L-012

マウスカーソルの軌跡を用いた個人認証

Personal authentication using mouse cursor path

岸 海渡　　堂薗 浩
Kaito Kishi　Hiroshi Dozono

1. はじめに

個人認証とは，システム利用者が登録者本人であるかを識別するプロセスで，情報化社会が加速する現代で重要な要素となる．特にパソコンの利用者が増加し機密情報を取り扱う機会が多くなり，その個人認証にはパスワード入力，指紋認証，顔認証が代表例として挙げられる．パスワード入力はキーボード入力による盗み見の危険性があることに加え，個人情報として入手しやすいものを使用していることからセキュリティ上の脆弱性，指紋，顔認証は高精度なものの専用の機器の必要性が課題となる．そこでマウスカーソルの軌跡からログイン認証時の画面上の座標分布，速度変化等の特徴量を利用し機械学習による個人の識別を行う個人認証システムの作成，精度の検討を行った．

2. 実験方法

2. 1　実験 1　座標データと速度データでの異常検知

本実験で作成した座標取得プログラムは，

1. 右クリックでデータの取得を開始
2. 3 秒間の経過時間，X 座標，Y 座標を配列に逐次記録
3. 3 秒経過後，配列データを npy ファイルで保存

の 3 工程が実行される．加えてスリープ状態から起動する際の再現として，座標取得プログラム実行と同時に全画面のウィンドウを疑似的なスリープ画面として表示させる(図 1).

.さらに，のぞき見された場合の対策を想定し，画面の録画を行う．その後録画データを本人以外の被験者が視聴し，カーソルの動きを真似して追加のデータを 2 データずつ取得し，本人データ 10 データ，他者の模倣データ 8 データをサンプルとして使用する．また，本人データと模倣データを合わせた際の異常検出を行う．

図 2 に実験 1 で取得した被験者のカーソル軌跡の例を示す．取得したデータを分類用に調整を行い 1 列のデータに再配列する．1 列のデータに並べられた 3 秒間の X 座標，Y 座標，速度データを特徴量として，教師データ，検証データに分割して機械学習の手法である IsolationForest[1]を用いる．

異常検出では各パラメータの組み合わせごとに 10 回平均を求める．各ユーザの認証率が最も高くなるパラメータを選別し，その 5 パターンの組み合わせで 5 人の自他分類を行った際の確率を求め，検証する．

2.3　実験 2　時間指定のない記録での異常検知

次に，記録プログラムを改良し，マウスの動きによって記録を開始，右クリックで記録を終了する仕組みに変更した．このプログラムは使用ライブラリを変更し，より短い時間刻みで正確にデータを記録できるものとなった．

加えて，図 3 で示す円，三角形，四角形といった図形を複数並べた背景に変更し，図形をなぞるように具体的な指示を行ったうえで軌跡を取得した．このプログラムを用いて，1 人当たり本人データ 30～90，模倣データ 20～30 を記録した(図 4). このサンプルを用いて，異常検知を行う．

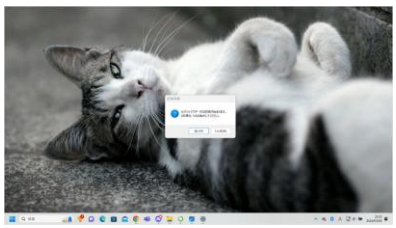

図 1:プログラム実行時のスクリーンショット

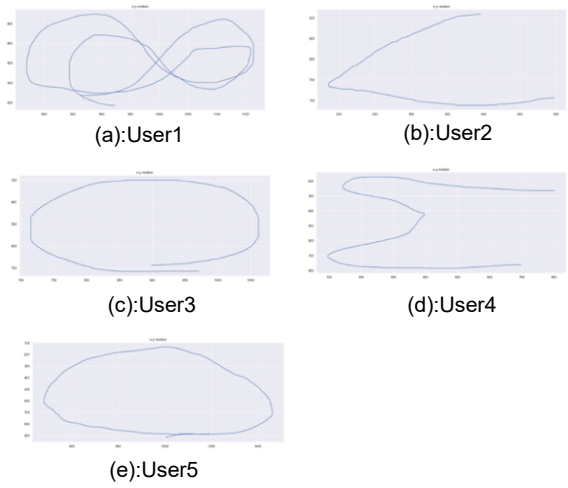

図 2:5 人の被験者軌跡データ

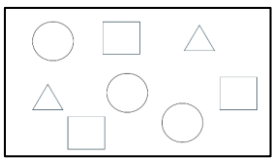

図 3:実験 2 実行時の背景

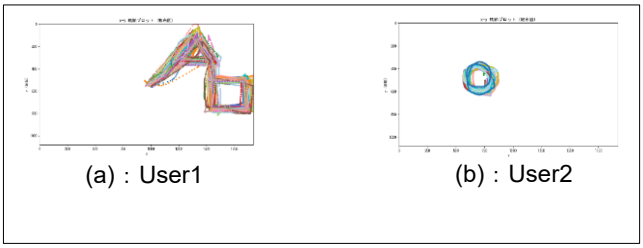

図 4:被験者記録データ

3. 実験結果と検討

実験 1 での異常検出の結果を表に示す．. 他人の軌跡を模倣したデータを各 2 データずつ取得し，本人データ 10，他人データ 8 で，本人データ 80%を学習データとして異常検知を行った．

評価のために調整するパラメータは n_estimators(決定木の本数決定), contamination(モデル感度)の 2 つである. n_estimators は利用する決定木の本数で, この数値が高いほど精度は上がるが, 高すぎると過学習の要因となる. contamination はデータセット内の異常の割合を変えることでモデルの感度を調整するパラメータで, 異常の割合が高いほど, モデルは異常を検知しやすくなるが、正常データを誤って異常と判定する可能性も増える.

最良パラメータの決定基準は, 本人認証率と他人拒否率がともに 80%以上となることとした. User2 から User4 は, このパラメータで本人認証率, 他人拒否率がともに 80%以上と高い精度での異常検知ができていたが, User5 は他人認証率が 26.25%, User1 に関しては異常検知の数値は高かったものの, 本人認証率が 50%を下回るという結果になった. 表 2(a)での User1 が最良のパラメータの数値は他の結果に比べ決定木の数値が高く, かつ 90%以上の本人認証率を出していないことから, 画像に沿った一筆書きのような図形を描く軌跡に比べ, 再現性の低い軌跡の識別は難しいことがわかる.

次に, 実験 2 で同様に異常検知を行った結果を表 2 に示す. 本実験では, 本人データのうち 30%を学習させ, 残りの本人データと模倣データを検証に用いる. 異常検知には引き続き IsolationForest を用いた. 本人認証率は 90%程度であったが, 他人認証率の数値が大きくなり, 異常検知ができていない結果となった. これは, プログラムの改良による, 速度成分の取得データ数に影響を受けていると考えられる. そこで, 特徴量抽出の 2 つの手法を用いて次元数を減らしたデータで, 同様に分類を行った. まずは, 先行研究[2]で行われていた表 3(a)に示す特徴量を抽出した手法である. この特徴量での分類結果を表 3(b)に示す. 先ほどの結果に比べ, 他人認証率の値が大きく下がっていることが分かる. そこで, 時系列解析で用いられる pycatch22[3]を用いて座標, 速度データをそれぞれ 22 次元の特徴量にする手法を用いた結果を表 4 に示す. この手法では, 各要素(x 座標, y 座標, 速度)をそれぞれ 22 次元, 合計 66 次元で学習させたものである. こちらも User1 が示すように他人認証率の値が大きく下がっていることが分かる. User1 より User2 の結果が悪い要因として, 模倣しやすい軌跡である点があげられる. 2 つ以上の図形に沿った軌跡と, 1 つの図形に沿った軌跡によって, 認証率がどれだけ関係するかを検討していく必要がある.

表 1 :各ユーザの最も評価の良いパラメータと確率

ユーザ名	n_estimators	contamination	本人認証率[%]	本人拒否率[%]	他人拒否率[%]	他人認証率[%]
user1	100	0.1	80.00	20.00	95.00	5.00
user2	110	0.2	100.00	0.00	98.75	1.25
user3	40	0.2	100.00	0.00	97.50	2.50
user4	40	0.2	100.00	0.00	96.25	3.75
user5	50	0.4	90.00	10.00	90.00	10.00

表 2 :各要素で分類した結果

	本人認証率[%]	本人拒否率[%]	他人認証率[%]	他人拒否率[%]
User1	95.80	4.20	80.30	19.70
User2	92.70	7.30	86.00	14.00

表 3(a) :抽出した特徴量

マウス操作に要した時間
カーソル速度の平均
カーソル速度の中央値
カーソル速度の標準偏差
軌跡サイズのx軸の大きさ
軌跡サイズのy軸の大きさ
操作の開始から終了までの総移動距離
データ取得量

表 3(b) :抽出特徴量での異常検知結果

	本人認証率[%]	本人拒否率[%]	他人認証率[%]	他人拒否率[%]
User1	94.57	5.43	19.35	80.65
User2	80.95	19.05	0.00	100.00

表 4 :pycatch22 での異常検知結果

	本人認証率[%]	本人拒否率[%]	他人認証率[%]	他人拒否率[%]
User1	90.90	9.10	6.50	93.50
User2	93.33	6.70	20.00	80.00

4. まとめ

本研究ではログイン認証時のマウスカーソルの軌跡を用いた個人認証プログラムの作成を行い, 座標, 速度データを用いた異常検出の検証と考察を行い, 実現性を示した. 秒数指定での本人データと異常データを識別した際の異常検出の精度は,全体でみると認証率は高い部類ではあったが, 本人認証率が 60%程度となってしまった結果も見られた.

また, 時間指定無しでの分類を行った結果, 実験 1 と比較して得られた速度成分の数が異なり, 直接データを用いて分類を行うと, 他人認証率が大きくなってしまう結果となった. そこで, 次元数を減らす手法を用いて異常検知を行ったところ, いずれの結果においても他人認証率の値は大きく下がる結果となった.

今後の課題として, 軌跡の複雑さが認証率にどれだけ関係するかの比較を行うとともに, 特徴量抽出の分類手法の結果を組み合わせた異常検知の検討を行う.また, 学習データの割合を変更した場合の影響, 及び常時軌跡の記録を行った場合についても調査を行う.

謝辞

本研究を行うにあたって, 日頃から親切かつ丁寧なご指導をいただいた堂薗 浩准教授並びに諸先生方に厚く御礼申し上げます. また, 本研究についてご助言, ご協力いただいた本研究室の学部生, 大学院生の皆様に深く感謝申し上げます.

参考文献

[1] 株式会社システムインテグレータ, " 異常検出アルゴリズムの代表格「Isolation Forest」とは?", https://products. sint. co. jp/aisia-ad/blog/what-is-isolation-forest
[2] 須田 恭平, "日常的な家電操作による人物識別のためのマウス操作による検討" (2022)
[3] Python Package Index, "pycatch22", https://pypi.org/project/pycatch22/0.4.1/