パッシブ認証の精度向上を目指した模倣データ自動生成
ースマートフォンを対象としてー

工藤 雅士[†1]　高橋 翼[‡2]　牛山 翔二郎[§3]　山名 早人[¶4]

†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 早稲田大学基幹理工学部 〒169-8555 東京都新宿区大久保 3-4-1
¶4 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, yamana, ushiyama}@yama.info.waseda.ac.jp, ‡tsubasa.takahashi@linecorp.com

あらまし 近年，スマートフォンの認証においてタッチストロークを利用したパッシブ認証が注目を集めている．タッチストロークを利用した認証は，認証のために特別な行動を要求しない一方で，第三者によるなりすましの危険性が報告されている．著者らの先行研究では，画面の覗き見によるストローク操作の模倣が，ストローク認証において脅威になり得る可能性を確認した．また，模倣による誤認証を防ぐ手法として，あらかじめ模倣データにより学習器を訓練する手法が有効であることを確認した．しかし，実運用を想定した場合，第三者による模倣データを用いて学習器を訓練することは困難である．そこで本研究では，模倣データを VAE（Variational AutoEncoder）を用いて自動生成し，自動生成された模倣データを学習に用いる手法を提案する．評価実験では，模倣者と被模倣者のストロークデータセットを基に構築した模倣データ自動生成モデルで模倣データを生成し，生成した模倣データを用いて EER による認証精度の評価を実施した．評価の結果，学習器の訓練時に実際の模倣データの代用として，自動生成した模倣データが使用可能であることを明らかにした．一方で，「ストローク操作の模倣」という行為をモデル化するためには，モデル構築用データの拡張や特徴量間の相関関係の学習など，さらなる改善の余地があることを確認した．

キーワード スマートフォン，パッシブ認証，深層学習，タッチストローク，模倣，データ生成

1. はじめに

近年，タッチスクリーンによって画面操作を行うデバイスは世界中で広く普及している．日本においても，その代表とされるスマートフォンの普及率は年々増加傾向にあり，総務省の「通信利用動向調査」[1]によると，令和元年における日本のスマートフォンの世帯保有率は 83.4%に上ると報告されている．また，個人のインターネット利用端末についても，スマートフォンの利用率がパソコンの利用率を 12.9%上回る 63.3%を記録しており，スマートフォンの普及率の高さを裏付ける結果が報告されている．

近年のキャッシュレス化や E コマースの広がりなどを受け，スマートフォンの利用が増加および多様化する一方で，スマートフォン所有者の個人情報をスマートフォン上で扱う場面も増加している．個人情報の保護を目的に，スマートフォンには「パスコード」や「指紋認証」，「顔認証」などの認証機能が標準的に搭載されている．しかし，これらの認証機能は画面の汚れによる推測[1]や，生体情報の人工生成[2][2]などによって，第三者に突破される危険性が存在する．こうした現状から，スマートフォンのセキュリティ性の向上が求められる一方で，スマートフォンの利用頻度の高さから，セキュリティ性と操作性のバランスは，新たな認証機能を導入する上で必要不可欠な要素となっている．

近年では，高い認証精度とユーザビリティを両立させた新しい認証方式として，タッチ座標やタッチ圧力，ストローク速度などのタッチストロークから抽出される特徴量を利用したパッシブ認証（以下，ストローク認証）が注目を集めている．タッチスクリーン上での操作はタッチストロークが主体となるため，ストローク認証では特別な操作を所有者に要求することなく本人判定を実施することができる．一方で，ストローク認証にも既存の認証機能と同様に，サイドチャネル攻撃やショルダーハック[3]といった，第三者のハッキングによるなりすましの危険性が報告されている[3]．

ストローク認証のセキュリティ向上を図るために，近年ではこうしたハッキングのリスクを軽減させるための手法が提案されている．サイドチャネル攻撃への対抗手法としては，デバイスに内蔵されているセンサーから取得されるデータにノイズをかける手法[4]や，

1 総務省，“令和元年通信利用動向調査”, 2019, https://www.soumu.go.jp/johotsusintokei/statistics/data/200529_1.pdf

2 welivesecurity, “Face unlock on many Android smartphones falls for a photo”, 2019, https://www.welivesecurity.com/2019/01/10/face-unlock-many-android-smartphones-falls-photo/

3 マカフィー株式会社，“社員のための抜け目ないショルダーハック対策と注意したい５つの事例”, 2017, https://blogs.mcafee.jp/shoulder-surfing-protect

タッチスクリーンに任意の画面倍率を適用し，座標データの読み取りを困難にする手法[5]が存在する．ショルダーハックへの対抗手法については，著者らの先行研究[6]が挙げられる．著者らの先行研究[6]では，画面の覗き見によるストローク操作の模倣が，ストローク認証において脅威になり得る可能性を実験的に確認し，模倣による誤認証を防ぐ手法として，あらかじめ模倣データを訓練する手法が有効であることを確認した．しかし，実運用を想定した場合，第三者による模倣データを用いて学習器を訓練することは困難である．

本稿では，著者らの先行研究[6]をベースラインとし，確率的なデータのぶれを生じさせることが可能な生成モデルである VAE（Variational AutoEncoder：変分自己符号化器）[7]を用いてタッチストロークの模倣データを自動生成する手法について検証を行う．また，提案手法によって生成された模倣データを訓練に使用した場合において EER（Equal Error Rate：等価エラー率）を算出し，パッシブ認証の認証精度と耐模倣性に関する評価を実施する．本稿では次の構成をとる．２節でストローク認証のなりすましに関連する研究を紹介し，３節で提案する認証システムについて詳述する．続いて，４節で評価実験の詳細と結果を示し，５節において提案手法の課題を述べ，６節で全体をまとめる．

2. 関連研究

本節では，関連研究としてストローク認証におけるサイドチャネル攻撃への対抗手法とショルダーハックへの対抗手法を提示する．

2.1. サイドチャネル攻撃への対抗手法

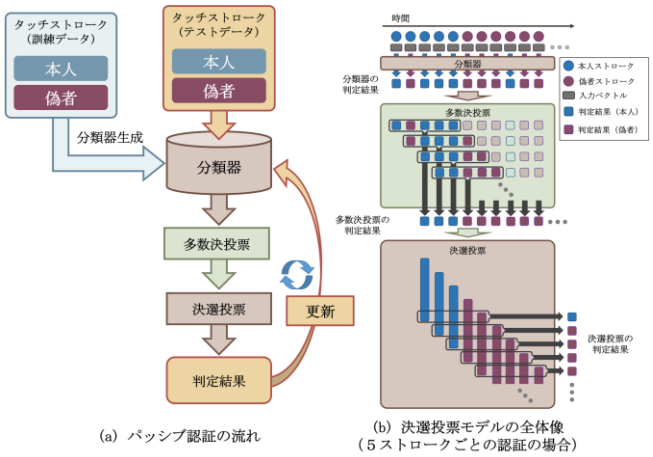
ストローク認証におけるサイドチャネル攻撃では，第三者がデバイス内蔵のセンサーデータを不正に抜き取ることによって所有者へのなりすましを図る．

Shrestha ら[4]は，2016 年にスマートフォン内蔵のモーションセンサーや位置センサーから取得されたデータに対して任意のノイズを付加できる SMASheD と呼ばれるフレームワーク[8]を使用し，「Slogger」と呼ばれるサイドチャネル攻撃への防御システムを提案した．Slogger は，任意のアプリケーションにおいて取得されるセンサーデータにノイズを付加することができ，不正なロギングが疑われるアプリケーションにおいて，タッチイベントの検出やストローク操作の推測を防ぐことを目的としている．Slogger の有効性を検証するために，1,200 回分の画面タップデータを使用した画面タップの検出実験と，ストローク操作が行われた画面領域を推定する検証が実施された．画面タップの検証実験では，Slogger を適用することにより 0%の Precision かつ 0%の Recall で，画面タップの検出が不可能になることが示された．画面領域を推測する検証実験では，スマートフォンの画面領域を二分割した場合において，ランダムに画面領域を推定する場合の精度である 50%から，Slogger の適用によって 35.5%の精度まで推定精度を引き下げることが可能であることが示された．

Gong ら[5]は，2016 年にスマートフォンの座標データの読み取りを困難にする手法として，スマートフォンの画面に対して X 軸方向と Y 軸方向それぞれに一定の時間間隔で任意の倍率をかける手法を提案した．評価実験では，画面の倍率として 0.8 倍，0.9 倍，1.0 倍，1.1 倍，1.2 倍を設定し，X 軸方向と Y 軸方向それぞれに画面倍率を適用させ，計 25 通りの設定について検証を実施した．25 人の実験参加者から，各倍率における水平方向ストロークと垂直方向ストロークのデータ収集を行ない，SVM を用いて構築した分類器を使用してなりすましの評価を実施した．評価の結果，画面に倍率をかけて座標データの読み取りを困難にすることにより，タッチストロークが不正に取得された場合においても本人へのなりすましが困難であることが示された．また，なりすましを行う攻撃者が X 軸方向と Y 軸方向に設定される倍率を知っていたとしてもなりすましが困難であることを確認した．

2.2. ショルダーハックへの対抗手法

ストローク認証におけるショルダーハックでは，第三者がデバイス所有者のストローク操作を覗き見し，操作の特徴を盗み取ることによって所有者へのなりすましを図る．著者らの先行研究[6]では，図 1 に示される 2 段階の多数決投票を導入した認証システムを使用して，画面の覗き見によるストロークの模倣によって所有者へのなりすましが可能かについての実験的な検証を実施した．認証の分類器として AROW[9]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%へと認証のエラー率が増加することを確認した．また，認証のエラー率が増加した点から，ストローク認証において画面の覗き見が深刻な攻撃になり得ることを示した．さらに，あらかじめストロークの模倣データで認証に使用する分類器の訓練を行うことで，EER と耐模倣性の両方が改善することを確認した．しかし，本手法は実運用を想定した場合，訓練データの準備が困難であるという問題がある．



(a) パッシブ認証の流れ　(b) 決選投票モデルの全体像（5ストロークごとの認証の場合）

図 1 認証の全体像（[6] Fig.1 を基に作成）

3. 提案手法

本節では，本稿で提案する模倣データの自動生成手法について詳しく説明する．

3.1. 提案概要

ストローク認証において，第三者から取得した模倣データを認証システム構築時に訓練データとして使用することは，ショルダーハックを防ぐ方法として有効である[6]．しかし，実運用を想定した場合，デバイス所有者のストロークに対応する模倣データを用意するためには，デバイス所有者のストローク操作を第三者に公開し，実際に操作方法の模倣を実施するという手順を踏まなければならない．これは，セキュリティ性とユーザビリティの両面から問題となり得る．

そこで本稿では，「ストローク操作の模倣」という行為のモデル化を行い，入力したストロークデータに対応する模倣データを自動生成する手法を提案する．模倣のモデル化には，確率的なデータのぶれを生じさせることが可能な生成モデルである VAE（Variational AutoEncoder：変分自己符号化器）[7]を使用し，一つのストロークデータから複数の模倣データが生成可能なモデルの構築を行う．また本稿では，著者らの先行研究[6]で提案した図 1 に示される認証システムを使用して，模倣データの自動生成に関する検証と，模倣データを使用した認証精度評価を実施する．

3.2. VAE(Variational AutoEncoder)

本稿では，ニューラルネットワークを用いた生成モデルである VAE を使用して，模倣データの生成を行う．VAE は 2014 年に Kingma らによって提案された生成モデルであり，図 2(a)のようなグラフィカルモデルで表現されるデータ生成過程を，図 2(b)に示されるような AutoEncoder を発展させたニューラルネットワーク構造を用いて実現させる．

VAE において，encoder 部分では入力 x から潜在変数 z をサンプリングする分布のパラメータとして，平均値 μ と標準偏差 σ の出力を行う．decoder 部分では，encoder で出力された平均値μと標準偏差σに基づいてサンプリングされた潜在変数$z \sim N(\mu, \sigma^2)$を入力とし，入力xを復元する形で$f(z)$の出力行う．このような構造を用いることで，訓練データに含まれない未知のデータを生成することが可能になる．

VAE では，潜在変数zから入力xを復元する decoder 部分の確率モデル$p_\theta(x|z)$の尤度を最大にするパラメータ$\emptyset$およびθの推定を行うことで，ネットワーク全体の最適化を図る．最尤推定法と Jensen の不等式を用いると，ネットワークの最適化問題は，以下に示される式(1)の右辺を最大化する問題に変換される．

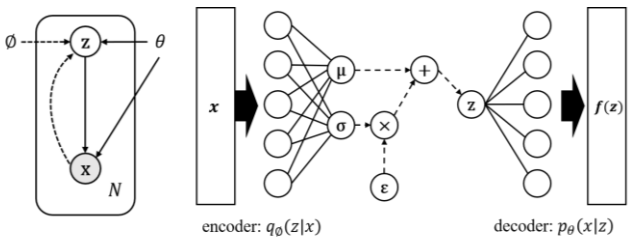
$$\log p_\theta(x) \geq -D_{KL}(q_\emptyset(z|x)||p_\theta(z)) + \int q_\emptyset(z|x) \log p_\theta(x|z)\, dz \quad (1)$$

式(1)右辺の第 1 項は，encoder で出力された平均値μと標準偏差σに基づいてサンプリングされた潜在変数zの事後分布$q_\emptyset(z|x)$と，潜在変数zの事前分布$p_\theta(z)$の KL ダイバージェンスを表しており，両者の分布が近いほど小さな値をとる．VAE では，事前分布$p_\theta(z)$を標準正規分布として仮定することが一般的であるため，第 1 項については潜在変数zの事後分布$q_\emptyset(z|x)$を標準正規分布に近づけるように encoder の学習を行うことで，最適化が達成される．

式(1)右辺の第 2 項は，入力xと，decoder で生成した$f(x)$の類似度を表しており，decoder によるデータの復元が正確であるほど大きな値をとる．第 2 項の最適化を考えた場合，encoder で出力された平均値μと標準偏差σに基づいて確率的に潜在変数zのサンプリングを実施すると，バックプロパゲーションが適用できないという問題が生じてしまう．この問題を解決するために，VAE では Reparametrization Trick と呼ばれる手法を用いて，標準正規分布からサンプリングされた値である$\varepsilon \sim N(0, 1^2)$を使い，式(2)に示される形で潜在変数$z$のサンプリングを行う．

$$z = \mu + \epsilon\sigma \quad (2)$$

以上の最適化を行うようにネットワークのパラメータを調整することで，入力データの特徴を捉えた未知のデータを出力することが可能になる．



(a) グラフィカルモデル　(b) ネットワーク構造

図 2 Variational AutoEncoder（[7] Figure 1 を基に(a)を作成）

3.3. 模倣データ生成モデルの構築方法

本稿で提案する模倣データの自動生成手法は，あらかじめ VAE 訓練用のデータとして，タッチストロークデータとそのタッチストロークを模倣した模倣データのペアが，それぞれのストロークを実施したユーザのラベルが付与された状態で複数用意されていることを前提としている．本手法では，VAE 訓練用データのペアを使用し，図 3 に示される 2 段階のステップを経て模倣データを生成する VAE の構築を行う．模倣データ生成モデルの構築手順を以下に示す．

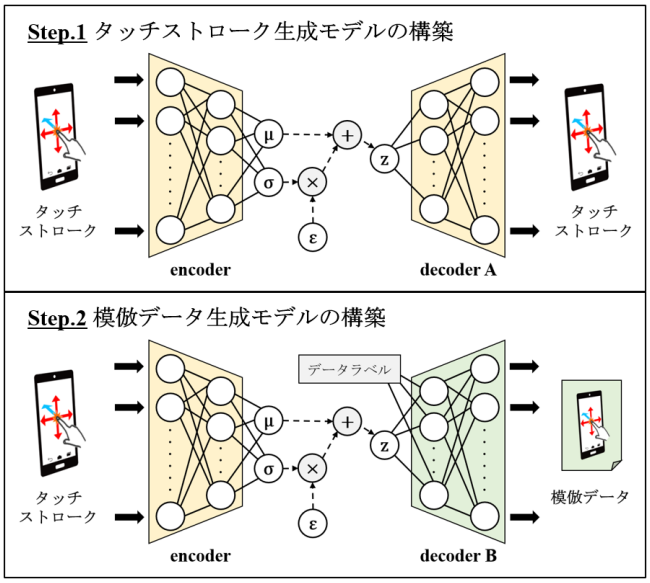


図 3 模倣データ生成モデルの構築手順概要

(1) タッチストローク生成モデルの構築（図 3 **Step.1**）

訓練データのペアのうち，タッチストロークデータのみを抽出し，VAE に入力する．入力されたタッチストロークを復元するように encoder と decoder A の学習を実施し，教師なし学習の方法でパラメータの調整とネットワークの最適化を行う．本手順では，タッチストロークの圧縮についての学習が行われる．

(2) 模倣データ生成モデルの構築（図 3 **Step.2**）

(1)で構築した VAE において，decoder A の破棄を行い，新たに decoder B をセットする．訓練データのペアのうち，タッチストロークデータを入力として与え，その対となる模倣データを生成するように decoder B のみ学習を実施する．このとき，模倣データに付与されたユーザのラベルについても decoder B に入力として与え，ラベル指定で模倣データの生成が行えるように半教師あり学習の方法でパラメータの調整とネットワークの最適化を行う．本手順では，入力に対する模倣データの生成についての学習が行われる．

3.4. 使用する特徴量

本稿では，スマートフォンの画面に指を触れ，画面から指を離すまでの一連の動きを 1 ストロークとし，著者らの先行研究[6]と同様に 1 ストロークから 26 種類のストローク特徴量を抽出し，模倣データの生成およびパッシブ認証の分類器構築を行う．本稿で使用するストローク特徴量とその取得方法をそれぞれ表 1 と図 4 に示す．

表 1 本稿で使用するストローク特徴量
（先行研究[6]と同様）

特徴量名	内容
startX, startY	ストローク開始時の XY 座標
stopX, stopY	ストローク終了時の XY 座標
x20, y20	ストローク 20%地点の XY 座標
x50, y50	ストローク 50%地点の XY 座標
x80, y80	ストローク 80%地点の XY 座標
startPressure	ストローク開始時の圧力
stopPressure	ストローク終了時の圧力
midPressure	ストロークの中間地点における圧力
maxPressure	ストローク中の最大圧力
averagePressure	ストローク中の平均圧力
averageVelocity	ストローク中の平均速度(pt/s)
vel20	ストローク 20%地点の速度(pt/s)
vel50	ストローク 50%地点の速度(pt/s)
vel80	ストローク 80%地点の速度(pt/s)
strokeDuration	ストロークにかかった時間(s)
interStrokeTime	ストローク間隔時間(s)
lengthEE	ストロークの開始地点と終了地点のユークリッド距離(pt)
angleEE	ストロークの開始地点と終了地点がなす角度(deg)
lengthTrj	ストローク軌跡の長さ(pt)
ratioTrj2EE	lengthEE と lengthTrj の比（lengthTrj/lengthEE）
direction	ストロークの方向（上方向/下方向の 2 値）

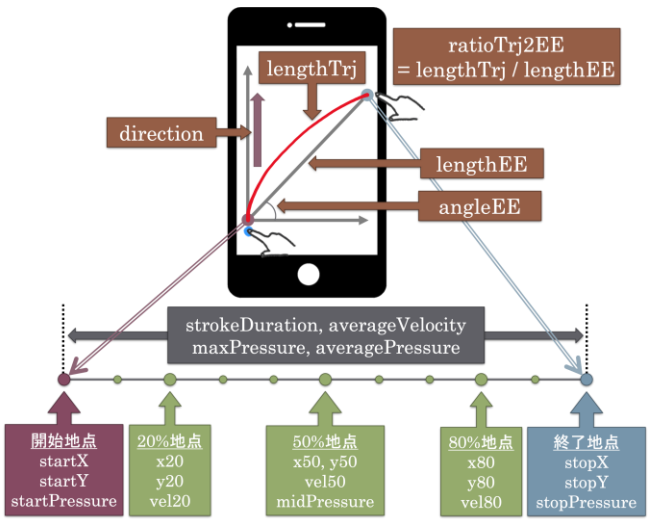


図 4 ストローク特徴量の取得方法
（先行研究[6]と同様）

4. 実験と評価

4.1. データ収集

本項では，評価実験に使用するタッチストロークの収集方法について詳しく説明する．

4.1.1. データ収集用アプリケーション

本稿では，模倣データの生成およびパッシブ認証の認証精度検証を行うために，ストロークデータの収集を実施した．著者らの先行研究[6]では，データ収集用に作成した独自の iOS アプリケーションを Apple 社製の iPhone8+にインストールし，対面形式でデータ収集を実施した．本稿では，時節への考慮と，データ収集の大規模化を目的に，データ収集のプラットフォームを個々のデバイスから web アプリケーションに移行し，これまでに使用していた写真マッチングゲームを web 上に実装する形でデータ収集を実施した．本稿で使用したデータ収集用アプリケーションの設計概要とアプリケーションの流れを以下に示す．

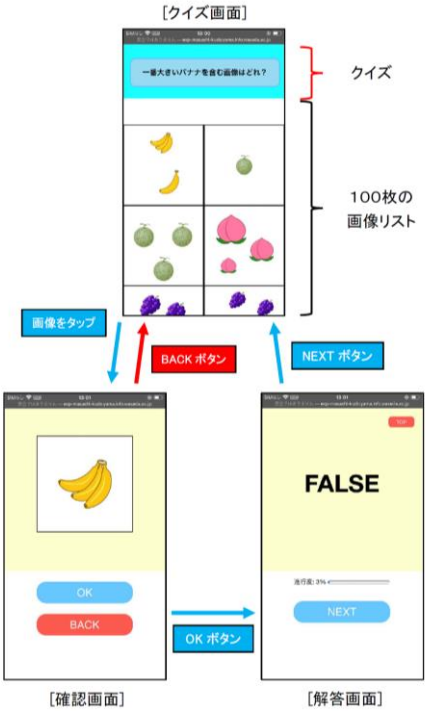


図 5 データ収集用アプリケーション

● **アプリケーションの設計概要**

本アプリケーションは，スマートフォンのユーザが，ショッピングアプリや Web ブラウザを使用して欲しい商品を検索し，検索結果で提示された商品画像群の中から該当する商品を上下方向の画面スクロール操作で探す場面を想定して設計されている．商品の検索結果から該当する商品を見つけるまでの流れを，提示されたクイズに対する答えを見つけるというタスクに置き換え，図 5 に示される画面及び遷移関係で実装が行われている．図 5 のクイズ画面において，クイズの解答候補となる画像は 100 枚存在し，100 枚の画像リストの中に正解画像は 1 枚のみ存在する．また，100 枚の画像リストは，クイズに解答するごとに配置がランダムに変わるよう設計されている．

● **アプリケーションの流れ**

1. クイズ画面：上部に表示されたクイズを確認する．
2. クイズ画面：画像リストをスクロールし，クイズの答えとなる画像を探す．
3. クイズ画面：画像をタップして選択する．
4. 確認画面：選択した画像を確認し，OK ボタンをタップする．
5. 解答画面：クイズの正解・不正解が表示される．
6. 解答画面：NEXT ボタンをタップし，次のクイズへと移行する．
7. 1 - 6 を規定数繰り返し，終了する．

4.1.2. データ収集の実施

本稿では，図 5 に示されるアプリケーションを用いて，被験者 23 人（男性 10 人，女性 13 人，21.4±2.4 歳）を対象に，オンライン上でデータ収集を実施した．データ収集で取得するタッチストロークは，図 5 のクイズ画面において，100 枚の画像リストをスクロールしている時のタッチストロークのみとし，画面タップやクイズ画面以外での画面操作はデータ収集の対象外とした．また，模倣データの自動生成を検証するにあたり，著者らの先行研究[6]と同様に，スマートフォン操作時の通常のタッチストローク（以下，Own strokes）に加えて，「第三者のタッチストロークを意図的に模倣した場合」のタッチストローク（以下，Imitation strokes）についてもデータの収集を行った．データ収集の手順を以下に示す．

1.1. タッチストローク模倣の対象となるユーザ（以下，被模倣者）を 2 名設定し，アプリケーションの操作を実施してもらう．

1.2. 操作中の Own strokes を取得し，Own strokes が上下それぞれ検証用データを構築する際に必要となる 650 ストローク以上取得された段階でアプリケーションを終了する．

1.3. 再度被模倣者にアプリケーションの操作を実施してもらう．この時，被模倣者がアプリケーションを操作している手元の様子を，後方からビデオカメラで 1 分程度撮影した動画（以下，ストローク動画）を取得する（図 6）．

2.1. 被模倣者を除くユーザ（以下，模倣者）にアプリケーションの操作を実施してもらい，手順 1.2 と同様に Own strokes の取得を行う．

2.2. 手順 1.3 で取得した 1 名分のストローク動画を模倣者にリピート再生で視聴してもらい，動画に収

められているストローク操作の特徴を学習してもらう.

2.3. 手順 2.2 で学習したストローク操作を用いながら, 再度アプリケーションの操作を実施してもらい, Own strokes と同様に, 第三者のタッチストロークを意図的に模倣した場合のストロークである Imitation strokes についても上下それぞれ 650 ストローク以上のデータを取得する.

2.4. 手順 2.2 と手順 2.3 を, もう一人の被模倣者のストローク動画で実施する.

図 6 ストローク動画（一部抜粋）

4.2. 模倣データ生成モデルの構築

4.1 項で取得した 2 種類のタッチストロークを使用して, 図 3 の手順で模倣データ生成モデルの構築を行う. 本稿で構築する VAE は, 入出力層のサイズを 26, 潜在変数の次元数を 10 とし, encoder と decoder それぞれにおいて 18 のニューロンを保持する隠れ層を 1 層持つ構造をとる. 学習の際の最適化には Adam を使用し, バッチサイズを 10, エポック数を 5 としてバッチ学習を行う. このとき, 学習率は 0.01 を初期値として設定し, エポックごとに学習率を 0.1 倍して学習を行う. 模倣データ生成モデル構築の詳細な手順を以下に示す.

1.1. 被模倣者の Own strokes と模倣者の Imitation strokes を時系列順に紐づけし, VAE 訓練用のペアデータセットを, 被模倣者毎にそれぞれ 21 セットずつ生成する. このとき, どの模倣者がどの被模倣者を模倣したデータセットかについて区別が行えるように, 被模倣者と模倣者にはそれぞれ 1 から 23 の一意のラベルを割り当てる.

1.2. 模倣者と被模倣者を合わせた 23 人分の Own strokes から, それぞれ上下方向を区別せずに 800 ストロークずつ抽出を行う.

1.3. 合計 18,400 ストローク分のデータに対して, 26 種類のストローク特徴量毎に最大値・最小値を用いた正規化を実施する.

1.4. 正規化したデータを用いてタッチストローク生成モデルの学習及び構築を行う（図 3 Step.1）.

2.1. 手順 1.4 で構築したタッチストローク生成モデルの decoder を破棄し, 学習を実施していない decoder をセットする. なお, 新たにセットする decoder は, 模倣者に割り振った 21 のラベルを One-hot 表現した形で入力できるように, 入力層のサイズを 21 だけ拡張した構造をとる.

2.2. 被模倣者 2 名の中から, 模倣データ生成モデルを学習するためのユーザを 1 名選出する.

2.3. 選出した被模倣者と模倣者のペアデータセット（21 セット存在）に対して, 26 種類のストローク特徴量毎に手順 1.3 で使用した最大値・最小値を用いた正規化を実施する.

2.4. 模倣者 21 人毎に上下方向を区別せずに 800 のペアデータを抽出し, 合計 16,800 のペアデータを用いて模倣データ生成モデルの学習及び構築を行う. 学習の際は, 入力として, 選出した被模倣者の Own strokes と模倣者のラベルを与え, 出力されたデータと入力に対応する Imitation strokes 間の交差エントロピー誤差を類似度として使用し, この誤差を小さくするように学習を行う.

3. 手順 2.4 で構築した模倣データ生成モデルに, 手順 2.2 で選出をしていない残りの被模倣者の Own strokes を入力し, 任意のデータラベルに対応した模倣データの生成を行う.

4.3. 評価実験

4.2 項で構築した模倣データ生成モデルから生成された模倣データを用いた評価実験について説明する.

4.3.1. 評価実験概要

本稿では, 4.2 項で構築した模倣データ生成モデルを用いて生成した模倣データ（以下, Generated strokes）が, 実際にユーザが模倣を行った際のストロークのような特性を備えているかについて, EER を用いた認証精度評価によって検証を行う. また, ストローク認証の耐模倣性を向上させる手法として, 著者らの先行研究[6]でその有効性を確認したあらかじめ模倣データで訓練を実施する手法について, Generated strokes で模倣データの代用が可能かについての検証を行う.

4.3.2. 評価用データセット

本稿では, Generated strokes の有用性の評価を行うために, 擬似的に被模倣者と模倣者をそれぞれ「スマートフォン所有者（以下, 本人）」,「スマートフォンの所有者ではない第三者（以下, 偽者）」として設定し, 以下の 4 つのデータで構成される実験データセットの構築を行った.

- 本人の訓練データおよびテストデータ
- 偽者の訓練データおよびテストデータ

表 2 評価用データセットの構成と評価結果

データセット									EER [%]
No.	訓練データ				テストデータ				
	本人データ		偽者データ		本人データ		偽者データ		
	ストローク種類	ストローク数	ストローク種類	ストローク数	ストローク種類	ストローク数	ストローク種類	ストローク数	
1	Own	16,000	Own	16,000	Own	400	Own	400	11.42
2-1	Own	16,000	Own	16,000	Own	400	Imitation	400	12.10
2-2	Own	16,000	Own	16,000	Own	400	Generated	400	0.60
3-1	Own	16,000	Own	8,000	Own	400	Imitation	400	9.32
			Imitation	8,000					
3-2	Own	16,000	Own	8,000	Own	400	Imitation	400	11.41
			Generated	8,000					

本人の訓練データおよびテストデータは，2 人の被模倣者から選出した 1 人のタッチストロークを使用して構築を行う．また，偽者のデータに関しては，21 人の模倣者から選出した 1 人のタッチストロークを使用してテストデータの構築を行い，残った 20 人の模倣者からタッチストロークを 800 ストロークずつ均等に使用して訓練データの構築を行う．なお，本人の訓練データ数と偽者の訓練データ数を揃えるために，本人の訓練データに関して 20 倍のオーバーサンプリングを行う．本稿で構築したデータセットを以下に示す．

- **データセット１（表 2 No.1）**

Own strokes のみを用いて本人データと偽者データを構築したデータセット．

- **データセット２（表 2 No.2-1, 2-2）**

データセット１の偽者テストデータを，本人データに対応する Imitation strokes と Generated strokes にそれぞれ置き換えて構築したデータセット．

- **データセット３（表 2 No.3-1, 3-2）**

データセット２―１の偽者訓練データのうち半分を，本人データに対応する Imitation strokes と Generated strokes にそれぞれ置き換えて構築したデータセット．

上記のデータセットについて，著者らの先行研究[6]で使用した認証システムを用いて EER の算出を行う．なお，本人データに使用する被模倣者と，偽者のテストデータに使用する模倣者を変えて各データセットでそれぞれ 42 通りの交差検証を行い，平均の EER をもとに評価を実施する．

4.4. 実験結果と考察

4.3 項で構築したデータセットの EER 評価結果を表 2 および図 7 にそれぞれ示す．

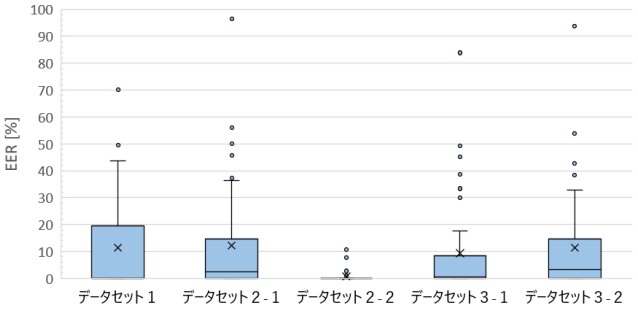


図 7 EER 評価結果
（×は平均を示す）

データセット１とデータセット２－１の比較より，著者らの先行研究[6]で示されたストローク操作の模倣による誤認証率の増加が本稿においても確認された．一方で，データセット２－１とデータセット２－２を比較すると，EER に 10%以上の差が見られた．したがって，実際の模倣データと比較して，生成した模倣データは偽者の判別が容易であることが確認された．

データセット２－１とデータセット３－１の比較より，著者らの先行研究[6]において有効性が示されたあらかじめ模倣データを訓練する手法が，本稿においてもストローク模倣に対して有効であることが確認された．また，データセット２－１とデータセット３－２を比較すると，生成した模倣データを訓練データの一部に使用することで EER の低下が見られた．したがって，著者らの先行研究[6]でショルダーハックに対する有効性を確認した，あらかじめ模倣データを訓練する手法において，疑似的に生成した模倣データの代用は有効であることが確認された．

5. 模倣データ生成モデルの性能向上に向けて

本稿で提案した模倣データ生成モデルから生成した模倣データは，認証システム構築時の訓練用データとして使用した場合，耐模倣性向上に貢献することが確認された．一方で，表 2 および図 7 において，実際の模倣データを使用した場合と生成した模倣データを使用した場合の評価結果を比較すると，データセット２では EER 11.50%，データセット３では EER 2.09%の差が見られる．データセット２－１とデータセット２－２，およびデータセット３－１とデータセット３－２における EER の差について t 検定を実施した結果，データセット３間では両 EER に有意な差は認められなかった（$t(82)=0.51$, n.s.）．一方で，データセット２間では両 EER に有意な差が認められた（$t(82)=3.60$, $p<0.05$）．これは，人工生成した模倣データと実際の模倣データが異なる性質を持つことを示している．したがって,「ストローク操作の模倣」という行為をモデル化するためにはさらなる改善が必要となる．本稿で提案した模倣データ生成モデルにおける今後の課題を以下に示す．

- 被模倣者の人数を増やし，各模倣者から取得する Imitation strokes のバリエーションを増やす．
- ストローク方向を区別して模倣データ生成モデルの構築を行う．
- 特徴量のグルーピングや特徴量間の相関分析などを実施し，新たなパラメータの開発を行う．

6. おわりに

本稿では，ショルダーハックによるストローク模倣に耐性のあるストローク認証を実運用する際に問題となる，模倣データの生成方法に関する問題を解決するために，タッチストロークの模倣データを，VAE を用いて自動生成する手法提案した．提案手法を用いて生成した模倣データを使用した評価実験では，認証システム構築時の訓練用データとして，生成した模倣データ使用した場合，耐模倣性向上に貢献することが確認された．一方で,「ストローク操作の模倣」という行為をモデル化するためには，さらなる改善が必要であることが明らかになった．

今後の課題としては，模倣データ生成モデルを構築する際に使用するデータ数の拡張や，ストローク特徴量の相関関係の学習などが挙げられる．

謝　辞

この研究は２０２０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] A. Aviv, K. Gibson, and E. Mossop, “Smudge Attacks on Smartphone TouchScreens”, Proc. of the 4th USENIX Conf. Offensive Technol., pp. 1-7, 2010.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Kingma, Diederik P and Welling, Max. Auto-encoding variational Bayes. In Proceedings of the International Conference on Learning Representations (ICLR), 2014.

[8] M. Mohamed, B. Shrestha, and N. Saxena, "SMASheD: Sniffing and Manipulating Android Sensor Data for Offensive Purposes", IEEE Trans. Inf. Forensics Secur., vol.12, no.4, pp.901-913, 2017.

[9] 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.