度取得間隔と得られ

る加速度の関係について述べる。

本稿では2台の Android スマートフォン Nexus5x を用いて実験を行っている。それぞれ、端末1および端末2と呼ぶ。

3.1 スマートフォンの加速度取得間隔

本節では Android スマートフォンの加速度取得間隔の種類について説明する。Android スマートフォンでは加速度取得間隔として4つのモードを設定することができる[10]。表1に Android スマートフォンの4つの加速度取得間隔を示す。

表1 Android スマートフォンの加速度取得間隔

SENSOR_DELAY_FASTEST	最高頻度の取得間隔(デバイスの性能や仕様に依存)
SENSOR_DELAY_NORMAL	画面の向きの変更に適した取得間隔(デフォルト)
SENSOR_DELAY_GAME	ゲームに適した取得間隔
SENSOR_DELAY_UI	ユーザインタフェースに適した取得間隔

本稿では、過去の研究[3]で用いた SENSOR_DELAY_GAME と、最も加速度取得間隔の短い SENSOR_DELAY_FASTEST に関して調査を行い、こちらを用いて評価と考察を行う。

本稿で用いた両端末(Nexus5x)の SENSOR_DELAY_GAME では約20msごとに加速度を取得し、SENSOR_DELAY_FASTEST では約2.5msごとに加速度を取得する。よって同じ長さの時間に渡り加速度を取得したとき SENSOR_DELAY_FASTEST では SENSOR_DELAY_GAME に比べて約8倍の加速度データを取得することになる。

3.2 加速度取得間隔とノイズの大きさの関係

本節では加速度取得間隔の変更によって得られる加速度のノイズがどのように変化するかについて述べる。

端末1および端末2の、各取得間隔における取得加速度に含まれるノイズの大きさの調査として、各端末を机の上に静止させた状態の加速度を取得し、その加速度の各軸の値の標準偏差を求めた。加速度は SENSOR_DELAY_GAME では16分間机の上で静止させて取得し、SENSOR_DELAY_FASTEST では2分間机の上で静止させて取得した。端末が静止しているため、測定ノイズが存在しなければ各端末では重力加速度の値が固定的に取得されるはずであり、取得加速度の変化は測定ノイズであると考えられる。端末1の取得加速度の標準偏差を表2に、端末2の標準偏差を表3に示す。

表 2 端末 1 のノイズ調査結果

	標準偏差		
	X 軸	Y 軸	Z 軸
SENSOR_DELAY_FASTEST	0.0440	0.0429	0.0473
T	3	0	8
SENSOR_DELAY_GAME	0.0079	0.0090	0.0086
	7	5	9

表 3 端末 2 のノイズ調査結果

	標準偏差		
	X 軸	Y 軸	Z 軸
SENSOR_DELAY_FASTEST	0.0438	0.0429	0.0551
T	1	9	6
SENSOR_DELAY_GAME	0.0103	0.0126	0.0105
	5	9	3

表 2, 表 3 の結果から, SENSOR_DELAY_GAME で取得した加速度は, SENSOR_DELAY_FASTEST で取得した加速度に比べてノイズが少ないことがわかる。

このことから, 取得間隔 SENSOR_DELAY_GAME にてアプリケーションに提供される加速度データは, 取得間隔 SENSOR_DELAY_FASTEST にて得られた加速データの一部を選択して抽出したもの(例えば 8 個に 1 個を選択したもの)ではなく, 平均などの統計処理が施されたものである, あるいは統計処理を施したものと同等のものになっていると予想できる。

3.3 加速度取得間隔と取得加速度の大きさの関係

この節では, SENSOR_DELAY_GAME で取得できる加速度と, SENSOR_DELAY_FASTEST で取得できる加速度の大きさの比較を行う。

最初に, 取得間隔 SENSOR_DELAY_GAME と SENSOR_DELAY_FASTEST で取得した加速度の絶対値の分布を(ヒストグラム)を図 1, 図 2 に示す。図 1 が端末 1 におけるヒストグラム, 図 2 が端末 2 におけるヒストグラムである。

次に, 取得間隔 SENSOR_DELAY_GAME にて得られた加速度の絶対値の分布と, SENSOR_DELAY_FASTEST にて得られた加速度を統計処理した値の分布を示す。具体的には, 以下の 1 から 5 の様に加速度を取得して統計処理を行った。

- 1.SENSOR_DELAY_GAME と SENSOR_DELAY_FASTEST の両取得間隔で 2 分間机の上に静止した状態の加速度を集計する

- 2.SENSOR_DELAY_FASTEST で取得した各軸の加速度の移動平均(ウィンドウサイズ 3)を取得する
- 3.移動平均を施した各軸の加速度の絶対値を取得する
- 4.上記の絶対値を 8 個ごとにグループ化し, 各グループにおける平均値または中央値または最小値を取得する
- 5.上記の値(平均値, 中央値, 最小値)の移動平均(ウィンドウサイズ 5)を取得する

処理 4 にて「平均値」を用いて処理した加速度の分布を図 3~4 に、「中央値」を用いた分布を図 5~6 に、「最小値」を用いた分布を図 7~8 に示す。

図 1 と図 2 より, 取得間隔 SENSOR_DELAY_GAME にて得られる加速度は, 取得間隔 SENSOR_DELAY_FASTEST にて得られる加速度より標準偏差が小さく安定しており, 狭い領域の加速度に取得値が集中していることが分かる。

また, 取得される値のピーク(最頻値)にも差異が確認され, 取得間隔 SENSOR_DELAY_FASTEST にて得られる加速度の方が絶対値が大きいことが分かる。すなわち, 分布が異なるだけでなく, 得られる絶対値の平均にも差があることが分かる。

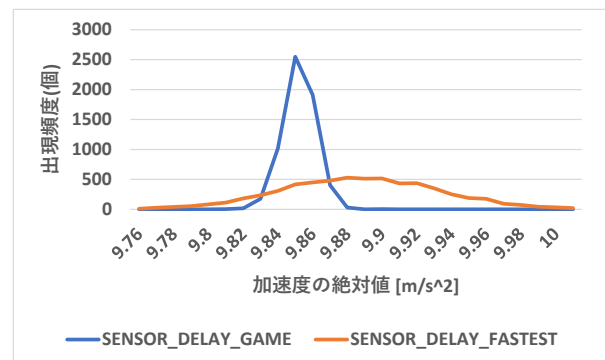


図 1 端末 1 における SENSOR_DELAY_GAME の絶対値と SENSOR_DELAY_FASTEST の絶対値の関係

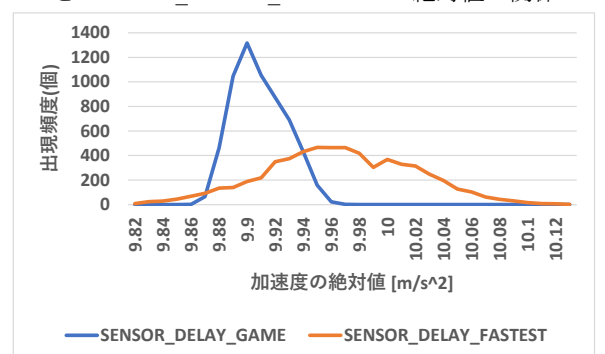


図 2 端末 2 における SENSOR_DELAY_GAME の絶対値と SENSOR_DELAY_FASTEST の絶対値の関係

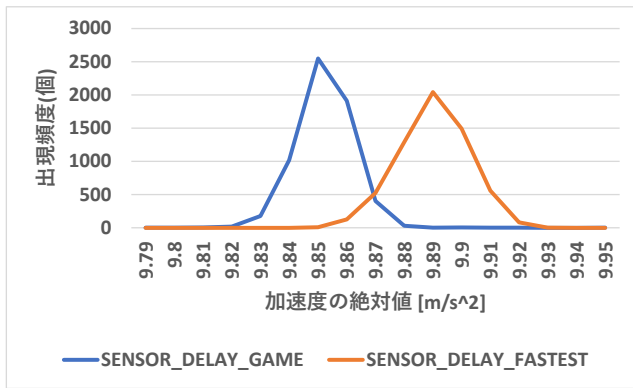


図 3 端末 1 における 3.3 節の処理のうち 4 の処理を平均値に変えた比較

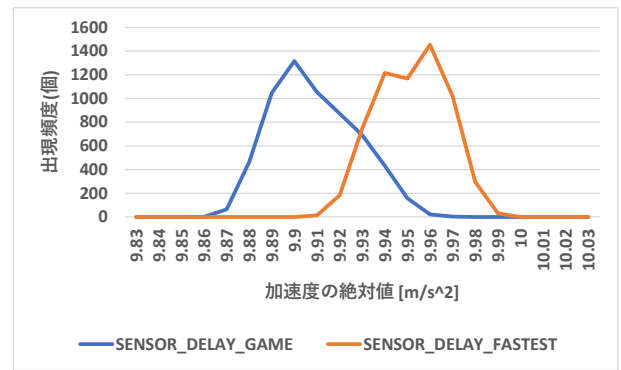


図 6 端末 2 における 3.3 節の処理のうち 4 の処理を中央値に変えた比較

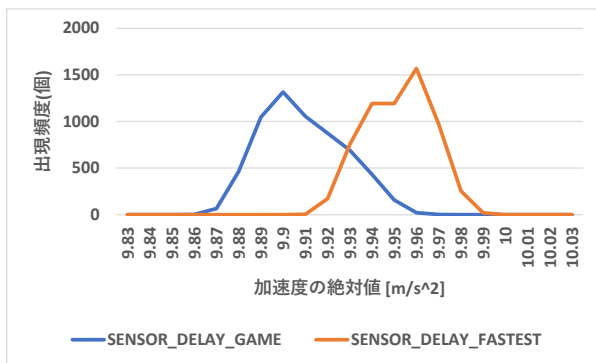


図 4 端末 2 における 3.3 節の処理のうち 4 の処理を平均値に変えた比較

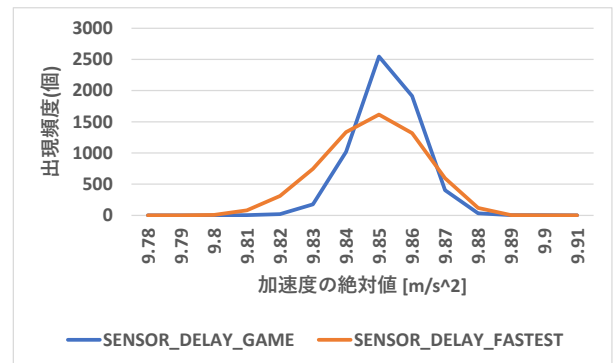


図 7 端末 1 で 3.3 節の処理を行った比較

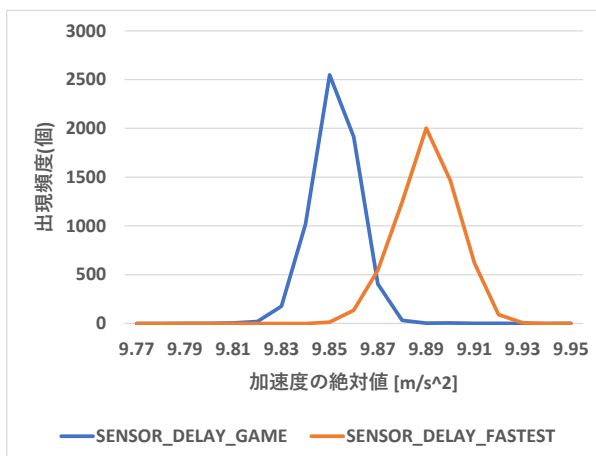


図 5 端末 1 における 3.3 節の処理のうち 4 の処理を中央値に変えた比較

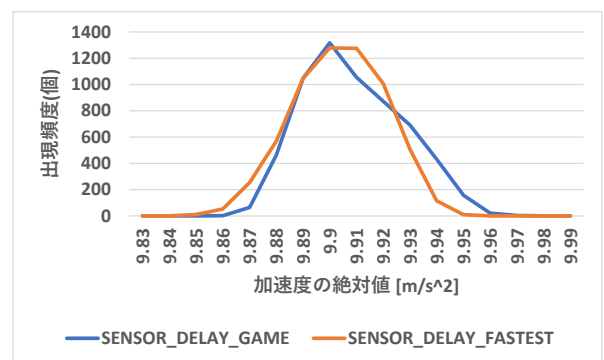


図 8 端末 2 で 3.3 節の処理を行った比較

次に、図 3～8 の比較を行う。図 1 と図 2 の結果にて示した様に両取得間隔では得られる加速度の大きさに差異があり、図 3～6 の結果が示すように取得間隔 SENSOR_DELAY_FASTEST にて得られた加速度の平均値や中央値を取得しても取得間隔 SENSOR_DELAY_GAME にて得られる加速度の分布に近くならないことが分かる。また、図 1 と図 2 で示した様に SENSOR_DELAY_FASTEST で得られる加速度の絶対値より SENSOR_DELAY_GAME にて得られる絶対値の方が小さく、図 7 と図 8 が示すように SENSOR_DELAY_FASTEST で得られた加速度に最小値の取得を含めた統計処理を施すことにより SENSOR_DELAY_GAME で得られる加速度に近い分布になることが分かる。

4. LSTMによる個人推定手法

本章にて、スマートフォン加速度センサから得られた加速度を LSTM による解析し個人推定を行う手法[3]を説明する。本手法では、推定対象ユーザが既知のユーザ群の中のいずれのユーザであるかの推定を行う。

4.1 推定手法

推定手法は三段階に分かれている。第一段階は推定モデルの事前学習用に各推定対象者の加速度を取得する。加速度取得時のユーザの動作状況としては、歩きながらスマートフォンを操作している状況を想定する。第二段階では、事前学習用に収集した加速度データを LSTM を用いて学習し、推定用モデルを構築する。第三段階では推定用加速度データを LSTM に入力し、上記のモデルを用いて解析し、入力データのユーザの推定を行う。第二、第三段階で入力するデータは過去の研究[3]に従い加速度の3軸全ての値とする。研究[3]では、推定結果として「不明」を出力する場合と、「不明」を出力しない場合が提案されているが、本稿では後者の「不明」を出力しない場合のみを想定する。よって、推定手法は必ず推定結果を出力し、出力結果は正解か不正解かで評価される。

4.2 推定対象者

本稿の推定モデルは、推定システムがあるユーザの歩行時の加速度データを与えられ、それが既知の n 人の中のどのユーザであるかを推定する。既知の n 人の学習用の加速度データはモデルに与えられており、推定前にモデルを構築することが可能であることを前提とする。

5. 性能評価

本章にて性能評価を行う。取得間隔 `SENSOR_DELAY_GAME` および `SENSOR_DELAY_FASTEST` にて実ユーザの使用時の加速度を取得し、4章の推定手法にてユーザの推定を行い、その推定精度を評価した。

5.1 実験環境

本節では実験環境について述べる。

最初に入力データとして使用する加速度を取得した際の環境について説明する。各推定対象者に「普段通りに歩きながらスマートフォンを操作してください」と伝えて、安全な場所を歩いてもらった。その際に2台のスマートフォン(前述の端末1と端末2)を背面合わせで固定したものを被験者には1台のスマートフォンとして扱ってもらった。これにより、両端末にて同一の加速度がかかる状況を構築した。この2台のスマートフォンでは、それぞれで取得間隔 `SENSOR_DELAY_GAME` と `SENSOR_DELAY_FASTEST` で加速度を同時に取得した。被験者が歩行する際にはスタッフが同行し、被験者の安全を確保した状態で行った。全被験者が歩いた場所はすべて同じ道である。計測を行った

2台のスマートフォンは Nexus5x であり、OS として Android 7.1.1 を搭載している。本端末の各軸の正の向きは図9の通りである。

本稿では加速度測定のための Android アプリケーションを作成し、それをインストールしたスマートフォンを推定対象の被験者に使用させて加速度を計測した。この Android アプリケーションは Google API を用いて加速度を取得する。アプリケーションで取得される加速度は X 軸、Y 軸、Z 軸の三軸方向である。計測される加速度方向は図9の通りである。

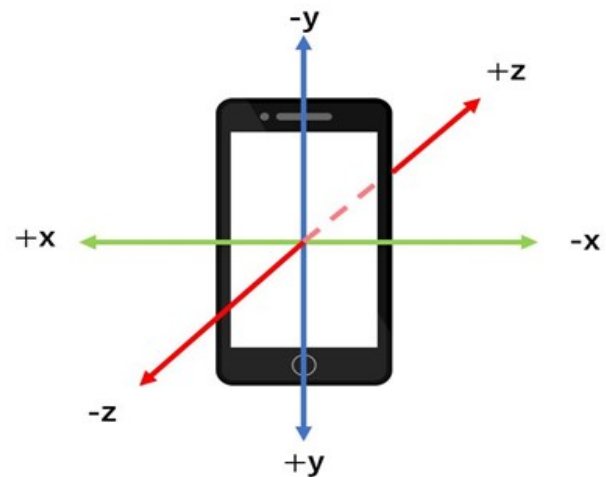


図9 スマートフォンの加速度方向

性能評価では8人の加速度データを使用している。8人の内訳は男性6名、女性2名であり、全員22~24歳である。また各ユーザ名をA~Hとする。

本稿の性能評価では取得した加速度の最初の2秒間分と最後の2秒間分のデータを削除している。これは加速度集計時に使用した Android アプリケーション操作時の加速度を削除するためである。この削除処理は `SENSOR_DELAY_GAME` と `SENSOR_DELAY_FASTEST` の両取得間隔の取得加速度データに対して適応している。

本稿で用いた LSTM のハイパーパラメータは表4の通りである。

表 4 モデルのハイパーパラメータ
ネットワークのハイパーパラメータ

活性化関数	ソフトマックス関数
誤差関数	クロスエントロピー
最適化関数	Adagrad
中間層次元数	100
中間層数	1

5.2 取得間隔が異なる加速度を使用した性能評価

本節にて、取得間隔が異なる加速度を使用した推定の精度の比較を行う。4章の推定手法の第一段階における加速度を取得間隔 SENSOR_DELAY_GAME および SENSOR_DELAY_FASTEST で取得した。そして第二、第三段階では、第一段階で取得した加速度を使用して、ユーザの推定を行った。

推定手法を用いて8クラス分類の推定を行い、その推定精度を評価した。学習と評価には交差検証を用いた。取得間隔 SENSOR_DELAY_GAME と取得間隔 SENSOR_DELAY_FASTEST の取得データを、LSTM への入力データ長 40 秒にて評価した結果を図 10 と図 11 に示す。図 10 が全ユーザの平均推定精度であり、図 11 がユーザごとの推定精度である。入力データ長 20 秒にて評価した結果を図 12 と図 13 に示す。同様に、図 12 が全ユーザの平均推定精度を、図 13 がユーザごとの推定精度を示している。

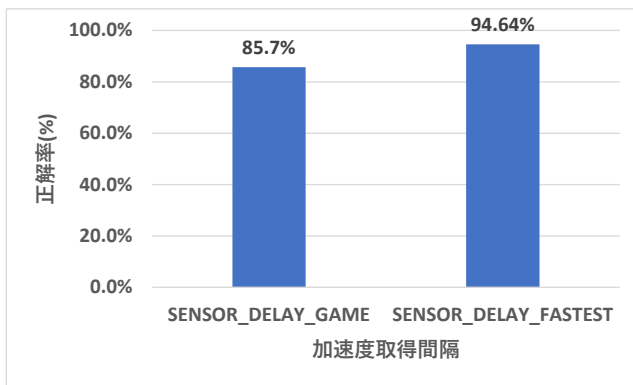


図 10 平均推定精度(8 クラス分類, 入力データ 40 秒)

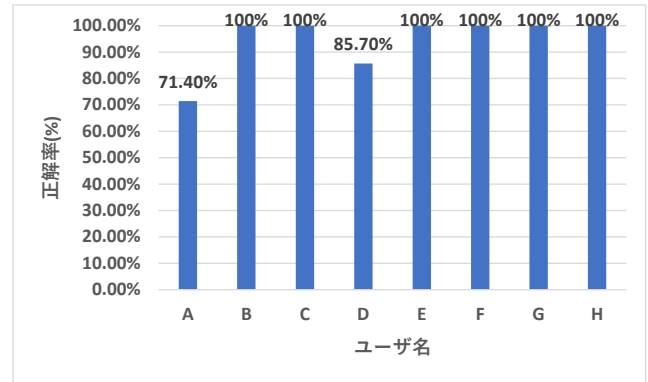


図 11 ユーザごとの推定精度(8 クラス分類, 入力データ 40 秒)

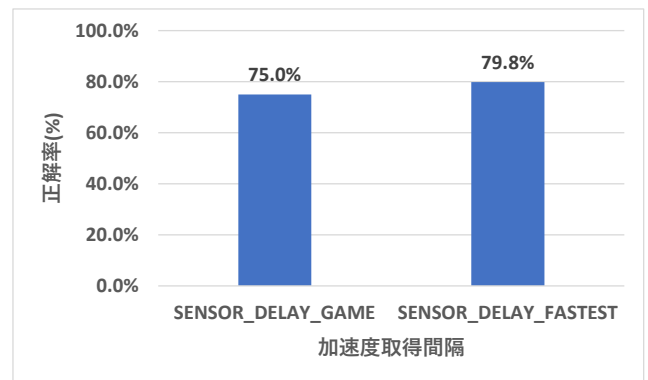


図 12 平均推定精度(8 クラス分類, 入力データ 20 秒)

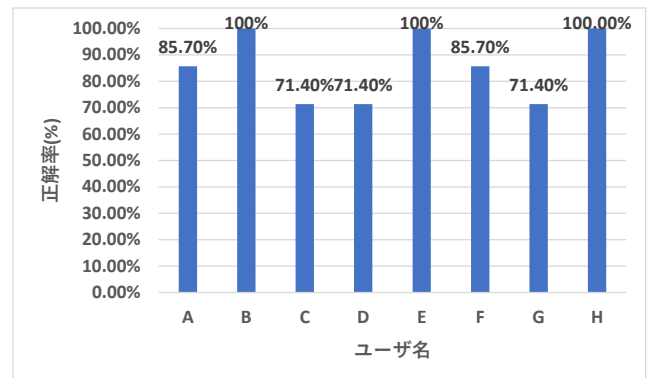


図 13 ユーザごとの推定精度(8 クラス分類, 入力データ 20 秒)

図 10, 図 11 の結果から、取得間隔 SENSOR_DELAY_FASTEST を用いて取得した加速度を用いて推定した方が、SENSOR_DELAY_GAME で取得した加速度を用いて推定するよりも精度が高いことがわかる。このことから、同じ時間で高い頻度で加速度を取得し、より多くの加速度値を用いて推定した方が、より高い精度で推定ができることが分かる。

同様に図 12, 図 13 より、入力データの時間が短い場合であっても高い頻度で加速度を取得して学習と分類を行った方が高い精度で推定ができることが分かる。

6. まとめ

本稿では加速度よりスマートフォンを保持しているユーザを推定する手法に着目し、加速度取得間隔の変更が推定精度に与える影響を調査した。そして、加速度取得間隔を短くし、加速度データを増やすことで推定精度が向上することを確認した。

今後の課題としては、異なる使用状況における評価などを考えている。

謝辞

本研究は JSPS 科研費 18K11277, 21K11854 の助成を受けたものである。

本研究は、JST, CREST JPMJCR1503 の支援を受けたものである。

参考文献

- [1] Arei Kobayashi, Shigeki Muramatsu, Daisuke Kamisaka, Takafumi Watanabe, Atsunori Minamikawa, Takeshi Iwamoto, and Hiroyuki Yokoyama. Shaka: User movement estimation considering reliability, power saving, and latency using mobile phone. IEICE TRANSACTIONS on Information and Systems, Vol. E94-D, No. 6, pp. 1153--1163, 201
- [2] H. Eren, S. Makinist, E. Akin and A. Yilmaz. 2012. Estimating driving behavior by a smartphone. 2012 IEEE Intelligent Vehicles Symposium, Alcalá de Henares. pp. 234-239. doi: 10.1109/IVS.2012.6232298
- [3] 高橋 良颯, 中村 鴻介, 神山 剛, 小口 正人, 山口 実靖, "LSTM による加速度に基づく個人推定", IPSJ 研究報告コンシューマ・デバイス&システム (CDS), 2020-CDS-28 巻, 2 号, pp. 1 - 7, 2020
- [4] Vincent Becker, Linus Fessler, and Gábor Sörös: GestEar: combining audio and motion sensing for gesture recognition on smartwatches. *Proceedings of the 23rd International Symposium on Wearable Computers (ISWC '19)*. Association for Computing Machinery, pp. 10-19 (2019) DOI:<https://doi.org/10.1145/3341163.3347735>
- [5] Heike Brock, Yuji Ohgi, and James Lee. 2017: Learning to judge like a human: convolutional networks for classification of ski jumping errors. *Proceedings of the 2017 ACM International Symposium on Wearable Computers (ISWC '17)*. Association for Computing Machinery, pp. 106-113 (2017). DOI:<https://doi.org/10.1145/3123021.3123038>
- [6] Gino Brunner, Darya Melnyk, Birkir Sigfússon, and Roger Wattenhofer.: Swimming style recognition and lap counting using a smartwatch and deep learning. *Proceedings of the 23rd International Symposium on Wearable Computers (ISWC '19)*. Association for Computing Machinery, pp. 23-31 (2019). DOI:<https://doi.org/10.1145/3341163.3347719>
- [7] W. Shi, J. Yang, Yifei Jiang, Feng Yang and Yingen Xiong. 2011. SenGuard: Passive user identification on smartphones using multiple sensors, 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Wuhan. pp. 141-148. doi: 10.1109/WiMOB.2011.6085412
- [8] Muhammad Ahmad, Adil Mehmood Khan, Joseph Alexander Brown, Stanislav Protasov, Asad Masood Khattak. 2016. Gait fingerprinting-based user identification on smartphones. Published in: 2016 International Joint Conference on Neural Networks (IJCNN). DOI: 10.1109/IJCNN.2016.7727588
- [9] 佐藤 悠祐, 神山 剛, 福田 晃, 小口 正人, 山口 実靖, "加速度センサによるスマートフォンユーザの推定に関する考察", IPSJ 研究報告コンシューマ・デバイス&システム (CDS), 2019-CDS-26 巻, 2 号, pp. 1 - 6, 2019
- [10] Yusuke Sato, Shun Kurihara, Shoki Fukuda, Masato Oguchi, and Saneyasu Yamaguchi. 2017. Height Estimation based on Sensor Data on Smartphone. In *Proceedings of the 15th International Conference on Advances in Mobile Computing & Multimedia (MoMM2017)*. Association for Computing Machinery, New York, NY, USA, 102-106. DOI: <https://doi.org/10.1145/3151848.3151877>
- [11] "SensorManager | Android Developers" <https://developer.android.com/reference/android/hardware/SensorManager.html> <<accessed 2021/04/21>>