「マルチメディア，分散，協調とモバイル
(DICOMO2019) シンポジウム」 令和元年7月

スマートロックにおける異常検知を用いた二つの端末の加速度による歩行認証の提案

渡辺 一樹[1] 長友 誠[1] 油田 健太郎[2] 岡崎 直宣[2] 朴 美娘[1]

概要：電子的に鍵の開閉を行うスマートロックと呼ばれる IoT(Internet of Things) 製品がある．そのスマートロックの認証方式として，従来のパスワード方式ではユーザの記憶負荷やパスワード漏洩などの危険性がある．また，顔認証方式では家の前で数秒間立ち止まる必要があり，マスクやサングラスなどを身に付けていればそれらを外す必要がある．そこで，我々は歩行状態の行動的特徴を用いる歩行認証に着目した．これまで，スマートフォンから得られる加速度データや角速度データをもとに，機械学習の分類を用いた歩行認証の研究が行われてきたが，教師あり学習の分類を用いた認証を行っているため未知のデータに対して有効な認証方式とは言えない．本研究では，スマートフォンとウェアラブル端末の二つの端末の加速度センサからデータを取得し，機械学習の異常検知を用いて識別器を作成することで未知のデータに対して有効な歩行認証のシステムモデルを提案する．また，システムモデルの FAR(False Acceptance Rate) や FRR(False Rejection Rate) を確認する実験を行った結果，異常検知アルゴリズムに IsolationForset を用いた場合，平均 FAR が 8.3%，平均 FRR が 9.5%であった．さらに，各被験者の FAR と FRR を算出し，被験者によって FAR や FRR が低くなる異常検知アルゴリズムが異なることを確認した．

Gait-authentication by Acceleration of Two Devices Using Anomaly Detection in Smart Lock

KAZUKI WATANABE[1] MAKOTO NAGATOMO[1] KENTARO ABURADA[2] NAONOBU OKAZAKI[2] MIRANG PARK[1]

1. はじめに

近年，物のネットワーク化が進んでおり，例えばインターネットと繋がる自動車や，インターネットと繋がる AI スピーカーなどの製品が開発されている．このような製品は IoT 製品と呼ばれ，私たちの生活の中に浸透しつつある．その中で，スマートロックと呼ばれる家のドアの鍵に取り付けパスワード入力などを用いて鍵の開閉を行う IoT 製品がある．2015 年はスマートロック元年と言われ，現在までに August[1] や Akerun[2]，Qrio[3] などのスマートロック製品が開発されてきた．しかし，そのスマートロックの認証方式として，従来のパスワード方式では，パスワードを記憶しなくてはならないためユーザに記憶負荷がかかるという問題点があり，推測されやすいパスワードを設定した場合，パスワードクラックを受けてしまう問題点がある．また，カードやスマートフォンなどを用いた所持認証による認証方式では，盗難や置き忘れ，紛失などの問題点がある．これらの問題を解消する方式として個人の身体的特徴を用いる生体認証方式がある．しかし，この生体認証方式でもスマートロックにおける認証の煩わしさを完全に排除できていない現状がある．例えば，指紋認証方式では，認証を行う際に家のドアの前で数秒間立ち止まる必要があり，手袋などを身に付けていればそれを外す必要がある．また，顔認証方式においても認証を行う際にカメラに顔を近づけ数秒間立ち止まる必要があり，マスクやサングラスなどを身に付けていればそれらを外す必要がある．

そこで，我々は歩行状態の行動的特徴を用いる歩行認証に着目した．スマートロックにおける認証方式として歩行認証を用いることで，スマートロックから離れた場所で認

[1] 神奈川工科大学
Kanagawa Institute of Technology
[2] 宮崎大学
University of Miyazaki

証を開始し，スマートロックに近づいた場所で認証を終了することで自動的に認証を行うことができ，スマートロックにおける認証の煩わしさを解消できると考える．しかし，行動的特徴を用いる認証方式は，他人に行動を真似されなりすまし攻撃を受けてしまう可能性がある．この問題に対して Muaaz ら [4] は，スマートフォンから取得する加速度データをもとに行う歩行認証方式が，なりすまし攻撃に対して有効か確認を行っている．実験では，アクターズスクールの学生が攻撃者として登録者の歩行を真似し，認証が成功するか確認を行っている．結果としてほとんどの場合，攻撃者は認証を成功させることができず，歩行認証がなりすまし攻撃に対して有効であることを示している．

これまで，歩行認証の精度向上のため多くの研究が行われてきた [5], [6], [7]．これらの研究では，機械学習や複数のセンサを用いることで認証精度の向上を行っている．しかし，登録したときの端末の向きと認証するときの向きを同じにする必要があるという問題点があり，機械学習の分類を用いた認証を行っているため想定ユーザ以外の未知のデータに対して有効な認証方式とは言えない．また，体に装着した 8 つのセンサからデータを取得することで歩行認証を行う研究も行われている [8]．単一のセンサを用いる場合より認証精度が大きく向上しているが，大量のセンサを身に付けなくてはならないためユーザにとって望ましい認証方式とは言えない．

そこで，以前我々は普段身に着けているスマートフォンや増加が予測されるウェアラブル端末の二つの端末を用いて認証を行う歩行認証方式の提案を行った [9]．二つの端末から取得した合成加速度を用い特徴量を抽出することで端末の向きを考慮せずに認証を行い，その正解率を確認した．しかし，認証には教師あり学習の分類を用いているため，想定したユーザ以外のデータに対して，登録したユーザのいずれかに分類されてしまい，未知のデータに対して有効な認証方式とは言えない．

そこで，本論文では未知のデータに対して有効な認証を行うため，教師なし学習の異常検知を用いた歩行認証の提案を行う．提案方式では，以前我々が提案した方式 [9] と同様にスマートフォンとウェアラブル端末の二つの端末から加速度データを取得し，それぞれの合成加速度を用いた特徴量の抽出を行う．また，実験を行い取得した被験者の歩行データをもとにそれぞれの識別器を作成し，想定していないユーザのデータに対して異常を検知しすることでその FAR(False Acceptance Rate) と FRR(False Rejection Rate) の確認を行う．

以降，2 章では，関連研究について述べ，3 章では，提案方式を示し，加速度処理や特徴量抽出について述べる．4 章では，評価実験について述べる．5 章では，まとめと今後の課題について述べる．

2. 関連研究

2.1 加速度センサを用いた歩行認証

彭ら [5] は，スマートフォンの加速度データをもとに，x, y, z の 3 軸加速度データからそれぞれ平均値 (3 軸)，標準偏差 (3 軸)，平均絶対偏差 (3 軸)，平均合成加速度，ピーク間の時間 (3 軸)，ビン分布 (3 軸 $\times$ 10 個) の合計 43 個の特徴量を抽出し機械学習を用いて分類を行っている．機械学習にはデータマイニングソフトである WEKA を用い，正解率の確認を行っている．また，特徴量ごとの正解率を確認し認証に貢献する特徴量や貢献しない特徴量について確認を行っている．結果として，機械学習の分類アルゴリズムに決定木 J48 を用いた場合は FAR が 0.6%，FRR が 8.7%であり，分類アルゴリズムにニューラルネットワークを用いた場合は FAR が 0.3%，FRR が 3.8%であった．また，認証に貢献する特徴量としては平均値を挙げ，認証に貢献しない特徴量としては平均合成加速度やピーク間の時間を挙げている．しかし，この研究では 3 軸加速度データをそれぞれ用いた特徴量の抽出を行っているため，端末の向きを考慮しなくてはならないという問題点がある．

2.2 加速度センサと角速度センサを用いた歩行認証

今野ら [6] は，加速度センサと角速度センサを用いた歩行認証の提案をしている．ここでは，加速度データと角速度データを取得し，登録したそれぞれのデータと取得したデータの距離を DTW(Dynamic Time Warping) を用いて計算し，機械学習の SVM(Support Vector Machine) を用いて認証精度の確認を行っている．また，センサを装着する位置として，腰部と脚部でそれぞれ装着した場合の認証精度の確認も行っている．結果として，脚部と腰部ではそれぞれ同程度の精度で認証可能であることを示し，脚部認証に関して EER(Equal Error Rate) が 0.8%という結果であった．しかし，3 軸それぞれの加速度データや角速度データを用いているためセンサの向きを考慮しなくてはならないという煩わしさがある．

2.3 3 つの推定を行う歩行認証

スマートフォンの加速度データに対して，状態推定 (静止状態，歩行状態，走行状態), 所持位置推定 (前ポケット，後ろポケット，胸ポケット，画面を見る位置，腕を振る位置)，ユーザの推定の 3 つの推定を行う研究が行われている [7]．状態推定に関しては特徴量に合成加速度の最大値や最小値，高速フーリエ変換のスペクトルを用い，所持位置推定に関しては 3 軸加速度データそれぞれの平均や変化量，合成加速度の最大値や最小値を用いている．また，ユーザの推定に関しては合成加速度の最大値や最小値，高速フーリエ変換のスペクトルを特徴量としている．ここ

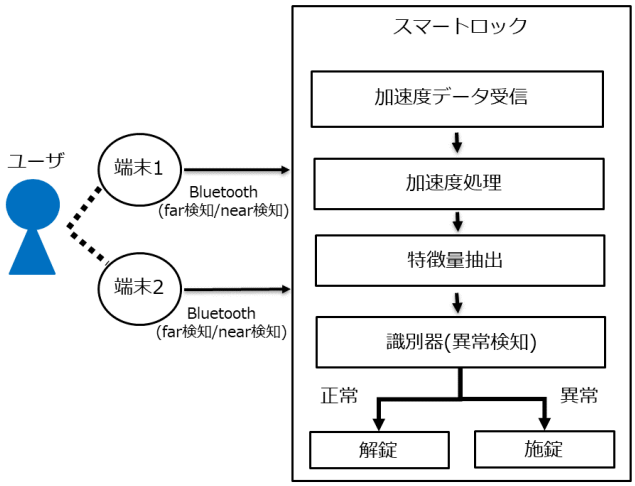

図 1　提案システムモデル

で，得られた特徴量から機械学習の分類を用いてそれぞれの正解率の確認を行っている．結果として，状態推定に関しては 99.7%，所持位置推定に関しては 99.4%，ユーザ推定に関しては 97.0%という正解率であった．所持位置推定の結果から，同一身体上に身に付けた端末でも端末を所持する位置によって異なったデータを取得できると考えられる．そのため，複数の異なった部位に身に付けたセンサからデータを取得し，それらのデータを統合することで歩行認証の精度向上が期待できる．

2.4　複数のセンサ端末を用いた歩行認証

体に複数の端末を装着した状態で歩行し認証を行う研究が行われている [8]．この研究では，体の関節 8 箇所 (右肩，左肩，右腕，左腕，右腰，左腰，右足，左足) にそれぞれセンサを装着し得られる角度データをもとに特徴量を抽出し歩行認証を行っている．結果として，ANN(Artificial Neural Network) や LDA(Linear Discriminant Analysis) を用い，100%の精度で分類を行っている．しかし，体に大量のセンサを身に付けなくてはならないため，ユーザにとって望ましい認証方式とは言えない．

そこで，以前我々は多くの人が身に付けると考えられるスマートフォンとウェアラブル端末の二つの端末を用いた歩行認証の提案を行った [9]．二つの端末の加速度データから特徴量を抽出し，教師あり学習の分類を用いてその正解率を確認した．結果として，分類アルゴリズムに Random Forest を用いた場合，正解率が 95.3%であり，極大値間隔や最小値，極小値間隔が認証に貢献することを示した．しかし，認証には教師あり学習の分類を用いたため，想定ユーザ以外の未知のデータに対して有効な認証方式とは言えない．

3.　提案方式

3.1　システムモデル

提案方式では，二つの端末から得られる加速度データをもとに，教師なし学習の異常検知を用いることで認証を行う．提案システムモデルを**図 1** に示す．提案システムモデルは，加速度センサが搭載されたスマートフォンやスマートウォッチなどの二つの端末と，加速度データを受信し認証を行うスマートロックで構成される．二つの端末とスマートロックは Bluetooth で接続し，BLE(Bluetooth Low Energy) の近接検知を用いて加速度計測を行う．加速度計測は，スマートロックが二つの端末との近接検知を行い，far(約 1m〜約 50m) を検知した際に加速度計測を開始し，near(約 2cm〜約 1m) を検知した際に加速度計測を終了することで行う．計測した加速度データはそれぞれスマートロックへ送信し，スマートロック内で加速度処理を行う．ここで，端末の向きを考慮せず認証を行うため合成加速度を算出し，特徴量を抽出しやすくするため歩行状態の抽出やノイズの削除を行う．また，加速度処理を行ったデータから特徴量の抽出を行い，機械学習の異常検知を用いて識別器を作成する．作成した識別器を用いて認証を行い，正常データとして判定された場合に解錠し，異常データとして判定された場合施錠することで鍵の開閉を行う．

3.2　加速度処理

本節では，ユーザが身に付けた二つの端末の加速度データから特徴量を抽出するための加速度処理について述べる．まず，端末の向きを考慮せずに認証を行うため合成加速度の算出を行う．ここで，端末 1 と端末 2 の合成加速度 r_i^1, r_i^2 を以下に示す．なお，取得した x, y, z 軸の加速度データを端末 1 では x_i^1, y_i^1, z_i^1 と表し，端末 2 では x_i^2, y_i^2, z_i^2 と表す．

$$r_i^1 = \sqrt{(x_i^1)^2 + (y_i^1)^2 + (z_i^1)^2} \tag{1}$$

$$r_i^2 = \sqrt{(x_i^2)^2 + (y_i^2)^2 + (z_i^2)^2} \tag{2}$$

また，二つの端末から取得した合成加速度データの集合 d_1, d_2 を以下に示す．ここで，スマートフォンとウェアラブル端末の加速度データの計測開始から i 番目にデータを取得した時刻を t_i^1, t_i^2 とし，計測開始から終了までに取得したデータ数を n_1, n_2 と表す．

$$d_1 = \left\{ (t_i^1, r_i^1) | i \in \{1, ..., n_1\} \right\} \tag{3}$$

$$d_2 = \left\{ (t_i^2, r_i^2) | i \in \{1, ..., n_2\} \right\} \tag{4}$$

二つの端末から得らえれる加速度データの類似度を特徴

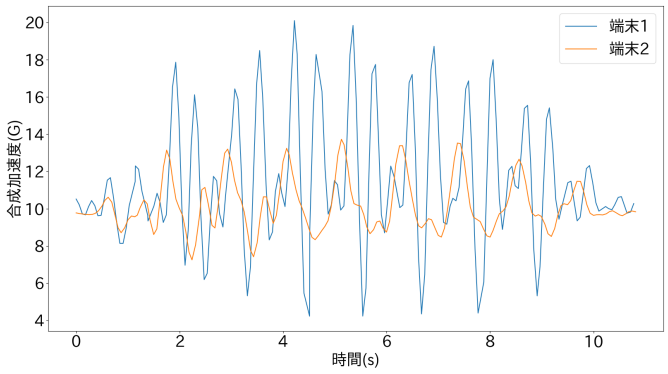

図 2　加速度処理後の合成加速度データ

量とする際に，サンプル数の違いによりずれが生じるため，二つの端末の加速度データを取得した時刻差からサンプル数を合わせる．例えば，2 つのサンプル数の大小関係が $n_1 > n_2$ であった場合の，サンプル数を合わせたときのデータの集合 d'_1, d'_2 を以下の式で示す．

$$d'_1 = \left\{ (t'^1_i, r'^1_i) | i \in \{1, ..., n_2\} \right\} \tag{5}$$

$$d'_2 = \left\{ (t'^2_i, r'^2_i) | i \in \{1, ..., n_2\} \right\} \tag{6}$$

ここで，t'^1_i，r'^1_i，t'^2_i，r'^2_i は以下の式で計算される．

$$t'^1_i = t^1_{\underset{j \in \{1,...,n_1\}}{\arg\min} (|t^1_j - t^2_i|)} \tag{7}$$

$$r'^1_i = r^1_{\underset{j \in \{1,...,n_1\}}{\arg\min} (|t^1_j - t^2_i|)} \tag{8}$$

$$t'^2_i = t^2_i \tag{9}$$

$$r'^2_i = r^2_i \tag{10}$$

また，歩行状態を抽出するため，一定区間の加速度データで標準偏差が高い区間を歩行データとして抽出する．その後，特徴量を抽出しやすくするため，フーリエ変換を用いたローパスフィルタで加速度データのノイズを除去する．**図 2** に上記の加速度データ処理を行った後の波形データを示す．青い波形は右ポケットに入れたスマートフォンの合成加速度データを示し，オレンジ色の波形はウェアラブル端末であるスマートウォッチを左腕に装着した場合の合成加速度データを示す．ここで，加速度処理が行われ，歩行状態が抽出されたことが確認できる．

3.3　特徴量抽出

機械学習の異常検知を用いた歩行認証を行うため，二つの端末から得られる加速度データをもとに特徴量の抽出を行う．特徴量抽出には，二つの端末の特徴量の差や二つの端末の類似度などを用い，一つの端末では取得できない特徴量の抽出も行う．以下に抽出する特徴量を示す．

- 平均値 (端末 1，端末 2)
- 二つの端末の平均値の差
- 標準偏差 (端末 1，端末 2)
- 二つの端末の標準偏差の差
- 最大値 (端末 1，端末 2)
- 二つの端末の最大値の差
- 最小値 (端末 1，端末 2)
- 二つの端末の最小値の差
- 極大値間隔の中央値 (端末 1，端末 2)
- 極小値間隔の中央値 (端末 1，端末 2)
- 類似度 (SSD: Sum of Squared Difference)

$$SSD = \min_{k \in \{1,...,\frac{2}{3}n\}} \frac{1}{n-k} \sum_{i=1}^{n-k} (r'^1_i - r'^2_{i+k})^2 \tag{11}$$

- 類似度 (NCC: Normalized Cross Correlation)

$$NCC = \max_{k \in \{1,...,\frac{2}{3}n\}} \frac{\sqrt{\sum_{i=1}^{n-k} r'^1_i r'^2_{i+k}}}{\sqrt{\sum_{i=1}^{n-k} (r'^1_i)^2} \sqrt{\sum_{i=1}^{n-k} (r'^2_{i+k})^2}} \tag{12}$$

- 類似度 (DTW: Dynamic Time Warping)

$$DTW = f(n, n)$$

$$f(i,j) = |r'^1_i - r'^2_j| + min \begin{cases} f(i-1, j-1) \\ f(i-1, j) \\ f(i, j-1) \end{cases} \tag{13}$$

なお，サンプル数を n とし，$i, j \in \{1, ..., n\}$ とする．また，SSD と NCC の類似度算出に関して二つの端末の動かし方が似ているかを特徴量とするため，二つの加速度データを時間軸方向にずらした時の最も高い類似度を特徴量とする．ここで，二つの端末の合成加速度をサンプル数の数ずらすと類似度を比較するサンプル数が少なくなってしまうため，ずらす個数を $k \in \{1, ..., \frac{2}{3}n\}$ とする．上記の特徴量をもとに教師なし学習の異常検知を用いて識別器を作成し認証を行う．また，認証精度向上のため特徴量の選択を行うことで FAR と FRR の確認を行う．

4.　評価実験

4.1　実装

提案システムの実験を行うため，加速度を取得する二つの端末として，スマートフォン (Xperia XZs) と時計型のウェアラブル端末であるスマートウォッチ (SmartWatch 3) を用い，それぞれの Android システムから加速度データを取得するプログラムを Java で作成する．また，スマートロックの代わりにサーバを用意し，二つの端末から加速度データを受信するプログラムを Python で作成する．なお，

図 3 実験の様子

ここでは BLE の近接検知の精度により異なった歩行距離のデータが取得されることを防ぐため，手動で各端末へ計測開始命令と計測終了命令を送信するプログラムを作成する．さらに，得られた二つの端末の加速度データから特徴量を抽出し，教師なし学習の異常検知を用いて認証を行うプログラムを Python のモジュール Scikit-learn[10] を用いて作成する．異常検知には Elliptic Envelope，GMM(Gaussian Mixture Model)，Isolation Forest，KDE(Kernel Density Estimation)，LOF(Local Outlier Factor)，One Class SVM の 6 つの異常検知アルゴリズムを用いて FAR と FRR の確認を行う．

4.2 実験

提案方式の有効性を確認するための実験を行う．実験は，**図 3** のように平坦な廊下で行い，右ポケットにスマートフォン (Xperia XZs) を入れ，左腕にスマートウォッチ (SmartWatch 3) を装着した状態で約 10m 歩行してもらった．被験者 11 名に 50 回歩行してもらい，その内の 30 回を学習データとし，20 回を正常データとした．また，他の被験者 10 名に異常データとして 20 回の歩行を普段通り行ってもらった．実験の手順を以下に示す．

(1) 二つの端末がそれぞれ加速度計測を開始する．
(2) 被験者はその場で 5 秒間静止する．
(3) 被験者は廊下を 10m 歩行する．
(4) 被験者はその場で 5 秒間静止する．
(5) 二つの端末がそれぞれ加速度計測を終了する．

なお，手順 (2), (4) では，手順 (3) の歩行データをより正確に抽出するため 5 秒間の静止を行ってもらった．以上の手順により得られた加速度データから加速度処理を行い特徴量の抽出を行う．その後，教師なし学習の異常検知を用いて正常データか異常データかを判定することで FAR と FRR の確認を行う．

4.3 結果と考察

実験で得られた加速度データをもとに機械学習の異常検知を用いて FAR と FRR の確認を行う．機械学習には Elliptic Envelope，GMM，Isolation Forest，KDE，LOF，One Class SVM の 6 つの異常検知アルゴリズムを用いる．学習データとして実験を行った 11 名の 30 回のデータをもとにそれぞれ識別器を作成する．作成した識別器にそれぞれの正常データと，異常データとして計測を行った 10 名のデータを識別させることで FAR と FRR の確認を行う．

特徴量選択による認証精度向上のため，抽出した 19 個の特徴量を 13 グループ (平均値，平均値の差，標準偏差，標準偏差の差，最大値，最大値の差，最小値，最小値の差，極大値間隔の中央値，極小値間隔の中央値，SSD，NCC，DTW) に分割し，それぞれの特徴量の組み合わせに対して FAR と FRR を算出する．例えば，平均値のグループでは端末 1 の平均値，端末 2 の平均値をそのグループとする．**表 1** に学習アルゴリズムごとの最も認証精度が良くなった特徴量の組み合わせと，全ての被験者の平均 FAR と平均 FRR を示す．ここで，機械学習のパラメータはこの平均 FAR と平均 FRR が小さくなるように調整した．結果として，異常検知アルゴリズムごとに最適な特徴量の組み合わせが異なり，認証精度に差があることが確認できた．また，GMM や Isolation Forest を用いた場合に FAR と FRR が低くなり，GMM を用いた場合に FAR が 13.6%，FRR が 3.6%であり，Isolation Forest を用い場合に FAR が 8.3%，FRR が 9.5%であった．また，提案方式では，学習に本人のデータのみを学習させるため，個人の歩行のばらつきによって認証精度が大きく低下することが予測される．そのため，GMM と Isolation Forest の被験者ごとの FAR と FRR の確認を行った．**表 2** に被験者ごとの FAR と FRR を示す．ここで，同じ識別器の作成方法でも，被験者ごとに認証精度に差があることが確認できた．さらに，同じ被験者でも認証精度が良くなる識別器が異なることが確認できた．以上より，複数の異常検知アルゴリズムの結果をもとに異常を判定することによって，認証精度の向上が期待できる．

5. おわりに

本研究では，二つの端末を用い未知のデータに対しても有効な歩行認証を行うため，教師なし学習の異常検知を用いた認証方式を提案した．提案方式では，体に身に付けた二つの端末から加速度データを取得し，それぞれの合成加速度を用いた特徴量の抽出を行う．実験として，被験者 11 名の加速度データを学習データと正常データとして取得し，別の被験者 10 名の加速度データを異常データとして取得した．取得した学習データから機械学習の異常検知アルゴリズムを用いそれぞれの識別器を作成し，正常データと異常データの判定を行うことで FAR と FRR の確認

表 1 異常検知を用いた提案システムの FAR と FRR

機械学習 (異常検知)					
Elliptic Envelope	GMM	Isolation Forest	KDE	LOF	One Class SVM
特徴量の選択					
標準偏差の差	平均値	平均値	平均値	標準偏差の差	平均値
最大値	平均値の差	標準偏差	平均値の差	最大値	平均値の差
最大値の差	標準偏差	標準偏差の差	標準偏差	最大値の差	標準偏差
最小値の差	標準偏差の差	最大値	標準偏差の差	最小値	最大値
DTW	最大値	最小値	最小値	最小値の差	最大値の差
極大値間隔	最小値	最小値の差	最小値の差	極大値間隔	最小値
	最小値の差	SSD	NCC	極小値間隔	最小値の差
	SSD	NCC	DTW		SSD
	NCC	DTW	極大値間隔		NCC
	極大値間隔	極小値間隔			極大値間隔
FAR					
7.8%	13.6%	8.3%	15.2%	12.4%	6.4%
FRR					
12.2%	3.6%	9.5%	17.7%	13.1%	17.2%

表 2 被験者ごとの FAR と FRR

	機械学習 (異常検知)			
	GMM		Isolation Forest	
	FAR	FRR	FAR	FRR
被験者 A	6.0%	5.0%	4.0%	10.0%
被験者 B	42.5%	0.0%	16.5%	0.0%
被験者 C	28.5%	0.0%	6.5%	25.0%
被験者 D	0.0%	10.0%	0.0%	5.0%
被験者 E	21.5%	0.0%	3.5%	5.0%
被験者 F	2.5%	0.0%	7.5%	5.0%
被験者 G	0.0%	5.0%	0.0%	0.0%
被験者 H	0.5%	0.0%	13.0%	10.0%
被験者 I	37.5%	15.0%	23.0%	20.0%
被験者 J	8.5%	5.0%	9.5%	25.0%
被験者 K	3.0%	0.0%	8.5%	0.0%

を行った．また，認証精度向上のため，特徴量をグループに分割しそのグループの組み合わせに対して平均 FAR と平均 FRR を計算することで特徴量の選択を行った．結果として，異常検知アルゴリズムに GMM を用いた場合は FAR が 13.6%，FRR が 3.6%となり，Isolation Forest を用いた場合は FAR が 8.3%，FRR が 9.5%となった．また，GMM と Isolation Forest の被験者ごとの FAR と FRR を確認し，被験者ごとに FAR と FRR が低くなる異常検知アルゴリズムが異なることを確認した．このことから複数の異常検知アルゴリズムの予測結果から異常を判定することによって認証精度の向上が期待できる．

今後は，スマートウォッチなどの端末の向きを考慮しなくてもよいウェアラブルデバイスから得られる加速度データから，3 軸それぞれの特徴量を抽出し認証精度が向上するか確認を行う予定である．また，端末ごとの ID などによる所持認証と，本提案システムである歩行認証の二要素認証についての検討を行う予定である．

参考文献

[1] august: August Smart Lock, ⟨https://august.com⟩ (参照 2019-05-07).

[2] 株式会社フォトシンス: 入退室管理システムなら Akerun オフィス向けスマートロック, ⟨https://akerun.com/feature ⟩ (参照 2019-05-07).

[3] Qrio: Qrio Smart Lock, ⟨https://qrio.me/smart lock⟩ (参照 2019-05-07).

[4] M. Muaaz, R. Mayrhofer: Smartphone-Based Gait Recognition: From Authentication to Imitation, IEEE Transactions on Mobile Computing, Vol. 16, No. 11, (2017).

[5] 彭 龍, 渡邉 裕司: スマートフォンの加速度センサを用いた歩行時の認証に関する一考察, Computer Security Symposium, pp.21-23 (2013).

[6] 今野 慎介，中村 嘉隆, 白石 陽，高橋 修: 複数のウェアラブルセンサを用いた歩行動作による本人認証法の精度向上, 情報処理学会論文誌, Vol.57, No.1 pp.109-122 (2016).

[7] 岩本 健嗣, 杉森 大輔, 松本 三千人: 3 軸加速度センサを用いた歩行者推定手法, 情報処理学会論文誌, Vol.55, No.2 pp.734-749 (2014).

[8] S. Mondal, A. Nandy, P. Chakraborty, et al.: Gait Based Personal Identification System Using Rotation Sensor, Journal of Emerging Trends in Computing and Information Sciences, Vol.3, No.3, pp.395-402 (2012).

[9] 渡辺 一樹，長友 誠, 油田 健太郎，岡崎 直宣，朴 美娘: スマートフォンとウェアラブル端末の加速度センサを用いたスマートロックにおける歩行認証，Computer Security Symposium, pp.173-178 (2018).

[10] scikit-learn: scikit-learn machine learning in Python scikit-learn 0.19.1 documentation, ⟨http://scikit-learn.org/stable/index.html⟩ (参照 2019-05-07).