スマートフォンにおける 耐模倣性向上を目指したパッシブ認証学習手法の提案

工藤 雅士†1　高橋 翼‡2　牛山 翔二郎†1　山名 早人§3

†1 早稲田大学大学院基幹理工学研究科 〒169-8555 東京都新宿区大久保 3-4-1

‡2 LINE 株式会社 〒160-0004 東京都新宿区四谷一丁目 6 番 1 号 四谷タワー23 階

§3 早稲田大学理工学術院 〒169-8555 東京都新宿区大久保 3-4-1

E-mail: †§ {kudoma34, ushiyama, yamana}@yama.info.waseda.ac.jp, ‡ tsubasa.takahashi@linecorp.com

あらまし　近年，スマートフォンのセキュリティ性向上のため，顔認証や指紋認証に代表される標準的な認証に加えてタッチストロークを利用したパッシブ認証の導入が検討されている．タッチストロークを利用したパッシブ認証は，スマートフォンの操作性を損ねない一方で，画面の覗き見による模倣攻撃のリスクが報告されている．著者らの先行研究では，訓練データに第三者による模倣データを含めて学習を行うことで，模倣攻撃への耐性が得られることを確認した．しかし，実運用を想定した場合，模倣データを取得することは困難である．そこで本研究では，模倣はデータの類似性を高める行為であるという前提のもと，あらかじめ訓練用に用意した第三者のストロークデータについて誤認証率を算出し，誤認証率に基づいて学習を行う新たなパッシブ認証の学習手法を提案する．23 人分のデータを用いた評価実験の結果より，学習させるストローク操作の種類を増やし，本人として誤認証しやすいユーザのデータをオーバーサンプリングして拡張する手法が，誤認証率および耐模倣性の観点から有効であることを確認した．

キーワード　パッシブ認証，機械学習，タッチストローク，模倣，オーバーサンプリング，スマートフォン

1. はじめに

近年，スマートフォンを利用するユーザ数が増加し，スマートフォンの普及率が高まりを見せている．総務省の「令和２年通信利用動向調査」[1]によると，スマートフォンを保有している世帯の割合が 86.8%と，令和元年度の 83.4%から堅調に増加しており，個人の所有率も増加傾向にある．スマートフォンの普及率の増加に伴い，スマートフォン上で個人情報を利用する場面も増加している．スマートフォン所有者の個人情報を保護するためにスマートフォンには標準的に認証機能が搭載されている．一方で，標準的な認証機能は悪意のあるユーザによって突破されるリスクが報告されている．例えば，PIN（Personal Identification Number）は画面の覗き見によって推測が可能であり[1][2], パターン認証は皮脂による画面の汚れによって推測が可能であることが報告されている[3]．近年使用率が高まっている指紋認証[4]や顔認証[5]も，生体情報を人工的に複製することで突破できてしまうことが報告されている[6]．こうした背景から，新たなスマートフォンの認証機能の需要が高まりを見せている．

新たなスマートフォンの認証機能として，座標や圧力，ストローク速度といった画面操作時のストロークデータを使用したパッシブ認証が注目を集めている．認証に画面操作時のストロークデータを使用することで，ユーザに対して特別な操作を要求せずに，現在のスマートフォンの使用者が所有者本人であるかを継続的に判定することができる．また，既存の認証機能を実施した後にパッシブ認証を導入し，二重に認証を実施することで，手軽にスマートフォンのセキュリティ性を向上させることができる．

一方で，ユーザの行動に基づいた認証は模倣による攻撃に弱いとされ，ストロークデータに基づく認証も画面の覗き見による操作方法の模倣によって攻撃されるリスクが存在する[7]．著者らの先行研究[8]では，画面の覗き見によるストローク操作の模倣によって，パッシブ認証の誤認証率が上昇することを確認した．これは，模倣によって攻撃者のストロークと本人のストローク間の類似性が高まったことを意味する．また，このような模倣による攻撃を防ぐ手法として，あらかじめ用意した模倣データを訓練時に学習させる手法が有効であることを確認した．これは，本人と偽者の分類を学習する際に，その境界付近をより重点的に学習できた結果であると考えられる．しかしながら，実運用を想定した場合，模倣データは生成が困難なデータであるため，分類器を構築する際にあらかじめ学習させるのは困難である考えられる．

そこで本稿では，模倣データは本人データとの類似性が高く，誤認証を引き起こしやすいデータであると

[1] 総務省，“令和２年通信利用動向調査”，2021, https://www.soumu.go.jp/johotsusintokei/statistics/data/210618_1.pdf

いう前提のもと，著者らの先行研究[8]で提案した「模倣データの学習」を，「誤認証率が高いユーザの学習」に置き換え，実運用にも対応可能な新たなパッシブ認証の学習手法を提案する．

本稿では次の構成をとる．２節でスマートフォンにおける認証の脆弱性とその対抗手法に関する関連研究を述べ，３節で提案手法について詳述する．続いて，４節で予備知識として本手法において使用するオーバーサンプリング手法と評価指標を説明し，５節で提案手法を評価するために実施する評価実験の方法について詳述し，６節においてパッシブ認証の最適な学習手法について議論を行い，７節で本稿をまとめる．

2. 関連研究

スマートフォンは携帯性が高く，公共の場所で使用する場面も多い為，悪意のある第三者によって不正に個人情報や認証に使用される情報が盗まれる危険性が存在する．中でも標準的な認証として広く普及しているパターンや PIN などの認証は，SSA（Shoulder Surfing Attacks：ショルダーサーフィン攻撃）を受けやすいとされている[1][2][7]．United States Naval Academy の Davin ら[1]は 2017 年に，SSA が行われる場面を再現するために，PIN やパターンを入力する操作を多方向からビデオカメラで撮影し，撮影された映像から PIN やパターンの推測が可能か実践的な検証を実施した．検証の結果，ビデオ視聴後の初回の推測で，PIN は 45.8%，パターンは 87.5%，線の描写があるパターンは 95.8%の精度で推測が可能であることを確認した．また，University Tenaga Nasional の Aris ら[2]は，2019 年にスマートフォンの PIN やパターン認証などの画面ロック機能は覗き見による攻撃を受けやすいとして，2009 年から 2018 年に提案された生体認証を使用しない画面ロック手法を対象に，SSA を防ぐ技術に関するサーベイを実施した．Aris らのサーベイにより，SSA に対する耐性が得られると期待される 10 項目の技術が明らかになり，生体情報を用いない認証において，SSA への耐性を向上させる指針が示された．

生体認証においては，第三者による Spoofing Attacks（なりすまし攻撃）が脅威として報告されており[7]，その対抗手法の提案が行われている．National Institute of Informatics の越前ら[6]は，2018 年に写真から偽造した指紋によって，0.01%の FAR（False Acceptance Rate：他人受入率）で指紋認証における他者へのなりすましが可能であることを示した．その上で，指紋センサーによる認証が可能で，かつ写真による指紋情報の不正な取得を防ぐことが可能な皮膚装着型のウェアラブルデバイスを提案し，その有用性を示した．

ストロークデータのようなユーザの行動や操作を使用した認証では，なりすましや SSA が脅威として報告されている[7]．Texas Tech University の Serwadda ら[9]は，2016 年にプログラミングで操作が可能な市販のレゴロボットを用いて，タッチストロークを利用した認証においてユーザへのなりすましが可能であるかについての検証を行った．Serwadda らは，なりすましの対象として設定したユーザから最大で 10 ストローク分のデータを不正に盗んだ場合を想定し，盗んだデータから生成したストローク操作をロボットに実行させることにより，なりすましの検証を実施した．7 種類の機械学習モデルを使用した評価実験では，盗んだデータを使用しない zero-effort attack と比較して，最大で FAR が 5 倍増加することを確認した．このようなセンサデータを不正に取得するなりすまし攻撃に対しては，画面タップの検出を困難にする手法[10]や，画面に内部的な倍率をかけて座標データの読み取りを困難にする手法[11]を用いることで，なりすまし攻撃を防ぐことが可能である．

SSA を防ぐ手法としては，著者らの先行研究[8]が挙げられる．著者らの先行研究[8]では，SSA によるストローク操作の模倣によって，スマートフォン所有者へのなりすましが可能かについての検証を実施した．認証の分類器として AROW[12]を使用し，23 人の大学生から取得したタッチストロークを用いて行った評価実験では，画面の覗き見によるストロークの模倣によって，模倣未実施時の EER 0.67%から EER 0.75%へと誤認証率が増加することを確認した．模倣による誤認証率増加の結果を踏まえて，SSA を防ぐ手法としてあらかじめ第三者によるストロークの模倣データを用意し，認証に使用する分類器の訓練時に模倣データを学習させる手法を提案した．本手法を適用させることで，模倣データを学習させない場合と比較して，誤認証率と耐模倣性の両方が向上することを確認した．

3. 提案手法

本稿では，パッシブ認証の認証精度および耐模倣性の向上を目的に，実運用にも対応可能な新たなパッシブ認証の学習手法を提案する．

既存のスマートフォンの認証機能において，新たな認証機能が提案されるとともに，想定される攻撃への防御手法についても提案や検証が行われている．一方で，タッチストロークを用いたパッシブ認証に関する研究では，新たな認証手法や既存手法の認証精度向上を目指した研究は多く存在するが，認証への攻撃に対する防御手法を実験的に検証した研究は，著者の知る限り著者らの先行研究[8]以外に存在しない．

著者らの先行研究[8]では，模倣データをあらかじめ訓練させることによってパッシブ認証の認証精度と耐模倣性が向上することを明らかにした．しかしながら，実運用を想定した場合，模倣データは模倣を行う第三

者を設定し，その第三者に対して自身のストローク操作を公開しなければならないため，セキュリティやコストの面から生成が困難なデータであると考えられる．そこで本稿では，模倣データは本人データとの類似性が高く，誤認証を引き起こしやすいデータであるという前提のもと，著者らの先行研究[8]で提案した「模倣データの学習」を，「誤認証率が高いユーザの学習」に置き換え，実運用にも対応可能な新たなパッシブ認証の学習手法を提案する．具体的には，あらかじめ用意した訓練用ユーザについて，本人との誤認証率を算出し，算出した誤認証率に基づいて重点的に学習を行うユーザを能動的に選定した上で分類器の構築を行う手法を提案する．選定したユーザの重点的な学習については，既存のデータから新規のデータを生成することによりオーバーサンプリングを行う手法を採用する．本稿で提案するパッシブ認証の学習手順を以下に示す．

1. スマートフォン所有者（本人）のストロークデータと，分類器を訓練する際に偽者役のデータとして使用するN人分のストロークデータを取得する．
2. 偽者役のN人のユーザの中から，本人との誤認証率を算出するユーザを 1 人選出する．
3. 本人のストロークデータに「positive」ラベル，2 で選出したユーザを除いた$N-1$人分の偽者役のストロークデータに「negative」ラベルをそれぞれ付与し，本人のストロークデータかを判定する二値分類器を生成する．
4. 3 で使用していない本人のストロークデータと，2 で選出したユーザのストロークデータをもとに評価用データセットを生成し，本人との誤認証率を算出する．
5. 選出する偽者役のユーザを変え，2 から 4 の流れで偽者役のN人のユーザすべてについて，本人との誤認証率の算出を行う．
6. 5 で得られた偽者ユーザ毎の誤認証率をもとに，重点的に学習するユーザの選定を行い，認証に使用する分類器の構築を行う．

本稿では，先行研究[8]で使用した二値分類器である AROW[12]を使用して提案手法の検証を行う．また，ユーザの選定および学習方法として以下の 3 項目を設定し，最適な学習手法を検証する．

A) 誤認証率の高い上位M人のユーザを選出し，選出したユーザのみを学習
B) 誤認証率の高い上位M人のユーザを選出し，選出したユーザのデータをオーバーサンプリングして学習
C) 誤認証率に閾値を設定し，誤認証率が閾値以上であるユーザのデータをオーバーサンプリングして学習

データのオーバーサンプリングには，既存のデータに基づいて新たなデータを生成することによりオーバーサンプリングを実施する SMOTE[13] と ADASYN[14]を採用する．SMOTE と ADASYN の詳細については，4.1 項と 4.2 項でそれぞれ説明する．学習手法の評価は，各手法を用いて構築した分類器の誤認証率および耐模倣性の 2 つの観点から実施する．

4. 予備知識

本節では，本稿において使用するオーバーサンプリング手法および認証精度評価指標について説明する．

4.1. SMOTE

SMOTE(Synthetic Minority Over-sampling Technique)[13]は，不均衡データセットを均衡化することを目的に，Chawla らによって 2002 年に提案されたオーバーサンプリング手法である．SMOTE では，少数派クラスに属する各データについて，同じ少数派クラスに属する近傍のデータを利用して新規のデータを生成することによりオーバーサンプリングを行う．SMOTE のアルゴリズムをアルゴリズム 1 に示す．

アルゴリズム 1　Algorithm SMOTE in [13]

Input: Number of minority class samples T; Amount of SMOTE N%; Number of nearest neighbors k
Output: $(N/100) * T$ synthetic minority class samples
(* If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd. *)
if $N < 100$
  then Randomize the T minority class samples
    $T = (N/100) * T$
    $N = 100$
endif
$N = (int)(N/100)$ (* The amount of SMOTE is assumed to be in integral multiples of 100. *)
k = Number of nearest neighbors
$numattrs$ = Number of attributes
$Sample$[][]: array for original minority class samples
$newindex$: keeps a count of number of synthetic samples generated, initialized to 0
$Synthetic$[][]: array for synthetic samples
(* Compute k nearest neighbors for each minority class sample only. *)
for $i \leftarrow 1$ **to** T
  Compute k nearest neighbors for i, and save the indices in the $nnarray$
  Populate(N, i, $nnarray$)
endfor

Populate(N, i, $nnarray$) (* Function to generate the synthetic samples *)
while $N \neq 0$
  Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
  for $attr \leftarrow 1$ **to** $numattrs$
    Compute: $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$
    Compute: $gap =$ random number between 0 and 1
    $Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$
  endfor
  $newindex++$
  $N = N - 1$
endwhile
return (* End of Populate. *)

4.2. ADASYN

ADASYN[14]は He らによって 2008 年に提案された SMOTE の改良手法である．SMOTE では，不均衡データの少数派クラスに属するデータを対象に，同クラスの近傍データに基づいて新規データの生成することによりオーバーサンプリングを行うのに対し，ADASYN

では多数派クラスに属するデータも考慮して新規データの生成およびオーバーサンプリングを実施する．ADASYN では，近傍にある多数派クラスのデータ数に応じて生成するデータ数を増減させるため，クラスの境界に近いデータをより重点的にオーバーサンプリングすることができる．ADASYN のアルゴリズムをアルゴリズム 2 に示す．

アルゴリズム 2　[Algorithm – ADASYN] in [14]

Input

(1) Training data set D_{tr} with m samples $\{\boldsymbol{x_i}, y_i\}$, $i = 1, \cdots, m$, where $\boldsymbol{x_i}$ is an instance in the n dimensional feature space $\boldsymbol{X}$ and $y_i \in Y = \{1, -1\}$ is the class identity label associated with $\boldsymbol{x_i}$. Define m_s and m_l as the number of minority class examples and the number of majority class examples, respectively. Therefore, $m_s \leq m_l$ and $m_s + m_l = m$.

Procedure

(1) Calculate the degree of class imbalance:

$$d = m_s / m_l \tag{1}$$

where $d \in (0, 1)$.

(2) If $d < d_{th}$ then (d_{th} is a preset threshold for the maximum tolerated degree of class imbalance ratio):

(a) Calculate the number of synthetic data examples that need to be generated for the minority class:

$$G = (m_l - m_s) \times \beta \tag{2}$$

Where $\beta \in [0, 1]$ is a parameter used to specify the desired balance level after generation of the synthetic data. $\beta = 1$ means a fully balanced data set is created after the generalization process.

(b) For each example $\boldsymbol{x_i} \in minorityclass$, find K nearest neighbors based on the Euclidean distance in n dimensional space, and calculate the ratio r_i defined as:

$$r_i = \Delta_i / K, \quad i = 1, \cdots, m_s \tag{3}$$

where Δ_i is the number of examples in the K nearest neighbors of $\boldsymbol{x_i}$ that belong to the majority class, therefore $r_i \in [0, 1]$;

(c) Normalize r_i according to $\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i$, so that $\hat{r}_i$ is a density distribution ($\sum_i \hat{r}_i = 1$)

(d) Calculate the number of synthetic data examples that need to be generated for each minority example x_i:

$$g_i = \hat{r}_i \times G \tag{4}$$

where G is the total number of synthetic data examples that need to be generated for the minority class as defined in Equation (2).

(e) For each minority class data example $\boldsymbol{x_i}$, generate g_i synthetic data examples according to the following steps:

Do the **Loop** from 1 to g_i:

(i) Randomly choose one minority data example, $\boldsymbol{x_{zi}}$, from the K nearest neighbors for data $\boldsymbol{x_i}$.

(ii) Generate the synthetic data example:

$$\boldsymbol{s_i} = \boldsymbol{x_i} + (\boldsymbol{x_{zi}} - \boldsymbol{x_i}) \times \lambda \tag{5}$$

where $(\boldsymbol{x_{zi}} - \boldsymbol{x_i})$ is the difference vector in n dimensional spaces, and λ is a random number: $\lambda \in [0, 1]$.

End **Loop**

4.3. 認証精度評価指標

本稿では，二値分類器の認証精度を評価するにあたり，評価指標として EER（Equal Error Rate：等価エラー率）を使用する．EER は認証システムの精度評価を行う際に一般的に使用される指標であり，二値分類の閾値を変化させたとき，FRR（False Rejection Rate：本人拒否率）と FAR（False Acceptance Rate：他人受入率）が等しくなる点におけるエラー率を示す．FRR（本人拒否率）と FAR（他人受入率）はそれぞれ以下の式(1), (2)で表される．このとき，式(1)と式(2)における TP, FN, FP, TN はそれぞれクラス分類の予測結果を表し，表 1 に示される混同行列で定義される．

$$FRR = \frac{FN}{TP + FN} \tag{1}$$

$$FAR = \frac{FP}{FP + TN} \tag{2}$$

表 1　混同行列

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

5. 評価実験

5.1. ストロークデータ

本稿では，著者らの先行研究[8]で使用した 26 次元のストローク特徴量を用いて二値分類器の生成を行う．本稿で使用するストローク特徴量およびその算出方法を表 2，図 1 にそれぞれ示す．表 2 のストローク特徴量を含むストロークデータは，著者らの先行研究[8]において，独自のアプリケーションを用いて 23 人のユーザから収集したデータを使用する．なお，本データには 23 人のユーザの通常のストロークデータ（通常ストローク）に加えて，模倣者役として設定した 21 人のユーザが，被模倣者役として設定した 2 人のユーザのストローク操作を模倣したストロークデータ（模倣ストローク）が含まれる．また，通常ストロークと模倣ストロークは，特徴量毎に全データを用いて min-max normalization を実施し，正規化を行った上で使用する．

表 2 ストローク特徴量

No.	特徴量名	内容
1	startX	ストローク開始地点の X 座標
2	startY	ストローク開始地点の Y 座標
3	stopX	ストローク終了地点の X 座標
4	stopY	ストローク終了地点の Y 座標
5	startPressure	ストローク開始地点の圧力
6	stopPressure	ストローク終了地点の圧力
7	midPressure	ストローク中間地点の圧力
8	averageVelocity	ストローク中の平均速度(pt/s)
9	vel20	ストローク 20%地点の速度(pt/s)
10	vel50	ストローク 50%地点の速度(pt/s)
11	vel80	ストローク 80%地点の速度(pt/s)
12	strokeDuration	ストロークにかかった時間(s)
13	interStrokeTime	ストローク間隔時間(s)
14	lengthEE	ストローク開始地点と終了地点のユークリッド距離(pt)
15	angleEE	ストローク開始地点と終了地点がなす角度(deg)
16	lengthTrj	ストローク軌跡の長さ(pt)
17	ratioTrj2EE	lengthEE と lengthTrj の比 (lengthTrj / lengthEE)
18	direction	ストロークの方向（上方向/下方向の二値）
19	x20	ストローク 20%地点の X 座標
20	x50	ストローク 50%地点の X 座標
21	x80	ストローク 80%地点の X 座標
22	y20	ストローク 20%地点の Y 座標
23	y50	ストローク 50%地点の Y 座標
24	y80	ストローク 80%地点の Y 座標
25	maxPressure	ストローク中の最大圧力
26	averagePressure	ストローク中の平均圧力

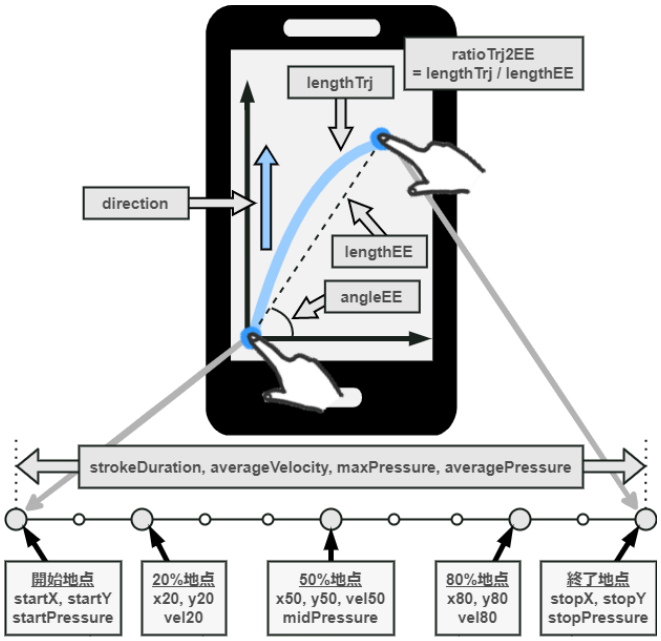

図 1 ストローク特徴量の取得方法
（先行研究[8]と同様）

5.2. 学習手法の検証内容

本稿で提案する，あらかじめ用意した訓練用ユーザの誤認証率に基づくパッシブ認証の学習手法の有用性を評価するために，以下の 3 つの項目に関しての検証および評価を実施する．

A) 誤認証率の高い上位*M*人のユーザを選出し，選出したユーザのみを学習

B) 誤認証率の高い上位*M*人のユーザを選出し，選出したユーザのデータをオーバーサンプリングして学習

C) 誤認証率に閾値を設定し，閾値以上のユーザのデータをオーバーサンプリングして学習

検証項目 A では，誤認証率が高い順に訓練用ユーザを選出し，選出したユーザのみ学習する手法を検証する．本稿では，選出するユーザ数を 5 人，10 人，15 人に設定した場合それぞれについて評価を行う．検証項目 B および検証項目 C では，SMOTE[13]および ADASYN[14]を用いて誤認証率が高いユーザのデータを 2 倍にオーバーサンプリングすることにより，本人と類似するユーザの重点的な学習を行う．検証項目 B では人数の観点から，検証項目 C では誤認証率の観点からそれぞれユーザの選出を行い，選出したユーザのオーバーサンプリングを実施する．本稿では，検証項目 B で選出するユーザ数を 5 人，10 人，15 人に設定した場合，検証項目 C で用いる閾値を EER 5%，EER 10%，EER 15% に設定した場合でそれぞれ評価を行う．

各検証項目では，著者らの先行研究[8]で認証精度の向上を確認したストローク方向を上方向と下方向で区別して学習を行う手法を用いて，それぞれの方向ごとに分類器の構築および評価を実施する．また，各検証項目では，通常ストロークに対する EER を用いた誤認証率の評価と，模倣ストロークに対する EER を用いた誤認証率の評価を実施し，2 つの観点から評価を行う．

5.3. オーバーサンプリングデータの確認

評価実験を行うにあたり，5.2 項で示した検証項目 B および検証項目 C で使用する SMOTE[13]および ADASYN[14]について，各手法によって生成されるデータの確認を行った．評価実験で使用する 23 人のユーザから 2 ユーザ（ユーザ A とユーザ B）を選出し，26 次元のストローク特徴量を含む通常ストロークを，ユーザ A とユーザ B からそれぞれ 400 ストロークずつ抽出した．ユーザ A の通常ストロークを，オーバーサンプリングを行う対象のデータとして設定し，SMOTE および ADASYN を用いて 2 倍のオーバーサンプリングを行った．ユーザ A の通常ストローク，ユーザ B の通常ストローク，SMOTE によるオーバーサンプリングデータ，ADASYN によるオーバーサンプリングデータを，それぞれ PCA（主成分分析）を用いて 2 次元に圧縮し，同一平面上にプロットした結果を図 2 に示す．図 2 より，SMOTE ではユーザ A の元データに基づいて満遍なくデータが生成されるのに対し，ADASYN ではユーザ A とユーザ B の境界付近のデータが重点的に生成される様子が確認された．

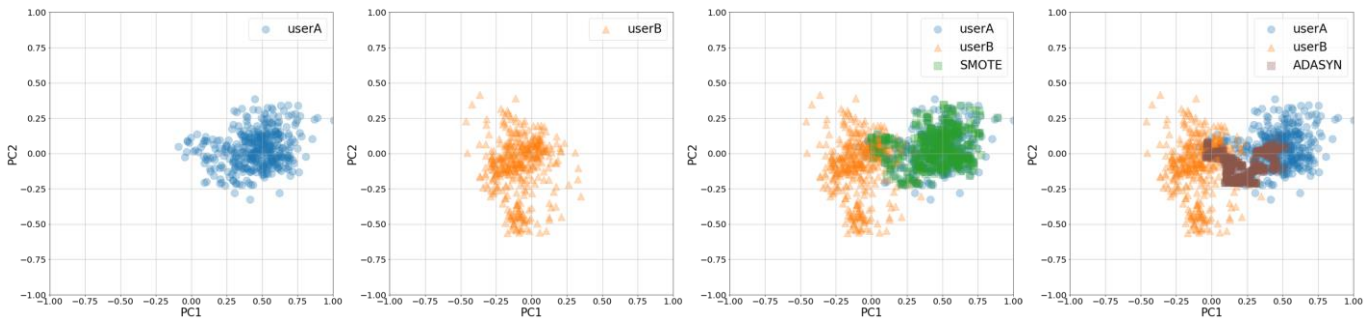

図 2 ストロークデータのプロット結果（PCA を用いて 2 次元に圧縮）

5.4. 評価実験の流れ

5.2 項で示した各学習手法を評価する具体的な手順を以下に示す.

1. 23 人のユーザから，本人役のユーザ（本人）を 1 人選出する.
2. 本人を除く 22 人のユーザから，攻撃者役のユーザ（攻撃者）を 1 人選出する.
3. 本人と攻撃者を除く 21 人のユーザから，偽者役のユーザ（偽者）を 1 人選出し，残りの 20 人のユーザを訓練用ユーザとして設定する.
4. 各訓練用ユーザから通常ストロークをそれぞれ 400 ストロークずつ抽出する.
5. 本人から通常ストローク 400 ストローク抽出する.
6. 4 で抽出したデータに「-1」のラベルを，5 で抽出したデータに「+1」のラベルをそれぞれ付与する.
7. 各ラベルのデータ数を揃えるために，本人のデータに対してオーバーサンプリングを実施する．本手順では，本人の実データを学習させることを重視し，既存のデータを複製することによりオーバーサンプリングを実施する.
8. 本人のデータと訓練用ユーザのデータをもとに二値分類器を構築する.
9. 評価用データセットを構築するために，本人と偽者からそれぞれ新たに通常ストロークを 200 ストロークずつ抽出する.
10. 8 で構築した二値分類器を用いて，9 で構築した評価用データセットの誤認証率（EER）を算出する．この時，短期的な操作のブレによる誤認証を防ぐために著者らの先行研究[8]で採用したスライディングウィンドウを用いて認証結果を出力する手法を本稿においても採用する．また，本稿ではウィンドウサイズを 15 に設定し，分類器による 15 ストローク分の判定結果から算出される平均値をもとに認証結果の出力を行う.
11. 3 で選出する偽者役のユーザを変えて 4-10 の手順を実施し，本人に対する 21 人分の誤認証率および誤認証率の順位付けを取得する.
12. 本人と攻撃者を除く 21 人のユーザを訓練用ユーザとして設定し，4-8 と同様の手順でパッシブ認証に使用する分類器の構築を行う．この時，11 で得られた誤認証率および誤認証率の順位付けに基づいて，各学習手法の適用を行う.
13. 本人から通常ストロークを 200 ストローク，攻撃者から通常ストロークもしくは模倣ストロークを 200 ストローク抽出し，評価用データセットを構築する.
14. 12 で構築した二値分類器を用いて，10 と同様の方法で，13 で構築した評価用データセットの誤認証率（EER）を算出する.
15. 1 で選出する本人と，2 で選出する攻撃者を変えて計 506 通りの交差検証を実施し，各学習手法における誤認証率（EER）の平均値を評価結果の代表値として採用して評価結果の比較を行う.

5.5. 学習手法の評価結果

5.4 項で示した手順に沿ってパッシブ認証に使用する分類器の構築を行い，5.2 項で示した学習手法毎に誤認証率に関する評価を実施した．各学習手法を評価する際に構築した訓練用データセットと評価用データセットにおける本人および偽者のデータ数を表 3 に示す．表 3 で示したデータセットを用いて，学習手法毎に通常ストロークおよび模倣ストロークを使用して上下方向別に誤認証率を算出した結果を表 4 および図 3 に示す.

表 3　各データセットを構成する本人及び偽者のデータ数

検証項目	学習手法	訓練用データセット		評価用データセット	
		本人	偽者	本人	偽者
	全ユーザを均等に学習（ベースライン）	8,400	8,400	200	200
A	上位 5 ユーザのみ学習	2,000	2,000	200	200
	上位 10 ユーザのみ学習	4,000	4,000	200	200
	上位 15 ユーザのみ学習	6,000	6,000	200	200
B	上位 5 ユーザを SMOTE	8,400	10,400	200	200
	上位 10 ユーザを SMOTE	8,400	12,400	200	200
	上位 15 ユーザを SMOTE	8,400	14,400	200	200
	上位 5 ユーザを ADASYN	8,400	10,400	200	200
	上位 10 ユーザを ADASYN	8,400	12,400	200	200
	上位 15 ユーザを ADASYN	8,400	14,400	200	200
C	EER 5%以上を SMOTE	8,400	≥8,400	200	200
	EER 10%以上を SMOTE	8,400	≥8,400	200	200
	EER 15%以上を SMOTE	8,400	≥8,400	200	200
	EER 5%以上を ADASYN	8,400	≥8,400	200	200
	EER 10%以上を ADASYN	8,400	≥8,400	200	200
	EER 15%以上を ADASYN	8,400	≥8,400	200	200

表 4　各学習手法の誤認証率評価結果

検証項目	学習手法	通常ストローク（上方向）		通常ストローク（下方向）		模倣ストローク（上方向）		模倣ストローク（下方向）	
		EER [%]	SD	EER [%]	SD	EER [%]	SD	EER [%]	SD
	全ユーザを均等に学習（ベースライン）	7.16	14.44	6.12	13.03	8.33	10.42	7.16	12.56
A	上位 5 ユーザのみ学習	11.67**	20.39	10.17**	19.28	10.45	15.15	11.25	22.06
	上位 10 ユーザのみ学習	7.67	15.01	6.40	13.30	7.95	12.19	7.56	14.72
	上位 15 ユーザのみ学習	7.28	14.40	6.13	12.58	8.33	10.97	7.33	13.45
B	上位 5 ユーザを SMOTE	7.03	14.14	5.94	12.46	8.05	10.30	5.99**	11.48
	上位 10 ユーザを SMOTE	6.99	14.08	5.85*	12.21	7.87*	10.18	6.32*	11.87
	上位 15 ユーザを SMOTE	6.96*	14.12	5.93	12.31	8.08	10.25	6.40*	11.91
	上位 5 ユーザを ADASYN	6.91	13.90	6.08	12.62	8.52	11.04	5.73**	10.82
	上位 10 ユーザを ADASYN	6.62**	13.40	6.12	12.75	7.89	10.02	6.13*	11.46
	上位 15 ユーザを ADASYN	6.71*	13.28	6.33	13.12	8.33	10.27	6.05	10.98
C	EER 5%以上を SMOTE	6.96	14.00	5.90	12.33	8.45	10.45	6.51*	11.89
	EER 10%以上を SMOTE	7.01	14.19	5.93	12.27	8.29	10.63	6.48**	11.87
	EER 15%以上を SMOTE	7.01	14.06	6.00	12.44	8.38	10.67	6.74	12.06
	EER 5%以上を ADASYN	6.64**	13.40	6.15	12.99	8.20	10.14	6.13**	11.27
	EER 10%以上を ADASYN	6.64**	13.67	6.21	13.01	7.89	10.06	6.20*	11.46
	EER 15%以上を ADASYN	6.68**	13.78	6.20	12.89	8.31	10.85	6.47	11.67

* p<0.05　** p<0.01　（各学習手法とベースライン間の検定）

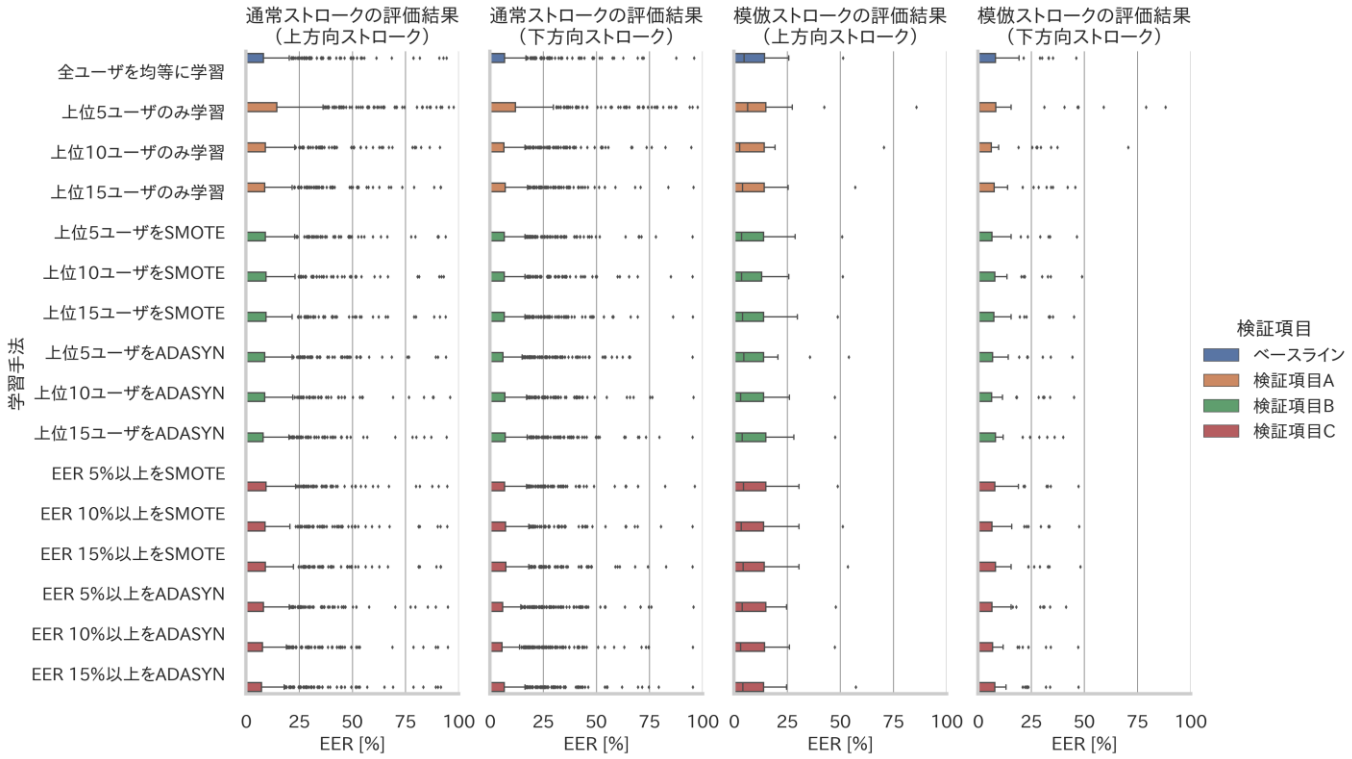

図 3　各学習手法における誤認証率の分布

6. パッシブ認証学習手法の検討

5 節で実施した評価実験の結果（表 4 および図 3）をもとに，ストロークデータに基づくパッシブ認証に適する学習手法について検討を行う．

検証項目 A では，誤認証を引き起こしやすいユーザのみを訓練時に学習させる手法について評価を行った．訓練用ユーザのうち，誤認証率が高い上位 5 ユーザのみを学習させる手法と，全ユーザ（22 ユーザ）を学習させる手法の比較により，通常ストローク評価時では上下方向共に $p<0.01$ で有意に全ユーザを学習させる手法の誤認証率が低いことが認められた．模倣ストロークの評価においても，有意差は認められないが，上下方向共に全ユーザを学習させる手法の誤認証率が低いことが確認された．また，誤認証率が高い上位 15 ユーザのみを学習させる手法についても，上位 5 ユーザのみを学習させる手法と同様の比較結果が得られた．従って，訓練時に学習させるユーザ数，すなわちが学習させるストローク操作の種類が多いほど誤認証率が低下する傾向にあることが示された．

検証項目 B および検証項目 C では，誤認証率が高い訓練用ユーザを対象にオーバーサンプリングを実施し，誤認証を引き起こしやすいデータを追加で学習する手法について評価を行った．採用する閾値やユーザ数によっては，一部ベースラインと比較して誤認証率の増加が見られたが，有意に誤認証率の増加が認められる例は確認されなかった．一方で，有意に誤認証率の低下が認められる例は複数確認されたため，訓練用ユーザの誤認証率に基づいてユーザの選定を行い，選定したユーザが持つデータを拡張する手法は，ストロークデータに基づくパッシブ認証に対して有効な学習手法であるといえる．中でも，誤認証率が高い訓練用ユーザに対して SMOTE を用いてオーバーサンプリングを行う手法は，通常ストロークと模倣ストロークの評価において上下方向共に誤認証率の低下が確認されたため，誤認証率と耐模倣性の観点から有効な手法である．

7. おわりに

本稿では，スマートフォンにおいてストロークデータを使用したパッシブ認証を対象に，あらかじめ訓練用に用意した第三者のストロークデータについて誤認証率を算出し，誤認証率に基づいて重点的に学習するユーザを選定する新たなパッシブ認証の学習手法の検討を行った．評価実験では，23 人分のデータを用いて，本人に対する誤認証率に基づいてユーザの選定およびデータの拡張を行う手法の評価を実施した．評価の結果，学習させるストローク操作の種類を増やし，本人として誤認証しやすいユーザのデータをオーバーサンプリングして拡張する手法が，誤認証率および耐模倣性の観点から有効であることを確認した．

謝　辞

この研究は 2021 年度国立情報学研究所 CRIS 共同研究の助成を受けています。

参　考　文　献

[1] J. T. Davin, A. J. Aviv, F. Wolf, and R. Kuber “Baseline measurements of shoulder surfing analysis and comparability for smartphone unlock authentication”, Proc. of the 35th Annual ACM Conf. on Human Factors in Comput. Syst. (CHI 2017), pp.2496–2503, 2017.

[2] H. Aris, and W. F. Yaakob, “Shoulder surf resistant screen locking for smartphones: A review of fifty non-biometric methods”, IEEE Conf. on Application, Information and Network Security (AINS 2018), pp.7-14, 2018.

[3] A. Aviv, K. Gibson, and E. Mossop, “Smudge Attacks on Smartphone TouchScreens”, Proc. of the 4th USENIX Conf. Offensive Technol., pp.1-7, 2010.

[4] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, “Handbook of Fingerprint Recognition”, Springer Science & Business Media, 2009.

[5] D. Crouse, H. Han, D. Chandra, B. Barbello, and A. K. Jain, “Continuous authentication of mobile user: Fusion of face image and inertial Measurement Unit data,” Proc. of the Int. Conf. Biometrics (ICB 2015), pp.135–142, 2015.

[6] I. Echizen, and T. Ogane, “BiometricJammer: Method to Prevent Acquisition of 40 Biometric Information by Surreptitious Photography on Fingerprints”, IEICE Trans. Inf. Syst., vol. E101D, no.1, pp.2-12, 2018.

[7] W. Meng, D. S. Wong, S. Furnell, and J. Zhou, "Surveying the development of biometric user authentication on mobile phones", IEEE Commun. Surv. Tutorials, vol.17, no. 3, pp.1268-1293, 2015.

[8] M. Kudo, and H. Yamana, “imitation-Resistant Passive Authentication Interface for Strokebased Touch Screen Devices”, Proc. of the 22nd Int. Conf. on Human-Computer Interaction (HCI International), pp.558-565, 2020.

[9] A. Serwadda, V. V. Phoha, Z. Wang, R. Kumar, and D. Shukla, “Toward robotic robbery on the touch screen,” ACM Trans. Inf. Syst. Secur., vol.18, no. 4, 2016.

[10] P. Shrestha, M. Mohamed, and N. Saxena, “Slogger: Smashing motion-based touchstroke logging with transparent system noise”, Proc. of the 9th ACM Conf. Secur. Priv. Wirel. Mob. Netw., pp.67-77, 2016.

[11] N. Z. Gong, R. Moazzezi, M. Payer, and M. Frank, “Forgery-resistant touch-based authentication on mobile devices”, Proc. of the 11th ACM Asia Conf. Comput. Commun. Secur., pp.499-510, 2016.

[12] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive Regularization of Weight Vectors", Proc. of the 23rd Adv. Neural Inf. Process. Syst. 22, pp.414-422, 2009.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique Nitesh”, Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[14] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning”, Proc. of the 2008 IEEE Int. Joint Conf. Neural Netw. (IJCNN’08), pp.1322-1328, 2008.